Unmanned Maritime Autonomy Architecture (UMAA)
Maneuver Operations (MO)
Interface Control Document (ICD)
(UMAA-SPEC-MOICD)

MDE Version 4.2.1 Commit 7c2b513

UMAA Spec Commit f2fa426

Version 3.0.1
25 February 2021

UMAA Maneuver Operations ICD

(UMAA-SPEC-MO-ICD)

Signature Page

Submitted by: Rothgeb.Mark.E.ORC3011008580.ID — Digitally signed by Rothgeb.Mark.E.ORC3011008580.ID Date: 2021.05.05 11:54:46 -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 14:51:20 -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 17:18:37 -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 as part of the Unmanned Maritime Autonomy Architecture (UMAA). The services and their corresponding interfaces covered in this ICD encompass the functionality to control and maneuver an Unmanned Maritime Vehicle (UMV) (surface or undersea). As such, it includes the commands and status to/from a vehicle's control systems for controlling all aspects of maneuvering and its associated dynamics. This includes both low-level controls such as heading and speed, as well as higher-level behaviors for loitering or traversing waypoints. This ICD also includes managing driving constraints such as setting bounds on desired speed range or setting a desired maximum turn rate. This document is generated automatically from data models that define its services and their interfaces as part of the Unmanned Systems (UxS) Control Segment (UCS) Architecture as extended by UMAA to provide autonomy services for 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.

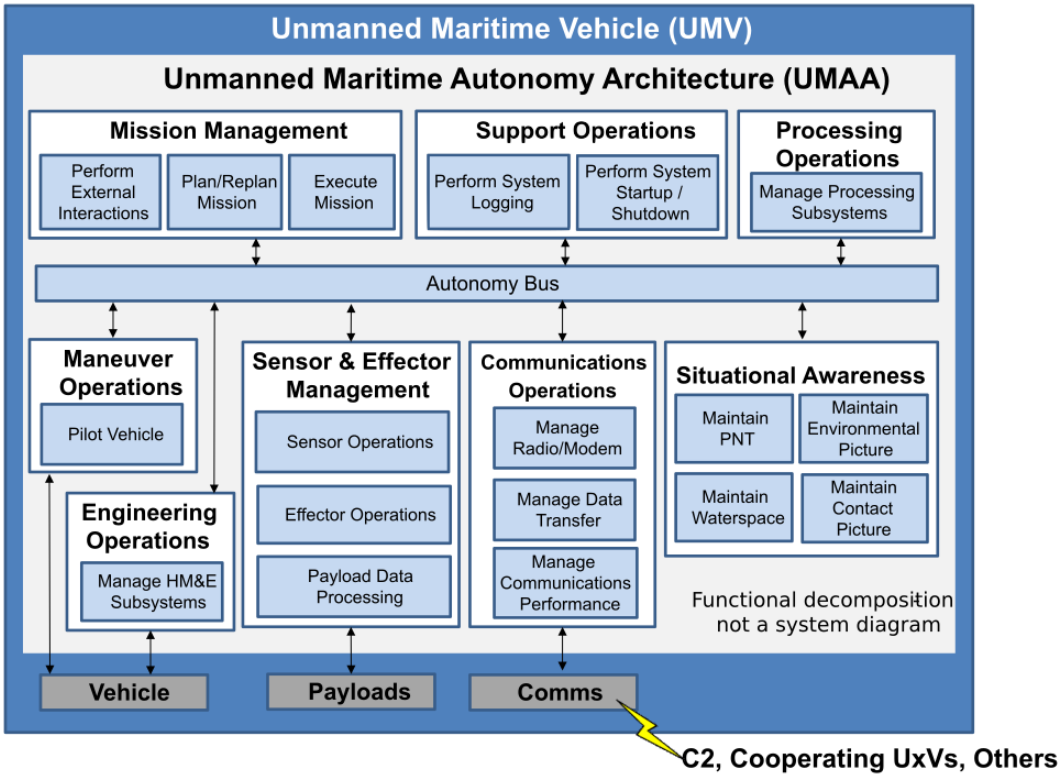

Figure 1: UMAA Functional Organization

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.

Maneuver Operations defines the services required to drive an UMV. Figure 2 depicts an example of various levels of maneuvering behaviors in relation to navigation sensing and Hull, Mechanical, & Electrical (HM&E) control services provided in separate ICDs. Figure 2 depicts an example of a possible component service grouping is shown with the dashed lines.

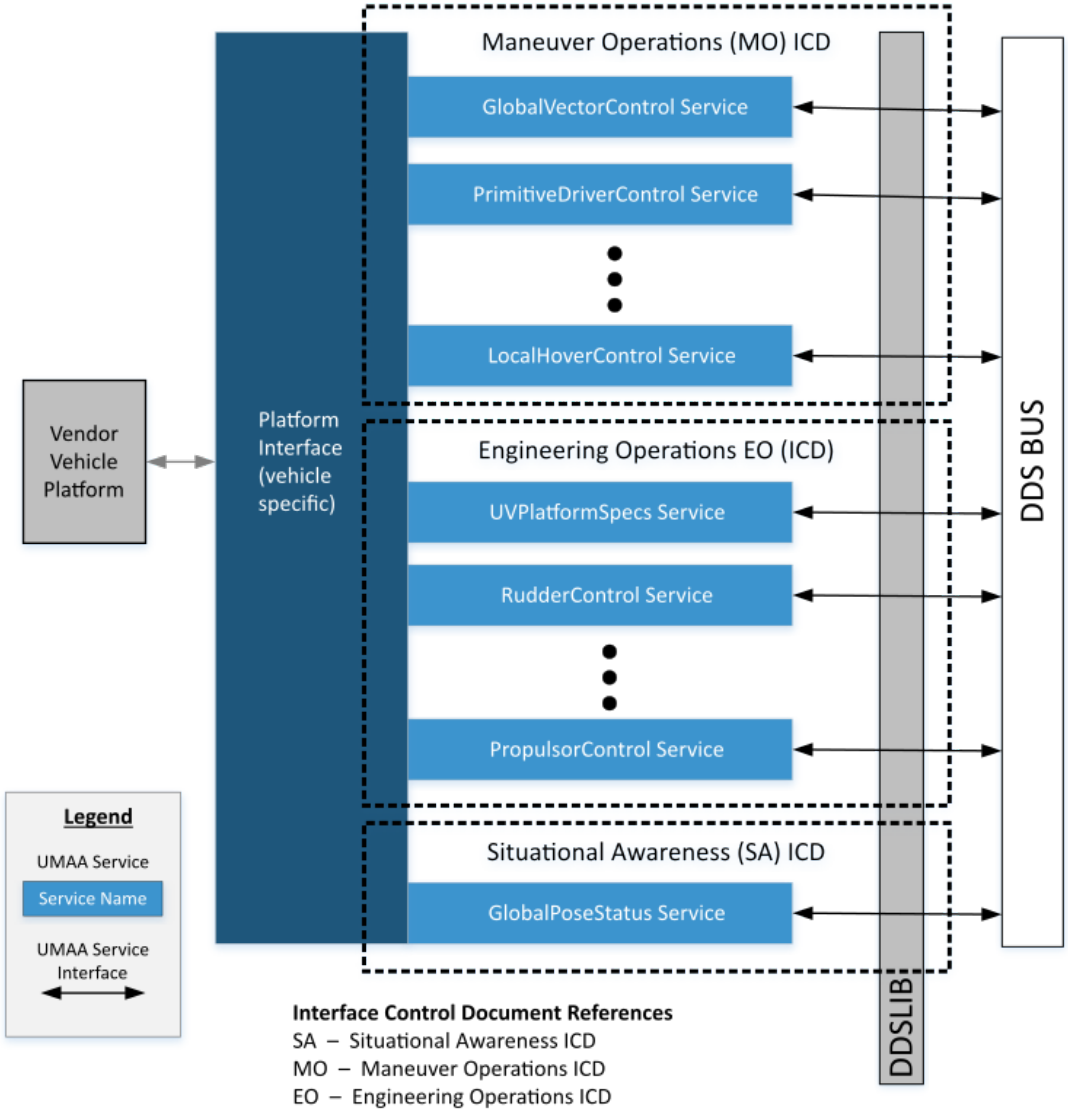

Figure 2: UMAA Services and Interfaces Example

1.3 Document Organization

This interface control document is organized as follows:

Section 1 – Scope: A brief purview of this document

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

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

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

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

Section 6 – Maneuver Operations (MO) 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.

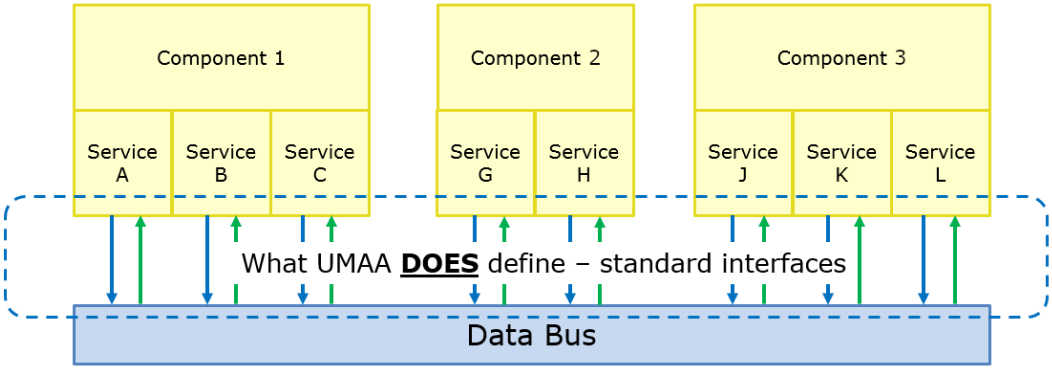

Figure 3: Services and Interfaces Exposed on the UMAA Data Bus

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).

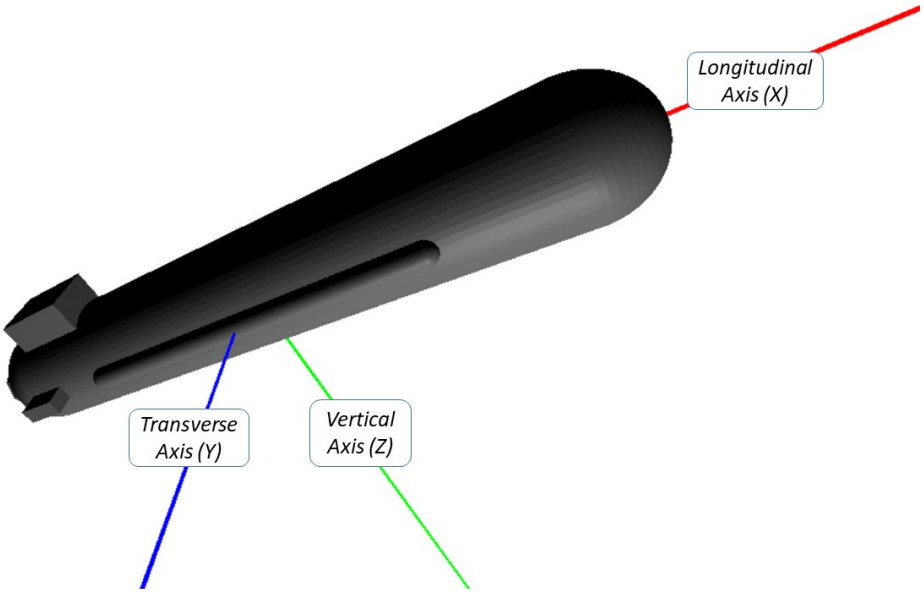

Figure 4: Given a vehicle in arbitrary orientation

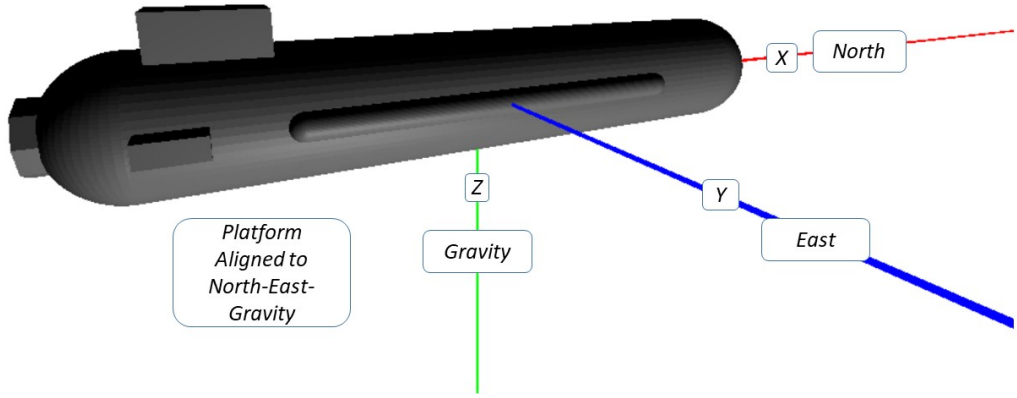

Figure 5: Align the vehicle with the reference frame axes

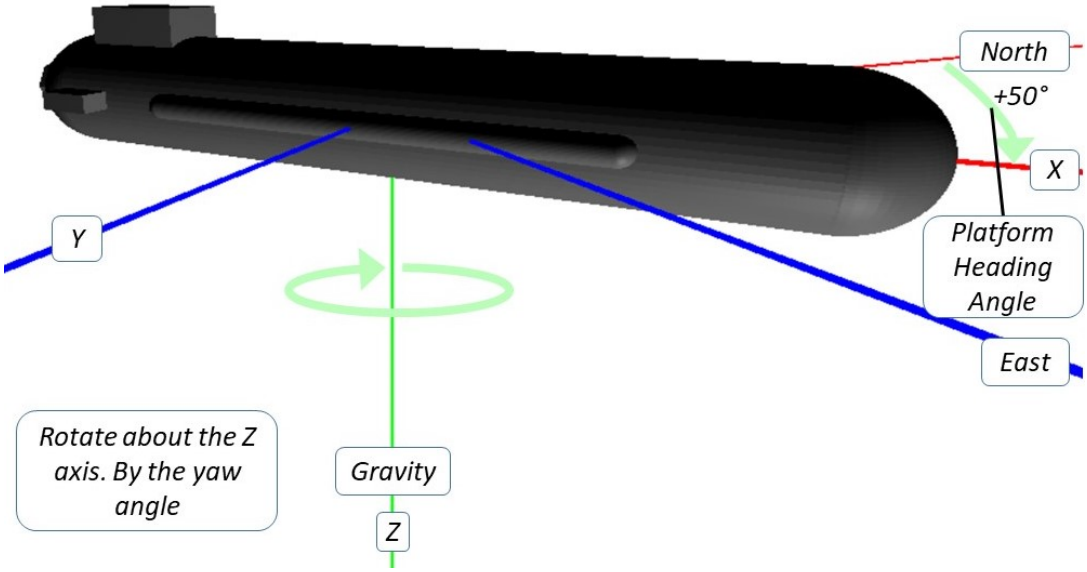

Figure 6: Rotate the vehicle by the Yaw angle

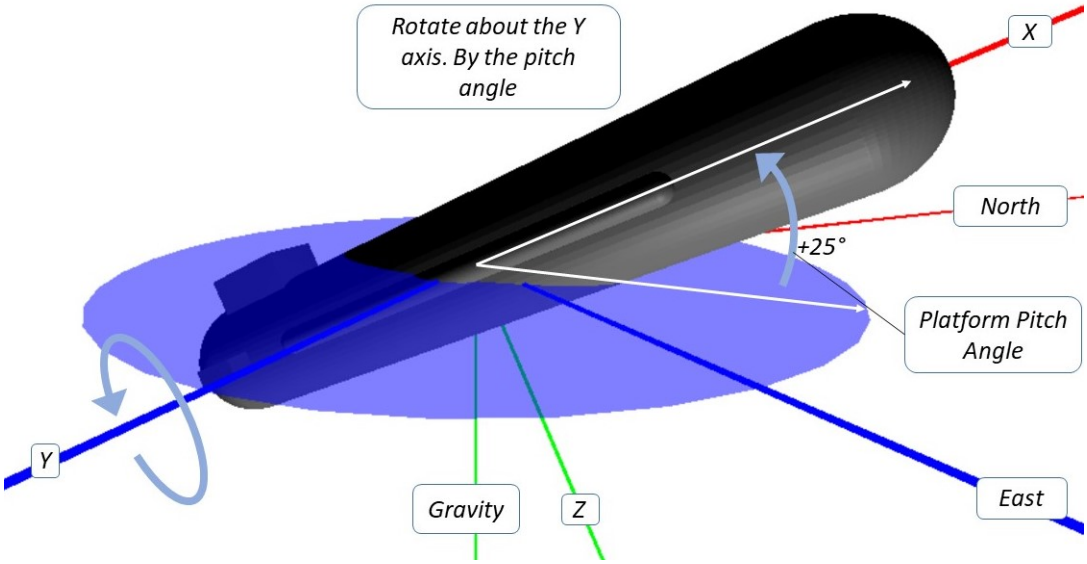

Figure 7: Rotate the vehicle by the Pitch angle

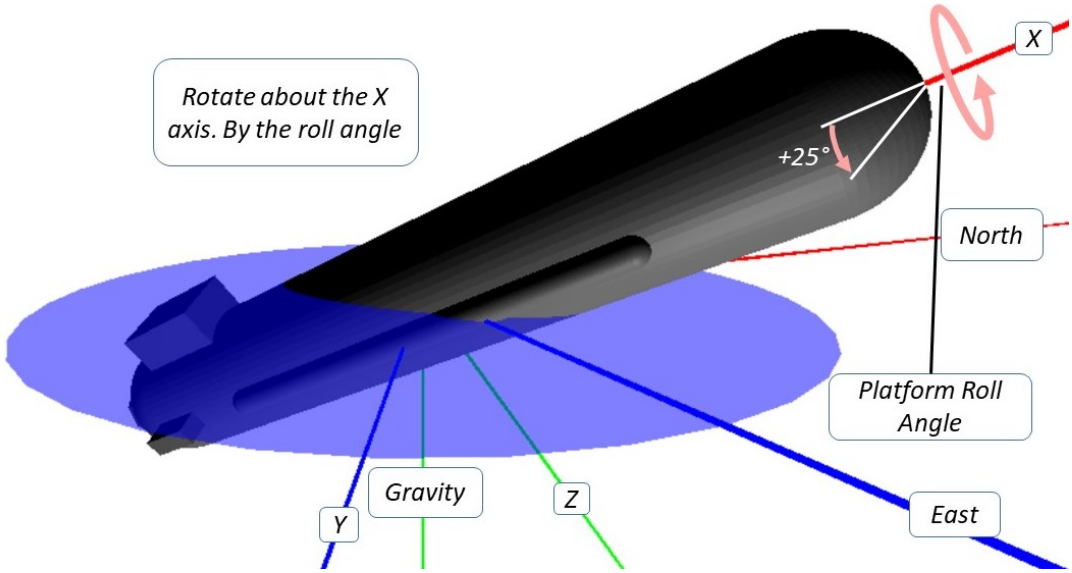

Figure 8: Rotate the vehicle by the Roll angle

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).

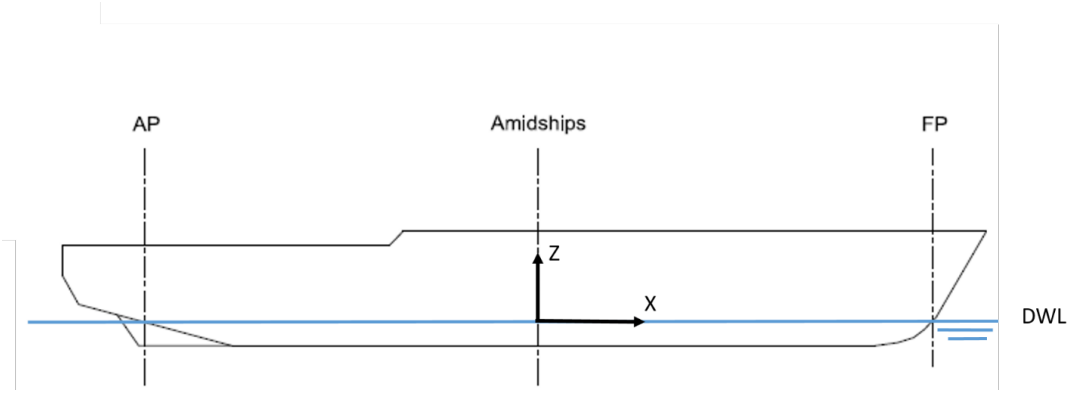

Figure 9: Origin location on a USV as example

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

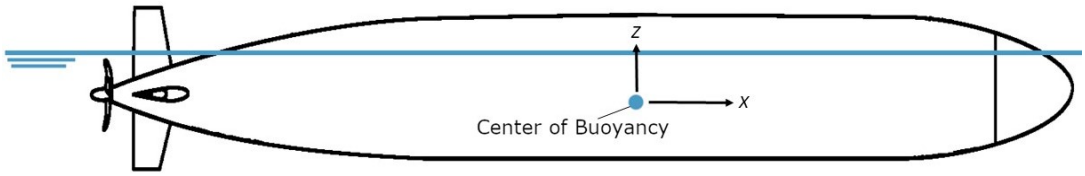

Figure 10: Origin location on a UUV as example

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.

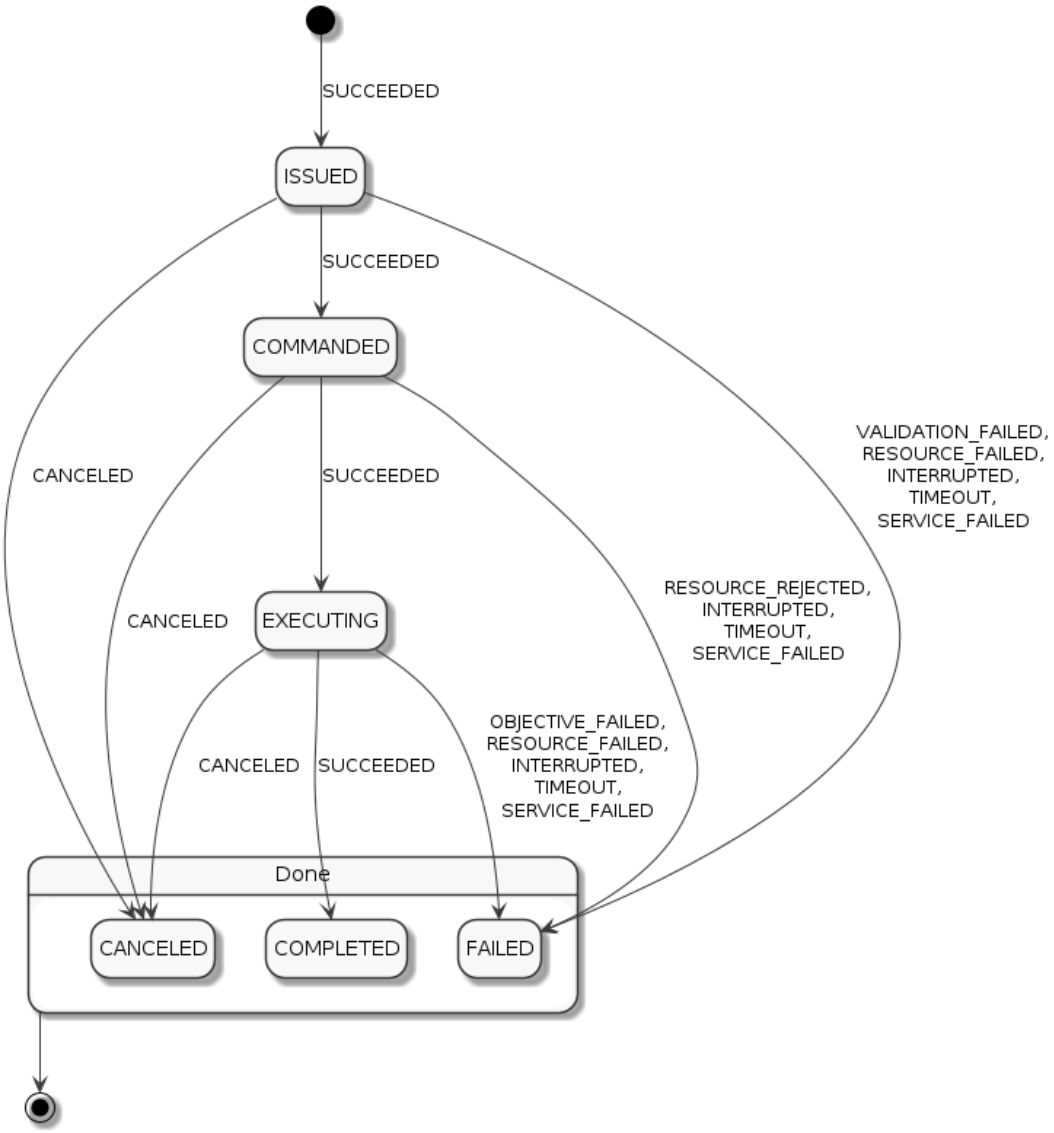

Figure 11: The state transitions of the `commandStatus` as commands are processed. Labels on the arrows represent valid `commandStatusReason` values for each transition.

