Unmanned Maritime Autonomy Architecture (UMAA)
Engineering Operations (EO)
Interface Control Document (ICD)
(UMAA-SPEC-EOICD)

MDE Version 4.2.1 Commit 7c2b513

UMAA Spec Commit f2fa426

Version 3.0.1
25 February 2021

UMAA Engineering Operations ICD

(UMAA-SPEC-EO-ICD)

Signature Page

Submitted by: Rothgeb.Mark.E.ORC3011008580.ID Digitally signed by Rothgeb.Mark.E.ORC3011008580.ID Date: 2021.05.05 11:49:36 -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 14:48:39 -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 17:16:00 -04'00' **Date Signed:** _____

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

Contents

1 Scope **11**
- 1.1 Identification . . . 11
- 1.2 Overview . . . 11
- 1.3 Document Organization . . . 13

2 Referenced Documents **14**

3 Introduction to Data Model, Services, and Interfaces **15**
- 3.1 Data Model . . . 15
- 3.2 Definitions . . . 15
- 3.3 Data Distribution Service (DDS™) . . . 15
- 3.4 Naming Conventions . . . 16
- 3.5 Namespace Conventions . . . 17
- 3.6 Cybersecurity . . . 17
- 3.7 GUID algorithm . . . 18
- 3.8 Large Sets . . . 18

4 Introduction to Coordinate Reference Frames and Position Model **19**
- 4.1 Platform Reference Frame . . . 19
- 4.2 Platform Orientation . . . 19
- 4.3 Vehicle Coordinate Reference Frame Origin . . . 21

5 Flow Control **23**
- 5.1 Command / Response . . . 23
  - 5.1.1 High-Level Flow . . . 24
  - 5.1.2 Command Startup Sequence . . . 25
    - 5.1.2.1 Service Provider Startup Sequence . . . 26
    - 5.1.2.2 Service Consumer Startup Sequence . . . 27
  - 5.1.3 Command Execution Sequences . . . 27
  - 5.1.4 Command Start Sequence . . . 28
    - 5.1.4.1 Command Execution . . . 28
    - 5.1.4.2 Command Execution Success . . . 29
    - 5.1.4.3 Command Execution Failure . . . 30
    - 5.1.4.4 Command Canceled . . . 31
  - 5.1.5 Command Cleanup . . . 32
  - 5.1.6 Command Shutdown Sequence . . . 33
    - 5.1.6.1 Service Provider Shutdown Sequence . . . 33
    - 5.1.6.2 Service Consumer Shutdown Sequence . . . 34
- 5.2 Request / Reply . . . 35
  - 5.2.1 Request/Reply without Query Data . . . 35
    - 5.2.1.1 Service Provider Startup Sequence . . . 36
    - 5.2.1.2 Service Consumer Startup Sequence . . . 36
    - 5.2.1.3 Service Provider Shutdown . . . 36
    - 5.2.1.4 Service Consumer Shutdown . . . 36
  - 5.2.2 Request/Reply with Query Data . . . 37

6 Engineering Operations (EO) Services and Interfaces **38**
- 6.1 Services and Interfaces . . . 38
  - 6.1.1 AnchorControl . . . 38
    - 6.1.1.1 reportAnchorCommandAck . . . 38
    - 6.1.1.2 reportAnchorCommandStatus . . . 39
    - 6.1.1.3 setAnchor . . . 39
  - 6.1.2 AnchorSpecs . . . 39
    - 6.1.2.1 reportAnchorSpecs . . . 40
  - 6.1.3 AnchorStatus . . . 40
    - 6.1.3.1 reportAnchor . . . 41

6.1.4 BallastTank . . . 41
6.1.4.1 reportBallastPump . . . 42
6.1.4.2 reportBallastPumpCommand . . . 42
6.1.4.3 reportBallastPumpCommandStatus . . . 42
6.1.4.4 reportBallastPumpSpecs . . . 43
6.1.4.5 reportBallastTank . . . 43
6.1.4.6 reportBallastTankCommand . . . 44
6.1.4.7 reportBallastTankCommandStatus . . . 44
6.1.4.8 reportBallastTankSpecs . . . 45
6.1.4.9 setBallastPump . . . 45
6.1.4.10 setBallastTank . . . 45
6.1.5 BatterySpecs . . . 46
6.1.5.1 reportBatterySpecs . . . 46
6.1.6 BatteryStatus . . . 47
6.1.6.1 reportBattery . . . 47
6.1.7 BellControl . . . 47
6.1.7.1 reportBellControl . . . 48
6.1.7.2 reportBellControlCommand . . . 48
6.1.7.3 reportBellControlCommandStatus . . . 49
6.1.7.4 setBellControl . . . 49
6.1.8 EngineControl . . . 49
6.1.8.1 reportEngineCommandAck . . . 50
6.1.8.2 reportEngineCommandStatus . . . 50
6.1.8.3 setEngine . . . 51
6.1.9 EngineSpecs . . . 51
6.1.9.1 reportEngineSpecs . . . 51
6.1.10 EngineStatus . . . 52
6.1.10.1 reportEngine . . . 52
6.1.11 FuelTankSpecs . . . 53
6.1.11.1 reportFuelTankSpecs . . . 53
6.1.12 FuelTankStatus . . . 54
6.1.12.1 reportFuelTank . . . 54
6.1.13 GeneratorSpecs . . . 54
6.1.13.1 reportGeneratorSpecs . . . 55
6.1.14 GeneratorStatus . . . 55
6.1.14.1 reportGenerator . . . 55
6.1.15 GongControl . . . 56
6.1.15.1 reportGongControl . . . 56
6.1.15.2 reportGongControlCommand . . . 57
6.1.15.3 reportGongControlCommandStatus . . . 57
6.1.15.4 setGongControl . . . 57
6.1.16 LightControl . . . 58
6.1.16.1 reportLightCommandAck . . . 58
6.1.16.2 reportLightCommandStatus . . . 59
6.1.16.3 setLight . . . 59
6.1.17 LightSpecs . . . 60
6.1.17.1 reportLightSpecs . . . 60
6.1.18 LightStatus . . . 60
6.1.18.1 reportLight . . . 61
6.1.19 MastControl . . . 61
6.1.19.1 reportMastCommandAck . . . 61
6.1.19.2 reportMastCommandStatus . . . 62
6.1.19.3 setMast . . . 62
6.1.20 MastStatus . . . 62
6.1.20.1 reportMast . . . 63
6.1.21 PowerControl . . . 63
6.1.21.1 reportPowerCommandAck . . . 63
6.1.21.2 reportPowerCommandStatus . . . 64

6.1.21.3 setPower . . . 64
6.1.22 PowerStatus . . . 65
6.1.22.1 reportPower . . . 65
6.1.23 PropulsorConfig . . . 65
6.1.23.1 reportPropulsorConfig . . . 65
6.1.24 PropulsorControl . . . 66
6.1.24.1 reportPropulsorCommandAck . . . 66
6.1.24.2 reportPropulsorCommandStatus . . . 67
6.1.24.3 setPropulsor . . . 67
6.1.25 PropulsorSpecs . . . 67
6.1.25.1 reportPropulsorSpecs . . . 68
6.1.26 PropulsorStatus . . . 68
6.1.26.1 reportPropulsor . . . 68
6.1.27 RudderControl . . . 69
6.1.27.1 reportRudderCommandAck . . . 69
6.1.27.2 reportRudderCommandStatus . . . 70
6.1.27.3 setRudder . . . 70
6.1.28 RudderSpecs . . . 70
6.1.28.1 reportRudderSpecs . . . 71
6.1.29 RudderStatus . . . 71
6.1.29.1 reportRudder . . . 71
6.1.30 SituationalSignalControl . . . 72
6.1.30.1 reportSituationalSignalCommandAck . . . 72
6.1.30.2 reportSituationalSignalCommandStatus . . . 73
6.1.30.3 setSituationalSignal . . . 73
6.1.31 SituationalSignalSpecs . . . 74
6.1.31.1 reportSituationalSignalSpecs . . . 75
6.1.32 SituationalSignalStatus . . . 76
6.1.32.1 reportSituationalSignal . . . 76
6.1.33 ThrusterControl . . . 77
6.1.33.1 reportThrusterCommandAck . . . 77
6.1.33.2 reportThrusterCommandStatus . . . 78
6.1.33.3 setThruster . . . 78
6.1.34 ThrusterSpecs . . . 79
6.1.34.1 reportThrusterSpecs . . . 79
6.1.35 ThrusterStatus . . . 79
6.1.35.1 reportThruster . . . 79
6.1.36 UVPlatformSpecs . . . 80
6.1.36.1 reportUVPlatformCapabilities . . . 80
6.1.36.2 reportUVPlatformSpecs . . . 81
6.1.37 WhistleControl . . . 82
6.1.37.1 reportWhistleControl . . . 82
6.1.37.2 reportWhistleControlCommand . . . 83
6.1.37.3 reportWhistleControlCommandStatus . . . 83
6.1.37.4 setWhistleControl . . . 83
6.2 Common Data Types . . . 85
6.2.1 UCSMDEInterfaceSet . . . 85
6.2.2 UMAACommand . . . 85
6.2.3 UMAAStatus . . . 85
6.2.4 UMAACommandStatusBase . . . 86
6.2.5 UMAACommandStatus . . . 86
6.2.6 DateTime . . . 86
6.2.7 BallastFillType . . . 87
6.2.8 BallastMassType . . . 87
6.2.9 BallastPumpFlowRateType . . . 87
6.2.10 BatteryCellDataType . . . 87
6.2.11 BlastPropertiesType . . . 88
6.2.12 LevelType . . . 88

6.2.13 MassBallastFlowRateType . . . 88
6.2.14 Position3D_PlatformXYZ . . . 89
6.2.15 PropulsionRotationType . . . 89
6.2.16 PropulsionType . . . 89
6.2.17 PropulsiveEffortType . . . 90
6.2.18 PropulsiveRPMType . . . 90
6.2.19 PropulsiveRotationAngleType . . . 90
6.2.20 PropulsiveRotationEffortType . . . 90
6.2.21 Quaternion . . . 91
6.2.22 RotationalEffort . . . 91
6.2.23 ThrusterControlType . . . 91
6.2.24 ThrusterSpecsType . . . 92
6.2.25 UVPlatformCapabilitiesType . . . 92
6.2.26 VolumeBallastFlowRateType . . . 92
6.2.27 WhistlePropertiesType . . . 93
6.3 Enumerations . . . 94
6.3.1 AnchorActionEnumType . . . 94
6.3.2 AnchorKindEnumType . . . 94
6.3.3 AnchorLocationEnumType . . . 94
6.3.4 AnchorStateEnumType . . . 95
6.3.5 BlastConditionEnumType . . . 95
6.3.6 BlastKindEnumType . . . 95
6.3.7 CommandStatusReasonEnumType . . . 96
6.3.8 EngineKindEnumType . . . 96
6.3.9 IgnitionStateEnumType . . . 97
6.3.10 MastActionEnumType . . . 97
6.3.11 MastStateEnumType . . . 97
6.3.12 CommandStatusEnumType . . . 97
6.3.13 MountModeEnumType . . . 98
6.3.14 OnOffStatusEnumType . . . 98
6.3.15 PowerPlantStateEnumType . . . 99
6.3.16 PumpStateEnumType . . . 99
6.4 Type Definitions . . . 100

A Appendices **104**
A.1 Acronyms . . . 104

List of Figures

1 UMAA Functional Organization . . . 11
2 UMAA Services and Interfaces Example . . . 12
3 Services and Interfaces Exposed on the UMAA Data Bus . . . 15
4 Given a vehicle in arbitrary orientation . . . 19
5 Align the vehicle with the reference frame axes . . . 20
6 Rotate the vehicle by the Yaw angle . . . 20
7 Rotate the vehicle by the Pitch angle . . . 21
8 Rotate the vehicle by the Roll angle . . . 21
9 Origin location on a USV as example . . . 22
10 Origin location on a UUV as example . . . 22
11 The state transitions of the `commandStatus` as commands are processed. Labels on the arrows represent valid `commandStatusReason` values for each transition. . . . 24
12 The sequence diagram for the high-level description of a command exeuction. . . . 25
13 The sequence diagram for command startup. . . . 26
14 The sequence diagram for command startup for Service Providers. . . . 26
15 The sequence diagram for command startup for Service Consumers. . . . 27
16 The sequence diagram for the start of a command execution. . . . 28
17 The beginning sequence diagram for a command execution. . . . 29
18 The sequence diagram for a command that completes successfully. . . . 30