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.

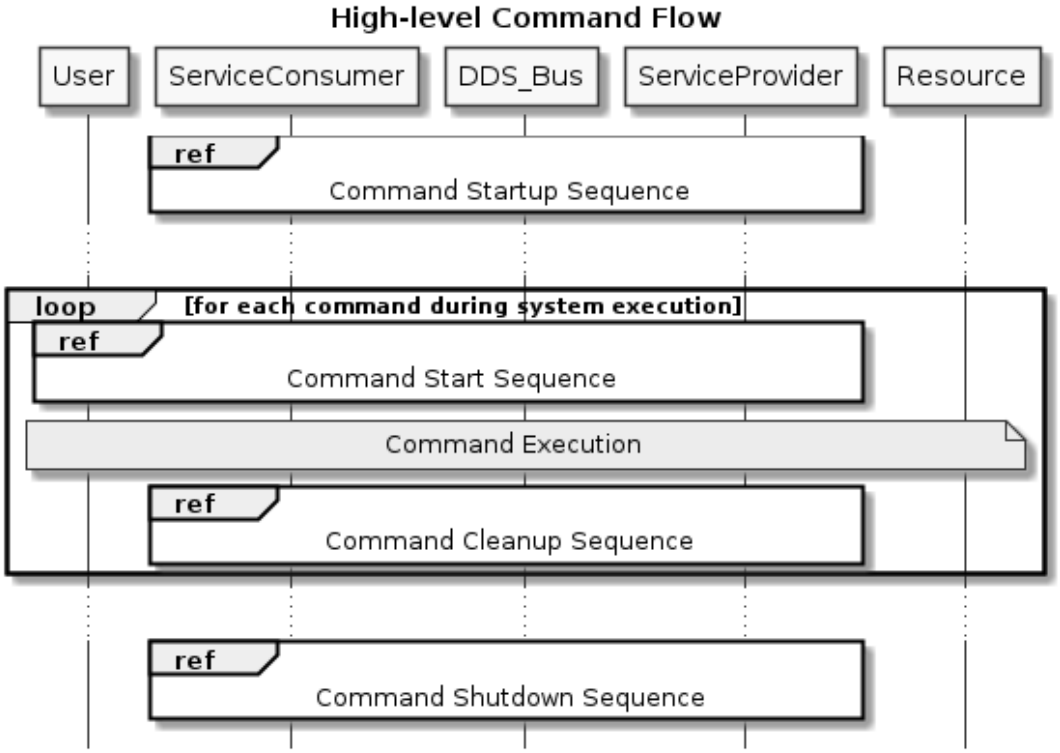

Figure 12: The sequence diagram for the high-level description of a command exeuction.

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.

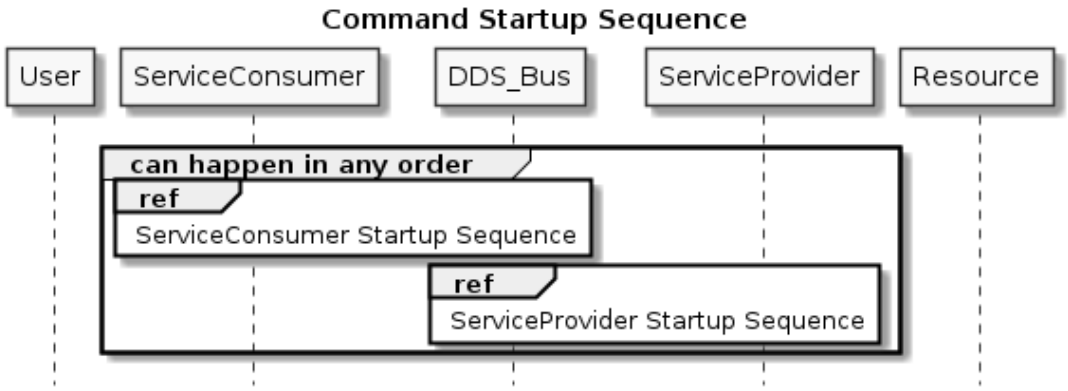

Figure 13: The sequence diagram for command startup.

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.

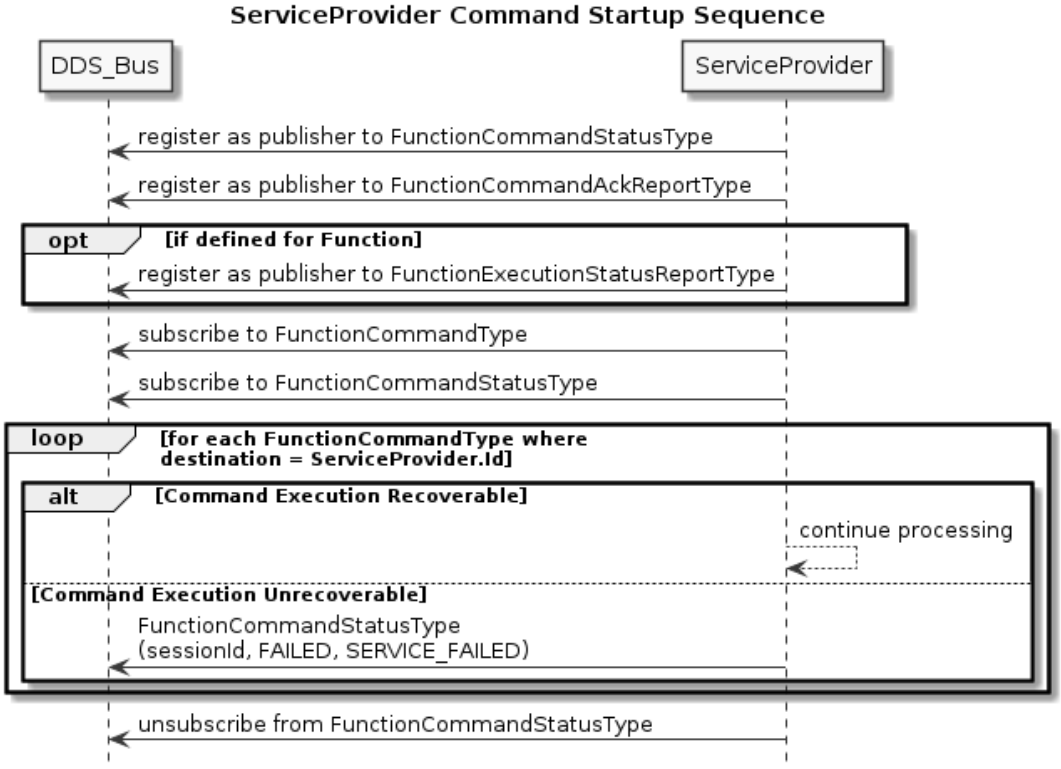

Figure 14: The sequence diagram for command startup for Service Providers.

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.

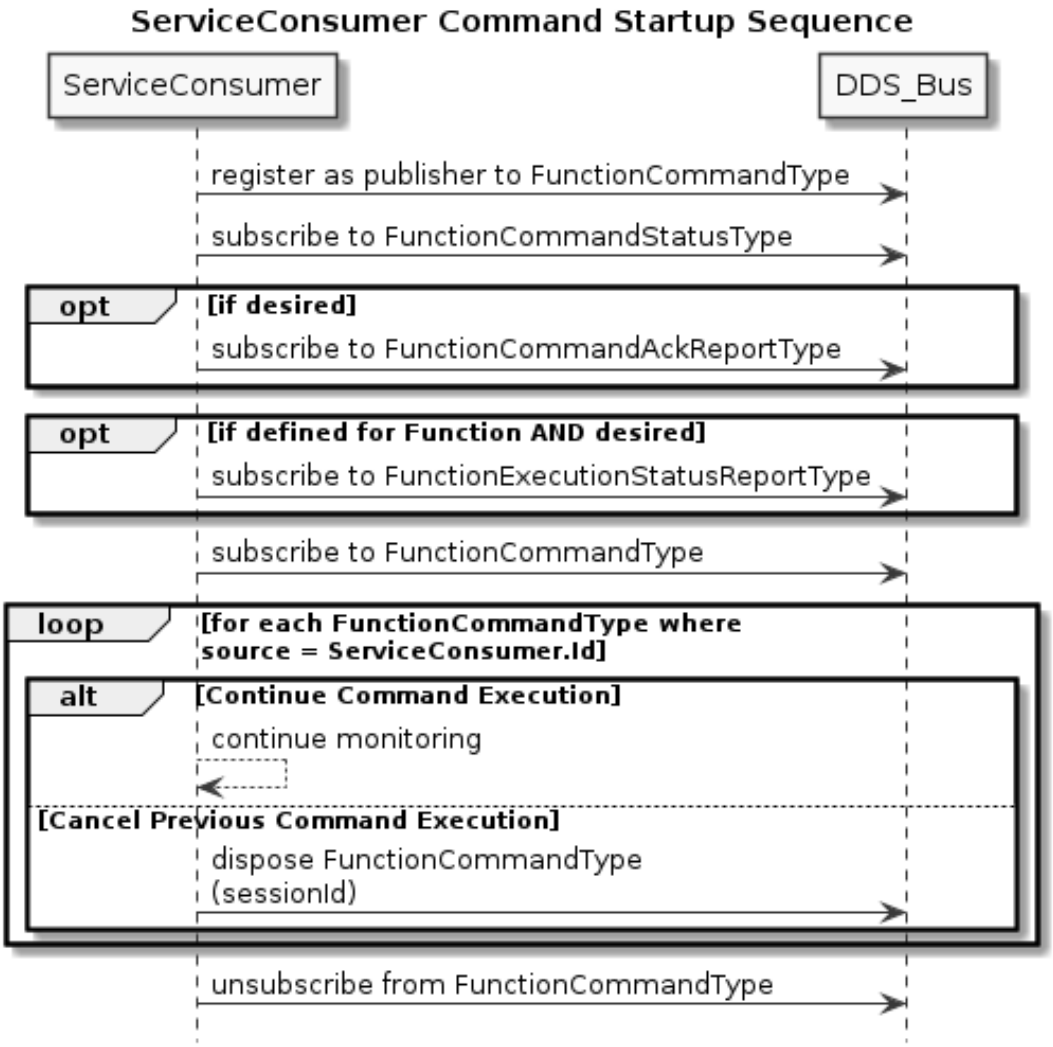

Figure 15: The sequence diagram for command startup for Service Consumers.

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).

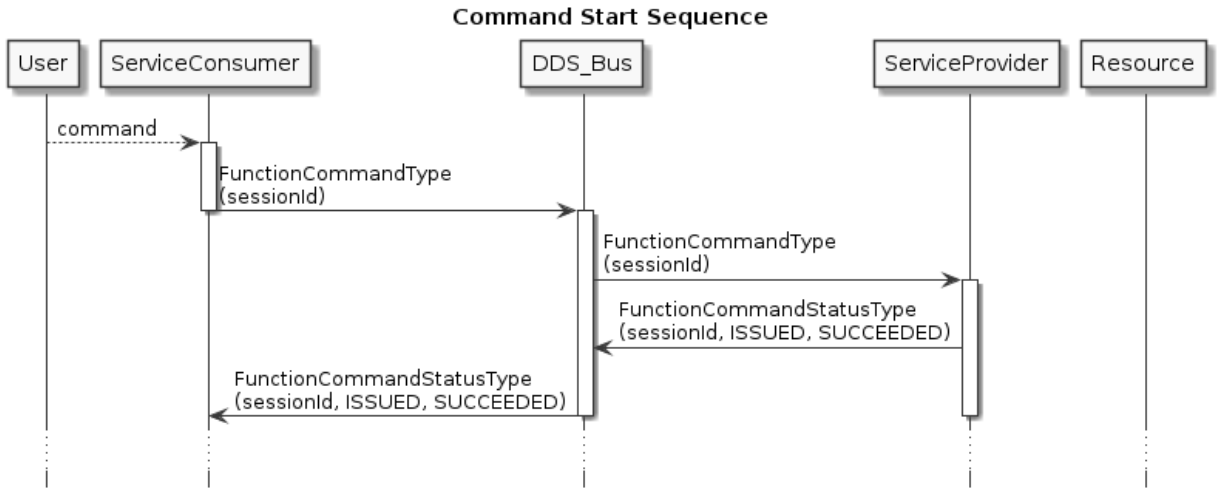

Figure 16: The sequence diagram for the start of a command execution.

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.

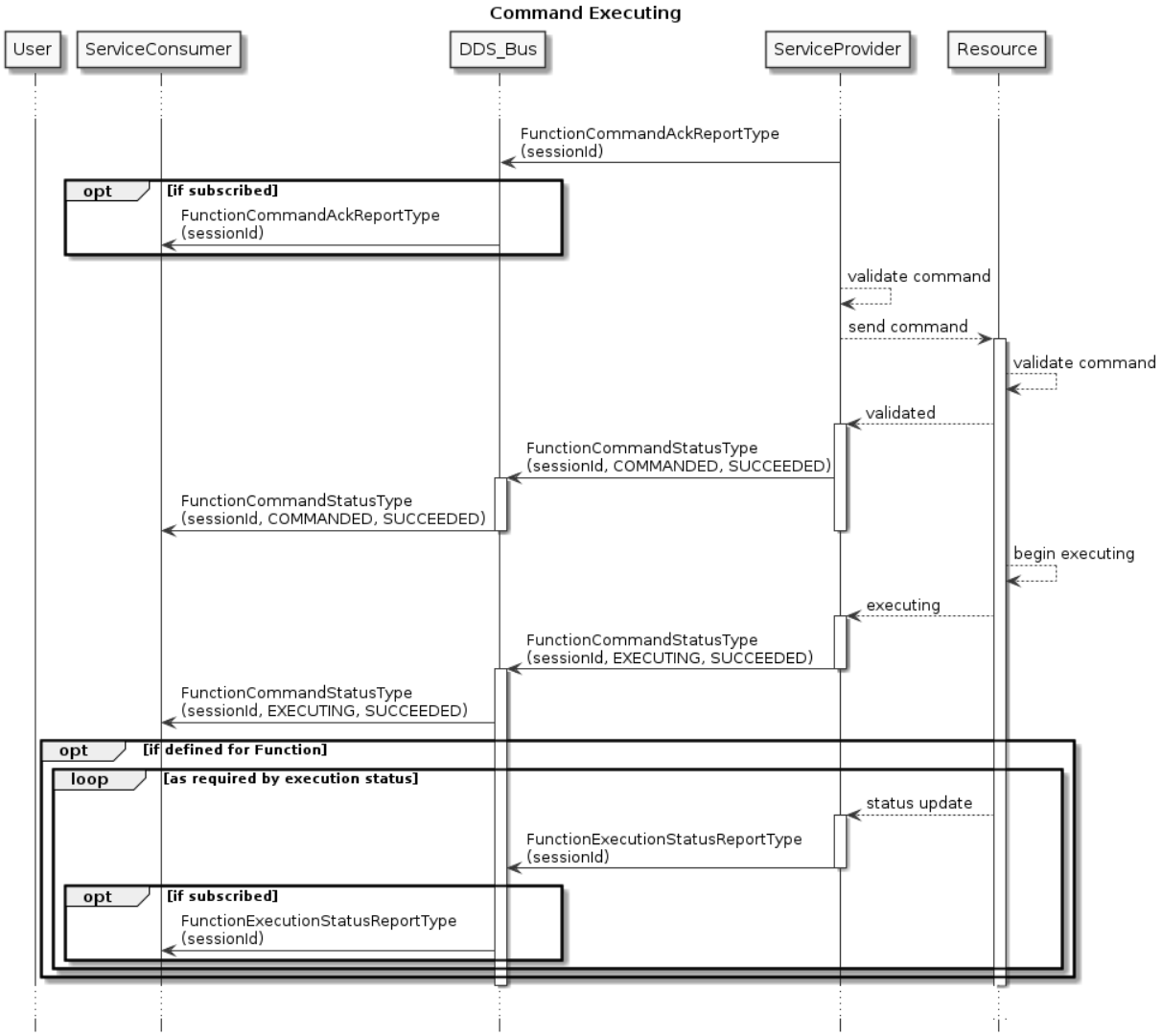

Figure 17: The beginning sequence diagram for a command execution.

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.

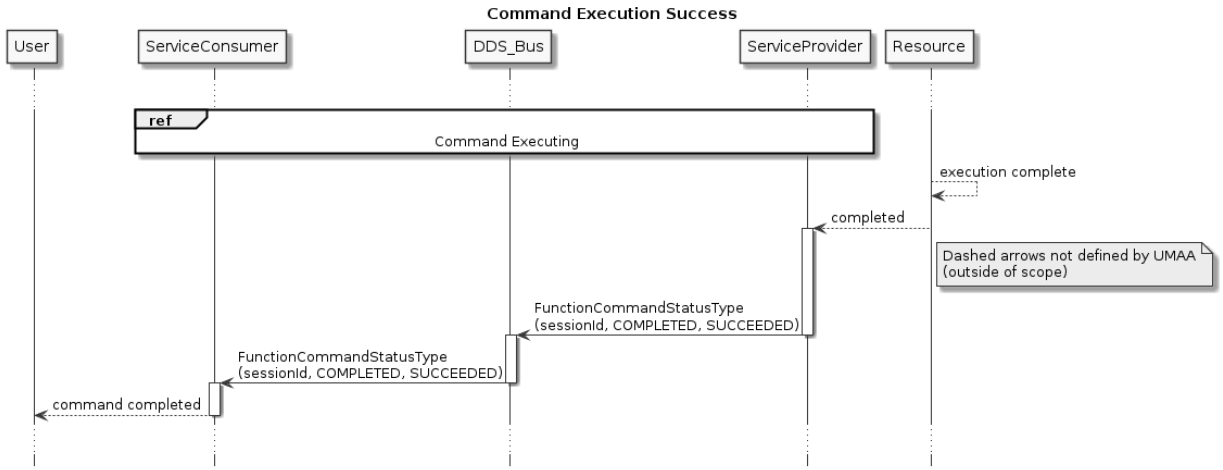

Figure 18: The sequence diagram for a command that completes successfully.

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.

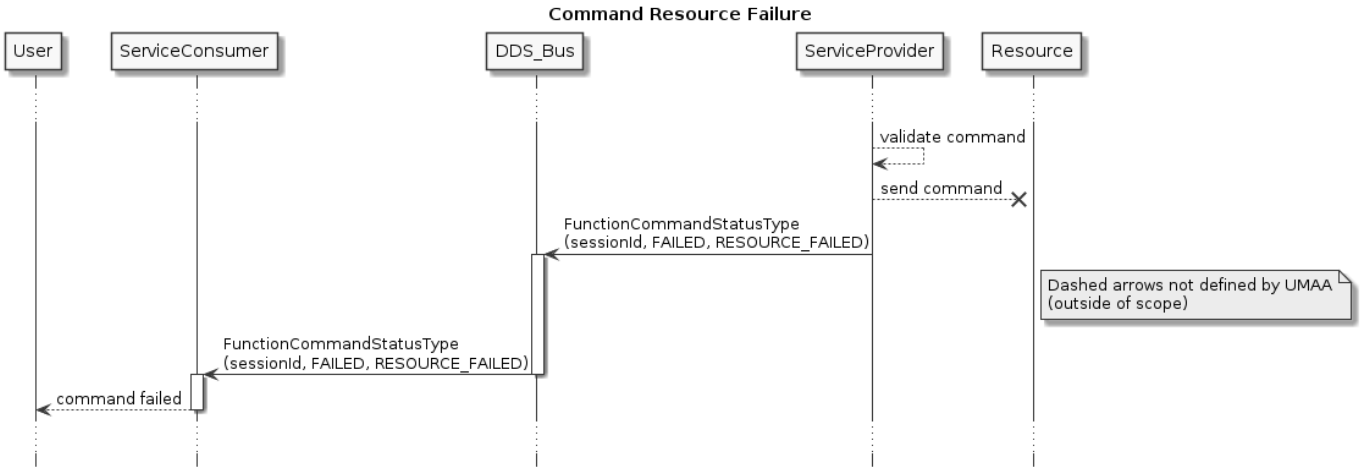

Figure 19: The sequence diagram for a command that fails due to Resource failure.

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.

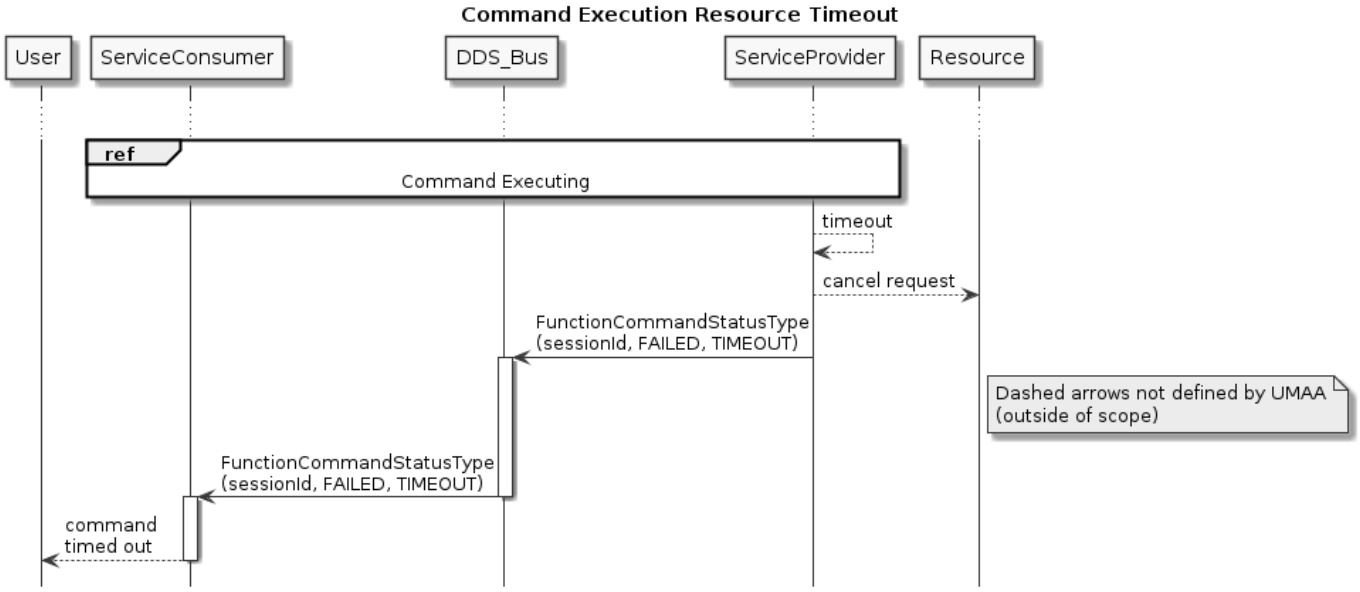

Figure 20: The sequence diagram for a command that times out before completing.

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.

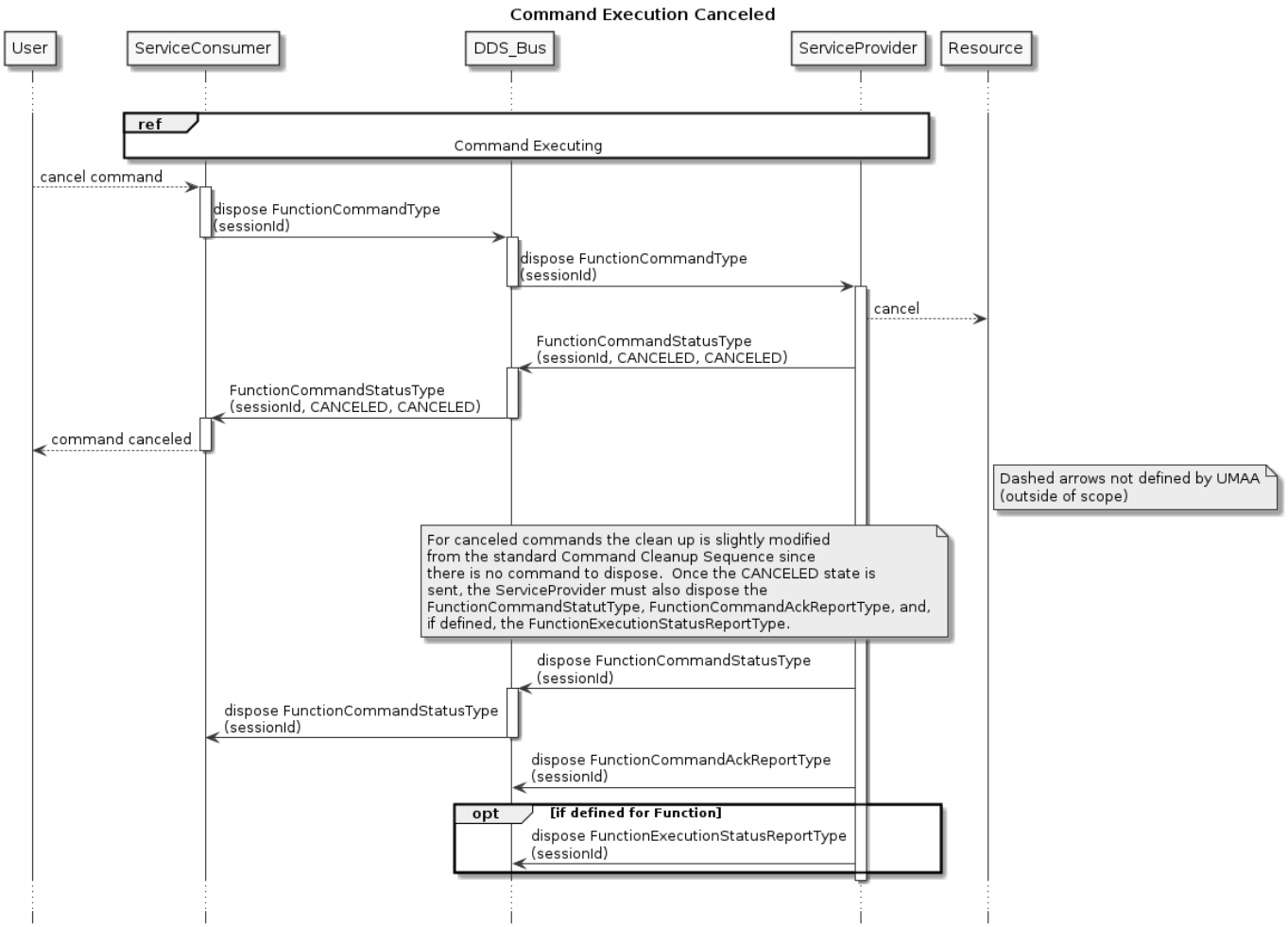

Figure 21: The sequence diagram for a command that is canceled by the Service Consumer before the Service Provider is able to complete it.

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.

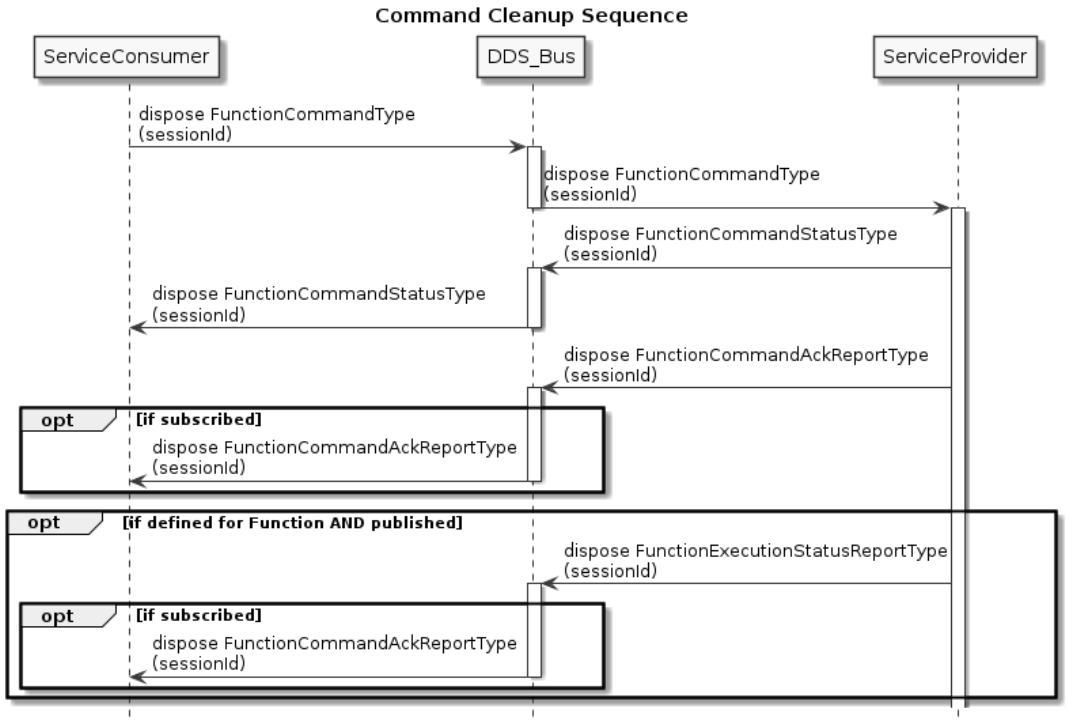

Figure 22: The sequence diagram showing cleanup of the bus when a command has been completed and the Service Consumer no longer wishes to maintain the commanded state.

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.

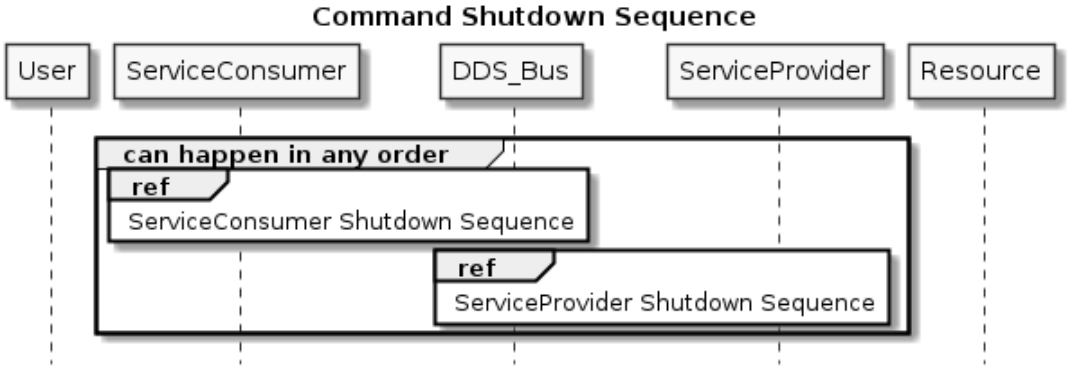

Figure 23: The sequence diagram for command shutdown.

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.

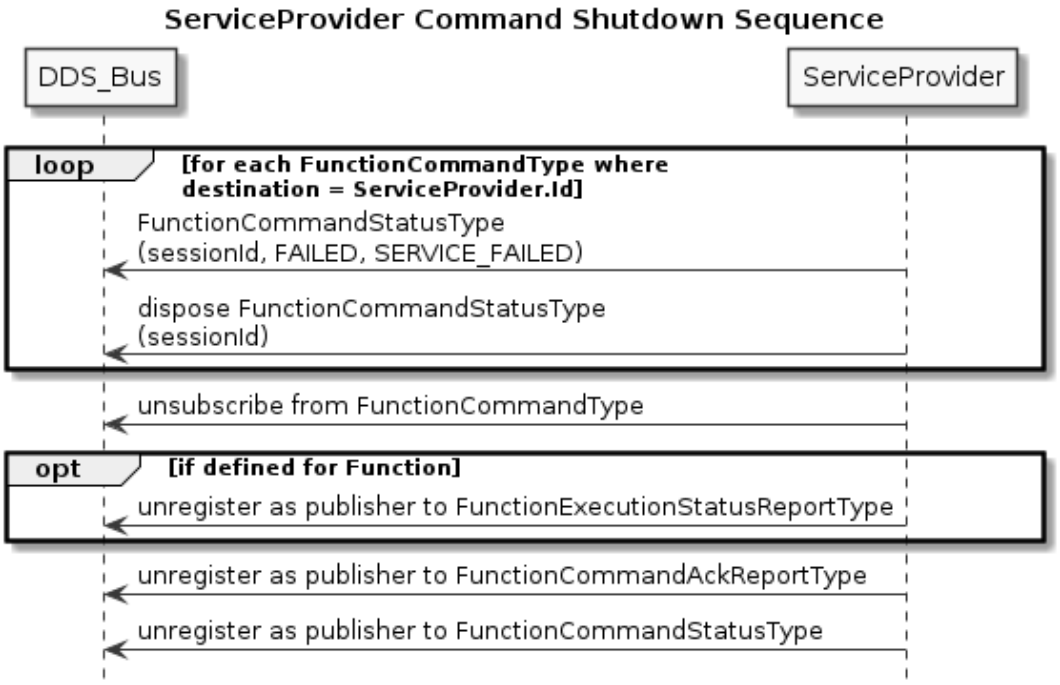

Figure 24: The sequence diagram for command shutdown for Service Providers.

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.

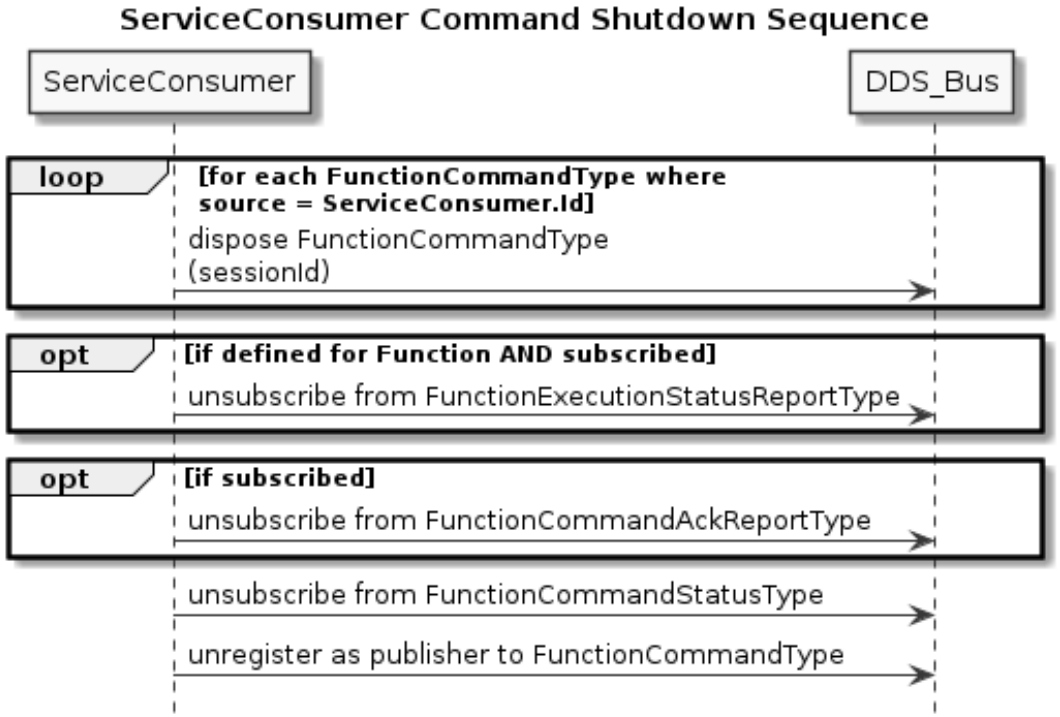

Figure 25: The sequence diagram for command shutdown for Service Consumers.

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.

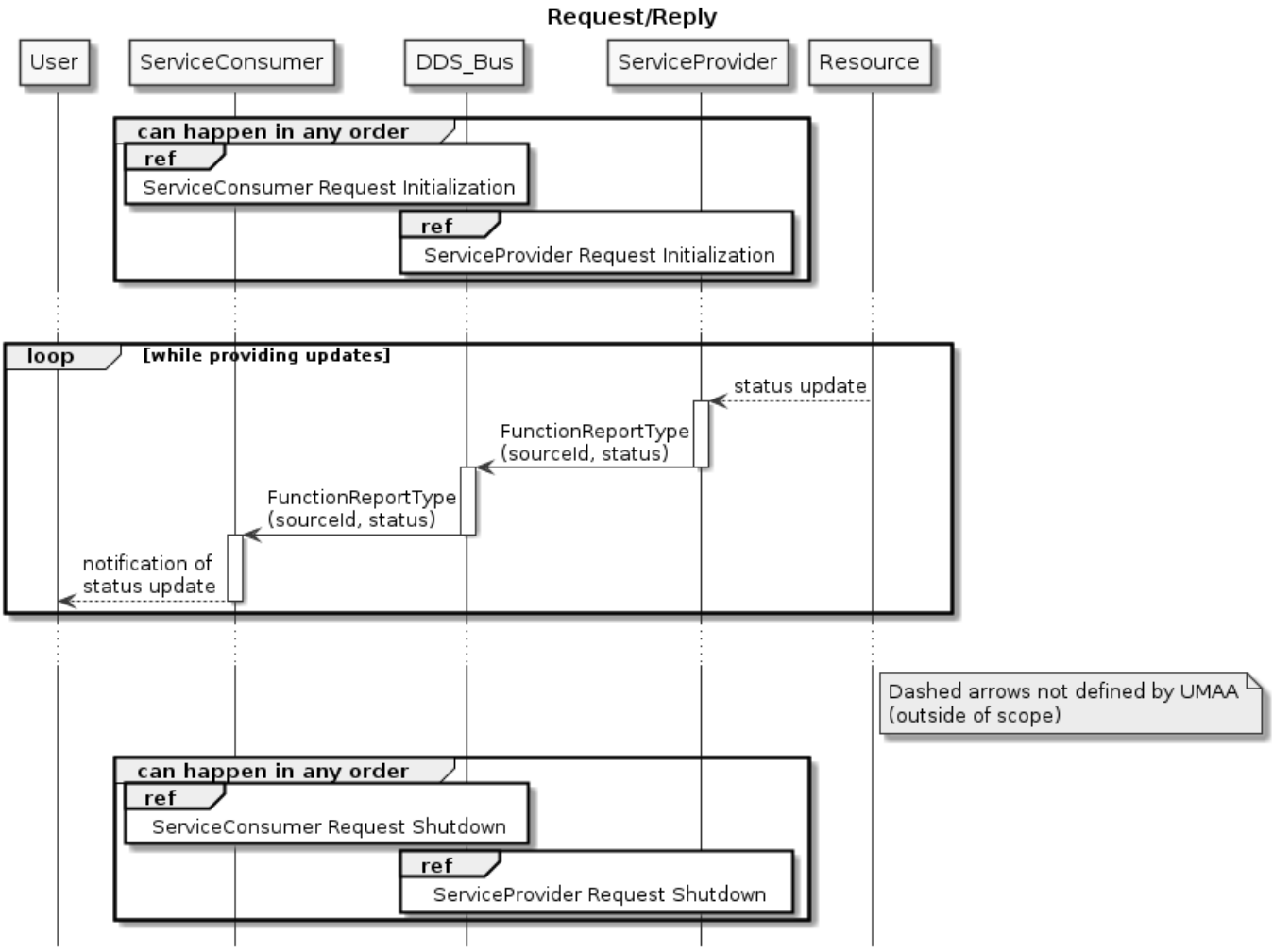

Figure 26: The sequence diagram for a request/reply for report data that does not require any specific query data.

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.

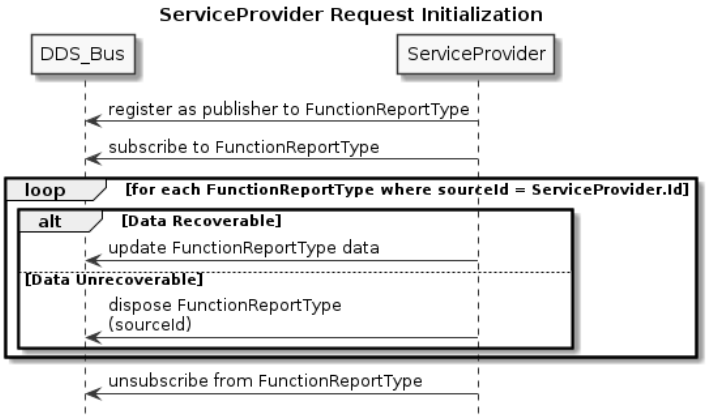

Figure 27: The sequence diagram for initialization of a Service Provider to provide FunctionReportTypes.

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.

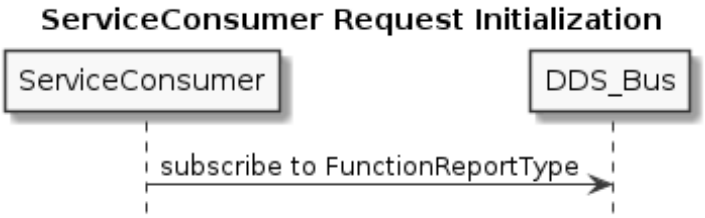

Figure 28: The sequence diagram for initialization of a Service Consumer to request FunctionReportTypes.

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.

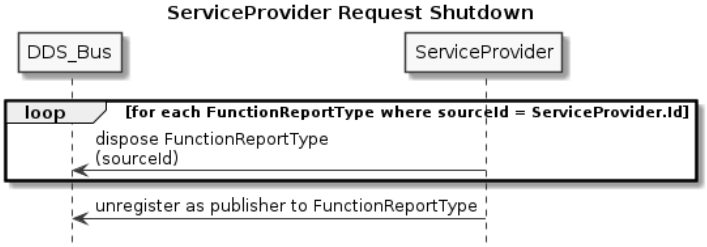

Figure 29: The sequence diagram for shutdown of a Service Provider.

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

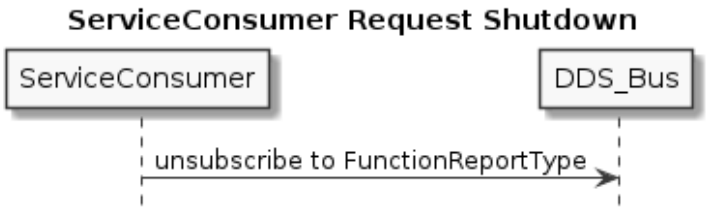

Figure 30: The sequence diagram for shutdown of a Service Consumer.

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 Maneuver Operations (MO) 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.

A standard feature of the maneuver operations driver services is that a new driving control command to a service overrides the previous driver command to that service.

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 ContactManeuverInfluenceStatus

The purpose of this service is to provide the influence of contacts on the maneuvering an unmanned platform.

Table 6: ContactManeuverInfluenceStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryContactManeuverInfluence⊕	reportContactManeuverInfluence

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

6.1.1.1 reportContactManeuverInfluence

Description: This operation is a response to retrieve the current influence of contacts on the maneuvering of the unmanned platform.

Namespace: UMAA::MO::ContactManeuverInfluenceStatus

Topic: ContactManeuverInfluenceReport

Data Type: ContactManeuverInfluenceReportType

Table 7: ContactManeuverInfluenceReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
contactID	NumericGUID	An identifier of the contact
maneuver	ContactManeuverInfluenceEnumType	Specifies the maneuvers the unmanned platform is taking in response to a contact

6.1.2 DriverConfig

This service provides the ability to dynamically report bounds on various driving attributes. Whereas specification services are inherent limitations of the vehicle system and component capabilities, driving constraints can set limits based on mission profile. For example, setting a maximum speed below the vehicle's inherent capabilities in order to preserve power, or turn rate limitations to enable a sensor to continue to operate effectively through a slow turn. Nothing prohibits multiple implementations of this service to be implemented as needed.

A consumer of the constraints service may use the information to limit how it commands the Maneuver Operations driving services. It is intended as a means of specifying safety bounds, optimizing performance (e.g. speed x.x gives optimal efficiency), allowing the payload to affect driving to enable sensor performance (e.g. turn rate limit y.y prevents the towed array from being tangled), and limiting any other use of a vehicles full performance capabilities. In addition, multiple constraints services can be instantiated by different components within a system. It is intended to be informative to higher level of reasoning (e.g. mission management) which may then determine to use the constraint(s) or ignore based on other considerations such as weighing mission importance versus optimizing operation.

Table 8: DriverConfig Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryDriverConfig⊕	reportDriverConfig

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

6.1.2.1 reportDriverConfig

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

Namespace: UMAA::MO::DriverConfig

Topic: DriverConfigReport

Data Type: DriverConfigReportType

Table 9: DriverConfigReportType Message Definition

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

Attribute Name	Attribute Type	Attribute Description
maxAllowableAcceleration†	Acceleration3D_PlatformXYZ	The maximum allowed value for the linear acceleration of the unmanned platform.
maxAllowableAttitudeAcceleration†	OrientationAcceleration3D	The maximum allowed value for the rotational acceleration of the unmanned platform.
maxAllowableElevation†	ElevationType	The maximum allowed value for the elevation of the unmanned platform.
maxAllowableSpeed†	SpeedControlType	The maximum allowed value for the linear speed of the unmanned platform.
maxAllowableTurnRate†	AngleRate	The maximum turn rate.
maxElevationChangeRate†	Speed_BSL_Capability	specifies the maximum rate of change of the vehicle's elevation.
minAllowableAcceleration†	Acceleration3D_PlatformXYZ	The minimum allowed value for the linear acceleration of the unmanned platform.
minAllowableAttitudeAcceleration†	OrientationAcceleration3D	The minimum allowed value for the rotational acceleration of the unmanned platform.
minAllowableElevation†	ElevationType	The minimum allowed value for the elevation of the unmanned platform.
minAllowableSpeed†	SpeedControlType	The minimum allowed value for the linear speed of the unmanned platform.
recommendedAcceleration†	Acceleration3D_PlatformXYZ	The recommended value for the linear acceleration of the unmanned platform.
recommendedAttitudeAcceleration†	OrientationAcceleration3D	The recommended value for the rotational acceleration of the unmanned platform.
recommendedElevation†	ElevationType	The recommended value for the elevation of the unmanned platform.
recommendedElevationChangeRate†	Speed_BSL_Capability	specifies the recommended rate of change of the vehicle's elevation.
recommendedSpeed†	SpeedControlType	The recommended value for the linear speed of the unmanned platform.
recommendedTurnRate†	AngleRate	The recommended turn rate.

6.1.3 GlobalDriftControl

The purpose of this service is to maintain a position within the global reference frame and within a defined drift radius. See figure for reference.

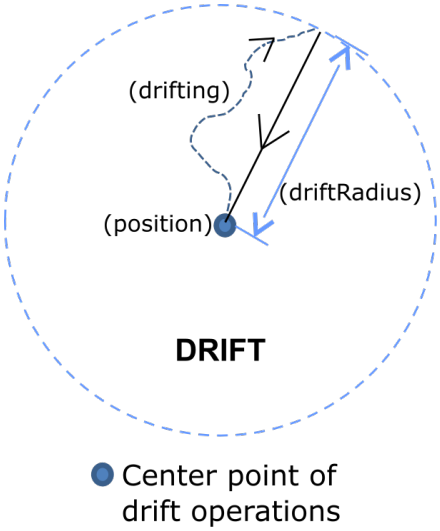

Figure 31: Example Drift Pattern

Table 10: GlobalDriftControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setGlobalDrift	reportGlobalDriftCommandStatus
queryGlobalDriftCommandAck⊕	reportGlobalDriftCommandAck
queryGlobalDriftExecutionStatus⊕	reportGlobalDriftExecutionStatus
cancelGlobalDriftCommand⊕	reportGlobalDriftCancelCommandStatus⊕

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

6.1.3.1 reportGlobalDriftCommandAck

Description: This operation is used to report the commanded values of the position and global drift and/or time that were commanded to the unmanned platform in the global coordinate system.

Namespace: UMAA::MO::GlobalDriftControl

Topic: GlobalDriftCommandAckReport

Data Type: GlobalDriftCommandAckReportType

Table 11: GlobalDriftCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
driftTolerance	Distance	Defines the drift radius that specifies the maximum distance from the reference position the vehicle is allowed to drift.
elevation	ElevationType	Defines the elevation for loitering
elevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded elevation.

Attribute Name	Attribute Type	Attribute Description
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
position†	Position2D	Defines the reference position for loitering. When not specified, means at current location.
positionTolerance	Distance	Defines the capture radius that specifies the minimum distance from the reference position the vehicle must achieve while maneuvering to it.
speed	VariableSpeedControlType	The desired speed to return to the drift position when tolerance exceeded.
speedTolerance	GroundSpeed	The amount of speed error allowed relative to the commanded speed.
transitElevation	ElevationType	The elevation used while driving to the loiter track (USVs must specify 0 as it is a required field).
transitElevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded transit elevation.
transitSpeed	VariableSpeedControlType	The speed at which one drives to the drift position.
transitSpeedTolerance	GroundSpeed	The amount of speed offset allowed relative to the commanded transit speed.