19 The sequence diagram for a command that fails due to Resource failure. . . . 30
20 The sequence diagram for a command that times out before completing. . . . 31
21 The sequence diagram for a command that is canceled by the Service Consumer before the Service Provider is able to complete it. . . . 32
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. . . . 33
23 The sequence diagram for command shutdown. . . . 33
24 The sequence diagram for command shutdown for Service Providers. . . . 34
25 The sequence diagram for command shutdown for Service Consumers. . . . 34
26 The sequence diagram for a request/reply for report data that does not require any specific query data. . . . 35
27 The sequence diagram for initialization of a Service Provider to provide FunctionReportTypes. . . . 36
28 The sequence diagram for initialization of a Service Consumer to request FunctionReportTypes. . . . 36
29 The sequence diagram for shutdown of a Service Provider. . . . 36
30 The sequence diagram for shutdown of a Service Consumer. . . . 37

List of Tables

3 Standards Documents . . . 14
4 Government Documents . . . 14
5 Service Requests and Associated Responses . . . 16
6 AnchorControl Operations . . . 38
7 AnchorCommandAckReportType Message Definition . . . 39
8 AnchorCommandStatusType Message Definition . . . 39
9 AnchorCommandType Message Definition . . . 39
10 AnchorSpecs Operations . . . 40
11 AnchorSpecsReportType Message Definition . . . 40
12 AnchorStatus Operations . . . 41
13 AnchorReportType Message Definition . . . 41
14 BallastTank Operations . . . 41
15 BallastPumpReportType Message Definition . . . 42
16 BallastPumpCommandReportType Message Definition . . . 42
17 BallastPumpCommandStatusType Message Definition . . . 43
18 BallastPumpSpecsReportType Message Definition . . . 43
19 BallastTankReportType Message Definition . . . 44
20 BallastTankCommandReportType Message Definition . . . 44
21 BallastTankCommandStatusType Message Definition . . . 44
22 BallastTankSpecsReportType Message Definition . . . 45
23 BallastPumpCommandType Message Definition . . . 45
24 BallastTankCommandType Message Definition . . . 46
25 BatterySpecs Operations . . . 46
26 BatterySpecsReportType Message Definition . . . 46
27 BatteryStatus Operations . . . 47
28 BatteryReportType Message Definition . . . 47
29 BellControl Operations . . . 48
30 BellControlReportType Message Definition . . . 48
31 BellControlCommandReportType Message Definition . . . 48
32 BellControlCommandStatusType Message Definition . . . 49
33 BellControlCommandType Message Definition . . . 49
34 EngineControl Operations . . . 50
35 EngineCommandAckReportType Message Definition . . . 50
36 EngineCommandStatusType Message Definition . . . 50
37 EngineCommandType Message Definition . . . 51
38 EngineSpecs Operations . . . 51
39 EngineSpecsReportType Message Definition . . . 51
40 EngineStatus Operations . . . 52
41 EngineReportType Message Definition . . . 53
42 FuelTankSpecs Operations . . . 53

43 FuelTankSpecsReportType Message Definition . . . 54
44 FuelTankStatus Operations . . . 54
45 FuelTankReportType Message Definition . . . 54
46 GeneratorSpecs Operations . . . 55
47 GeneratorSpecsReportType Message Definition . . . 55
48 GeneratorStatus Operations . . . 55
49 GeneratorReportType Message Definition . . . 56
50 GongControl Operations . . . 56
51 GongControlReportType Message Definition . . . 56
52 GongControlCommandReportType Message Definition . . . 57
53 GongControlCommandStatusType Message Definition . . . 57
54 GongControlCommandType Message Definition . . . 58
55 LightControl Operations . . . 58
56 LightCommandAckReportType Message Definition . . . 58
57 LightCommandStatusType Message Definition . . . 59
58 LightCommandType Message Definition . . . 59
59 LightSpecs Operations . . . 60
60 LightSpecsReportType Message Definition . . . 60
61 LightStatus Operations . . . 60
62 LightReportType Message Definition . . . 61
63 MastControl Operations . . . 61
64 MastCommandAckReportType Message Definition . . . 62
65 MastCommandStatusType Message Definition . . . 62
66 MastCommandType Message Definition . . . 62
67 MastStatus Operations . . . 63
68 MastReportType Message Definition . . . 63
69 PowerControl Operations . . . 63
70 PowerCommandAckReportType Message Definition . . . 64
71 PowerCommandStatusType Message Definition . . . 64
72 PowerCommandType Message Definition . . . 64
73 PowerStatus Operations . . . 65
74 PowerReportType Message Definition . . . 65
75 PropulsorConfig Operations . . . 65
76 PropulsorConfigReportType Message Definition . . . 66
77 PropulsorControl Operations . . . 66
78 PropulsorCommandAckReportType Message Definition . . . 66
79 PropulsorCommandStatusType Message Definition . . . 67
80 PropulsorCommandType Message Definition . . . 67
81 PropulsorSpecs Operations . . . 67
82 PropulsorSpecsReportType Message Definition . . . 68
83 PropulsorStatus Operations . . . 68
84 PropulsorReportType Message Definition . . . 69
85 RudderControl Operations . . . 69
86 RudderCommandAckReportType Message Definition . . . 70
87 RudderCommandStatusType Message Definition . . . 70
88 RudderCommandType Message Definition . . . 70
89 RudderSpecs Operations . . . 71
90 RudderSpecsReportType Message Definition . . . 71
91 RudderStatus Operations . . . 71
92 RudderReportType Message Definition . . . 72
93 SituationalSignalControl Operations . . . 72
94 SituationalSignalCommandAckReportType Message Definition . . . 72
95 SituationalSignalCommandStatusType Message Definition . . . 73
96 SituationalSignalCommandType Message Definition . . . 74
97 SituationalSignalSpecs Operations . . . 75
98 SituationalSignalSpecsReportType Message Definition . . . 75
99 SituationalSignalStatus Operations . . . 76
100 SituationalSignalReportType Message Definition . . . 76

101 ThrusterControl Operations . . . 77
102 ThrusterCommandAckReportType Message Definition . . . 78
103 ThrusterCommandStatusType Message Definition . . . 78
104 ThrusterCommandType Message Definition . . . 78
105 ThrusterSpecs Operations . . . 79
106 ThrusterSpecsReportType Message Definition . . . 79
107 ThrusterStatus Operations . . . 79
108 ThrusterReportType Message Definition . . . 80
109 UVPlatformSpecs Operations . . . 80
110 UVPlatformCapabilitiesReportType Message Definition . . . 80
111 UVPlatformSpecsReportType Message Definition . . . 81
112 WhistleControl Operations . . . 82
113 WhistleControlReportType Message Definition . . . 82
114 WhistleControlCommandReportType Message Definition . . . 83
115 WhistleControlCommandStatusType Message Definition . . . 83
116 WhistleControlCommandType Message Definition . . . 84
117 UCSMDEInterfaceSet Structure Definition . . . 85
118 UMAACommand Structure Definition . . . 85
119 UMAAStatus Structure Definition . . . 85
120 UMAACommandStatusBase Structure Definition . . . 86
121 UMAACommandStatus Structure Definition . . . 86
122 DateTime Structure Definition . . . 86
123 BallastFillType Union(s) . . . 87
124 BallastMassType Structure Definition . . . 87
125 BallastPumpFlowRateType Union(s) . . . 87
126 BatteryCellDataType Structure Definition . . . 88
127 BlastPropertiesType Structure Definition . . . 88
128 LevelType Structure Definition . . . 88
129 MassBallastFlowRateType Structure Definition . . . 88
130 Position3D_PlatformXYZ Structure Definition . . . 89
131 PropulsionRotationType Union(s) . . . 89
132 PropulsionType Union(s) . . . 89
133 PropulsiveEffortType Structure Definition . . . 90
134 PropulsiveRPMType Structure Definition . . . 90
135 PropulsiveRotationAngleType Structure Definition . . . 90
136 PropulsiveRotationEffortType Structure Definition . . . 91
137 Quaternion Structure Definition . . . 91
138 RotationalEffort Structure Definition . . . 91
139 ThrusterControlType Structure Definition . . . 92
140 ThrusterSpecsType Structure Definition . . . 92
141 UVPlatformCapabilitiesType Structure Definition . . . 92
142 VolumeBallastFlowRateType Structure Definition . . . 93
143 WhistlePropertiesType Structure Definition . . . 93
144 AnchorActionEnumType Enumeration . . . 94
145 AnchorKindEnumType Enumeration . . . 94
146 AnchorLocationEnumType Enumeration . . . 95
147 AnchorStateEnumType Enumeration . . . 95
148 BlastConditionEnumType Enumeration . . . 95
149 BlastKindEnumType Enumeration . . . 96
150 CommandStatusReasonEnumType Enumeration . . . 96
151 EngineKindEnumType Enumeration . . . 96
152 IgnitionStateEnumType Enumeration . . . 97
153 MastActionEnumType Enumeration . . . 97
154 MastStateEnumType Enumeration . . . 97
155 CommandStatusEnumType Enumeration . . . 98
156 MountModeEnumType Enumeration . . . 98
157 OnOffStatusEnumType Enumeration . . . 98
158 PowerPlantStateEnumType Enumeration . . . 99

159 PumpStateEnumType Enumeration 99
160 Type Definitions 100

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 interact with the Hull, Mechanical & Electrical (HM&E) systems of an Unmanned Maritime Vehicle (UMV) (surface or undersea). As such, it includes low-level control and/or status of individual components on the vehicle such as the propulsor, battery, engine, gong, and rudder. It includes physical constraints and specifications of the system, and its components. Also, it includes fault handling and health of the vehicle. Services provided here may be invoked by the Maneuver Operations ICD services depending on how those higher level services are implemented (i.e. either using intrinsic vehicle capabilities or interacting at the component level with the UMV component services included herein). 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.

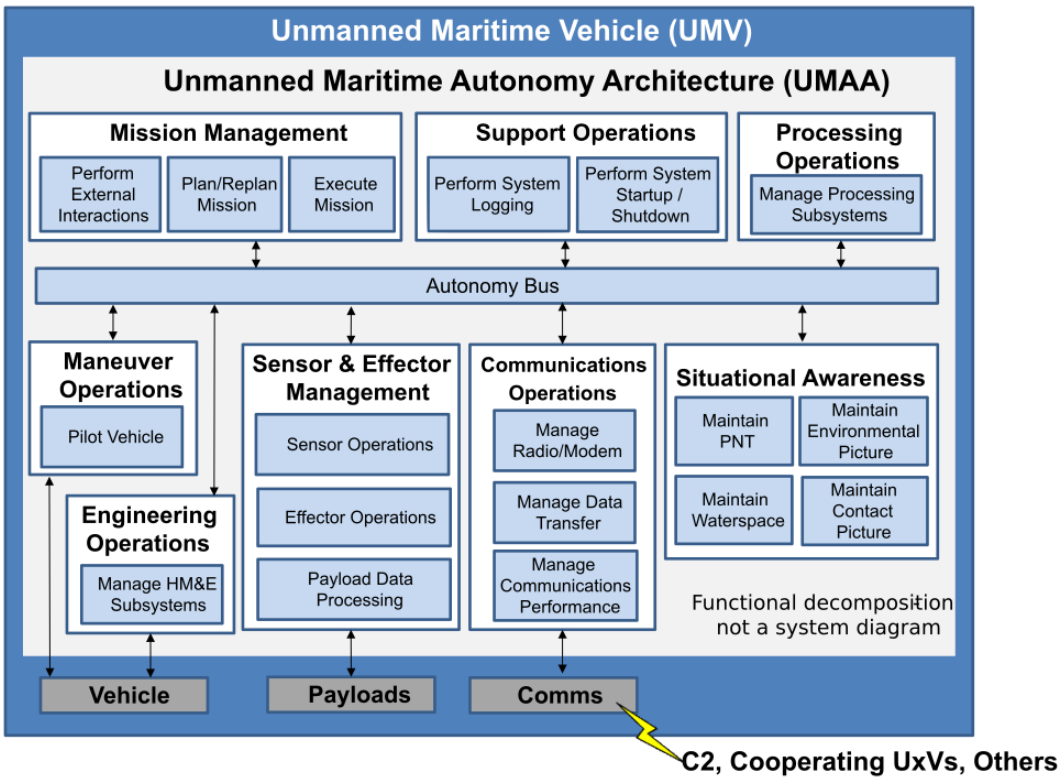

Figure 1: UMAA Functional Organization

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.

Engineering Operations (EO) includes the services required to interact with the physical systems of an UMV. Figure 2 depicts an example of various services in this EO ICD in relation to the maneuvering behavior services (in the Maneuver Operations ICD) and navigation sensing services (in the Situational Awareness ICD).

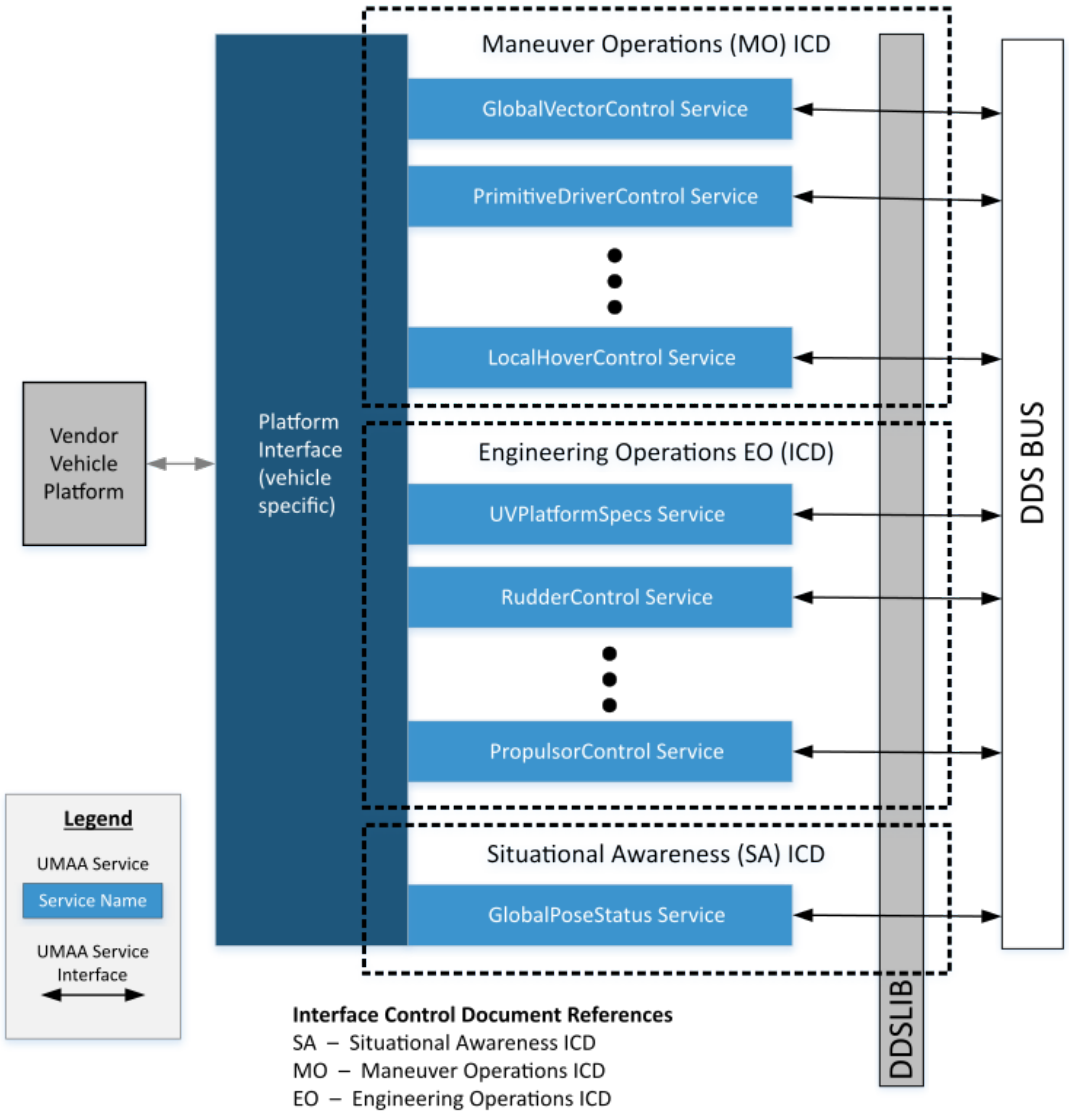

Figure 2: UMAA Services and Interfaces Example

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 – Engineering Operations (EO) 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.

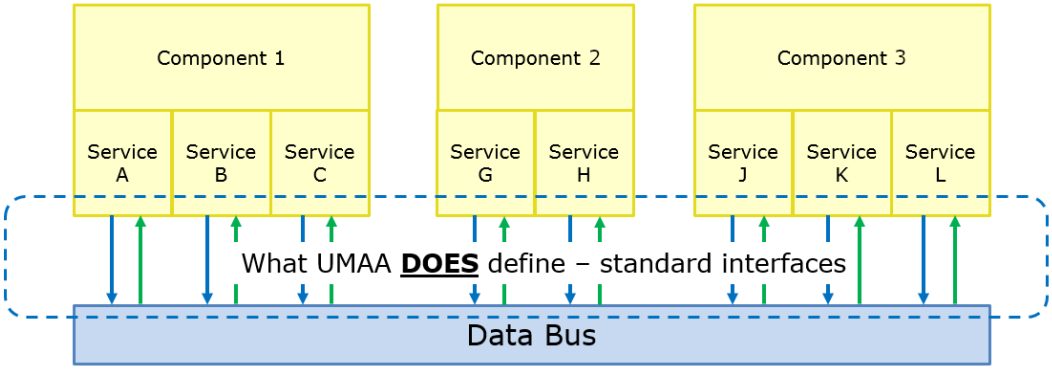

Figure 3: Services and Interfaces Exposed on the UMAA Data Bus

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).