6.1.3.2 reportGlobalDriftCommandStatus

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

Namespace: UMAA::MO::GlobalDriftControl

Topic: GlobalDriftCommandStatus

Data Type: GlobalDriftCommandStatusType

Table 12: GlobalDriftCommandStatusType Message Definition

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

6.1.3.3 reportGlobalDriftExecutionStatus

Description: This operation is used to report the current state of the unmanned platform drift in the global coordinate system.

Namespace: UMAA::MO::GlobalDriftControl

Topic: GlobalDriftExecutionStatusReport

Data Type: GlobalDriftExecutionStatusReportType

Table 13: GlobalDriftExecutionStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
distanceFromReference	Distance	Defines the distance from the reference position.
globalDriftState	GlobalDriftStateType	Defines the state of the global drift.
timeDriftAchieved	DateTime	Defines the absolute time at which loiter is estimated to be achieved or was actually first achieved
timeDriftCompleted†	DateTime	Defines the absolute time at which the loiter is estimated to be completed (optional in case duration is forever)

6.1.3.4 setGlobalDrift

Description: This operation is used to set the desired position in the global coordinate system given the specified global drift and/or time. If the command attributes do not specify a determinate end of execution, the consumer must perform a "cancel" of the command to initiate the end of command execution.

Namespace: UMAA::MO::GlobalDriftControl

Topic: GlobalDriftCommand

Data Type: GlobalDriftCommandType

Table 14: GlobalDriftCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
driftTolerance	Distance	Defines the drift radius that specifies the maximum distance from the reference position the vehicle is allowed to drift.
elevation	ElevationType	Defines the elevation for loitering
elevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
position†	Position2D	Defines the reference position for loitering. When not specified, means at current location.
positionTolerance	Distance	Defines the capture radius that specifies the minimum distance from the reference position the vehicle must achieve while maneuvering to it.
speed	VariableSpeedControlType	The desired speed to return to the drift position when tolerance exceeded.
speedTolerance	GroundSpeed	The amount of speed error allowed relative to the commanded speed.
transitElevation	ElevationType	The elevation used while driving to the loiter track (USVs must specify 0 as it is a required field).

Attribute Name	Attribute Type	Attribute Description
transitElevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded transit elevation.
transitSpeed	VariableSpeedControlType	The speed at which one drives to the drift position.
transitSpeedTolerance	GroundSpeed	The amount of speed offset allowed relative to the commanded transit speed.

6.1.4 GlobalFigure8Control

Intended to command the platform to loiter about a desired position in the global coordinate frame using a figure 8 loiter pattern.
See figure for reference.

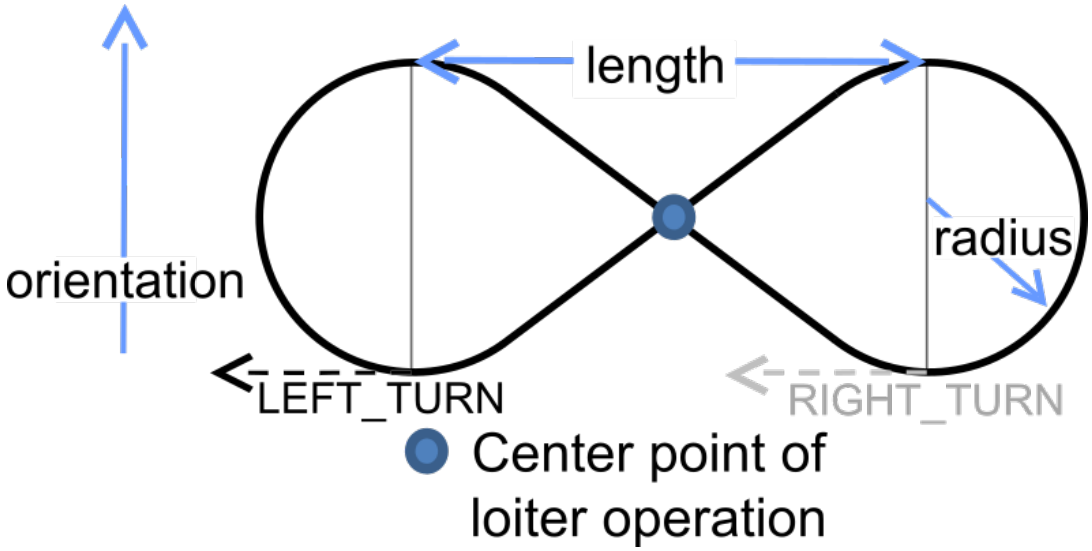

Figure 32: Example Figure 8 Pattern

Table 15: GlobalFigure8Control Operations

Service Requests (Inputs)	Service Responses (Outputs)
setGlobalFigure8	reportGlobalFigure8CommandStatus
queryGlobalFigure8CommandAck⊕	reportGlobalFigure8CommandAck
queryGlobalFigure8ExecutionStatus⊕	reportGlobalFigure8ExecutionStatus
cancelGlobalFigure8Command⊕	reportGlobalFigure8CancelCommandStatus⊕

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

6.1.4.1 reportGlobalFigure8CommandAck

Description: This operation is used to report the commanded values of the position and pattern and/or time that were commanded to the unmanned platform in the global coordinate system.

Namespace: UMAA::MO::GlobalFigure8Control

Topic: GlobalFigure8CommandAckReport

Data Type: GlobalFigure8CommandAckReportType

Table 16: GlobalFigure8CommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
crossTrackTolerance	Distance	The amount of error in position allowed from the pattern being executed.
elevation	ElevationType	The optional elevation used for the unmanned maritime platform.
elevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded loitering elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
length	Distance	Describes the length between the semicircles at either end of the figure 8 the unmanned platform should stay in.
orientation	HeadingType	The orientation of the figure 8, measured perpendicular to the length axis. If not specified, aligns with True North.
position†	Position2D	The desired loiter position (latitude, longitude) of the unmanned platform in the global coordinate system. When not specified, means at current location.
radius	Distance	Describes the radius of the semicircles at either end of the figure 8 the unmanned platform should stay in.
speed	VariableSpeedControlType	The desired speed to loiter of the unmanned platform.
speedTolerance	GroundSpeed	The amount of speed error allowed relative to the commanded pattern execution speed.
transitElevation	ElevationType	The elevation used while driving to the loiter track (USVs must specify 0 as it is a required field).
transitElevationTolerance	Distance	The amount of elevation error allowed relative to the commanded transit elevation.
transitSpeed	VariableSpeedControlType	The speed at which one drives to the loiter track.
transitSpeedTolerance	GroundSpeed	The amount of speed error allowed relative to the transit speed.
turnDirection	WaterTurnDirectionEnumType	The desired turn direction for the loiter of the unmanned platform.

6.1.4.2 reportGlobalFigure8CommandStatus

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

Namespace: UMAA::MO::GlobalFigure8Control

Topic: GlobalFigure8CommandStatus

Data Type: GlobalFigure8CommandStatusType

Table 17: GlobalFigure8CommandStatusType Message Definition

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

6.1.4.3 reportGlobalFigure8ExecutionStatus

Description: This operation is used to report the current position and pattern and/or time that the unmanned platform loitering based in the global coordinate system.

Namespace: UMAA::MO::GlobalFigure8Control

Topic: GlobalFigure8ExecutionStatusReport

Data Type: GlobalFigure8ExecutionStatusReportType

Table 18: GlobalFigure8ExecutionStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
globalFigure8State	GlobalFigure8StateType	Defines the state of the global figure 8.
timePatternAchieved	DateTime	The absolute time at which the loiter pattern is estimated to be achieved or was actually first achieved.
timePatternCompleted†	DateTime	The absolute time at which the loiter pattern is estimated to be completed.

6.1.4.4 setGlobalFigure8

Description: This operation is used to set the desired position in the global coordinate system given the specified loiter pattern and/or time. If the command attributes do not specify a determinate end of execution, the consumer must perform a "cancel" of the command to initiate the end of command execution.

Namespace: UMAA::MO::GlobalFigure8Control

Topic: GlobalFigure8Command

Data Type: GlobalFigure8CommandType

Table 19: GlobalFigure8CommandType Message Definition

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

Attribute Name	Attribute Type	Attribute Description
crossTrackTolerance	Distance	The amount of error in position allowed from the pattern being executed.
elevation	ElevationType	The optional elevation used for the unmanned maritime platform.
elevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded loitering elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
length	Distance	Describes the length between the semicircles at either end of the figure 8 the unmanned platform should stay in.
orientation	HeadingType	The orientation of the figure 8, measured perpendicular to the length axis. If not specified, aligns with True North.
position†	Position2D	The desired loiter position (latitude, longitude) of the unmanned platform in the global coordinate system. When not specified, means at current location.
radius	Distance	Describes the radius of the semicircles at either end of the figure 8 the unmanned platform should stay in.
speed	VariableSpeedControlType	The desired speed to loiter of the unmanned platform.
speedTolerance	GroundSpeed	The amount of speed error allowed relative to the commanded pattern execution speed.
transitElevation	ElevationType	The elevation used while driving to the loiter track (USVs must specify 0 as it is a required field).
transitElevationTolerance	Distance	The amount of elevation error allowed relative to the commanded transit elevation.
transitSpeed	VariableSpeedControlType	The speed at which one drives to the loiter track.
transitSpeedTolerance	GroundSpeed	The amount of speed error allowed relative to the transit speed.
turnDirection	WaterTurnDirectionEnumType	The desired turn direction for the loiter of the unmanned platform.

6.1.5 GlobalHoverControl

The function of this service is to command the platform to hover in a desired position in the global coordinate frame.

Table 20: GlobalHoverControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setGlobalHover	reportGlobalHoverCommandStatus
queryGlobalHoverCommandAck⊕	reportGlobalHoverCommandAck
queryGlobalHoverExecutionStatus⊕	reportGlobalHoverExecutionStatus
cancelGlobalHoverCommand⊕	reportGlobalHoverCancelCommandStatus⊕

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

6.1.5.1 reportGlobalHoverCommandAck

Description: This operation is used to report the commanded values of the position or time that was commanded to the unmanned platform in the global coordinate system.

Namespace: UMAA::MO::GlobalHoverControl

Topic: GlobalHoverCommandAckReport

Data Type: GlobalHoverCommandAckReportType

Table 21: GlobalHoverCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
controlPriority	HoverKindEnumType	The desired priority to hover at the specified point.
elevation	ElevationType	The optional elevation used for the unmanned maritime platform.
elevationTolerance	Distance	The amount of elevation error allowed relative to the commanded hover elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
heading†	HeadingType	Defines the heading that the vehicle must maintain for hovering.
headingTolerance	RelativeAngle	The amount of angle error allowed relative to the commanded heading.
position†	Position2D	The desired hover position (latitude, longitude) of the unmanned platform in the global coordinate system. When not specified, means at current location.
positionTolerance	Distance	The amount of distance error (radius) allowed relative to the commanded position.
transitElevation	ElevationType	The elevation used while driving to the hover location (USVs must specify 0 as it is a required field).
transitElevationTolerance	Distance	The amount of elevation error allowed relative to the commanded transit elevation.
transitSpeed	VariableSpeedControlType	The speed at which one drives to the hover location.
transitSpeedTolerance	GroundSpeed	The amount of speed error allowed relative to the transit speed.

6.1.5.2 reportGlobalHoverCommandStatus

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

Namespace: UMAA::MO::GlobalHoverControl

Topic: GlobalHoverCommandStatus

Data Type: GlobalHoverCommandStatusType

Table 22: GlobalHoverCommandStatusType Message Definition

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

6.1.5.3 reportGlobalHoverExecutionStatus

Description: This operation is used to report the current position or time that the unmanned platform was hovering based in the global coordinate system.

Namespace: UMAA::MO::GlobalHoverControl

Topic:

Data Type: GlobalHoverExecutionStatusReport

Table 23: GlobalHoverExecutionStatusReport Message Definition

Attribute Name	Attribute Type	Attribute Description
globalHoverState	GlobalHoverStateType	Defines the state of the global hover.
timeHoverAchieved	DateTime	The absolute time at which hover is estimated to be achieved or was actually first achieved.
timeHoverCompleted†	DateTime	The absolute time at which the hover is estimated to be completed (optional in case duration is forever).

6.1.5.4 setGlobalHover

Description: This operation is used to set the desired hover position in the global coordinate system given the desired location and/or time. If the command attributes do not specify a determinate end of execution, the consumer must perform a "cancel" of the command to initiate the end of command execution.

Namespace: UMAA::MO::GlobalHoverControl

Topic: GlobalHoverCommand

Data Type: GlobalHoverCommandType

Table 24: GlobalHoverCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
controlPriority	HoverKindEnumType	The desired priority to hover at the specified point.

Attribute Name	Attribute Type	Attribute Description
elevation	ElevationType	The optional elevation used for the unmanned maritime platform.
elevationTolerance	Distance	The amount of elevation error allowed relative to the commanded hover elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
heading†	HeadingType	Defines the heading that the vehicle must maintain for hovering.
headingTolerance	RelativeAngle	The amount of angle error allowed relative to the commanded heading.
position†	Position2D	The desired hover position (latitude, longitude) of the unmanned platform in the global coordinate system. When not specified, means at current location.
positionTolerance	Distance	The amount of distance error (radius) allowed relative to the commanded position.
transitElevation	ElevationType	The elevation used while driving to the hover location (USVs must specify 0 as it is a required field).
transitElevationTolerance	Distance	The amount of elevation error allowed relative to the commanded transit elevation.
transitSpeed	VariableSpeedControlType	The speed at which one drives to the hover location.
transitSpeedTolerance	GroundSpeed	The amount of speed error allowed relative to the transit speed.

6.1.6 GlobalRacetrackControl

Intended to command the platform to loiter about a desired position in the global coordinate frame using a specified loiter pattern. The start location on the Racetrack and the path to this start location is system dependent.
See figure for reference.

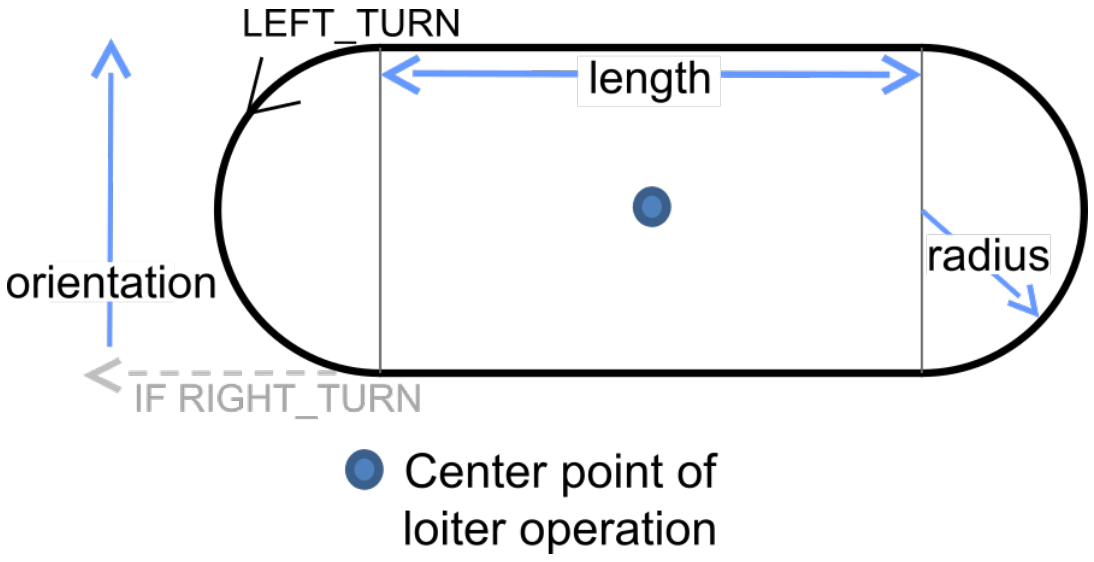

Figure 33: Example Racetrack Pattern

Table 25: GlobalRacetrackControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setGlobalRacetrack	reportGlobalRacetrackCommandStatus
queryGlobalRacetrackCommandAck⊕	reportGlobalRacetrackCommandAck
queryGlobalRacetrackExecutionStatus⊕	reportGlobalRacetrackExecutionStatus
cancelGlobalRacetrackCommand⊕	reportGlobalRacetrackCancelCommandStatus⊕

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

6.1.6.1 reportGlobalRacetrackCommandAck

Description: This operation is used to report the commanded values of the position and pattern and/or time that were commanded to the unmanned platform in the global coordinate system.

Namespace: UMAA::MO::GlobalRacetrackControl

Topic: GlobalRacetrackCommandAckReport

Data Type: GlobalRacetrackCommandAckReportType

Table 26: GlobalRacetrackCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
crossTrackTolerance	Distance	The amount of error in position allowed from the pattern being executed.
elevation	ElevationType	The optional elevation used for the unmanned maritime platform.
elevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded loitering elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
length	Distance	Describes the length between the semicircles at either end of the racetrack the unmanned platform should stay in.
orientation	HeadingType	The orientation of the racetrack, measured perpendicular to the length axis. If not specified, aligns with True North.
position†	Position2D	The desired loiter position (latitude, longitude) of the unmanned platform in the global coordinate system. When not specified, means at current location.
radius	Distance	Describes the radius of the semicircles at either end of the racetrack the unmanned platform should stay in.
speed	VariableSpeedControlType	The desired speed to loiter of the unmanned platform.
speedTolerance	GroundSpeed	The amount of speed error allowed relative to the commanded pattern execution speed.
transitElevation	ElevationType	The elevation used while driving to the loiter track (USVs must specify 0 as it is a required field).

Attribute Name	Attribute Type	Attribute Description
transitElevationTolerance	Distance	The amount of elevation error allowed relative to the commanded transit elevation.
transitSpeed	VariableSpeedControlType	The speed at which one drives to the loiter track.
transitSpeedTolerance	GroundSpeed	The amount of speed error allowed relative to the transit speed.
turnDirection	WaterTurnDirectionEnumType	The desired turn direction for the loiter of the unmanned platform.

6.1.6.2 reportGlobalRacetrackCommandStatus

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

Namespace: UMAA::MO::GlobalRacetrackControl

Topic: GlobalRacetrackCommandStatus

Data Type: GlobalRacetrackCommandStatusType

Table 27: GlobalRacetrackCommandStatusType Message Definition

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

6.1.6.3 reportGlobalRacetrackExecutionStatus

Description: This operation is used to report the current position and pattern and/or time that the unmanned platform loitering based in the global coordinate system.

Namespace: UMAA::MO::GlobalRacetrackControl

Topic: GlobalRacetrackExecutionStatusReport

Data Type: GlobalRacetrackExecutionStatusReportType

Table 28: GlobalRacetrackExecutionStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
globalRacetrackState	GlobalRacetrackStateType	Defines the state of the global racetrack.
timePatternAchieved	DateTime	The absolute time at which the loiter pattern is estimated to be achieved or was actually first achieved.
timePatternCompleted†	DateTime	The absolute time at which the loiter pattern is estimated to be completed.

6.1.6.4 setGlobalRacetrack

Description: This operation is used to set the desired position in the global coordinate system given the specified loiter pattern and/or time. If the command attributes do not specify a determinate end of execution, the consumer must perform a "cancel" of the command to initiate the end of command execution.

Namespace: UMAA::MO::GlobalRacetrackControl

Topic: GlobalRacetrackCommand

Data Type: GlobalRacetrackCommandType

Table 29: GlobalRacetrackCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
	Additional fields included from UMAA::UMAACommand	
crossTrackTolerance	Distance	The amount of error in position allowed from the pattern being executed.
elevation	ElevationType	The optional elevation used for the unmanned maritime platform.
elevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded loitering elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
length	Distance	Describes the length between the semicircles at either end of the racetrack the unmanned platform should stay in.
orientation	HeadingType	The orientation of the racetrack, measured perpendicular to the length axis. If not specified, aligns with True North.
position†	Position2D	The desired loiter position (latitude, longitude) of the unmanned platform in the global coordinate system. When not specified, means at current location.
radius	Distance	Describes the radius of the semicircles at either end of the racetrack the unmanned platform should stay in.
speed	VariableSpeedControlType	The desired speed to loiter of the unmanned platform.
speedTolerance	GroundSpeed	The amount of speed error allowed relative to the commanded pattern execution speed.
transitElevation	ElevationType	The elevation used while driving to the loiter track (USVs must specify 0 as it is a required field).
transitElevationTolerance	Distance	The amount of elevation error allowed relative to the commanded transit elevation.
transitSpeed	VariableSpeedControlType	The speed at which one drives to the loiter track.
transitSpeedTolerance	GroundSpeed	The amount of speed error allowed relative to the transit speed.
turnDirection	WaterTurnDirectionEnumType	The desired turn direction for the loiter of the unmanned platform.

6.1.7 GlobalRegularPolygonControl

Intended to command the platform to loiter about a desired position in the global coordinate frame using a regular polygon pattern circumscribed on a circle. The start location on the RegularPolygon and the path to this start location is system dependent.
See figure for reference.

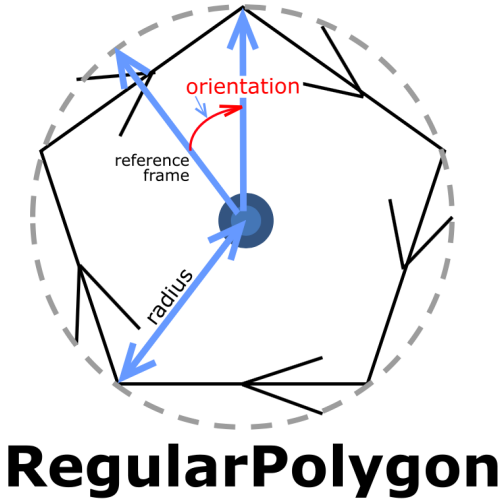

Figure 34: Example Loiter Pattern

Table 30: GlobalRegularPolygonControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setGlobalRegularPolygon	reportGlobalRegularPolygonCommandStatus
queryGlobalRegularPolygonCommandAck⊕	reportGlobalRegularPolygonCommandAck
queryGlobalRegularPolygonExecutionStatus⊕	reportGlobalRegularPolygonExecutionStatus
cancelGlobalRegularPolygonCommand⊕	reportGlobalRegularPolygonCancelCommandStatus⊕

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

6.1.7.1 reportGlobalRegularPolygonCommandAck

Description: This operation is used to report the commanded values of the position and pattern and/or time that were commanded to the unmanned platform in the global coordinate system.

Namespace: UMAA::MO::GlobalRegularPolygonControl

Topic: GlobalRegularPolygonCommandAckReport

Data Type: GlobalRegularPolygonCommandAckReportType

Table 31: GlobalRegularPolygonCommandAckReportType Message Definition

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

Attribute Name	Attribute Type	Attribute Description
crossTrackTolerance	Distance	The amount of error in position allowed from the pattern being executed.
diameter	Distance	The diameter of a circumscribed circle around the polygon.
elevation	ElevationType	The optional elevation used for the unmanned maritime platform.
elevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded loitering elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
numberSides	SidesCount	The number of sides on the polygon.
orientation	HeadingType	The orientation describes the bearing from the reference position of the polygon to one point on the polygon.
position†	Position2D	The desired loiter position (latitude, longitude) of the unmanned platform in the global coordinate system. When not specified, means at current location.
speed	VariableSpeedControlType	The desired speed to loiter of the unmanned platform.
speedTolerance	GroundSpeed	The amount of speed error allowed relative to the commanded pattern execution speed.
transitElevation	ElevationType	The elevation used while driving to the loiter track (USVs must specify 0 as it is a required field).
transitElevationTolerance	Distance	The amount of elevation error allowed relative to the commanded transit elevation.
transitSpeed	VariableSpeedControlType	The speed at which one drives to the loiter track.
transitSpeedTolerance	GroundSpeed	The amount of speed error allowed relative to the transit speed.
turnDirection	WaterTurnDirectionEnumType	The desired turn direction for the loiter of the unmanned platform.

6.1.7.2 reportGlobalRegularPolygonCommandStatus

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

Namespace: UMAA::MO::GlobalRegularPolygonControl

Topic: GlobalRegularPolygonCommandStatus

Data Type: GlobalRegularPolygonCommandStatusType

Table 32: GlobalRegularPolygonCommandStatusType Message Definition

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

6.1.7.3 reportGlobalRegularPolygonExecutionStatus

Description: This operation is used to report the current position and pattern and/or time that the unmanned platform loitering based in the global coordinate system.

Namespace: UMAA::MO::GlobalRegularPolygonControl

Topic: GlobalRegularPolygonExecutionStatusReport

Data Type: GlobalRegularPolygonExecutionStatusReportType

Table 33: GlobalRegularPolygonExecutionStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
globalRegularPolygonState	GlobalRegularPolygonStateType	Defines the state of the global regular polygon.
timePatternAchieved	DateTime	The absolute time at which the loiter pattern is estimated to be achieved or was actually first achieved.
timePatternCompleted†	DateTime	The absolute time at which the loiter pattern is estimated to be completed.