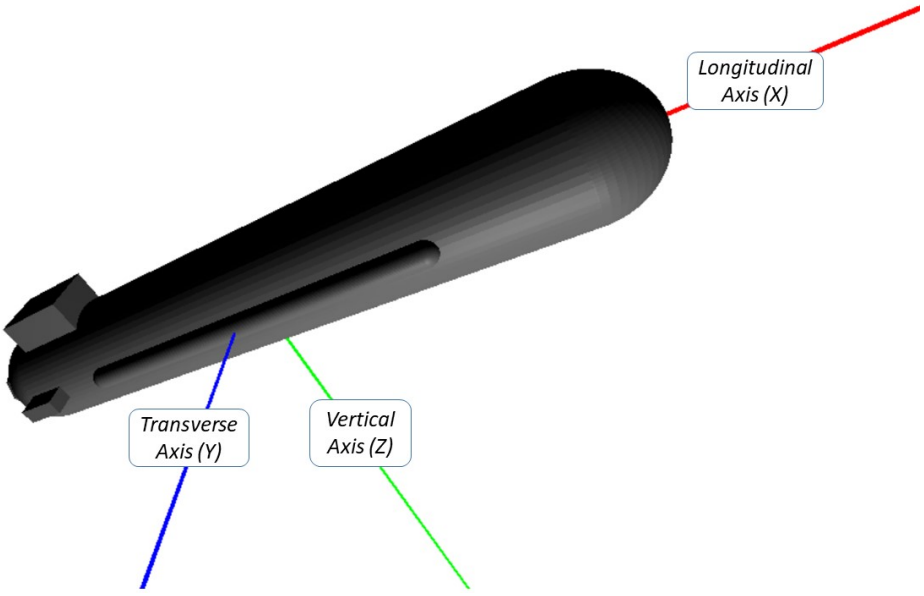

Figure 4: Given a vehicle in arbitrary orientation

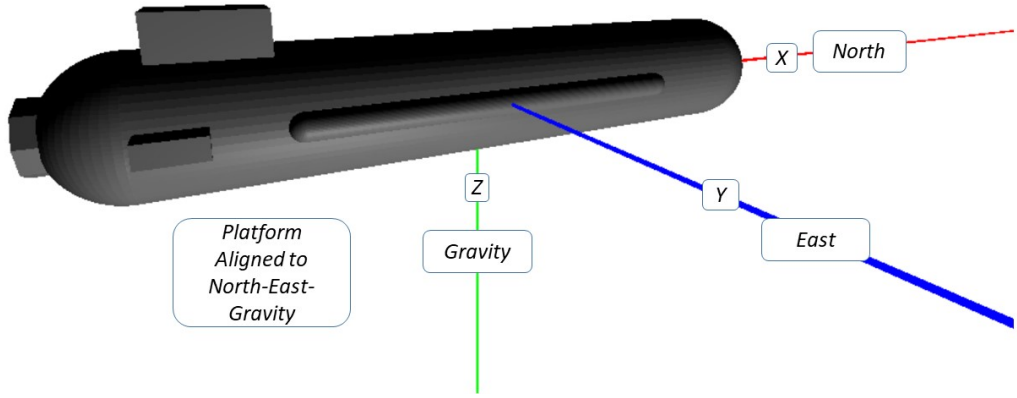

Figure 5: Align the vehicle with the reference frame axes

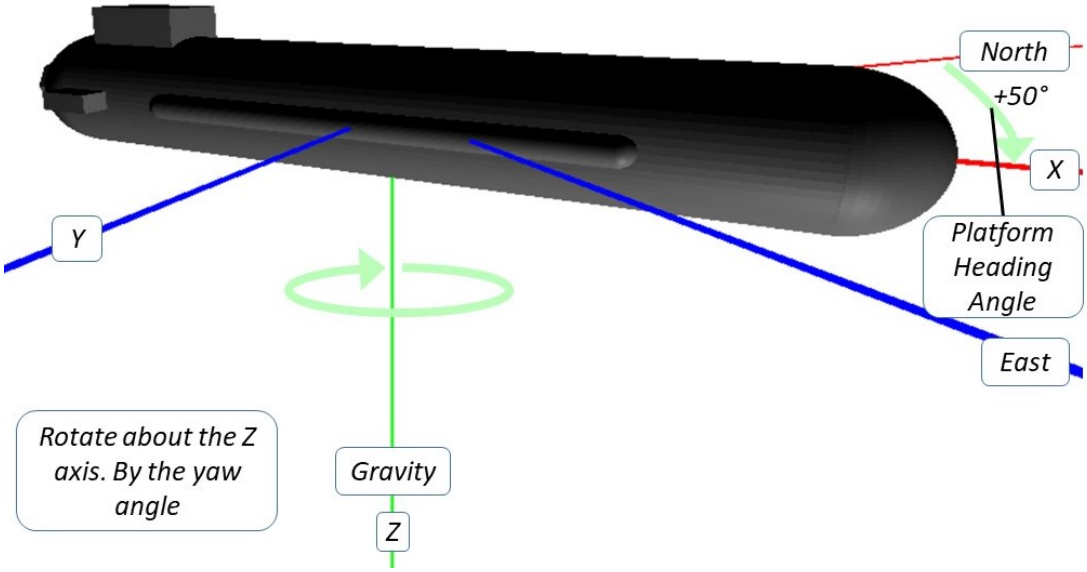

Figure 6: Rotate the vehicle by the Yaw angle

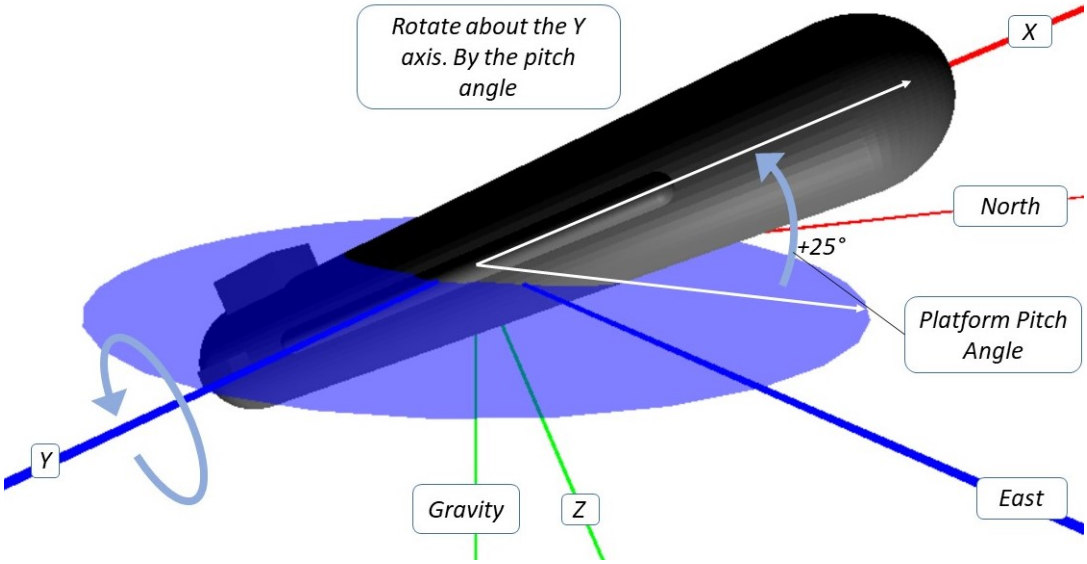

Figure 7: Rotate the vehicle by the Pitch angle

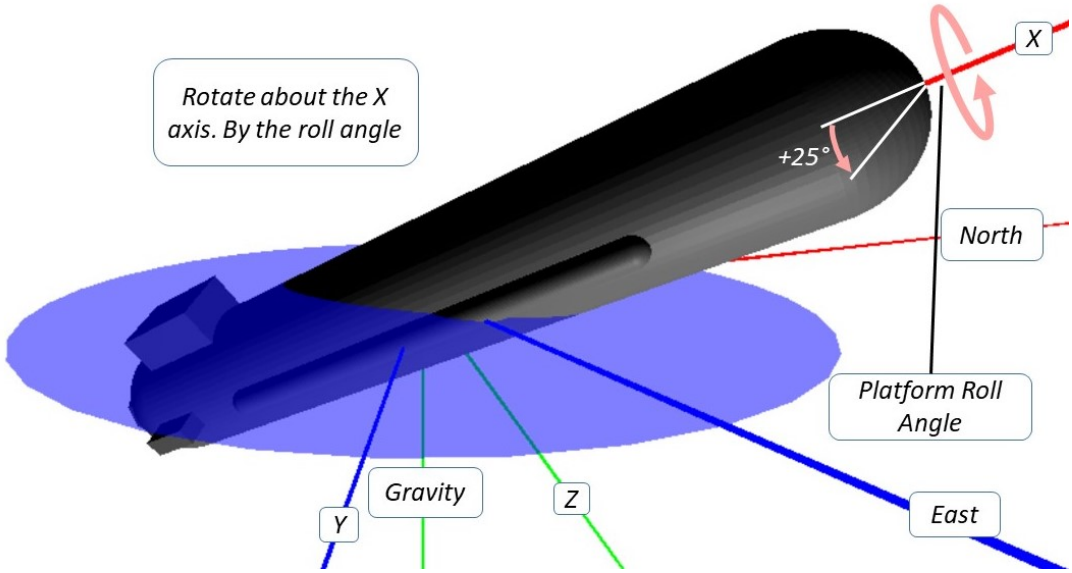

Figure 8: Rotate the vehicle by the Roll angle

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).

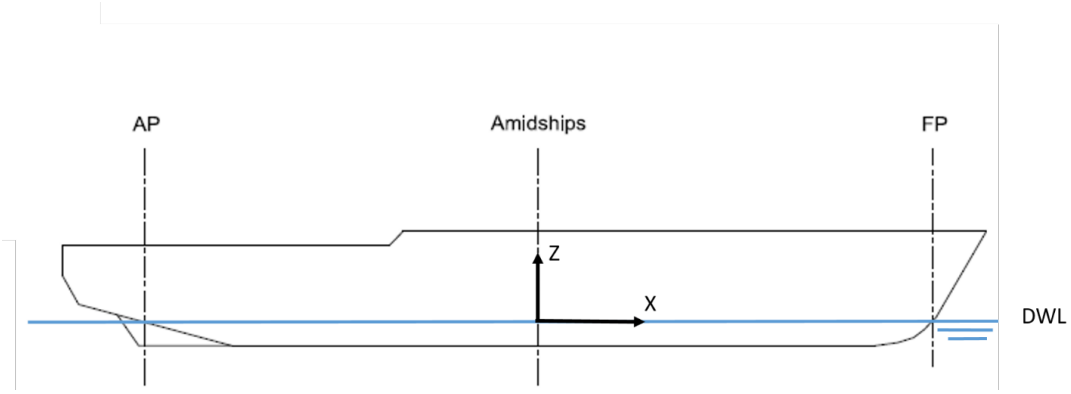

Figure 9: Origin location on a USV as example

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

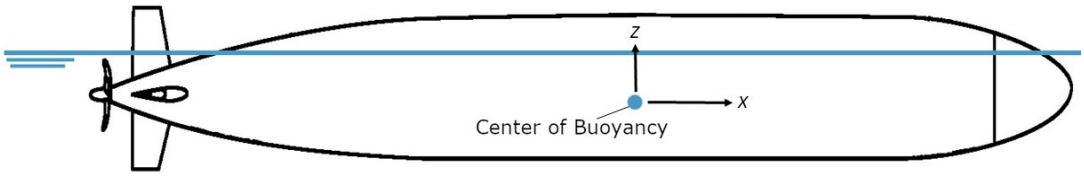

Figure 10: Origin location on a UUV as example

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.

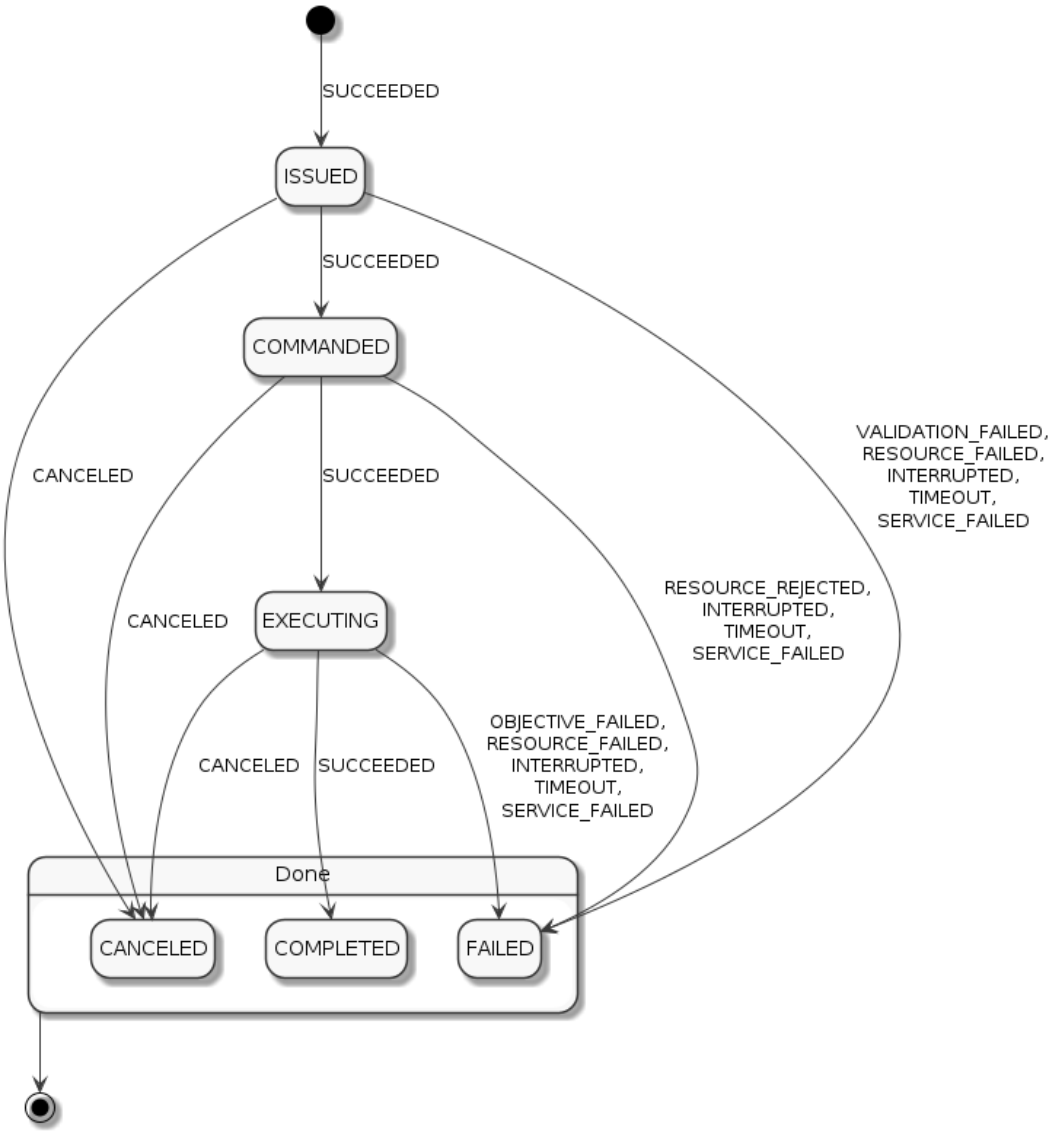

Figure 11: The state transitions of the `commandStatus` as commands are processed. Labels on the arrows represent valid `commandStatusReason` values for each transition.

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.

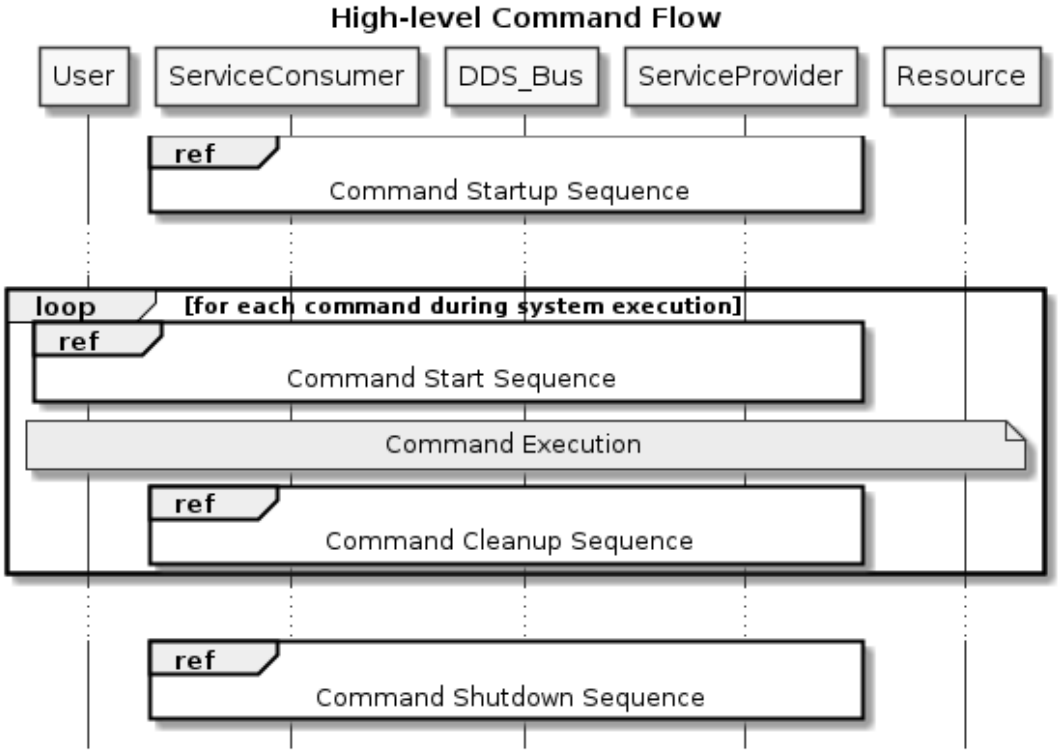

Figure 12: The sequence diagram for the high-level description of a command exeuction.