6.1.7.4 setGlobalRegularPolygon

Description: This operation is used to set the desired position in the global coordinate system given the specified loiter pattern and/or time. If the command attributes do not specify a determinate end of execution, the consumer must perform a "cancel" of the command to initiate the end of command execution.

Namespace: UMAA::MO::GlobalRegularPolygonControl

Topic: GlobalRegularPolygonCommand

Data Type: GlobalRegularPolygonCommandType

Table 34: GlobalRegularPolygonCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
crossTrackTolerance	Distance	The amount of error in position allowed from the pattern being executed.
diameter	Distance	The diameter of a circumscribed circle around the polygon.
elevation	ElevationType	The optional elevation used for the unmanned maritime platform.
elevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded loitering elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.

Attribute Name	Attribute Type	Attribute Description
numberSides	SidesCount	The number of sides on the polygon.
orientation	HeadingType	The orientation describes the bearing from the reference position of the polygon to one point on the polygon.
position†	Position2D	The desired loiter position (latitude, longitude) of the unmanned platform in the global coordinate system. When not specified, means at current location.
speed	VariableSpeedControlType	The desired speed to loiter of the unmanned platform.
speedTolerance	GroundSpeed	The amount of speed error allowed relative to the commanded pattern execution speed.
transitElevation	ElevationType	The elevation used while driving to the loiter track (USVs must specify 0 as it is a required field).
transitElevationTolerance	Distance	The amount of elevation error allowed relative to the commanded transit elevation.
transitSpeed	VariableSpeedControlType	The speed at which one drives to the loiter track.
transitSpeedTolerance	GroundSpeed	The amount of speed error allowed relative to the transit speed.
turnDirection	WaterTurnDirectionEnumType	The desired turn direction for the loiter of the unmanned platform.

6.1.8 GlobalVectorControl

The purpose of this service is to command the unmanned platform to maintain a provided speed, North-up course, roll, pitch, and altitude or depth (if supported).

Table 35: GlobalVectorControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setGlobalVector	reportGlobalVectorCommandStatus
queryGlobalVectorCommandAck⊕	reportGlobalVectorCommandAck
queryGlobalVectorExecutionStatus⊕	reportGlobalVectorExecutionStatus
cancelGlobalVectorCommand⊕	reportGlobalVectorCancelCommandStatus⊕

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

6.1.8.1 reportGlobalVectorCommandAck

Description: This operation is used to report the current commanded values of the speed, depth, North-up course, roll, and pitch to an unmanned platform in the global coordinate system.

Namespace: UMAA::MO::GlobalVectorControl

Topic: GlobalVectorCommandAckReport

Data Type: GlobalVectorCommandAckReportType

Table 36: GlobalVectorCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
attitude	OrientationType	The desired reference frame used for vehicle orientation commands.
attitudeTolerance	Orientation3D_Tolerance	The allowable attitude tolerance.
elevation	ElevationType	Specifies the elevation of the vector.
elevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
speed	SpeedControlType	The desired speed of the unmanned platform.
speedTolerance	GroundSpeed	The amount of speed error allowed relative to the commanded speed.

6.1.8.2 reportGlobalVectorCommandStatus

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

Namespace: UMAA::MO::GlobalVectorControl

Topic: GlobalVectorCommandStatus

Data Type: GlobalVectorCommandStatusType

Table 37: GlobalVectorCommandStatusType Message Definition

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

6.1.8.3 reportGlobalVectorExecutionStatus

Description: This operation is used to report the current vector data based in the global coordinate system.

Namespace: UMAA::MO::GlobalVectorControl

Topic: GlobalVectorExecutionStatusReport

Data Type: GlobalVectorExecutionStatusReportType

Table 38: GlobalVectorExecutionStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
attitudeAchieved	BooleanEnumType	When the vector is executing, this indicates that the attitude requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
elevationAchieved	BooleanEnumType	When the vector is executing, this indicates that the elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
speedAchieved	BooleanEnumType	When the vector is executing, this indicates that the speed requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

6.1.8.4 setGlobalVector

Description: This operation is used to command the speed, North-up course, roll, pitch, and altitude or depth of an unmanned platform in the global coordinate system. If the command attributes do not specify a determinate end of execution, the consumer must perform a "cancel" of the command to initiate the end of command execution.

Namespace: UMAA::MO::GlobalVectorControl

Topic: GlobalVectorCommand

Data Type: GlobalVectorCommandType

Table 39: GlobalVectorCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
attitude	OrientationType	The desired reference frame used for vehicle orientation commands.
attitudeTolerance	Orientation3D_Tolerance	The allowable attitude tolerance.
elevation	ElevationType	Specifies the elevation of the vector.
elevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
speed	SpeedControlType	The desired speed of the unmanned platform.
speedTolerance	GroundSpeed	The amount of speed error allowed relative to the commanded speed.

6.1.9 GlobalWaypointControl

The purpose of this service is to move the platform based on the defined single or a list of target waypoints, desired speed, current platform pose, and current velocity state. This service receives data from the Set Global Waypoint message to define its target waypoint (position and orientation of the platform); it also receives data from the Set Travel Speed message to change its speed. On a waypoint failure, the waypoint set is terminated with error. If during waypoint operations, the service cannot satisfy a waypoint, the service will publish a failed status state and the command issuer must determine whether to reissue the remaining waypoints or take some other action. A waypoint is achieved when it is within the waypointTolerance radius of the specified waypoint location.

Table 40: GlobalWaypointControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setGlobalWaypoint	reportGlobalWaypointCommandStatus
queryGlobalWaypointCommandAck⊕	reportGlobalWaypointCommandAck
queryGlobalWaypointExecutionStatus⊕	reportGlobalWaypointExecutionStatus
cancelGlobalWaypointCommand⊕	reportGlobalWaypointCancelCommandStatus⊕
setGlobalWaypointSpeed	reportGlobalWaypointSpeedCommandStatus
queryGlobalWaypointSpeedCommandAck⊕	reportGlobalWaypointSpeedCommandAck
cancelGlobalWaypointSpeedCommand⊕	reportGlobalWaypointSpeedCancelCommandStatus⊕

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

6.1.9.1 reportGlobalWaypointCommandAck

Description: This operation is used to report the commanded values of the waypoint data based in the global coordinate system.

Namespace: UMAA::MO::GlobalWaypointControl

Topic: GlobalWaypointCommandAckReport

Data Type: GlobalWaypointCommandAckReportType

Table 41: GlobalWaypointCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
waypointCount	Count	The total number of waypoint on the series of waypoints.
waypoints	sequence<GlobalWaypointType>	The desired series of waypoints in the global coordinate system.

6.1.9.2 reportGlobalWaypointCommandStatus

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

Namespace: UMAA::MO::GlobalWaypointControl

Topic: GlobalWaypointCommandStatus

Data Type: GlobalWaypointCommandStatusType

Table 42: GlobalWaypointCommandStatusType Message Definition

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

6.1.9.3 reportGlobalWaypointExecutionStatus

Description: This operation is used to report the current waypoint data based in the global coordinate system. All times and distances are provided based on waypoint location not including tolerances.

Namespace: UMAA::MO::GlobalWaypointControl

Topic: GlobalWaypointExecutionStatusReport

Data Type: GlobalWaypointExecutionStatusReportType

Table 43: GlobalWaypointExecutionStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
arrivalTime	DateTime	The arrival time of the end of the route.
crossTrackError	Distance	Defines the current cross track error (only valid if maintainTrack is true)
cumulativeDistance	Distance	Defines the ground distance travel from the start of the route to this point
distanceRemaining	Distance	Defines the amount of distance remaining from a point to the end of the route
distanceToWaypoint	Distance	Defines the remaining distance to the current waypoint
elevationAchieved	BooleanEnumType	When the waypoint is executing, this indicates that the elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
maintainTrack	BooleanEnumType	Indicates whether a track line is to be followed when transiting to the waypoint.
speedAchieved	BooleanEnumType	When the waypoint is executing, this indicates that the speed requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
timeToWaypoint	DateTime	The absolute time at which the waypoint is estimated to be achieved or was actually first achieved.
trackLineAchieved	BooleanEnumType	When the waypoint is executing, this indicates that the track line requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

Attribute Name	Attribute Type	Attribute Description
waypointsRemaining	Count	Defines the remaining number of waypoints, which includes the current waypoint.
waypointID*	NumericGUID	Defines the current waypoint ID.

6.1.9.4 reportGlobalWaypointSpeedCommandAck

Description: This operation is used to report the commanded values of the travel speed of the unmanned platform.

Namespace: UMAA::MO::GlobalWaypointControl

Topic: GlobalWaypointSpeedCommandAckReport

Data Type: GlobalWaypointSpeedCommandAckReportType

Table 44: GlobalWaypointSpeedCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
speed	VariableSpeedControlType	The desired travel speed of the unmanned platform.

6.1.9.5 reportGlobalWaypointSpeedCommandStatus

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

Namespace: UMAA::MO::GlobalWaypointControl

Topic: GlobalWaypointSpeedCommandStatus

Data Type: GlobalWaypointSpeedCommandStatusType

Table 45: GlobalWaypointSpeedCommandStatusType Message Definition

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

6.1.9.6 setGlobalWaypoint

Description: This operation is used to set a single waypoint data based in the global coordinate system.

Namespace: UMAA::MO::GlobalWaypointControl

Topic: GlobalWaypointCommand

Data Type: GlobalWaypointCommandType

Table 46: GlobalWaypointCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
	Additional fields included from UMAA::UMAACommand	
waypointCount	Count	The total number of waypoint on the series of waypoints.
waypoints	sequence<GlobalWaypointType>	The desired series of waypoints in the global coordinate system.

6.1.9.7 setGlobalWaypointSpeed

Description: This operation is used to set the desired travel speed as an absolute or relative value.

Namespace: UMAA::MO::GlobalWaypointControl

Topic: GlobalWaypointSpeedCommand

Data Type: GlobalWaypointSpeedCommandType

Table 47: GlobalWaypointSpeedCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
	Additional fields included from UMAA::UMAACommand	
speed	VariableSpeedControlType	The desired travel speed of the unmanned platform.

6.1.10 LocalDriftControl

The purpose of this service is to maintain a position within the local reference frame and within a defined drift radius. See figure for reference. (Figure 31)

Table 48: LocalDriftControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setLocalDrift	reportLocalDriftCommandStatus
queryLocalDriftCommandAck⊕	reportLocalDriftCommandAck
queryLocalDriftExecutionStatus⊕	reportLocalDriftExecutionStatus
cancelLocalDriftCommand⊕	reportLocalDriftCancelCommandStatus⊕

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

6.1.10.1 reportLocalDriftCommandAck

Description: This operation is used to report the commanded values of the position and pattern and/or time that were commanded to the unmanned platform in the local coordinate system.

Namespace: UMAA::MO::LocalDriftControl

Topic: LocalDriftCommandAckReport

Data Type: LocalDriftCommandAckReportType

Table 49: LocalDriftCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
driftTolerance	Distance	Defines the drift radius that specifies the maximum distance from the reference position the vehicle is allowed to drift.
elevation	ElevationType	Defines the elevation for loitering
elevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
position†	Position2D_PlatformXYZ	Defines the reference position in the system-defined coordinate system for loitering. When not specified, means at current location.
positionTolerance	Distance	Defines the capture radius that specifies the minimum distance from the reference position the vehicle must achieve while maneuvering to it.
speed	VariableSpeedControlType	The desired speed to return to the drift position when tolerance exceeded.
speedTolerance	GroundSpeed	The amount of speed error allowed relative to the commanded speed.
transitElevation	ElevationType	The elevation used while driving to the loiter track (USVs must specify 0 as it is a required field).
transitElevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded transit elevation.
transitSpeed	VariableSpeedControlType	The speed at which one drives to the drift position.
transitSpeedTolerance	GroundSpeed	The amount of speed offset allowed relative to the commanded transit speed.

6.1.10.2 reportLocalDriftCommandStatus

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

Namespace: UMAA::MO::LocalDriftControl

Topic: LocalDriftCommandStatus

Data Type: LocalDriftCommandStatusType

Table 50: LocalDriftCommandStatusType Message Definition

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

6.1.10.3 reportLocalDriftExecutionStatus

Description: This operation is used to report the current state of the unmanned platform drift in the local coordinate system.

Namespace: UMAA::MO::LocalDriftControl

Topic: LocalDriftExecutionStatusReport

Data Type: LocalDriftExecutionStatusReportType

Table 51: LocalDriftExecutionStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
distanceFromReference	Distance	Defines the distance from the reference position.
localDriftState	LocalDriftStateType	Defines the state of the local drift.
timeDriftAchieved	DateTime	Defines the absolute time at which the local drift is estimated to be achieved or was actually first achieved
timeDriftCompleted†	DateTime	Defines the absolute time at which the local drift is estimated to be completed (optional in case duration is forever)

6.1.10.4 setLocalDrift

Description: This operation is used to set the desired position in the local coordinate system given the specified drift pattern and/or time. If the command attributes do not specify a determinate end of execution, the consumer must perform a "cancel" of the command to initiate the end of command execution.

Namespace: UMAA::MO::LocalDriftControl

Topic: LocalDriftCommand

Data Type: LocalDriftCommandType

Table 52: LocalDriftCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
	Additional fields included from UMAA::UMAACommand	
driftTolerance	Distance	Defines the drift radius that specifies the maximum distance from the reference position the vehicle is allowed to drift.
elevation	ElevationType	Defines the elevation for loitering
elevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
position†	Position2D_PlatformXYZ	Defines the reference position in the system-defined coordinate system for loitering. When not specified, means at current location.
positionTolerance	Distance	Defines the capture radius that specifies the minimum distance from the reference position the vehicle must achieve while maneuvering to it.
speed	VariableSpeedControlType	The desired speed to return to the drift position when tolerance exceeded.
speedTolerance	GroundSpeed	The amount of speed error allowed relative to the commanded speed.
transitElevation	ElevationType	The elevation used while driving to the loiter track (USVs must specify 0 as it is a required field).
transitElevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded transit elevation.
transitSpeed	VariableSpeedControlType	The speed at which one drives to the drift position.
transitSpeedTolerance	GroundSpeed	The amount of speed offset allowed relative to the commanded transit speed.

6.1.11 LocalFigure8Control

Intended to command the platform to loiter about a desired position in the local coordinate frame using a figure 8 loiter pattern.
See figure for reference. (Figure 32)

Table 53: LocalFigure8Control Operations

Service Requests (Inputs)	Service Responses (Outputs)
setLocalFigure8	reportLocalFigure8CommandStatus
queryLocalFigure8CommandAck⊕	reportLocalFigure8CommandAck
queryLocalFigure8ExecutionStatus⊕	reportLocalFigure8ExecutionStatus
cancelLocalFigure8Command⊕	reportLocalFigure8CancelCommandStatus⊕

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

6.1.11.1 reportLocalFigure8CommandAck

Description: This operation is used to report the commanded values of the position and pattern and/or time that were commanded to the unmanned platform in the local coordinate system.

Namespace: UMAA::MO::LocalFigure8Control

Topic: LocalFigure8CommandAckReport

Data Type: LocalFigure8CommandAckReportType

Table 54: LocalFigure8CommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
crossTrackTolerance	Distance	The amount of error in position allowed from the pattern being executed.
elevation	ElevationType	The desired elevation used for the unmanned maritime platform.
elevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded loitering elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
length	Distance	Describes the length between the semicircles at either end of the figure 8 the unmanned platform should stay in.
orientation	HeadingType	The orientation of the figure 8, measured perpendicular to the length axis. If not specified, aligns with the local X axis.
position†	Position2D_PlatformXYZ	The desired loiter position (X, Y) of the unmanned platform in the local coordinate system. When not specified, means at current location.
radius	Distance	Describes the radius of the semicircles at either end of the figure 8 the unmanned platform should stay in.
speed	VariableSpeedControlType	The desired speed to loiter of the unmanned platform.
speedTolerance	GroundSpeed	The amount of speed error allowed relative to the commanded pattern execution speed.
transitElevation	ElevationType	The elevation used while driving to the loiter track (USVs must specify 0 as it is a required field).
transitElevationTolerance	Distance	The amount of elevation error allowed relative to the commanded transit elevation.
transitSpeed	VariableSpeedControlType	The speed at which one drives to the loiter track.
transitSpeedTolerance	GroundSpeed	The amount of speed error allowed relative to the transit speed.
turnDirection	WaterTurnDirectionEnumType	The desired direction to loiter of the unmanned platform.

6.1.11.2 reportLocalFigure8CommandStatus

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

Namespace: UMAA::MO::LocalFigure8Control

Topic: LocalFigure8CommandStatus

Data Type: LocalFigure8CommandStatusType

Table 55: LocalFigure8CommandStatusType Message Definition

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

6.1.11.3 reportLocalFigure8ExecutionStatus

Description: This operation is used to report the current position and pattern and/or time that the unmanned platform loitering based in the local coordinate system.

Namespace: UMAA::MO::LocalFigure8Control

Topic: LocalFigure8ExecutionStatusReport

Data Type: LocalFigure8ExecutionStatusReportType

Table 56: LocalFigure8ExecutionStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
localFigure8State	LocalFigure8StateType	Defines the state of the local figure 8.
timePatternAchieved	DateTime	The absolute time at which the loiter pattern is estimated to be achieved or was actually first achieved.
timePatternCompleted†	DateTime	The absolute time at which the loiter pattern is estimated to be completed.

6.1.11.4 setLocalFigure8

Description: This operation is used to set the desired position in the local coordinate system given the specified loiter pattern and/or time. If the command attributes do not specify a determinate end of execution, the consumer must perform a "cancel" of the command to initiate the end of command execution.

Namespace: UMAA::MO::LocalFigure8Control

Topic: LocalFigure8Command

Data Type: LocalFigure8CommandType

Table 57: LocalFigure8CommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
crossTrackTolerance	Distance	The amount of error in position allowed from the pattern being executed.
elevation	ElevationType	The desired elevation used for the unmanned maritime platform.
elevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded loitering elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
length	Distance	Describes the length between the semicircles at either end of the figure 8 the unmanned platform should stay in.
orientation	HeadingType	The orientation of the figure 8, measured perpendicular to the length axis. If not specified, aligns with the local X axis.
position†	Position2D_PlatformXYZ	The desired loiter position (X, Y) of the unmanned platform in the local coordinate system. When not specified, means at current location.
radius	Distance	Describes the radius of the semicircles at either end of the figure 8 the unmanned platform should stay in.
speed	VariableSpeedControlType	The desired speed to loiter of the unmanned platform.
speedTolerance	GroundSpeed	The amount of speed error allowed relative to the commanded pattern execution speed.
transitElevation	ElevationType	The elevation used while driving to the loiter track (USVs must specify 0 as it is a required field).
transitElevationTolerance	Distance	The amount of elevation error allowed relative to the commanded transit elevation.
transitSpeed	VariableSpeedControlType	The speed at which one drives to the loiter track.
transitSpeedTolerance	GroundSpeed	The amount of speed error allowed relative to the transit speed.
turnDirection	WaterTurnDirectionEnumType	The desired direction to loiter of the unmanned platform.

6.1.12 LocalHoverControl

The function of this service is to command the platform to hover in a desired position in the local coordinate frame.

Table 58: LocalHoverControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setLocalHover	reportLocalHoverCommandStatus
queryLocalHoverCommandAck⊕	reportLocalHoverCommandAck
queryLocalHoverExecutionStatus⊕	reportLocalHoverExecutionStatus

Service Requests (Inputs)	Service Responses (Outputs)
cancelLocalHoverCommand⊕	reportLocalHoverCancelCommandStatus⊕

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

6.1.12.1 reportLocalHoverCommandAck

Description: This operation is used to report the commanded values of the position or time that was commanded to the unmanned platform in the local coordinate system.

Namespace: UMAA::MO::LocalHoverControl

Topic: LocalHoverCommandAckReport

Data Type: LocalHoverCommandAckReportType

Table 59: LocalHoverCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
controlPriority	HoverKindEnumType	The desired priority to hover at the specified point.
elevation	ElevationType	The optional elevation used for the unmanned maritime platform.
elevationTolerance	Distance	The amount of elevation error allowed relative to the commanded hover elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
heading†	HeadingType	Defines the heading that the vehicle must maintain for loitering.
headingTolerance	RelativeAngle	The amount of angle error allowed relative to the commanded heading.
position†	Position2D_PlatformXYZ	Defines the reference position in the system-defined coordinate system for the hover. When not specified, means at current location.
positionTolerance	Distance	The amount of distance error (radius) allowed relative to the commanded position.
transitElevation	ElevationType	The elevation used while driving to the hover location (USVs must specify 0 as it is a required field).
transitElevationTolerance	Distance	The amount of elevation error allowed relative to the commanded transit elevation.
transitSpeed	VariableSpeedControlType	The speed at which one drives to the hover location.
transitSpeedTolerance	GroundSpeed	The amount of speed error allowed relative to the transit speed.

6.1.12.2 reportLocalHoverCommandStatus

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

Namespace: UMAA::MO::LocalHoverControl

Topic: LocalHoverCommandStatus

Data Type: LocalHoverCommandStatusType

Table 60: LocalHoverCommandStatusType Message Definition

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

6.1.12.3 reportLocalHoverExecutionStatus

Description: This operation is used to report the current position or time that the unmanned platform was hovering based in the local coordinate system.

Namespace: UMAA::MO::LocalHoverControl

Topic:

Data Type: LocalHoverExecutionStatusReport

Table 61: LocalHoverExecutionStatusReport Message Definition

Attribute Name	Attribute Type	Attribute Description
localHoverState	LocalHoverStateType	Defines the state of the local hover.
timeHoverAchieved	DateTime	The absolute time at which hover is estimated to be achieved or was actually first achieved.
timeHoverCompleted†	DateTime	The absolute time at which the hover is estimated to be completed (optional in case duration is forever).

6.1.12.4 setLocalHover

Description: This operation is used to set the desired hover position in the local coordinate system given the desired location and/or time. If the command attributes do not specify a determinate end of execution, the consumer must perform a "cancel" of the command to initiate the end of command execution.

Namespace: UMAA::MO::LocalHoverControl

Topic: LocalHoverCommand

Data Type: LocalHoverCommandType

Table 62: LocalHoverCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
controlPriority	HoverKindEnumType	The desired priority to hover at the specified point.
elevation	ElevationType	The optional elevation used for the unmanned maritime platform.
elevationTolerance	Distance	The amount of elevation error allowed relative to the commanded hover elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
heading†	HeadingType	Defines the heading that the vehicle must maintain for loitering.
headingTolerance	RelativeAngle	The amount of angle error allowed relative to the commanded heading.
position†	Position2D_PlatformXYZ	Defines the reference position in the system-defined coordinate system for the hover. When not specified, means at current location.
positionTolerance	Distance	The amount of distance error (radius) allowed relative to the commanded position.
transitElevation	ElevationType	The elevation used while driving to the hover location (USVs must specify 0 as it is a required field).
transitElevationTolerance	Distance	The amount of elevation error allowed relative to the commanded transit elevation.
transitSpeed	VariableSpeedControlType	The speed at which one drives to the hover location.
transitSpeedTolerance	GroundSpeed	The amount of speed error allowed relative to the transit speed.

6.1.13 LocalRacetrackControl

Intended to command the platform to loiter about a desired position in the local coordinate frame using a specified loiter pattern. The start location on the Racetrack and the path to this start location is system dependent.
See figure for reference. (Figure 33)

Table 63: LocalRacetrackControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setLocalRacetrack	reportLocalRacetrackCommandStatus
queryLocalRacetrackCommandAck⊕	reportLocalRacetrackCommandAck
queryLocalRacetrackExecutionStatus⊕	reportLocalRacetrackExecutionStatus
cancelLocalRacetrackCommand⊕	reportLocalRacetrackCancelCommandStatus⊕

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

6.1.13.1 reportLocalRacetrackCommandAck

Description: This operation is used to report the commanded values of the position and pattern and/or time that were commanded to the unmanned platform in the local coordinate system.