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.

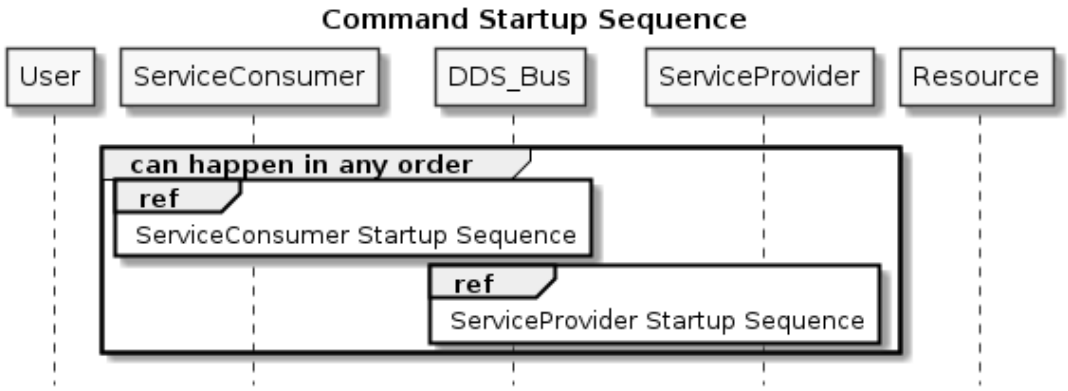

Figure 13: The sequence diagram for command startup.

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.

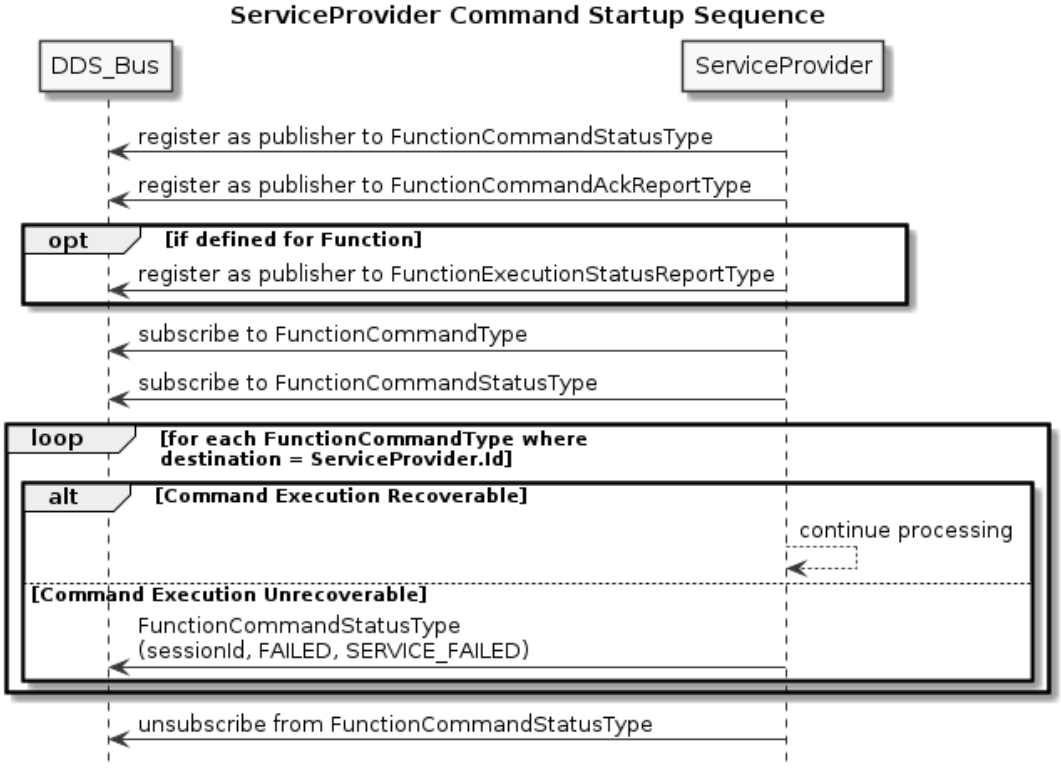

Figure 14: The sequence diagram for command startup for Service Providers.

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.

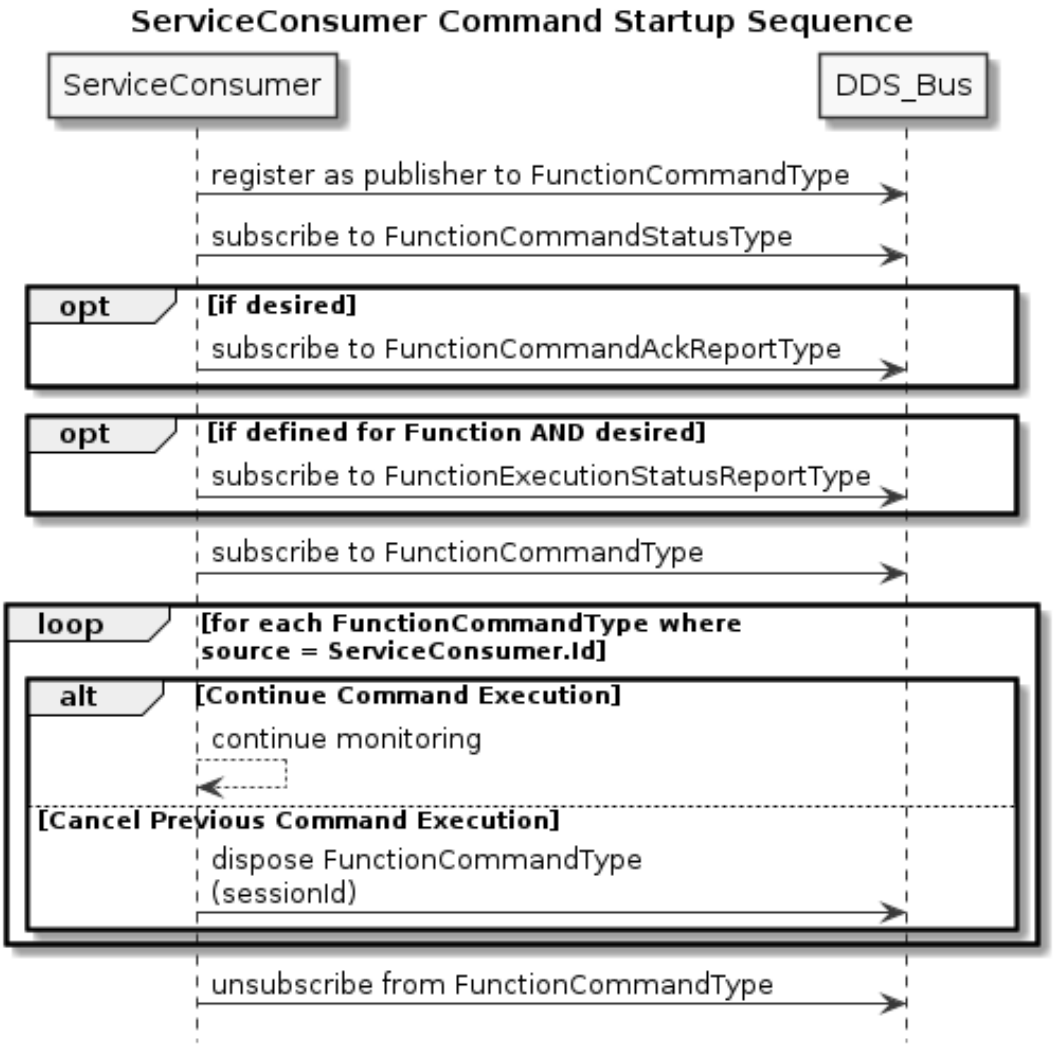

Figure 15: The sequence diagram for command startup for Service Consumers.

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).

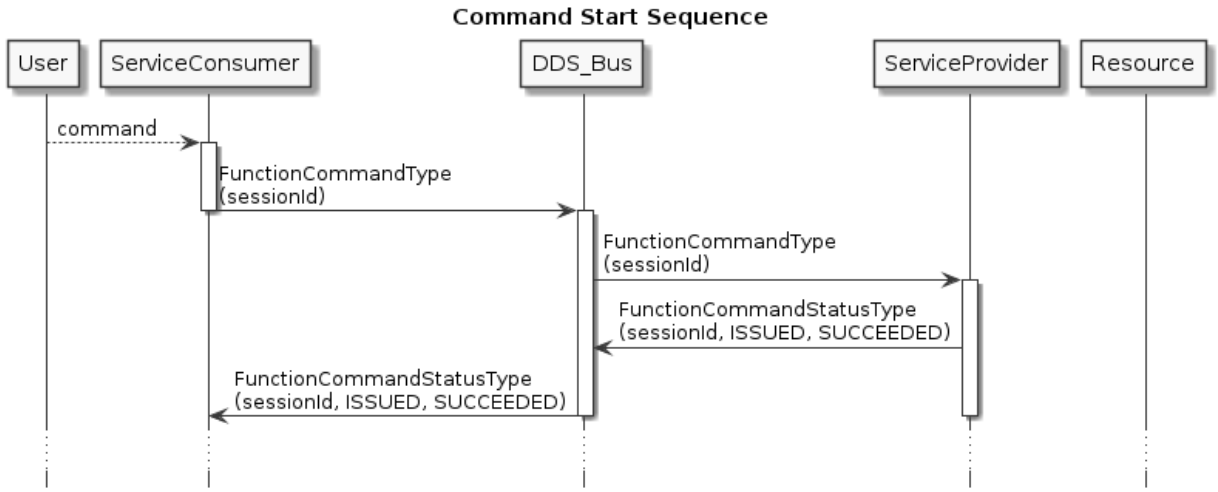

Figure 16: The sequence diagram for the start of a command execution.

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.

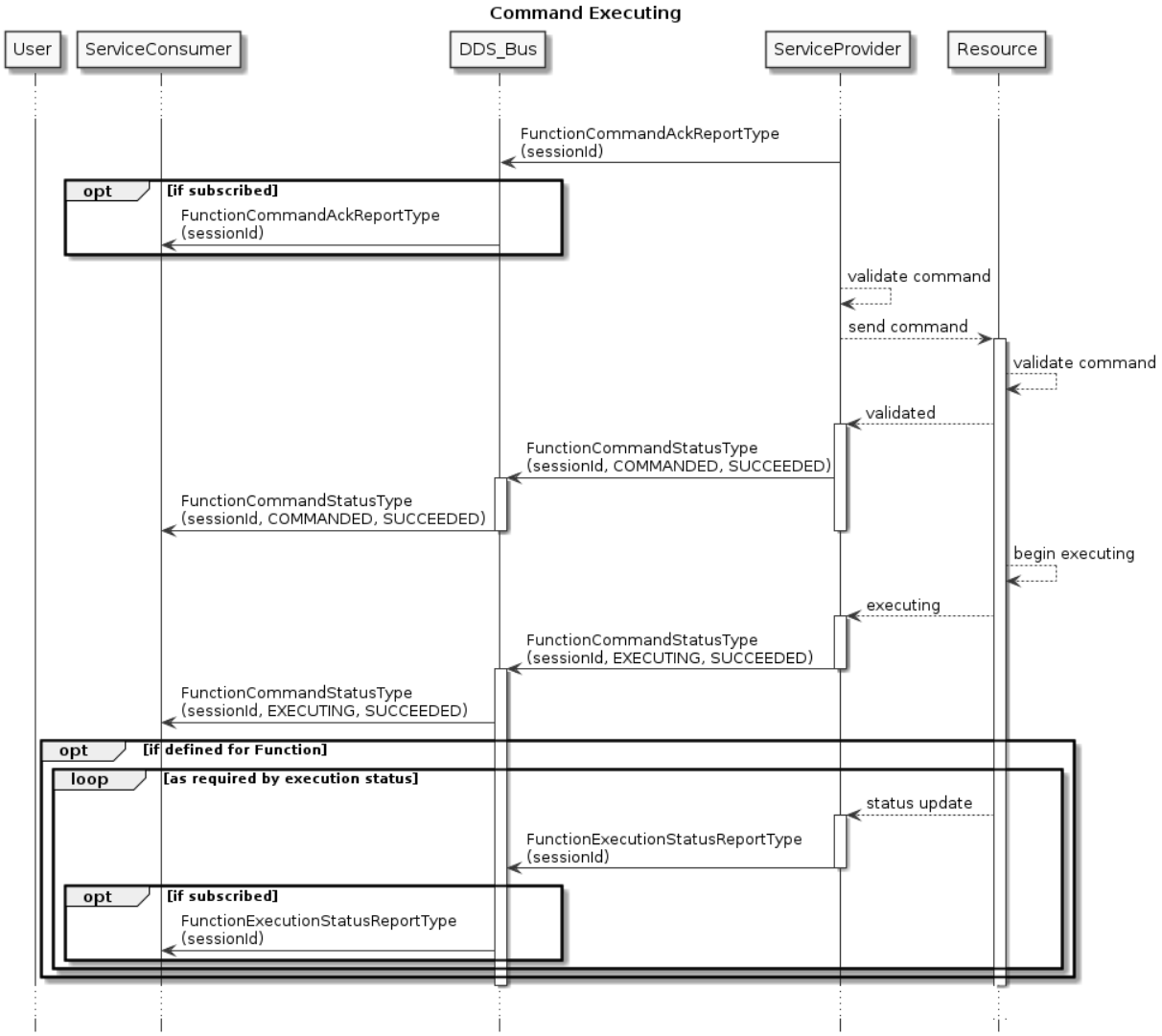

Figure 17: The beginning sequence diagram for a command execution.