Namespace: UMAA::MO::LocalRacetrackControl

Topic: LocalRacetrackCommandAckReport

Data Type: LocalRacetrackCommandAckReportType

Table 64: LocalRacetrackCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
crossTrackTolerance	Distance	The amount of error in position allowed from the pattern being executed.
elevation	ElevationType	The desired elevation used for the unmanned maritime platform.
elevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded loitering elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
length	Distance	The describes the length between the semicircles at either end of the racetrack the unmanned platform should stay in.
orientation	HeadingType	The orientation of the racetrack, measured perpendicular to the length axis. If not specified, aligns with the local X axis.
position†	Position2D_PlatformXYZ	The desired loiter position (X, Y) of the unmanned platform in the local coordinate system. When not specified, means at current location.
radius	Distance	The describes the radius of the semicircles at either end of the racetrack the unmanned platform should stay in.
speed	VariableSpeedControlType	The desired speed to loiter of the unmanned platform.
speedTolerance	GroundSpeed	The amount of speed error allowed relative to the commanded pattern execution speed.
transitElevation	ElevationType	The elevation used while driving to the loiter track (USVs must specify 0 as it is a required field).
transitElevationTolerance	Distance	The amount of elevation error allowed relative to the commanded transit elevation.
transitSpeed	VariableSpeedControlType	The speed at which one drives to the loiter track.
transitSpeedTolerance	GroundSpeed	The amount of speed error allowed relative to the transit speed.
turnDirection	WaterTurnDirectionEnumType	The desired direction to loiter of the unmanned platform.

6.1.13.2 reportLocalRacetrackCommandStatus

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

Namespace: UMAA::MO::LocalRacetrackControl

Topic: LocalRacetrackCommandStatus

Data Type: LocalRacetrackCommandStatusType

Table 65: LocalRacetrackCommandStatusType Message Definition

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

6.1.13.3 reportLocalRacetrackExecutionStatus

Description: This operation is used to report the current position and pattern and/or time that the unmanned platform loitering based in the local coordinate system.

Namespace: UMAA::MO::LocalRacetrackControl

Topic: LocalRacetrackExecutionStatusReport

Data Type: LocalRacetrackExecutionStatusReportType

Table 66: LocalRacetrackExecutionStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
localRacetrackState	LocalRacetrackStateType	Defines the state of the local racetrack.
timePatternAchieved	DateTime	The absolute time at which the loiter pattern is estimated to be achieved or was actually first achieved.
timePatternCompleted†	DateTime	The absolute time at which the loiter pattern is estimated to be completed.

6.1.13.4 setLocalRacetrack

Description: This operation is used to set the desired position in the local coordinate system given the specified loiter pattern and/or time. If the command attributes do not specify a determinate end of execution, the consumer must perform a "cancel" of the command to initiate the end of command execution.

Namespace: UMAA::MO::LocalRacetrackControl

Topic: LocalRacetrackCommand

Data Type: LocalRacetrackCommandType

Table 67: LocalRacetrackCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
crossTrackTolerance	Distance	The amount of error in position allowed from the pattern being executed.
elevation	ElevationType	The desired elevation used for the unmanned maritime platform.
elevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded loitering elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
length	Distance	The describes the length between the semicircles at either end of the racetrack the unmanned platform should stay in.
orientation	HeadingType	The orientation of the racetrack, measured perpendicular to the length axis. If not specified, aligns with the local X axis.
position†	Position2D_PlatformXYZ	The desired loiter position (X, Y) of the unmanned platform in the local coordinate system. When not specified, means at current location.
radius	Distance	The describes the radius of the semicircles at either end of the racetrack the unmanned platform should stay in.
speed	VariableSpeedControlType	The desired speed to loiter of the unmanned platform.
speedTolerance	GroundSpeed	The amount of speed error allowed relative to the commanded pattern execution speed.
transitElevation	ElevationType	The elevation used while driving to the loiter track (USVs must specify 0 as it is a required field).
transitElevationTolerance	Distance	The amount of elevation error allowed relative to the commanded transit elevation.
transitSpeed	VariableSpeedControlType	The speed at which one drives to the loiter track.
transitSpeedTolerance	GroundSpeed	The amount of speed error allowed relative to the transit speed.
turnDirection	WaterTurnDirectionEnumType	The desired direction to loiter of the unmanned platform.

6.1.14 LocalRegularPolygonControl

Intended to command the platform to loiter about a desired position in the local coordinate frame using a regular polygon pattern circumscribed on a circle. The start location on the RegularPolygon and the path to this start location is system dependent.
See figure for reference. (Figure 34)

Table 68: LocalRegularPolygonControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setLocalRegularPolygon	reportLocalRegularPolygonCommandStatus
queryLocalRegularPolygonCommandAck⊕	reportLocalRegularPolygonCommandAck
queryLocalRegularPolygonExecutionStatus⊕	reportLocalRegularPolygonExecutionStatus
cancelLocalRegularPolygonCommand⊕	reportLocalRegularPolygonCancelCommandStatus⊕

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

6.1.14.1 reportLocalRegularPolygonCommandAck

Description: This operation is used to report the commanded values of the position and pattern and/or time that were commanded to the unmanned platform in the local coordinate system.

Namespace: UMAA::MO::LocalRegularPolygonControl

Topic: LocalRegularPolygonCommandAckReport

Data Type: LocalRegularPolygonCommandAckReportType

Table 69: LocalRegularPolygonCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
crossTrackTolerance	Distance	The amount of error in position allowed from the pattern being executed.
diameter	Distance	The diameter of a circumscribed circle around the polygon.
elevation	ElevationType	The optional elevation used for the unmanned maritime platform.
elevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded loitering elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
numberSides	SidesCount	The number of sides on the polygon.
orientation	HeadingType	The orientation describes the bearing from the reference position of the polygon to one point on the polygon.
position†	Position2D_PlatformXYZ	Defines the reference position in the system-defined coordinate system for the polygon center. When not specified, means at current location.
speedTolerance	GroundSpeed	The amount of speed error allowed relative to the commanded pattern execution speed.
transitElevation	ElevationType	The elevation used while driving to the loiter track (USVs must specify 0 as it is a required field).
transitElevationTolerance	Distance	The amount of elevation error allowed relative to the commanded transit elevation.
transitSpeed	VariableSpeedControlType	The speed at which one drives to the loiter track.

Attribute Name	Attribute Type	Attribute Description
transitSpeedTolerance	GroundSpeed	The amount of speed error allowed relative to the transit speed.
turnDirection	WaterTurnDirectionEnumType	The desired turn direction for the loiter of the unmanned platform.

6.1.14.2 reportLocalRegularPolygonCommandStatus

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

Namespace: UMAA::MO::LocalRegularPolygonControl

Topic: LocalRegularPolygonCommandStatus

Data Type: LocalRegularPolygonCommandStatusType

Table 70: LocalRegularPolygonCommandStatusType Message Definition

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

6.1.14.3 reportLocalRegularPolygonExecutionStatus

Description: This operation is used to report the current position and pattern and/or time that the unmanned platform loitering based in the local coordinate system.

Namespace: UMAA::MO::LocalRegularPolygonControl

Topic: LocalRegularPolygonExecutionStatusReport

Data Type: LocalRegularPolygonExecutionStatusReportType

Table 71: LocalRegularPolygonExecutionStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
localRegularPolygonState	LocalRegularPolygonStateType	Defines the state of the local regular polygon.
timePatternAchieved	DateTime	The absolute time at which the loiter pattern is estimated to be achieved or was actually first achieved.
timePatternCompleted†	DateTime	The absolute time at which the loiter pattern is estimated to be completed.

6.1.14.4 setLocalRegularPolygon

Description: This operation is used to set the desired position in the local coordinate system given the specified loiter pattern and/or time. If the command attributes do not specify a determinate end of execution, the consumer must perform a "cancel" of the command to initiate the end of command execution.

Namespace: UMAA::MO::LocalRegularPolygonControl

Topic: LocalRegularPolygonCommand

Data Type: LocalRegularPolygonCommandType

Table 72: LocalRegularPolygonCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
	Additional fields included from UMAA::UMAACommand	
crossTrackTolerance	Distance	The amount of error in position allowed from the pattern being executed.
diameter	Distance	The diameter of a circumscribed circle around the polygon.
elevation	ElevationType	The optional elevation used for the unmanned maritime platform.
elevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded loitering elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
numberSides	SidesCount	The number of sides on the polygon.
orientation	HeadingType	The orientation describes the bearing from the reference position of the polygon to one point on the polygon.
position†	Position2D_PlatformXYZ	Defines the reference position in the system-defined coordinate system for the polygon center. When not specified, means at current location.
speedTolerance	GroundSpeed	The amount of speed error allowed relative to the commanded pattern execution speed.
transitElevation	ElevationType	The elevation used while driving to the loiter track (USVs must specify 0 as it is a required field).
transitElevationTolerance	Distance	The amount of elevation error allowed relative to the commanded transit elevation.
transitSpeed	VariableSpeedControlType	The speed at which one drives to the loiter track.
transitSpeedTolerance	GroundSpeed	The amount of speed error allowed relative to the transit speed.
turnDirection	WaterTurnDirectionEnumType	The desired turn direction for the loiter of the unmanned platform.

6.1.15 LocalVectorControl

The purpose of this service is to command the unmanned platform to maintain attitude and speed and altitude or depth (if supported) in local coordinate system.

Table 73: LocalVectorControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setLocalVector	reportLocalVectorCommandStatus
queryLocalVectorCommandAck⊕	reportLocalVectorCommandAck
queryLocalVectorExecutionStatus⊕	reportLocalVectorExecutionStatus
cancelLocalVectorCommand⊕	reportLocalVectorCancelCommandStatus⊕

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

6.1.15.1 reportLocalVectorCommandAck

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

Namespace: UMAA::MO::LocalVectorControl

Topic: LocalVectorCommandAckReport

Data Type: LocalVectorCommandAckReportType

Table 74: LocalVectorCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
attitude	OrientationType	The desired reference frame used for vehicle orientation commands.
attitudeTolerance	Orientation3D_Tolerance	The allowable attitude tolerance.
elevation	ElevationType	Specifies the elevation of the vector.
elevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
speed	SpeedControlType	The desired speed of the unmanned platform.
speedTolerance	GroundSpeed	The amount of speed error allowed relative to the commanded speed.

6.1.15.2 reportLocalVectorCommandStatus

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

Namespace: UMAA::MO::LocalVectorControl

Topic: LocalVectorCommandStatus

Data Type: LocalVectorCommandStatusType

Table 75: LocalVectorCommandStatusType Message Definition

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

6.1.15.3 reportLocalVectorExecutionStatus

Description: This operation is used to report the current vector data based in the local coordinate system.

Namespace: UMAA::MO::LocalVectorControl

Topic: LocalVectorExecutionStatusReport

Data Type: LocalVectorExecutionStatusReportType

Table 76: LocalVectorExecutionStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
attitudeAchieved	BooleanEnumType	When the vector is executing, this indicates that the attitude requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
elevationAchieved	BooleanEnumType	When the vector is executing, this indicates that the elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
speedAchieved	BooleanEnumType	When the vector is executing, this indicates that the speed requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

6.1.15.4 setLocalVector

Description: This operation is used to command the speed, yaw, roll, pitch, and altitude or depth of an unmanned platform in the local coordinate system. If the command attributes do not specify a determinate end of execution, the consumer must perform a "cancel" of the command to initiate the end of command execution.

Namespace: UMAA::MO::LocalVectorControl

Topic: LocalVectorCommand

Data Type: LocalVectorCommandType

Table 77: LocalVectorCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
	Additional fields included from UMAA::UMAACommand	
attitude	OrientationType	The desired reference frame used for vehicle orientation commands.
attitudeTolerance	Orientation3D_Tolerance	The allowable attitude tolerance.
elevation	ElevationType	Specifies the elevation of the vector.
elevationTolerance	Distance	The amount of elevation offset allowed relative to the commanded elevation.
endTime†	DateTime	Specifies the end of the valid time period for the loiter; if not specified runs indefinitely until command is changed externally.
speed	SpeedControlType	The desired speed of the unmanned platform.
speedTolerance	GroundSpeed	The amount of speed error allowed relative to the commanded speed.

6.1.16 LocalWaypointControl

The purpose of this service is to move the platform based on a defined single waypoint or a list of target waypoints, desired speed, current platform pose, and current velocity state in local coordinate system. If during waypoint operations, the service cannot satisfy a waypoint, the service will publish a failed status state and the command issuer must determine whether to reissue the remaining waypoints or take some other action. A waypoint is achieved when it is within the waypointTolerance radius of the specified waypoint location.

Table 78: LocalWaypointControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setLocalWaypoint	reportLocalWaypointCommandStatus
queryLocalWaypointCommandAck⊕	reportLocalWaypointCommandAck
queryLocalWaypointExecutionStatus⊕	reportLocalWaypointExecutionStatus
cancelLocalWaypointCommand⊕	reportLocalWaypointCancelCommandStatus⊕
setLocalWaypointSpeed	reportLocalWaypointSpeedCommandStatus
queryLocalWaypointSpeedCommandAck⊕	reportLocalWaypointSpeedCommandAck
cancelLocalWaypointSpeedCommand⊕	reportLocalWaypointSpeedCancelCommandStatus⊕

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

6.1.16.1 reportLocalWaypointCommandAck

Description: This operation is used to report the commanded values of the waypoint data based in the local coordinate system.

Namespace: UMAA::MO::LocalWaypointControl

Topic: LocalWaypointCommandAckReport

Data Type: LocalWaypointCommandAckReportType

Table 79: LocalWaypointCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
waypointCount	Count	The total number of waypoint on the series of waypoints.
waypoints	sequence<LocalWaypointType>	The desired series of waypoints in the local coordinate system.

6.1.16.2 reportLocalWaypointCommandStatus

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

Namespace: UMAA::MO::LocalWaypointControl

Topic: LocalWaypointCommandStatus

Data Type: LocalWaypointCommandStatusType

Table 80: LocalWaypointCommandStatusType Message Definition

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

6.1.16.3 reportLocalWaypointExecutionStatus

Description: This operation is used to report the current waypoint data based in the local coordinate system. All times and distances are provided based on waypoint location not including tolerances.

Namespace: UMAA::MO::LocalWaypointControl

Topic: LocalWaypointExecutionStatusReport

Data Type: LocalWaypointExecutionStatusReportType

Table 81: LocalWaypointExecutionStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
arrivalTime	DateTime	The arrival time of the end of the route.
crossTrackError	Distance	Defines the current cross track error. Only valid if maintainTrack is true.
cumulativeDistance	Distance	Defines the ground distance travel from the start of the route to this point

Attribute Name	Attribute Type	Attribute Description
distanceRemaining	Distance	Defines the amount of distance remaining from a point to the end of the route
distanceToWaypoint	Distance	Defines the remaining distance to the current waypoint
elevationAchieved	BooleanEnumType	When the waypoint is executing, this indicates that the elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
maintainTrack	BooleanEnumType	Indicates whether a track line is to be followed when transiting to the waypoint.
speedAchieved	BooleanEnumType	When the waypoint is executing, this indicates that the speed requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
timeToWaypoint	DateTime	The absolute time at which the waypoint is estimated to be achieved or was actually first achieved.
trackLineAchieved	BooleanEnumType	When the waypoint is executing, this indicates that the track line requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
waypointsRemaining	Count	Defines the remaining number of waypoints, which includes the current waypoint.
waypointID*	NumericGUID	Defines the current waypoint ID.

6.1.16.4 reportLocalWaypointSpeedCommandAck

Description: This operation is used to report the commanded values of the travel speed of the unmanned platform.

Namespace: UMAA::MO::LocalWaypointControl

Topic: LocalWaypointSpeedCommandAckReport

Data Type: LocalWaypointSpeedCommandAckReportType

Table 82: LocalWaypointSpeedCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
speed	VariableSpeedControlType	The desired travel speed of the unmanned platform.

6.1.16.5 reportLocalWaypointSpeedCommandStatus

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

Namespace: UMAA::MO::LocalWaypointControl

Topic: LocalWaypointSpeedCommandStatus

Data Type: LocalWaypointSpeedCommandStatusType

Table 83: LocalWaypointSpeedCommandStatusType Message Definition

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

6.1.16.6 setLocalWaypoint

Description: This operation is used to set a single waypoint data based in the local coordinate system.

Namespace: UMAA::MO::LocalWaypointControl

Topic: LocalWaypointCommand

Data Type: LocalWaypointCommandType

Table 84: LocalWaypointCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
waypointCount	Count	The total number of waypoint on the series of waypoints.
waypoints	sequence<LocalWaypointType>	The desired series of waypoints in the local coordinate system.

6.1.16.7 setLocalWaypointSpeed

Description: This operation is used to set the desired travel speed as an absolute or relative value.

Namespace: UMAA::MO::LocalWaypointControl

Topic: LocalWaypointSpeedCommand

Data Type: LocalWaypointSpeedCommandType

Table 85: LocalWaypointSpeedCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
speed	VariableSpeedControlType	The desired travel speed of the unmanned platform.

6.1.17 PrimitiveDriverControl

This service provides mobility in six degrees of freedom using a percent of available effort in each direction. Additionally, no power plant is implied and the service functions strictly in an open loop manner, i.e., a velocity is not commanded or held since that requires a speed sensor. The service definition makes no assertion about the preventative actions that must be taken to avoid unintended consequences, such as losing positive control when given a zero propulsive effort. This service uses "effort" as a relative measure of the amount of drive power. This measure is intentionally kept agnostic of the underlying control system for portability across hardware types. As a result, the implementation of an "effort" driver may map the request to a percent of maximum current of an electric motor, fluid pressure of a hydraulic system, duty-cycle of a pulse-width modulated controller, or position of a control lever. These examples are meant to be illustrative; the actual mapping is not restricted so long as it can be expressed as a percent of some maximum.

Table 86: PrimitiveDriverControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setPrimitiveDriver	reportPrimitiveDriverCommandStatus
queryPrimitiveDriverCommandAck⊕	reportPrimitiveDriverCommandAck
cancelPrimitiveDriverCommand⊕	reportPrimitiveDriverCancelCommandStatus⊕

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

6.1.17.1 reportPrimitiveDriverCommandAck

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

Namespace: UMAA::MO::PrimitiveDriverControl

Topic: PrimitiveDriverCommandAckReport

Data Type: PrimitiveDriverCommandAckReportType

Table 87: PrimitiveDriverCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
propulsiveLinearEffort	LinearEffort	The desired propulsive linear effort (X, Y, Z) in percent. Propulsive linear effort represents a thrusting action that results in a linear motion along the respective axis.
propulsiveRotationalEffort	RotationalEffort	The desired propulsive rotational effort (X, Y, Z) in percent. Propulsive rotational effort represents a thrusting action that results in a rotational motion about the respective axis.
resistiveLinearEffort	LinearEffort	The desired resistive linear effort (X, Y, Z) in percent. Resistive linear effort represents a braking action that impedes linear motion along the respective axis.
resistiveRotationalEffort	RotationalEffort	The desired resistive rotational effort (X, Y, Z) in percent. Resistive rotational effort represents a braking action that impedes rotational motion about the respective axis.

6.1.17.2 reportPrimitiveDriverCommandStatus

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

Namespace: UMAA::MO::PrimitiveDriverControl

Topic: PrimitiveDriverCommandStatus

Data Type: PrimitiveDriverCommandStatusType

Table 88: PrimitiveDriverCommandStatusType Message Definition

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

6.1.17.3 setPrimitiveDriver

Description: This operation is used to set the mobility of the unmanned platform using the effort. The consumer must perform a "cancel" of the command to initiate the end of command execution as this command has no determinate end of execution.

Namespace: UMAA::MO::PrimitiveDriverControl

Topic: PrimitiveDriverCommand

Data Type: PrimitiveDriverCommandType

Table 89: PrimitiveDriverCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
propulsiveLinearEffort	LinearEffort	The desired propulsive linear effort (X, Y, Z) in percent. Propulsive linear effort represents a thrusting action that results in a linear motion along the respective axis.
propulsiveRotationalEffort	RotationalEffort	The desired propulsive rotational effort (X, Y, Z) in percent. Propulsive rotational effort represents a thrusting action that results in a rotational motion about the respective axis.
resistiveLinearEffort	LinearEffort	The desired resistive linear effort (X, Y, Z) in percent. Resistive linear effort represents a braking action that impedes linear motion along the respective axis.
resistiveRotationalEffort	RotationalEffort	The desired resistive rotational effort (X, Y, Z) in percent. Resistive rotational effort represents a braking action that impedes rotational motion about the respective axis.

6.1.18 PrimitiveDriverStatus

This service provides the current status of the effort.

Table 90: PrimitiveDriverStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryPrimitiveDriver⊕	reportPrimitiveDriver

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

6.1.18.1 reportPrimitiveDriver

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

Namespace: UMAA::MO::PrimitiveDriverStatus

Topic: PrimitiveDriverReport

Data Type: PrimitiveDriverReportType

Table 91: PrimitiveDriverReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
propulsiveLinearEffort	LinearEffort	The current propulsive linear effort (X, Y, Z) in percent.
propulsiveRotationalEffort	RotationalEffort	The current propulsive rotational effort (X, Y, Z) in percent.
resistiveLinearEffort	LinearEffort	The current resistive linear effort (X, Y, Z) in percent.
resistiveRotationalEffort	RotationalEffort	The current resistive rotational effort (X, Y, Z) in percent.

6.1.19 StationkeepControl

The purpose of this service is to maintain position relative to a contact.

Table 92: StationkeepControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setStationkeep	reportStationkeepCommandStatus
queryStationkeepCommandAck⊕	reportStationkeepCommandAck
queryStationkeepExecutionStatus⊕	reportStationkeepExecutionStatus
cancelStationkeepCommand⊕	reportStationkeepCancelCommandStatus⊕

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

6.1.19.1 reportStationkeepCommandAck

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

Namespace: UMAA::MO::StationkeepControl

Topic: StationkeepCommandAckReport

Data Type: StationkeepCommandAckReportType

Table 93: StationkeepCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
angleType	BearingAngleEnumType	Defines angle reference frame
bearing	Angle	Defines bearing to contact for station keeping
bearingTolerance	Angle	Defines the amount of angle error allowed relative to the commanded bearing to contact
closingSpeed	GroundSpeed	Defines closingSpeed to contact for station keeping
contactTrackID	NumericGUID	Defines contactTrackID for station keeping
endTime†	DateTime	Specifies the end of the valid time period for the station-keep loiter; if not specified runs indefinitely until command is changed externally.
range	Distance	Defines distance to contact for station keeping
rangeTolerance	Distance	Defines the amount of distance error allowed relative to the commanded distance to contact for station keeping.
transitSpeed	VariableSpeedControlType	The desired waypoint travel speed of the unmanned platform.
transitSpeedTolerance	GroundSpeed	The amount of speed error allowed relative to the commanded transit speed.

6.1.19.2 reportStationkeepCommandStatus

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

Namespace: UMAA::MO::StationkeepControl

Topic: StationkeepCommandStatus

Data Type: StationkeepCommandStatusType

Table 94: StationkeepCommandStatusType Message Definition

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

6.1.19.3 reportStationkeepExecutionStatus

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

Namespace: UMAA::MO::StationkeepControl

Topic: StationkeepExecutionStatusReport

Data Type: StationkeepExecutionStatusReportType

Table 95: StationkeepExecutionStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
angleType	BearingAngleEnumType	Defines current bearing angle reference frame
bearing	Angle	Defines current bearing to contact
bearingTolerance	Angle	Defines the amount of angle error allowed relative to the commanded bearing to contact
closingSpeed	GroundSpeed	Defines current closingSpeed to contact
contactLost	BooleanEnumType	Indicates whether a contact has been lost when transiting to the waypoint.
contactTrackID	NumericGUID	Defines current contactTrackID
range	Distance	Defines current distance to contact
rangeTolerance	Distance	Defines the amount of distance error allowed relative to the commanded distance to contact for station keeping.
stationkeepState	StationkeepStateType	Defines the state of the station keeping.
timeStationkeepCompleted†	DateTime	The absolute time at which the station keep is estimated to be completed (optional in case duration is forever).
timeStationskeepAchieved	DateTime	The absolute time at which station keep is estimated to be achieved or was actually first achieved.

6.1.19.4 setStationkeep

Description: This operation is used to set the current Stationkeep command. The consumer must perform a "cancel" of the command to initiate the end of command execution as this command has no determinate end of execution.

Namespace: UMAA::MO::StationkeepControl

Topic: StationkeepCommand

Data Type: StationkeepCommandType

Table 96: StationkeepCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
angleType	BearingAngleEnumType	Defines angle reference frame

Attribute Name	Attribute Type	Attribute Description
bearing	Angle	Defines bearing to contact for station keeping
bearingTolerance	Angle	Defines the amount of angle error allowed relative to the commanded bearing to contact
closingSpeed	GroundSpeed	Defines closingSpeed to contact for station keeping
contactTrackID	NumericGUID	Defines contactTrackID for station keeping
endTime†	DateTime	Specifies the end of the valid time period for the station-keep loiter; if not specified runs indefinitely until command is changed externally.
range	Distance	Defines distance to contact for station keeping
rangeTolerance	Distance	Defines the amount of distance error allowed relative to the commanded distance to contact for station keeping.
transitSpeed	VariableSpeedControlType	The desired waypoint travel speed of the unmanned platform.
transitSpeedTolerance	GroundSpeed	The amount of speed error allowed relative to the commanded transit speed.

6.1.20 VelocityControl

The purpose of this service is to control the desired linear and rotational velocity of the unmanned platform. This service allows for low level control of platform mobility. This service does not imply any particular domain type such ground, air, surface, or underwater vehicles, but describes mobility in six degrees of freedom using velocity commands relative to the vehicle's coordinate system.

Table 97: VelocityControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setVelocity	reportVelocityCommandStatus
queryVelocityCommandAck⊕	reportVelocityCommandAck
queryVelocityExecutionStatus⊕	reportVelocityExecutionStatus
cancelVelocityCommand⊕	reportVelocityCancelCommandStatus⊕

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

6.1.20.1 reportVelocityCommandAck

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

Namespace: UMAA::MO::VelocityControl

Topic: VelocityCommandAckReport

Data Type: VelocityCommandAckReportType

Table 98: VelocityCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
attitudeRate	OrientationVelocity3D	The desired rotational rate of the unmanned platform.
velocity	Velocity3D_PlatformXYZ	The desired linear velocity (velocity X, Y, Z) of the unmanned platform.
commandType*	VelocityCommandTypeEnumType	The desired velocity reference to be set for velocity command type.

6.1.20.2 reportVelocityCommandStatus

Description: This operation is used to report the status of the linear and rotational velocity command.

Namespace: UMAA::MO::VelocityControl

Topic: VelocityCommandStatus

Data Type: VelocityCommandStatusType

Table 99: VelocityCommandStatusType Message Definition

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

6.1.20.3 reportVelocityExecutionStatus

Description: This operation is used to report the current velocity control state of the unmanned platform.

Namespace: UMAA::MO::VelocityControl

Topic: VelocityExecutionStatusReport

Data Type: VelocityExecutionStatusReportType

Table 100: VelocityExecutionStatusReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
downSpeedAchieved	BooleanEnumType	The down speed is achieved and is being maintained.
forwardSpeedAchieved	BooleanEnumType	The forward speed is achieved and is being maintained.
pitchRateAchieved	BooleanEnumType	The pitch rate is achieved and is being maintained.
rightSpeedAchieved	BooleanEnumType	The right speed is achieved and is being maintained.
rollRateAchieved	BooleanEnumType	The roll rate is achieved and is being maintained.

Attribute Name	Attribute Type	Attribute Description
yawRateAchieved	BooleanEnumType	The yaw rate is achieved and is being maintained.

6.1.20.4 setVelocity

Description: This operation is used to set the desired linear and rotational velocity of the unmanned platform. If the command attributes do not specify a determinate end of execution, the consumer must perform a "cancel" of the command to initiate the end of command execution.

Namespace: UMAA::MO::VelocityControl

Topic: VelocityCommand

Data Type: VelocityCommandType

Table 101: VelocityCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
attitudeRate	OrientationVelocity3D	The desired rotational rate of the unmanned platform.
velocity	Velocity3D_PlatformXYZ	The desired linear velocity (velocity X, Y, Z) of the unmanned platform.
commandType*	VelocityCommandTypeEnumType	The desired velocity reference to be set for velocity command type.

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 102: 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 103: 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 104: 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 105: 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 106: 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 107: 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 108: 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 109: 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 110: 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 111: 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 112: 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_MSL

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

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

Table 113: Altitude_MSL Structure Definition

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

6.2.13 AttitudeType

Namespace: UMAA::Common::Orientation::AttitudeType

Description: Defines the desired orientation of the unmanned platform.

Table 114: AttitudeType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::Common::Orientation::OrientationType		
attitude	Orientation3D	specifies desired orientation of the unmanned platform

6.2.14 CourseType

Namespace: UMAA::Common::Orientation::CourseType

Description: Defines the direction the vehicle is travelling

Table 115: CourseType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::Common::Orientation::DirectionType		
course	Course_TrueNorth	specifies the direction the vehicle is travelling

6.2.15 DepthType

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

Description: Defines the depth below sea level

Table 116: 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

6.2.16 DirectionType

Namespace: UMAA::Common::Orientation::DirectionType

Description: Union Type. Direction in either course the vehicle is travelling or heading the vehicle is facing.

Table 117: DirectionType Union(s)

Type Name	Type Description
CourseType	Defines the direction the vehicle is travelling
HeadingType	Defines the angle offset from the specified reference frame.

6.2.17 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 118: 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

Type Name	Type Description
DepthType	Defines the depth below sea level

6.2.18 EngineRPM

Namespace: UMAA::Common::Speed::EngineRPM

Description: Defines the engine RPM

Table 119: EngineRPM Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::Common::Speed::SpeedControlType		
RPM	EngineSpeed	specifies engine RPM

6.2.19 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 120: 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.

6.2.20 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 121: 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.

6.2.21 GlobalDriftStateType

Namespace: UMAA::MO::GlobalDriftState::GlobalDriftStateType

Description: Union Type. State of the global drift. While first transiting to the drift position, the selector will be GlobalTransitDriftType until the position and elevation are first achieved within their respective tolerances. Once achieved, the union selector will change to GlobalRegionDriftType. The selector will not change as a result of any of the GlobalRegionDriftType achievements states being lost and regained as a result of tolerance settings being violated. This is true until the service determines that the elevation or drift tolerances are violated by a sufficient margin that it is more effective for the vehicle to return to transiting to the drift location. In that case, the GlobalRegionDriftType reverts to the GlobalTransitDriftType selector and those transit achievements then are actively set.

Table 122: GlobalDriftStateType Union(s)

Type Name	Type Description
GlobalRegionDriftType	Indicates that the vehicle is in the global drift region
GlobalTransitDriftType	Indicates that vehicle is in transit to the global drift region

6.2.22 GlobalFigure8PatternType

Namespace: UMAA::MO::GlobalFigure8State::GlobalFigure8PatternType

Description: Indicates that the global figure 8 pattern is currently executing.

Table 123: GlobalFigure8PatternType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MO::GlobalFigure8State::GlobalFigure8StateType		
elevationAchieved	BooleanEnumType	Indicates that the elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
positionAchieved	BooleanEnumType	When the pattern is executing, this indicates that the position requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
speedAchieved	BooleanEnumType	When the pattern is executing, this indicates that the speed requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

6.2.23 GlobalFigure8StateType

Namespace: UMAA::MO::GlobalFigure8State::GlobalFigure8StateType

Description: Union Type. State of the global figure 8 pattern being executed. While first transiting to the figure 8 pattern to be performed, the selector will be GlobalFigure8TransitType until the pattern position, speed, and elevation are first achieved within their respective tolerances. Once achieved, the union selector will change to GlobalFigure8PatternType. The selector will not change as a result of any of the GlobalFigure8PatternType achievements states being lost and regained as a result of tolerance settings being violated. The service is expected to make driving adjustments to attempt to keep all achievement states satisfied. This is true until the service determines tolerance(s) are violated by a sufficient margin that

it is more effective for the vehicle to return to transiting to the pattern location. In that case, the GlobalFigure8StateType reverts to the GlobalFigure8TransitType selector and those transit achievements are then set.

Table 124: GlobalFigure8StateType Union(s)

Type Name	Type Description
GlobalFigure8PatternType	Indicates that the global figure 8 pattern is currently executing.
GlobalFigure8TransitType	Indicates that the vehicle is in transit to where the global figure 8 pattern is to be performed.

6.2.24 GlobalFigure8TransitType

Namespace: UMAA::MO::GlobalFigure8State::GlobalFigure8TransitType

Description: Indicates that the vehicle is in transit to where the global figure 8 pattern is to be performed.

Table 125: GlobalFigure8TransitType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MO::GlobalFigure8State::GlobalFigure8StateType		
transitElevationAchieved	BooleanEnumType	Indicates that the transit elevation requested is within the commanded tolerance. Achievement may be lost and re-gained resulting in multiple changes to this attribute.
transitSpeedAchieved	BooleanEnumType	Indicates that the transit speed requested is within the commanded tolerance. Achievement may be lost and re-gained resulting in multiple changes to this attribute.

6.2.25 GlobalHoverStateType

Namespace: UMAA::MO::GlobalHoverState::GlobalHoverStateType

Description: Union Type. State of the global hover. While first transiting to the hover location, the selector will be GlobalTransitHoverType until the position, heading, and elevation are first achieved within their respective tolerances. Once achieved, the union selector will change to GlobalHoveringHoverType. The selector will not change as a result of any of the GlobalHoveringHoverType achievements states being lost and regained as a result of tolerance settings being violated. The service is expected to make driving adjustments to attempt to keep all achievement states satisfied. This is true until the service determines that tolerance(s) are violated by a sufficient margin that it is more effective for the vehicle to return to transiting to the hover location. In that case, the GlobalHoverStateType reverts to the GlobalTransitHoverType selector and those transit achievements then are actively set.

Table 126: GlobalHoverStateType Union(s)

Type Name	Type Description
GlobalHoveringHoverType	Indicates that the global hover is currently executing.
GlobalTransitHoverType	Indicates that the vehicle is in transit to where the global hover is to be performed.

6.2.26 GlobalHoveringHoverType

Namespace: UMAA::MO::GlobalHoverState::GlobalHoveringHoverType

Description: Indicates that the global hover is currently executing.

Table 127: GlobalHoveringHoverType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MO::GlobalHoverState::GlobalHoverStateType		
elevationAchieved	BooleanEnumType	Indicates that the elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
headingAchieved	BooleanEnumType	Indicates that the heading requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
positionAchieved	BooleanEnumType	Indicates that the position requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

6.2.27 GlobalRacetrackPatternType

Namespace: UMAA::MO::GlobalRacetrackState::GlobalRacetrackPatternType

Description: Indicates that the global racetrack pattern is currently executing.

Table 128: GlobalRacetrackPatternType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MO::GlobalRacetrackState::GlobalRacetrackStateType		
elevationAchieved	BooleanEnumType	Indicates that the elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
positionAchieved	BooleanEnumType	Indicates that the position requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
speedAchieved	BooleanEnumType	Indicates that the speed requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

6.2.28 GlobalRacetrackStateType

Namespace: UMAA::MO::GlobalRacetrackState::GlobalRacetrackStateType

Description: Union Type. State of the global racetrack pattern being executed. While first transiting to the racetrack location to be performed, the selector will be GlobalRacetrackTransitType until the pattern position, speed, and elevation are first achieved within their respective tolerances. Once achieved, the union selector will change to GlobalRacetrackPatternType. The selector will not change as a result of any of the GlobalRacetrackPatternType achievements states being lost and regained as a result of tolerance settings being violated. The service is expected to make driving adjustments to attempt to keep all achievement states satisfied. This is true until the service determines that tolerance(s) are violated by a

sufficient margin that it is more effective for the vehicle to return to transiting to the racetrack location. In that case, the GlobalRacetrackStateType reverts to the GlobalRacetrackTransitType selector and those transit achievements are then set.

Table 129: GlobalRacetrackStateType Union(s)

Type Name	Type Description
GlobalRacetrackPatternType	Indicates that the global racetrack pattern is currently executing.
GlobalRacetrackTransitType	Indicates that the vehicle is in transit to where the global racetrack pattern is to be performed.

6.2.29 GlobalRacetrackTransitType

Namespace: UMAA::MO::GlobalRacetrackState::GlobalRacetrackTransitType

Description: Indicates that the vehicle is in transit to where the global racetrack pattern is to be performed.

Table 130: GlobalRacetrackTransitType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MO::GlobalRacetrackState::GlobalRacetrackStateType		
transitElevationAchieved	BooleanEnumType	Indicates that the transit elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
transitSpeedAchieved	BooleanEnumType	Indicates that the transit speed requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

6.2.30 GlobalRegionDriftType

Namespace: UMAA::MO::GlobalDriftState::GlobalRegionDriftType

Description: Indicates that the vehicle is in the global drift region

Table 131: GlobalRegionDriftType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MO::GlobalDriftState::GlobalDriftStateType		
elevationAchieved	BooleanEnumType	Indicates that the elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
positionAchieved	BooleanEnumType	Indicates that the position requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

6.2.31 GlobalRegularPolygonPatternType

Namespace: UMAA::MO::GlobalRegularPolygonState::GlobalRegularPolygonPatternType

Description: Indicates that the global regular polygon pattern is currently executing.

Table 132: GlobalRegularPolygonPatternType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MO::GlobalRegularPolygonState::GlobalRegularPolygonStateType		
elevationAchieved	BooleanEnumType	Indicates that the elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
positionAchieved	BooleanEnumType	Indicates that the position requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
speedAchieved	BooleanEnumType	Indicates that the speed requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

6.2.32 GlobalRegularPolygonStateType

Namespace: UMAA::MO::GlobalRegularPolygonState::GlobalRegularPolygonStateType

Description: Union Type. State of the global regular polygon pattern being executed. While first transiting to the regular polygon location to be performed, the selector will be GlobalRegularPolygonTransitType until the pattern position, speed, and elevation are first achieved within their respective tolerances. Once achieved, the union selector will change to GlobalRegularPolygonPatternType. The selector will not change as a result of any of the GlobalRegularPolygonPatternType achievements states being lost and regained as a result of tolerance settings being violated. The service is expected to make driving adjustments to attempt to keep all achievement states satisfied. This is true until the service determines that tolerance(s) are violated by a sufficient margin that it is more effective for the vehicle to return to transiting to regular polygon pattern location. In that case, the GlobalRegularPolygonStateType reverts to the GlobalRegularPolygonTransitType selector and those transit achievements are then set.

Table 133: GlobalRegularPolygonStateType Union(s)

Type Name	Type Description
GlobalRegularPolygonPatternType	Indicates that the global regular polygon pattern is currently executing.
GlobalRegularPolygonTransitType	Indicates that the vehicle is in transit to where the global regular polygon pattern is to be performed.

6.2.33 GlobalRegularPolygonTransitType

Namespace: UMAA::MO::GlobalRegularPolygonState::GlobalRegularPolygonTransitType

Description: Indicates that the vehicle is in transit to where the global regular polygon pattern is to be performed.

Table 134: GlobalRegularPolygonTransitType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MO::GlobalRegularPolygonState::GlobalRegularPolygonStateType		
transitElevationAchieved	BooleanEnumType	Indicates that the transit elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
transitSpeedAchieved	BooleanEnumType	Indicates that the transit speed requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

6.2.34 GlobalTransitDriftType

Namespace: UMAA::MO::GlobalDriftState::GlobalTransitDriftType

Description: Indicates that vehicle is in transit to the global drift region

Table 135: GlobalTransitDriftType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MO::GlobalDriftState::GlobalDriftStateType		
transitElevationAchieved	BooleanEnumType	Indicates that the transit elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
transitSpeedAchieved	BooleanEnumType	Indicates that the transit speed requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

6.2.35 GlobalTransitHoverType

Namespace: UMAA::MO::GlobalHoverState::GlobalTransitHoverType

Description: Indicates that the vehicle is in transit to where the global hover is to be performed.

Table 136: GlobalTransitHoverType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MO::GlobalHoverState::GlobalHoverStateType		
elevationAchieved	BooleanEnumType	Indicates that the transit elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
speedAchieved	BooleanEnumType	Indicates that the transit speed requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

6.2.36 GlobalWaypointType

Namespace: UMAA::MO::GlobalWaypointControl::GlobalWaypointType

Description: The structure is used to describe a waypoint in a global reference frame.

Table 137: GlobalWaypointType Structure Definition

Attribute Name	Attribute Type	Attribute Description
attitude†	Orientation3D_PlatformXYZ	The desired orientation (roll, pitch, yaw) of the unmanned platform at the waypoint.
elevation	ElevationType	The optional elevation used for the unmanned maritime platform.
maintainTrack	BooleanEnumType	Indicates whether a track line is to be followed when transiting to the waypoint. Use the vehicle position at time of command to define the track for the first waypoint.
position	Position2D	The desired waypoint position (latitude, longitude) in the global coordinate system.
speed	VariableSpeedControlType	The desired waypoint travel speed of the unmanned platform with reference to the medium, the ground, the air, RPM, or true speed.
trackTolerance†	Distance	The desired tolerance of the path measured by distance.
waypointID	NumericGUID	The desired id to keep track of the waypoint.
waypointTolerance	Distance	The desired tolerance of the waypoint measured by distance.

6.2.37 HeadingCurrentDirectionType

Namespace: UMAA::Common::Orientation::HeadingCurrentDirectionType

Description: Defines the heading offset relative to heading against the current

Table 138: HeadingCurrentDirectionType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::Common::Orientation::HeadingType		
headingCurrentDirection	Heading_CurrentDirection	specifies the heading offset angle relative to heading against the current

6.2.38 HeadingMagneticNorthType

Namespace: UMAA::Common::Orientation::HeadingMagneticNorthType

Description: Defines the heading reference angle relative to magnetic north

Table 139: HeadingMagneticNorthType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::Common::Orientation::HeadingType		
headingMagneticNorth	Heading_MagneticNorth	specifies the heading reference angle relative to magnetic north

6.2.39 HeadingTrueNorthType

Namespace: UMAA::Common::Orientation::HeadingTrueNorthType

Description: Defines the heading reference angle relative to true north

Table 140: HeadingTrueNorthType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::Common::Orientation::HeadingType		
headingTrueNorth	Heading_TrueNorth_Angle	specifies the heading reference angle relative to true north

6.2.40 HeadingType

Namespace: UMAA::Common::Orientation::HeadingType

Description: Union Type. Defines the angle offset from the specified reference frame.

Table 141: HeadingType Union(s)

Type Name	Type Description
HeadingCurrentDirectionType	Defines the heading offset relative to heading against the current
HeadingMagneticNorthType	Defines the heading reference angle relative to magnetic north
HeadingTrueNorthType	Defines the heading reference angle relative to true north
HeadingWindDirectionType	Defines the heading reference angle relative to the wind direction

6.2.41 HeadingWindDirectionType

Namespace: UMAA::Common::Orientation::HeadingWindDirectionType

Description: Defines the heading reference angle relative to the wind direction

Table 142: HeadingWindDirectionType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::Common::Orientation::HeadingType		
headingWindDirection	Heading_WindDirection	specifies the heading reference angle relative to the wind direction

6.2.42 LinearEffort

Namespace: UMAA::Common::Measurement::LinearEffort

Description: Defines the along-axes efforts as a percentage.

Table 143: LinearEffort Structure Definition

Attribute Name	Attribute Type	Attribute Description
xAxis	Effort	Linear effort along the x-axis, expressed as a percentage.
yAxis	Effort	Linear effort along the y-axis, expressed as a percentage.
zAxis	Effort	Linear effort along the z-axis, expressed as a percentage.

6.2.43 LocalDriftStateType

Namespace: UMAA::MO::LocalDriftState::LocalDriftStateType

Description: Union Type. State of the local drift. While first transiting to the drift position, the selector will be LocalTransitDriftType until the position and elevation are first achieved within their respective tolerances. Once achieved, the union selector will change to LocalRegionDriftType. The selector will not change as a result of any of the LocalRegionDriftType achievements states being lost and regained as a result of tolerance settings being violated. This is true until the service determines that the elevation or drift tolerances are violated by a sufficient margin that it is more effective for the vehicle to return to transiting to the drift location. In that case, the LocalRegionDriftType reverts to the LocalTransitDriftType selector and those transit achievements then are actively set.

Table 144: LocalDriftStateType Union(s)

Type Name	Type Description
LocalRegionDriftType	Indicates that the vehicle is in the local drift region.
LocalTransitDriftType	Indicates that the vehicle is in transit to the local drift region.

6.2.44 LocalFigure8PatternType

Namespace: UMAA::MO::LocalFigure8State::LocalFigure8PatternType

Description: Indicates that the local figure 8 pattern is currently executing.

Table 145: LocalFigure8PatternType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MO::LocalFigure8State::LocalFigure8StateType		
elevationAchieved	BooleanEnumType	Indicates that the elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

Attribute Name	Attribute Type	Attribute Description
positionAchieved	BooleanEnumType	When the pattern is executing, this indicates that the position requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
speedAchieved	BooleanEnumType	When the pattern is executing, this indicates that the speed requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

6.2.45 LocalFigure8StateType

Namespace: UMAA::MO::LocalFigure8State::LocalFigure8StateType

Description: Union Type. State of the local figure 8 pattern being executed. While first transiting to the figure 8 pattern to be performed, the selector will be LocalFigure8TransitType until the pattern position, speed, and elevation are first achieved within their respective tolerances. Once achieved, the union selector will change to LocalFigure8PatternType. The selector will not change as a result of any of the LocalFigure8PatternType achievements states being lost and regained as a result of tolerance settings being violated. The service is expected to make driving adjustments to attempt to keep all achievement states satisfied. This is true until the service determines tolerance(s) are violated by a sufficient margin that it is more effective for the vehicle to return to transiting to the pattern location. In that case, the LocalFigure8StateType reverts to the LocalFigure8TransitType selector and those transit achievements are then set.

Table 146: LocalFigure8StateType Union(s)

Type Name	Type Description
LocalFigure8PatternType	Indicates that the local figure 8 pattern is currently executing.
LocalFigure8TransitType	Indicates that the vehicle is in transit to where the local figure 8 pattern is to be performed.

6.2.46 LocalFigure8TransitType

Namespace: UMAA::MO::LocalFigure8State::LocalFigure8TransitType

Description: Indicates that the vehicle is in transit to where the local figure 8 pattern is to be performed.

Table 147: LocalFigure8TransitType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MO::LocalFigure8State::LocalFigure8StateType		
transitElevationAchieved	BooleanEnumType	Indicates that the transit elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
transitSpeedAchieved	BooleanEnumType	Indicates that the transit speed requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

6.2.47 LocalHoverStateType

Namespace: UMAA::MO::LocalHoverState::LocalHoverStateType

Description: Union Type. State of the local hover. While first transiting to the hover location, the selector will be LocalTransitHoverType until the position, heading, and elevation are first achieved within their respective tolerances. Once achieved, the union selector will change to LocalHoveringHoverType. The selector will not change as a result of any of the LocalHoveringHoverType achievements states being lost and regained as a result of tolerance settings being violated. The service is expected to make driving adjustments to attempt to keep all achievement states satisfied. This is true until the service determines that tolerance(s) are violated by a sufficient margin that it is more effective for the vehicle to return to transiting to the hover location. In that case, the LocalHoverStateType reverts to the LocalTransitHoverType selector and those transit achievements then are actively set.

Table 148: LocalHoverStateType Union(s)

Type Name	Type Description
LocalHoveringHoverType	Indicates that the local hover is currently executing.
LocalTransitHoverType	Indicates that vehicle is in transit to where the local hover is to be performed.

6.2.48 LocalHoveringHoverType

Namespace: UMAA::MO::LocalHoverState::LocalHoveringHoverType

Description: Indicates that the local hover is currently executing.

Table 149: LocalHoveringHoverType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MO::LocalHoverState::LocalHoverStateType		
elevationAchieved	BooleanEnumType	Indicates that the elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
headingAchieved	BooleanEnumType	Indicates that the heading requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
positionAchieved	BooleanEnumType	Indicates that the position requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

6.2.49 LocalRacetrackPatternType

Namespace: UMAA::MO::LocalRacetrackState::LocalRacetrackPatternType

Description: Indicates that the local racetrack pattern is currently executing.

Table 150: LocalRacetrackPatternType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MO::LocalRacetrackState::LocalRacetrackStateType		

Attribute Name	Attribute Type	Attribute Description
elevationAchieved	BooleanEnumType	Indicates that the elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
positionAchieved	BooleanEnumType	Indicates that the position requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
speedAchieved	BooleanEnumType	Indicates that the speed requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

6.2.50 LocalRacetrackStateType

Namespace: UMAA::MO::LocalRacetrackState::LocalRacetrackStateType

Description: Union Type. State of the local racetrack pattern being executed. While first transiting to the racetrack location to be performed, the selector will be LocalRacetrackTransitType until the pattern position, speed, and elevation are first achieved within their respective tolerances. Once achieved, the union selector will change to LocalRacetrackPatternType. The selector will not change as a result of any of the LocalRacetrackPatternType achievements states being lost and regained as a result of tolerance settings being violated. The service is expected to make driving adjustments to attempt to keep all achievement states satisfied. This is true until the service determines that tolerance(s) are violated by a sufficient margin that it is more effective for the vehicle to return to transiting to the racetrack location. In that case, the LocalRacetrackStateType reverts to the LocalRacetrackTransitType selector and those transit achievements are then set.

Table 151: LocalRacetrackStateType Union(s)

Type Name	Type Description
LocalRacetrackPatternType	Indicates that the local racetrack pattern is currently executing.
LocalRacetrackTransitType	Indicates that the vehicle is in transit to where the local racetrack pattern is to be performed.

6.2.51 LocalRacetrackTransitType

Namespace: UMAA::MO::LocalRacetrackState::LocalRacetrackTransitType

Description: Indicates that the vehicle is in transit to where the local racetrack pattern is to be performed.