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.

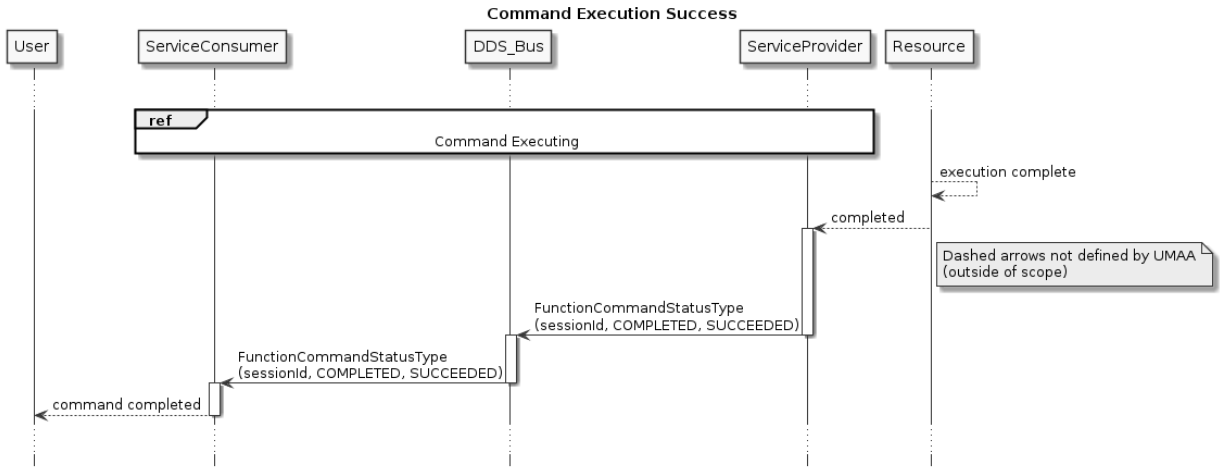

Figure 18: The sequence diagram for a command that completes successfully.

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.

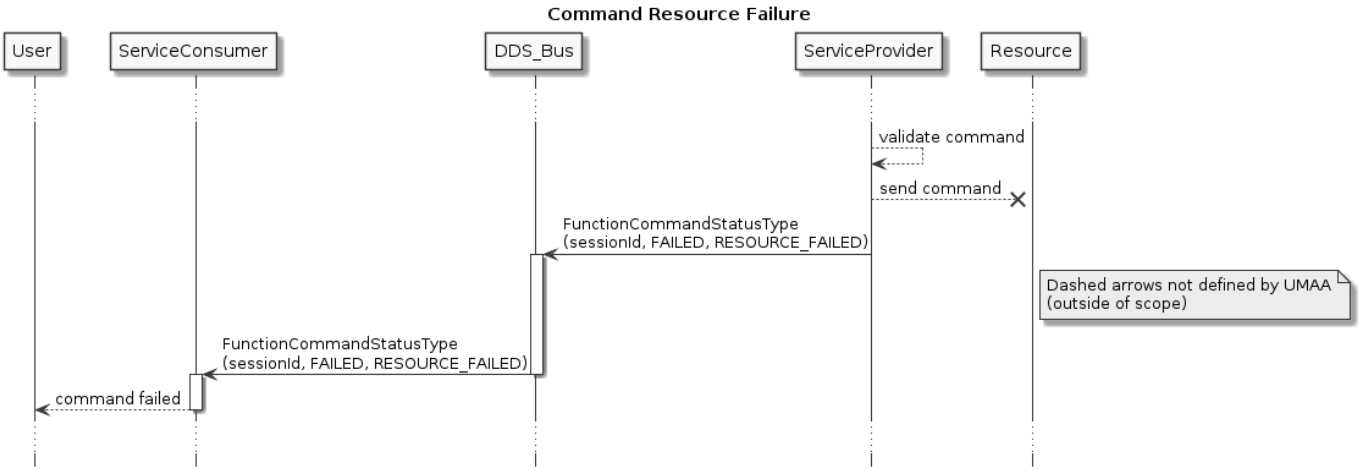

Figure 19: The sequence diagram for a command that fails due to Resource failure.

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.

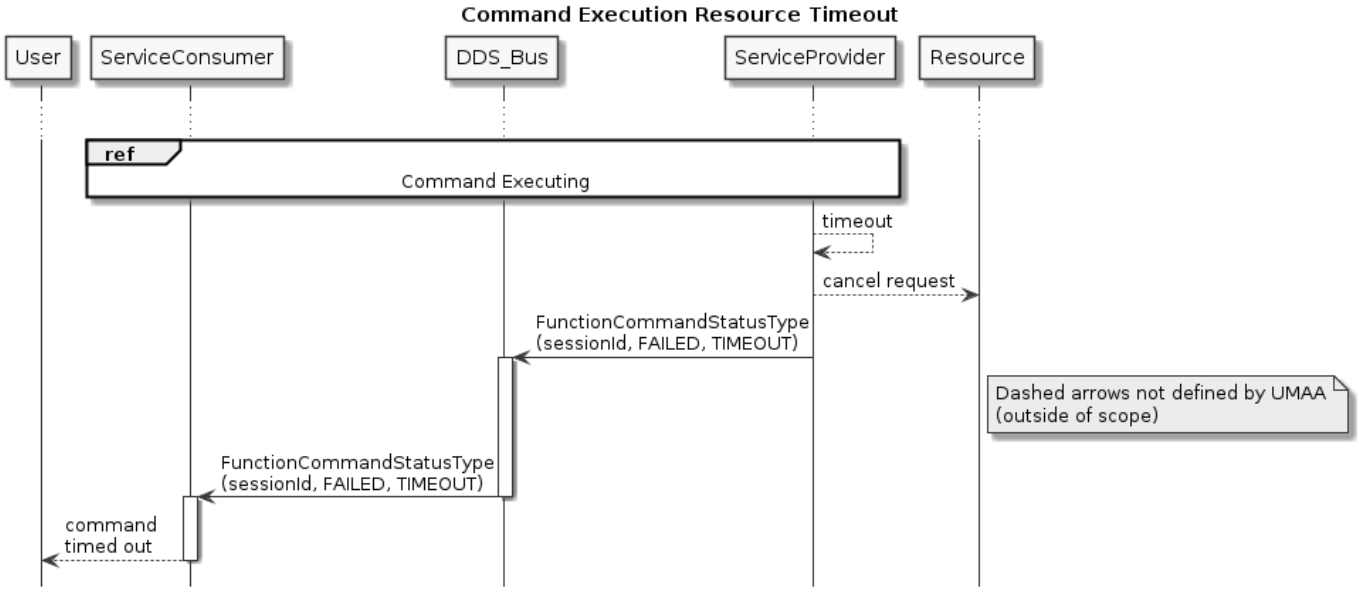

Figure 20: The sequence diagram for a command that times out before completing.

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.

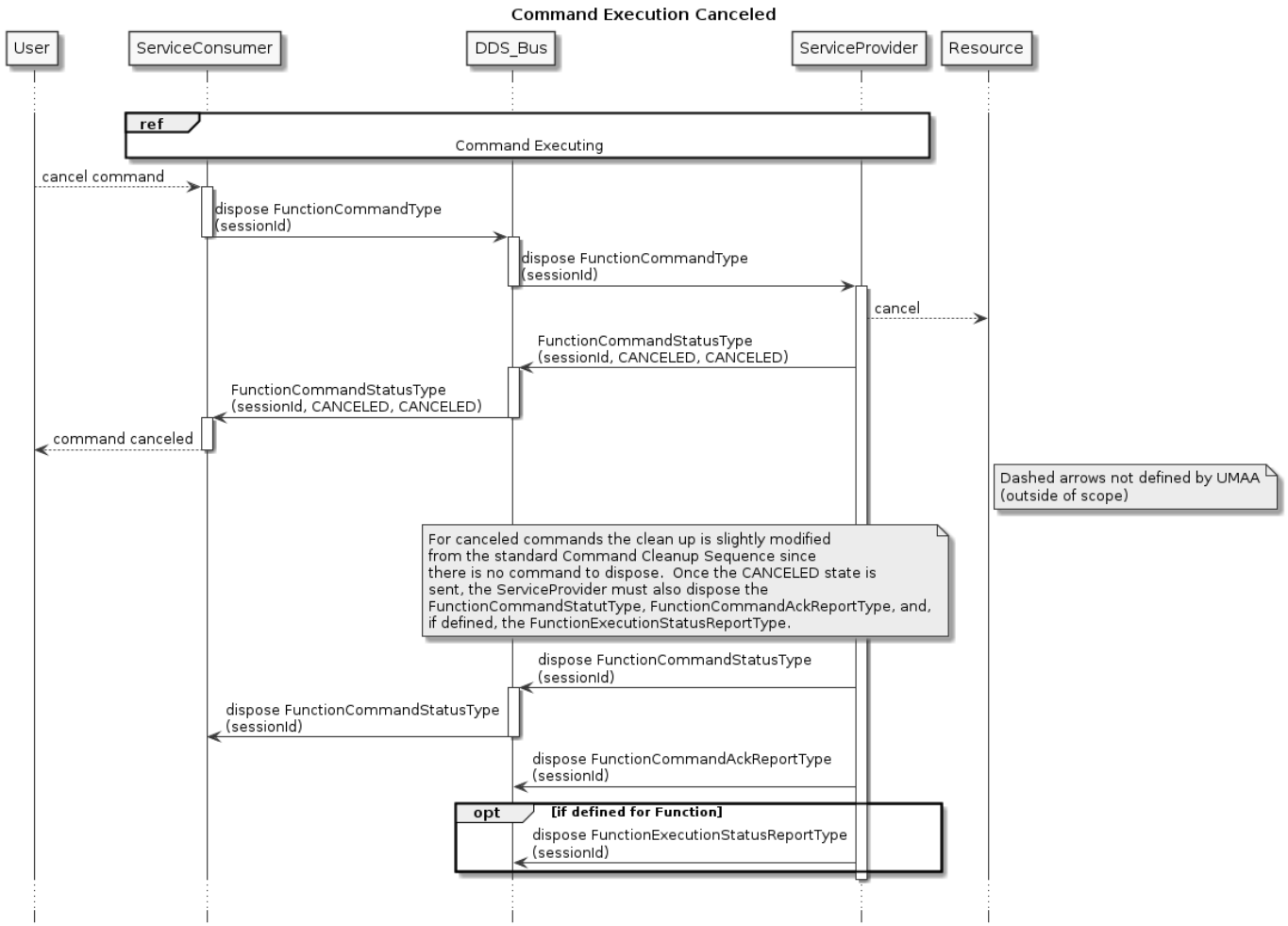

Figure 21: The sequence diagram for a command that is canceled by the Service Consumer before the Service Provider is able to complete it.

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.

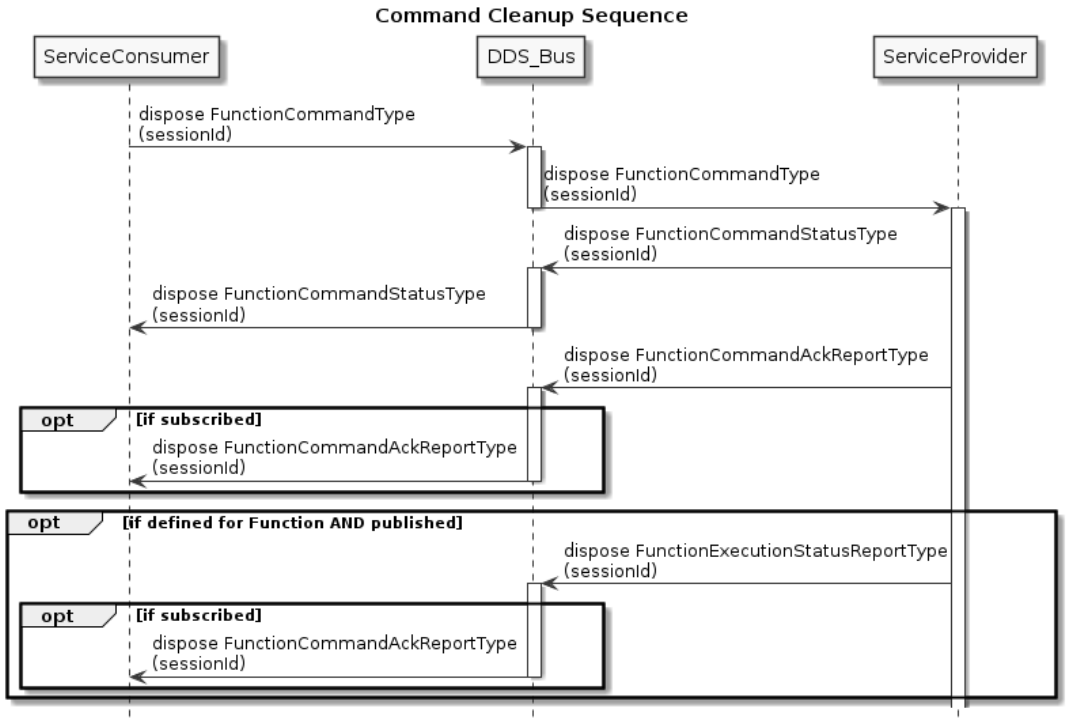

Figure 22: The sequence diagram showing cleanup of the bus when a command has been completed and the Service Consumer no longer wishes to maintain the commanded state.

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.

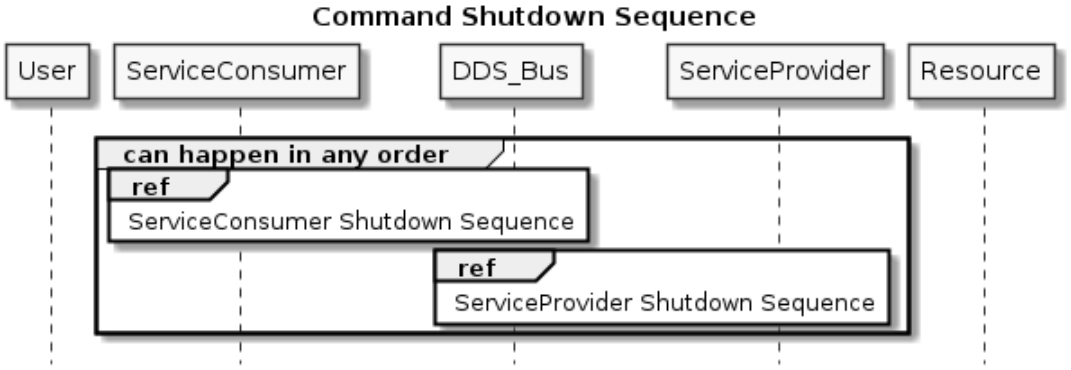

Figure 23: The sequence diagram for command shutdown.

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.

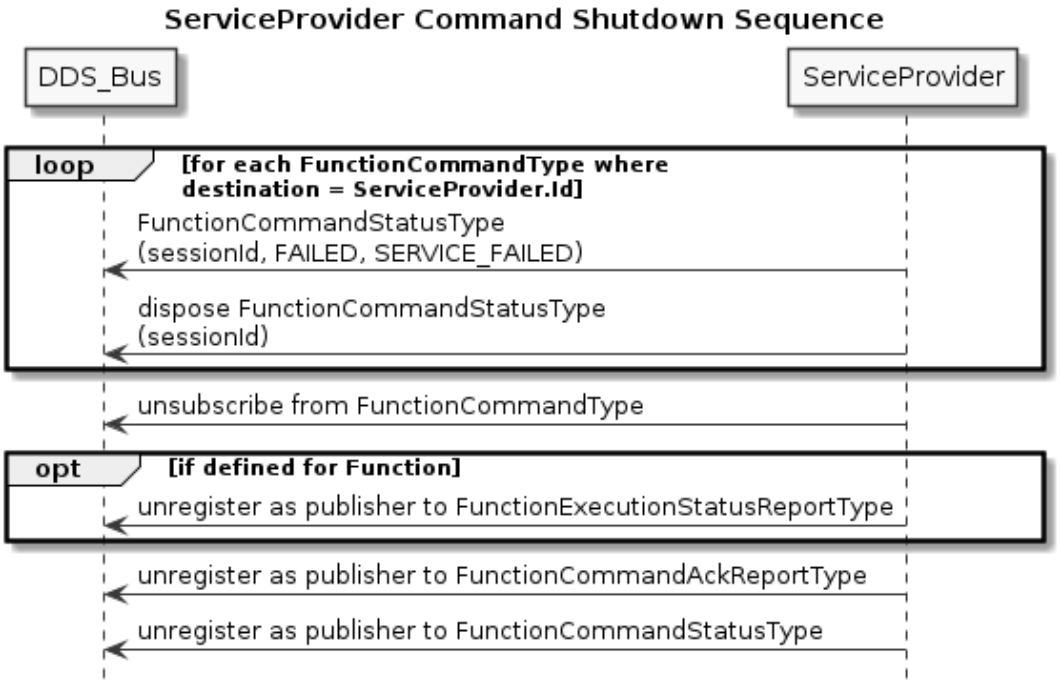

Figure 24: The sequence diagram for command shutdown for Service Providers.

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.

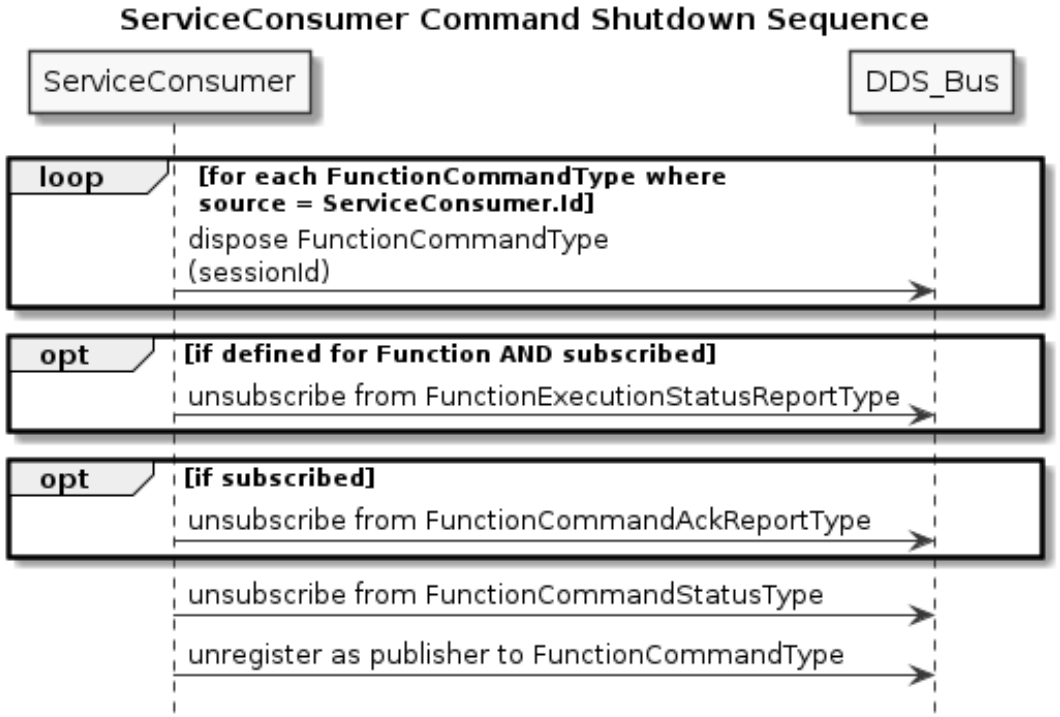

Figure 25: The sequence diagram for command shutdown for Service Consumers.

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.

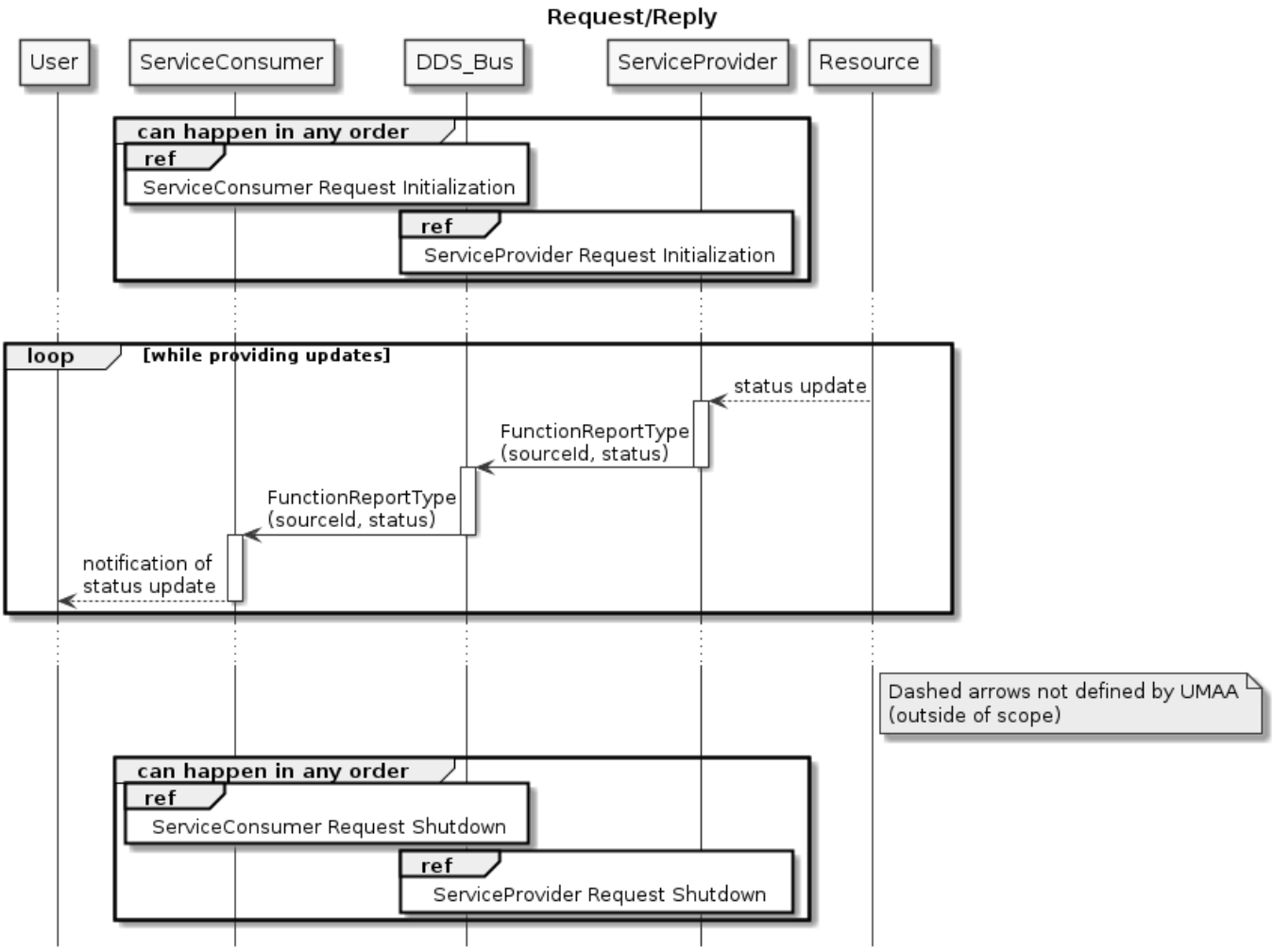

Figure 26: The sequence diagram for a request/reply for report data that does not require any specific query data.