Table 152: LocalRacetrackTransitType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MO::LocalRacetrackState::LocalRacetrackStateType		
transitElevationAchieved	BooleanEnumType	Indicates that the transit elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
transitSpeedAchieved	BooleanEnumType	Indicates that the transit speed requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

6.2.52 LocalRegionDriftType

Namespace: UMAA::MO::LocalDriftState::LocalRegionDriftType

Description: Indicates that the vehicle is in the local drift region.

Table 153: LocalRegionDriftType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MO::LocalDriftState::LocalDriftStateType		
elevationAchieved	BooleanEnumType	Indicates that the elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
positionAchieved	BooleanEnumType	Indicates that the position requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

6.2.53 LocalRegularPolygonPatternType

Namespace: UMAA::MO::LocalRegularPolygonState::LocalRegularPolygonPatternType

Description: Indicates that the local regular polygon pattern is currently executing.

Table 154: LocalRegularPolygonPatternType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MO::LocalRegularPolygonState::LocalRegularPolygonStateType		
elevationAchieved	BooleanEnumType	Indicates that the elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
positionAchieved	BooleanEnumType	Indicates that the position requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
speedAchieved	BooleanEnumType	Indicates that the speed requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

6.2.54 LocalRegularPolygonStateType

Namespace: UMAA::MO::LocalRegularPolygonState::LocalRegularPolygonStateType

Description: Union Type. State of the local regular polygon pattern being executed. While first transiting to the regular polygon location to be performed, the selector will be LocalRegularPolygonTransitType until the pattern position, speed, and elevation are first achieved within their respective tolerances. Once achieved, the union selector will change to LocalRegularPolygonPatternType. The selector will not change as a result of any of the LocalRegularPolygonPatternType achievements states being lost and regained as a result of tolerance settings being violated. The service is expected to make driving adjustments to attempt to keep all achievement states satisfied. This is true until the service determines that tolerance(s) are violated by a sufficient margin that it is more effective for the vehicle to return to transiting to regular polygon pattern location. In that case, the LocalRegularPolygonStateType reverts to the LocalRegularPolygonTransitType selector

and those transit achievements are then set.

Table 155: LocalRegularPolygonStateType Union(s)

Type Name	Type Description
LocalRegularPolygonPatternType	Indicates that the local regular polygon pattern is currently executing.
LocalRegularPolygonTransitTypeType	Indicates that the vehicle is in transit to where the local regular polygon pattern is to be performed.

6.2.55 LocalRegularPolygonTransitTypeType

Namespace: UMAA::MO::LocalRegularPolygonState::LocalRegularPolygonTransitTypeType

Description: Indicates that the vehicle is in transit to where the local regular polygon pattern is to be performed.

Table 156: LocalRegularPolygonTransitTypeType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MO::LocalRegularPolygonState::LocalRegularPolygonStateType		
transitElevationAchieved	BooleanEnumType	Indicates that the transit elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
transitSpeedAchieved	BooleanEnumType	Indicates that the transit speed requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

6.2.56 LocalTransitDriftType

Namespace: UMAA::MO::LocalDriftState::LocalTransitDriftType

Description: Indicates that the vehicle is in transit to the local drift region.

Table 157: LocalTransitDriftType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MO::LocalDriftState::LocalDriftStateType		
transitElevationAchieved	BooleanEnumType	Indicates that the transit elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
transitSpeedAchieved	BooleanEnumType	Indicates that the transit speed requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

6.2.57 LocalTransitHoverType

Namespace: UMAA::MO::LocalHoverState::LocalTransitHoverType

Description: Indicates that vehicle is in transit to where the local hover is to be performed.

Table 158: LocalTransitHoverType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MO::LocalHoverState::LocalHoverStateType		
elevationAchieved	BooleanEnumType	Indicates that the transit elevation requested is within the commanded tolerance. Achievement may be lost and re-gained resulting in multiple changes to this attribute.
speedAchieved	BooleanEnumType	Indicates that the transit speed requested is within the commanded tolerance. Achievement may be lost and re-gained resulting in multiple changes to this attribute.

6.2.58 LocalWaypointType

Namespace: UMAA::MO::LocalWaypointControl::LocalWaypointType

Description: is used to set a single waypoint based on the local coordinate system. A waypoint consists of a position (X, Y, and Z-position), an orientation, and a waypoint index.

Table 159: LocalWaypointType Structure Definition

Attribute Name	Attribute Type	Attribute Description
attitude	Orientation3D_PlatformXYZ	The desired orientation (roll, pitch, yaw) of the unmanned platform at the waypoint.
elevation	ElevationType	The desired elevation used for the unmanned platform.
maintainTrack	BooleanEnumType	Indicates whether a track line is to be followed when transiting to the waypoint.
pathTolerance	Distance	The desired tolerance of the path measured by distance.
position	Position2D_PlatformXYZ	The desired waypoint position (X, Y) in the local coordinate system.
speed	VariableSpeedControlType	Speed specifier for the unmanned vehicle which may be based on explicit speed, a recommended speed, a time window, or a time window with an optional recommended speed.
waypointTolerance	Distance	The desired tolerance of the waypoint measured by distance.
waypointID*	NumericGUID	The desired id to keep track of the waypoint.

6.2.59 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 160: 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.

6.2.60 Orientation3D_PlatformXYZ

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

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

Table 161: 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.

6.2.61 Orientation3D_Tolerance

Namespace: UMAA::Common::Measurement::Orientation3D_Tolerance

Description: Realizes OrientationToleranceType: an ObservableTolerance that specifies the range of allowable values for an orientation attribute.

Table 162: Orientation3D_Tolerance Structure Definition

Attribute Name	Attribute Type	Attribute Description
lowerLimit	Orientation3D	Lower limit tolerance bounds of the angular orientation of the platform's XYZ axes relative to the NED reference frame.
stepSize	Orientation3D	Step size of the tolerance bounds of the angular orientation of the platform's XYZ axes relative to the NED reference frame.
upperLimit	Orientation3D	Upper limit tolerance bounds of the angular orientation of the platform's XYZ axes relative to the NED reference frame.

6.2.62 OrientationAcceleration3D

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

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

Table 163: 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.

6.2.63 OrientationType

Namespace: UMAA::Common::Orientation::OrientationType

Description: Union Type. Orientation of the vehicle.

Table 164: OrientationType Union(s)

Type Name	Type Description
AttitudeType	Defines the desired orientation of the unmanned platform.
DirectionType	Direction in either course the vehicle is travelling or heading the vehicle is facing.

6.2.64 OrientationVelocity3D

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

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

Table 165: 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.

6.2.65 Position2D

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

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

Table 166: 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.

6.2.66 Position2D_PlatformXYZ

Namespace: UMAA::Common::Measurement::Position2D_PlatformXYZ

Description: Describes a two dimensional plane.

Table 167: Position2D_PlatformXYZ Structure Definition

Attribute Name	Attribute Type	Attribute Description
xAxis	Forward	The x axis in the two dimensional plane.
yAxis	Right	The y axis in the two dimensional plane.

6.2.67 Quaternion

Namespace: BasicTypes::Quaternion

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

Table 168: 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.

6.2.68 RecommendedSpeedControl

Namespace: UMAA::Common::VariableSpeedControl::RecommendedSpeedControl

Description: Defines the recommended speed mode

Table 169: RecommendedSpeedControl Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::Common::VariableSpeedControl::VariableSpeedControlType		
recommendedSpeedControl	SpeedControlType	specifies the recommended speed mode

6.2.69 RequiredSpeedControl

Namespace: UMAA::Common::VariableSpeedControl::RequiredSpeedControl

Description: Defines the required speed mode

Table 170: RequiredSpeedControl Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::Common::VariableSpeedControl::VariableSpeedControlType		
requiredSpeedControl	SpeedControlType	specifies the required speed mode

6.2.70 RotationalEffort

Namespace: UMAA::Common::Measurement::RotationalEffort

Description: Describes a set of efforts around each axis as a percentage, using the right-hand rule.

Table 171: RotationalEffort Structure Definition

Attribute Name	Attribute Type	Attribute Description
pitchEffort	Effort	Rotational effort around the y-axis, expressed as a percentage.
rollEffort	Effort	Rotational effort around the x-axis, expressed as a percentage.
yawEffort	Effort	Rotational effort around the z-axis, expressed as a percentage.

6.2.71 SpeedControlType

Namespace: UMAA::Common::Speed::SpeedControlType

Description: Union Type. Speed of the unmanned vehicle

Table 172: SpeedControlType Union(s)

Type Name	Type Description
EngineRPM	Defines the engine RPM
SpeedOverGround	Defines the speed over ground
SpeedThroughAir	Defines the speed through air
SpeedThroughWater	Defines the speed through water
VehicleSpeedMode	Defines the speed mode

6.2.72 SpeedOverGround

Namespace: UMAA::Common::Speed::SpeedOverGround

Description: Defines the speed over ground

Table 173: SpeedOverGround Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::Common::Speed::SpeedControlType		
speed	GroundSpeed	specifies speed over ground

6.2.73 SpeedThroughAir

Namespace: UMAA::Common::Speed::SpeedThroughAir

Description: Defines the speed through air

Table 174: SpeedThroughAir Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::Common::Speed::SpeedControlType		
speed	IndicatedAirspeed	specifies speed relative to the air

6.2.74 SpeedThroughWater

Namespace: UMAA::Common::Speed::SpeedThroughWater

Description: Defines the speed through water

Table 175: SpeedThroughWater Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::Common::Speed::SpeedControlType		
speed	Speed_LocalWaterMass	specifies speed relative to the water

6.2.75 Speed_BSL_Capability

Namespace: UMAA::Common::Capabilities::Speed_BSL_Capability

Description: Describes the Speed_BSL capability of the vehicle.

Table 176: Speed_BSL_Capability Structure Definition

Attribute Name	Attribute Type	Attribute Description
speed	Speed_BSL	Describes the Speed_BSL capability for the vehicle.
speedDomain	Speed_BSL_Specification	Describes the required Speed_BSL for the vehicle.
speedSetPoint	Speed_BSL_Requirement	Describes the range of values for the Speed_BSL of the vehicle.

6.2.76 Speed_BSL_Requirement

Namespace: UMAA::Common::Requirements::Speed_BSL_Requirement

Description: Describes the required value of a Speed_BSL.

Table 177: Speed_BSL_Requirement Structure Definition

Attribute Name	Attribute Type	Attribute Description
speed	Speed_BSL	Describes the required value of a Speed_BSL.
speedTolerance	Speed_BSL_Tolerance	Describes the required tolerance of a Speed_BSL.

6.2.77 Speed_BSL_Specification

Namespace: UMAA::Common::MeasurementSpecifications::Speed_BSL_Specification

Description: Describes the range of values for a Speed_BSL.

Table 178: Speed_BSL_Specification Structure Definition

Attribute Name	Attribute Type	Attribute Description
lowerLimit	sequence<Speed_BSL>	Describes the lower limit of values for a Speed_BSL.
stepSize	Speed_BSL	Describes the step size of values for a Speed_BSL.
upperLimit	sequence<Speed_BSL>	Describes the upper limit of values for a Speed_BSL.

6.2.78 Speed_BSL_Tolerance

Namespace: UMAA::Common::MeasurementTolerances::Speed_BSL_Tolerance

Description: Describes the tolerance for a Speed_BSL.

Table 179: Speed_BSL_Tolerance Structure Definition

Attribute Name	Attribute Type	Attribute Description
lowerLimit	Speed_BSL	Describes the lower limit of the tolerance for a Speed_BSL.
stepSize	Speed_BSL	Describes the step size of the tolerance for a Speed_BSL.
upperLimit	Speed_BSL	Describes the upper limit of the tolerance for a Speed_BSL.

6.2.79 StationkeepStateType

Namespace: UMAA::MO::StationkeepState::StationkeepStateType

Description: Union Type. State of the station keeping being performed. While first transiting to the station keeping position, the selector will be StationkeepingTransitType until the station keep range and bearing to target are first achieved within their respective tolerances. Once achieved, the union selector will change to StationkeepingStationkeepType. The selector will not change as a result of any of the StationkeepingStationkeepType achievements states being lost and regained as a result of tolerance settings being violated. The service is expected to make driving adjustments to attempt to keep all achievement states satisfied. This is true until the service determines that tolerance(s) are violated by a sufficient margin that it is more effective for the vehicle to return to transiting to the station keeping position. In that case, the StationkeepStateType reverts to the StationkeepingTransitType selector and those transit achievements are then set.

Table 180: StationkeepStateType Union(s)

Type Name	Type Description
StationkeepingStationkeepType	Indicates that the station keeping is currently executing.
TransitStationkeepType	Indicates that the vehicle is in transit to where the station keeping is to be performed.

6.2.80 StationkeepingStationkeepType

Namespace: UMAA::MO::StationkeepState::StationkeepingStationkeepType

Description: Indicates that the station keeping is currently executing.

Table 181: StationkeepingStationkeepType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MO::StationkeepState::StationkeepStateType		
bearingToContactAchieved	BooleanEnumType	When the station keeping is executing, this indicates that the contact bearing requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
elevationAchieved	BooleanEnumType	Indicates that the elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
rangeAchieved	BooleanEnumType	When the station keeping is executing, this indicates that the range requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

6.2.81 TimeWindowType

Namespace: UMAA::Common::TimeWindow::TimeWindowType

Description: Represents a range of time.

Table 182: TimeWindowType Structure Definition

Attribute Name	Attribute Type	Attribute Description
end	DateTime	End of the time window, inclusive.
start	DateTime	Start of the time window, inclusive.

6.2.82 TimeWithSpeed

Namespace: UMAA::Common::VariableSpeedControl::TimeWithSpeed

Description: Defines the time window and the recommended speed of an unmanned vehicle

Table 183: TimeWithSpeed Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::Common::VariableSpeedControl::VariableSpeedControlType		
recommendedSpeed†	SpeedControlType	specifies the recommended speed of the waypoint
timeWindow	TimeWindowType	specifies the time window of the waypoint

6.2.83 TransitStationkeepType

Namespace: UMAA::MO::StationkeepState::TransitStationkeepType

Description: Indicates that the vehicle is in transit to where the station keeping is to be performed.

Table 184: TransitStationkeepType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::MO::StationkeepState::StationkeepStateType		
transitElevationAchieved	BooleanEnumType	When in transit, this indicates whether the transit elevation requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.
transitSpeedAchieved	BooleanEnumType	When in transit, this indicates whether the transit speed requested is within the commanded tolerance. Achievement may be lost and regained resulting in multiple changes to this attribute.

6.2.84 VariableSpeedControlType

Namespace: UMAA::Common::VariableSpeedControl::VariableSpeedControlType

Description: Union Type. Speed specifier for the unmanned vehicle which may be based on explicit speed, a recommended speed, a time window, or a time window with an optional recommended speed.

Table 185: VariableSpeedControlType Union(s)

Type Name	Type Description
RecommendedSpeedControl	Defines the recommended speed mode
RequiredSpeedControl	Defines the required speed mode
TimeWithSpeed	Defines the time window and the recommended speed of an unmanned vehicle

6.2.85 VehicleSpeedMode

Namespace: UMAA::Common::Speed::VehicleSpeedMode

Description: Defines the speed mode

Table 186: VehicleSpeedMode Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::Common::Speed::SpeedControlType		
mode	VehicleSpeedModeEnumType	specifies the speed mode

6.2.86 Velocity3D_PlatformXYZ

Namespace: UMAA::Common::Measurement::Velocity3D_PlatformXYZ

Description: Velocity3D_PlatformXYZ specifies the velocity given by forward, right, and down vectors in an XYZ coordinate system centered on the platform.

Table 187: Velocity3D_PlatformXYZ Structure Definition

Attribute Name	Attribute Type	Attribute Description
downSpeed	DownSpeed	downSpeed specifies the down velocity vector in an XYZ coordinate system centered on the platform.
forwardSpeed	ForwardSpeed	forwardSpeed specifies the forward velocity vector in an XYZ coordinate system centered on the platform.
rightSpeed	RightSpeed	rightSpeed specifies the right velocity vector in an XYZ coordinate system centered on the platform.

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 BearingAngleEnumType

Namespace: UMAA::Common::MaritimeEnumeration::BearingAngleEnumType

Description: Defines a mutually exclusive set of values for the type of bearing angle.

Table 188: BearingAngleEnumType Enumeration

Enumeration Value	Description
OWNSHIP	Angle is relative to ownship
NORTH	Angle is relative to true north

6.3.2 CommandStatusReasonEnumType

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

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

Table 189: 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.

6.3.3 ContactManeuverInfluenceEnumType

Namespace: UMAA::Common::MaritimeEnumeration::ContactManeuverInfluenceEnumType

Description: A mutually exclusive set of values that defines the maneuver of a vessel in response to a contact.

Table 190: ContactManeuverInfluenceEnumType Enumeration

Enumeration Value	Description
BEING_OVERTAKEN_COMPLIANT	COLREGS being overtaken where the other vessel is determined to be compliant
BEING_OVERTAKEN_NONCOMPLIANT	COLREGS being overtaken where the other vessel is determined to be non-compliant
CROSSING_LEFT_COMPLIANT	COLREGS crossing left where the other vessel is determined to be compliant
CROSSING_LEFT_NONCOMPLIANT	COLREGS crossing left where the other vessel is determined to be non-compliant
CROSSING_RIGHT_COMPLIANT	COLREGS crossing right where the other vessel is determined to be compliant
CROSSING_RIGHT_NONCOMPLIANT	COLREGS crossing right where the other vessel is determined to be non-compliant
HEAD_ON_COMPLIANT	COLREGS head on where the other vessel is determined to be compliant
HEAD_ON_NONCOMPLIANT	COLREGS head on where the other vessel is determined to be non-compliant
OVERTAKING_COMPLIANT	COLREGS overtaking where the other vessel is determined to be compliant
OVERTAKING_NONCOMPLIANT	COLREGS overtaking where the other vessel is determined to be non-compliant
GUIDE	Contact is guiding or informing maneuvering (e.g., guide vessel for Stationkeep, cooperating swarm member)
IN_EXTREMIS	Determined in a situation where collision can no longer be avoided by one ship acting alone
COLLISION_AVOIDANCE	Maneuvering to avoid a dynamic obstacle
PREEMPTIVE	Maneuvering to avoid a perceived future state but not in direct response to configured obstacle avoidance thresholds
OBSTACLE_AVOIDANCE	Maneuvering to avoid a static obstacle
NONE	The contact has been examined and it was determined it has no influence on the maneuvering

6.3.4 HoverKindEnumType

Namespace: UMAA::Common::MaritimeEnumeration::HoverKindEnumType

Description: A mutually exclusive set of values that defines the loitering priority of the unmanned platform.

Table 191: HoverKindEnumType Enumeration

Enumeration Value	Description
LAT_LON_PRIORITY	Prioritize maintaining a latitude/longitude position
Z_PRIORITY	Prioritize maintaining an elevation

6.3.5 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 192: 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.

6.3.6 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 193: VehicleSpeedModeEnumType Enumeration

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

6.3.7 VelocityCommandTypeEnumType

Namespace: UMAA::Common::MaritimeEnumeration::VelocityCommandTypeEnumType

Description: A mutually exclusive set of values that defines the types of velocity command to the unmanned platform.

Table 194: VelocityCommandTypeEnumType Enumeration

Enumeration Value	Description
CURRENT_COMMAND_SOG	Current Command Speed Over Ground
CURRENT_COMMAND_SRM	Current Command Speed Relative to Medium
DEFAULT_COMMAND_SOG	Default Speed Over Ground command
DEFAULT_COMMAND_SRM	Default Speed Relative to Medium command
MAX_ALLOWED_SOG	Maximum Allowed Speed Over Ground
MAX_ALLOWED_SRM	Maximum Allowed Speed Relative to Medium
MIN_ALLOWED_SOG	Minimum Allowed Speed Over Ground
MIN_ALLOWED_SRM	Minimum Allowed Speed Relative to Medium

6.3.8 WaterTurnDirectionEnumType

Namespace: UMAA::Common::MaritimeEnumeration::WaterTurnDirectionEnumType

Description: A mutually exclusive set of values that define the types of turn directions applied by the vehicle during turns.

Table 195: WaterTurnDirectionEnumType Enumeration

Enumeration Value	Description
NO_VALID_TURN_DIRECTION	No valid turn direction is specified for the vehicle.
LEFT_TURN	The vehicle will make left turns.
RIGHT_TURN	The vehicle will make right turns.
VEHICLE_SPECIFIC	The vehicle will make turns as dictated by the vehicle's specific behavior.
INTO_THE_CURRENT	The vehicle will make turns into the current.
INTO_THE_WIND	The vehicle will make turns into the wind.

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 196: 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=3.141592653589 7932384626433832795 minInclusive=-3.141592653589 7931264626433832795 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.
AngleRate	double	units=RadianPerSecond minInclusive=-62.831 maxInclusive=62.831 fractionDigits=3	Represents the rate of change of angular displacement measured in radians per second.
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.
Count	long	units=N/A minInclusive=-2147483648 maxInclusive=2147483647 fractionDigits=0	Represents a whole (non-fractional) number that can be positive, negative or zero.
Course_TrueNorth	double	fractionDigits=3 maxInclusive=6.283185307179 586364925286766559 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.
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=18446744073709 500000 fractionDigits=0	seconds offset from the standard POSIX (IEEE Std 1003.1-2017) epoch reference point of January 1st, 1970 00:00:00 UTC.
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.
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.
Effort	double	fractionDigits=3 maxInclusive=100 minInclusive=-100 units=Percent referenceFrame=PlatformXYZ	Represents the level of effort measured in percent.
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.
ForwardSpeed	double	axisDirection=fore axisUnit=MeterPerSecond maximumValue=-10000 minimumValue=10000 rangeMeaning=exact resolution=0.000001	The ForwardSpeed axis is used for measuring speed and increases in magnitude as speed out the "front" of the reference body increases. ForwardSpeed measurements are expressed in meters per second.
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.

Type Name	Primitive Type	Range of Values	Description
GroundSpeed	double	units=MeterPerSecond minInclusive=0 maxInclusive=200 fractionDigits=6	This type stores speed in meters/s.
Heading_CurrentDirection	double	units=Radian referenceFrame=CurrentDirection minInclusive=-3.1415926535897932626433832795 maxInclusive=3.1415926535897932626433832795 fractionDigits=3	Describes heading as a value between -pi and pi with respect to the current direction.
Heading_MagneticNorth	double	units=Radian referenceFrame=MagneticNorth minInclusive=-3.1415926535897932626433832795 maxInclusive=3.1415926535897932626433832795 fractionDigits=3	Describes heading as a value between -pi and pi with respect to Magnetic North.
Heading_TrueNorth_Angle	double	units=Radian referenceFrame=TrueNorth minInclusive=-3.1415926535897932626433832795 maxInclusive=3.1415926535897932626433832795 fractionDigits=3	Describes heading as a value between -pi and pi with respect to True North.
Heading_WindDirection	double	units=Radian referenceFrame=WindDirection minInclusive=-3.1415926535897932626433832795 maxInclusive=3.1415926535897932626433832795 fractionDigits=3	Describes heading as a value between -pi and pi with respect to the wind direction.
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.
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.
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

Type Name	Primitive Type	Range of Values	Description
Pitch_HalfAngle	double	fractionDigits=3 maxInclusive=1.570796326794 8966192313216916398 minInclusive=-1.570796326794 8966192313216916398 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=RadianPerSecondSquare d 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.
RelativeAngle	double	fractionDigits=3 maxInclusive=3.141592653589 7932384626433832795 minInclusive=-3.141592653589 7931264626433832795 units=Radian referenceFrame=Counting	RelativeAngle specifies the angle between two intersecting rays. The measurement is stated in radians between -pi and pi.
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.
RightSpeed	double	axisDirection=starboard axisUnit=MeterPerSecond maximumValue=-10000 minimumValue=10000 rangeMeaning=exact resolution=0.000001	The RightSpeed axis is used for measuring speed and increases in magnitude as speed out the "right" of the reference body increases. RightSpeed measurements are expressed in meters per second.

Type Name	Primitive Type	Range of Values	Description
Roll_Angle	double	fractionDigits=3 maxInclusive=3.1415926535897932384626433832795 minInclusive=-3.1415926535897932384626433832795 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.
SidesCount	long	units=N/A minInclusive=3 maxInclusive=255 fractionDigits=0	Represents the number of sides a polygon has using a positive integer.
Speed_BSL	double	fractionDigits=6 units=MeterPerSecond referenceFrame=BSL	This type stores speed in meters/s in a below sea level reference frame.
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.
Yaw_PosAngle	double	fractionDigits=3 maxInclusive=6.283185307179586364925286766559 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.

Type Name	Primitive Type	Range of Values	Description
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.
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