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.

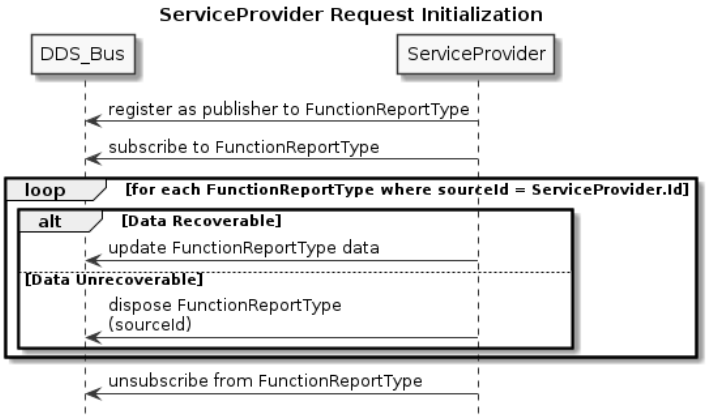

Figure 27: The sequence diagram for initialization of a Service Provider to provide FunctionReportTypes.

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.

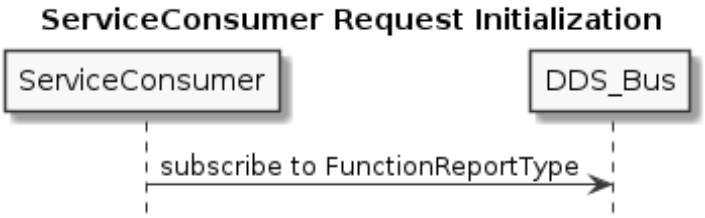

Figure 28: The sequence diagram for initialization of a Service Consumer to request FunctionReportTypes.

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.

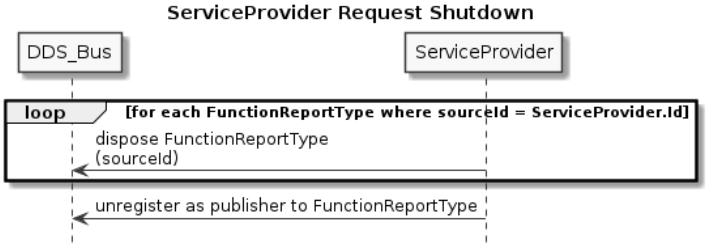

Figure 29: The sequence diagram for shutdown of a Service Provider.

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

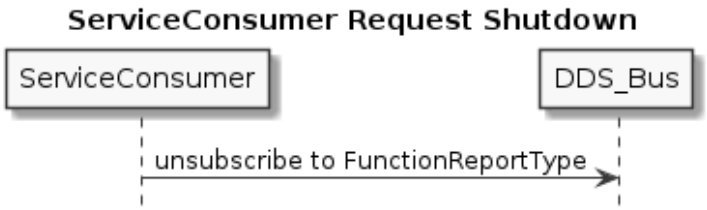

Figure 30: The sequence diagram for shutdown of a Service Consumer.

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 Engineering Operations (EO) Services and Interfaces

6.1 Services and Interfaces

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

Each interface topic is referenced by a UMAA service and is defined as either an input or output interface.
Attributes ending in one or more asterisk(s) denote the following:
* = Key (annotated with @key in IDL file, vendors may use different notation to indicate a key field)
†= Optional (annotated with @optional in IDL file, vendors may use different notation to indicate an optional field)

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

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

Operations without DDS Topics

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

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

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

report<...>CancelCommandStatus - all cancel reports are included here to show completeness of the MDE model mapping to UMAA. For UMAA, this operation is not used.

Instead, the cancel status is inferred from the associated command status. If the cancel command is successful, the corresponding command will fail with a command status and reason of CANCELED. If the corresponding command status reports COMPLETED, then this cancel command has failed.

6.1.1 AnchorControl

The purpose of this service is to provide the operations and interfaces to control the anchor on the vehicle. Three modes of operation: stop, lower, and raise are supported per anchor.

Table 6: AnchorControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setAnchor	reportAnchorCommandStatus
queryAnchorCommandAck⊕	reportAnchorCommandAck
cancelAnchorCommand⊕	reportAnchorCancelCommandStatus⊕

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

6.1.1.1 reportAnchorCommandAck

Description: This operation is used to report the current commanded anchor action.

Namespace: UMAA::EO::AnchorControl

Topic: AnchorCommandAckReport

Data Type: AnchorCommandAckReportType

Table 7: AnchorCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
action	AnchorActionEnumType	Defines the attributes used to control the anchor.

6.1.1.2 reportAnchorCommandStatus

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

Namespace: UMAA::EO::AnchorControl

Topic: AnchorCommandStatus

Data Type: AnchorCommandStatusType

Table 8: AnchorCommandStatusType Message Definition

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

6.1.1.3 setAnchor

Description: This operation is used to set the control parameters for the Anchor service.

Namespace: UMAA::EO::AnchorControl

Topic: AnchorCommand

Data Type: AnchorCommandType

Table 9: AnchorCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
action	AnchorActionEnumType	Defines the attributes used to control the anchor.

6.1.2 AnchorSpecs

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

Table 10: AnchorSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryAnchorSpecs⊕	reportAnchorSpecs

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

6.1.2.1 reportAnchorSpecs

Description: This operation is used to report the specification parameters for the Anchor service.

Namespace: UMAA::EO::AnchorSpecs

Topic: AnchorSpecsReport

Data Type: AnchorSpecsReportType

Table 11: AnchorSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
anchorHoldingPower	Mass	Defines the anchor holding power as determined by full scale anchor drag tests in a firm sand bottom.
anchorHoldingPowerRatio	Size_Numeral	The anchor holding power ratio is the ratio of the anchor holding power to the anchor size.
anchorKind	AnchorKindEnumType	Defines the type of anchor (e.g., commercial stockless, standard navy stockless, etc.)
anchorLocation	AnchorLocationEnumType	Defines the anchor location (i.e., bower anchor, stern anchor, keel anchor).
anchorSize	Mass	Defines the anchor size and is expressed in terms of the mass of the anchor.
rodeLength	Distance	The length of chain/rope that can be paid out by the anchoring system.
rodeSize	Distance	For a chain road it defines the link or chain size (the nominal diameter of the link material in the grip area). For a rope rode it defines the diameter of the rope.
rodeWorkingLoadLimit	Force	The rated working load limit for the chain/rope of the anchoring system.

6.1.3 AnchorStatus

The purpose of this service is to provide the operations and interfaces to monitor the anchor on the vehicle.

Table 12: AnchorStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryAnchor⊕	reportAnchor

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

6.1.3.1 reportAnchor

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

Namespace: UMAA::EO::AnchorStatus

Topic: AnchorReport

Data Type: AnchorReportType

Table 13: AnchorReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
rodeLengthPaidOut	Distance	The current length of chain/rope that is paid out.
state	AnchorStateEnumType	The current operational state of the anchor.

6.1.4 BallastTank

The purpose of this service is to provide the operations and interfaces to control and monitor the ballast tanks and their supporting pumps on the vehicle. The ballast tank control is measured by either volume or mass. The filled level of each tank can be commanded and monitored. Four modes of operation, Off, Fill, Empty, and Trim are supported per ballast pump. When trim mode is set, the BallastControl service is trying to keep the vehicle steady in a particular orientation at a desired depth.

Table 14: BallastTank Operations

Service Requests (Inputs)	Service Responses (Outputs)
setBallastPump	reportBallastPumpCommandStatus
queryBallastPumpCommand⊕	reportBallastPumpCommand
cancelBallastPumpCommand⊕	reportBallastPumpCancelCommandStatus⊕
queryBallastPump⊕	reportBallastPump
setBallastTank	reportBallastTankCommandStatus
queryBallastTankCommand⊕	reportBallastTankCommand
cancelBallastTankCommand⊕	reportBallastTankCancelCommandStatus⊕
queryBallastTank⊕	reportBallastTank
queryBallastPumpSpecs⊕	reportBallastPumpSpecs
queryBallastTankSpecs⊕	reportBallastTankSpecs

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

6.1.4.1 reportBallastPump

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

Namespace: UMAA::EO::BallastTank

Topic: BallastPumpReport

Data Type: BallastPumpReportType

Table 15: BallastPumpReportType Message Definition

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

6.1.4.2 reportBallastPumpCommand

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

Namespace: UMAA::EO::BallastTank

Topic: BallastPumpCommandReport

Data Type: BallastPumpCommandReportType

Table 16: BallastPumpCommandReportType Message Definition

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

6.1.4.3 reportBallastPumpCommandStatus

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

Namespace: UMAA::EO::BallastTank

Topic: BallastPumpCommandStatus

Data Type: BallastPumpCommandStatusType

Table 17: BallastPumpCommandStatusType Message Definition

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

6.1.4.4 reportBallastPumpSpecs

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

Namespace: UMAA::EO::BallastTank

Topic: BallastPumpSpecsReport

Data Type: BallastPumpSpecsReportType

Table 18: BallastPumpSpecsReportType Message Definition

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

6.1.4.5 reportBallastTank

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

Namespace: UMAA::EO::BallastTank

Topic: BallastTankReport

Data Type: BallastTankReportType

Table 19: BallastTankReportType Message Definition

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

6.1.4.6 reportBallastTankCommand

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

Namespace: UMAA::EO::BallastTank

Topic: BallastTankCommandReport

Data Type: BallastTankCommandReportType

Table 20: BallastTankCommandReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
ballastFill	BallastFillType	The optional elevation used for the unmanned maritime platform.

6.1.4.7 reportBallastTankCommandStatus

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

Namespace: UMAA::EO::BallastTank

Topic: BallastTankCommandStatus

Data Type: BallastTankCommandStatusType

Table 21: BallastTankCommandStatusType Message Definition

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

6.1.4.8 reportBallastTankSpecs

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

Namespace: UMAA::EO::BallastTank

Topic: BallastTankSpecsReport

Data Type: BallastTankSpecsReportType

Table 22: BallastTankSpecsReportType Message Definition

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

6.1.4.9 setBallastPump

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

Namespace: UMAA::EO::BallastTank

Topic: BallastPumpCommand

Data Type: BallastPumpCommandType

Table 23: BallastPumpCommandType Message Definition

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

6.1.4.10 setBallastTank

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

Namespace: UMAA::EO::BallastTank

Topic: BallastTankCommand

Data Type: BallastTankCommandType

Table 24: BallastTankCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
ballastFill	BallastFillType	The optional elevation used for the unmanned maritime platform.

6.1.5 BatterySpecs

The purpose of this service is to provide the battery specifications.

Table 25: BatterySpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryBatterySpecs⊕	reportBatterySpecs

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

6.1.5.1 reportBatterySpecs

Description: This operation is used to report the system specifications of the batteries of the unmanned platform.

Namespace: UMAA::EO::BatterySpecs

Topic: BatterySpecsReport

Data Type: BatterySpecsReportType

Table 26: BatterySpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
cellMinimumVoltage	PowerBusVoltage	The minimum safe voltage of a cell in the battery system.
maxCapacity	Charge	The Coulombs available by the battery system when fully charged
maxChargingCurrent	PowerBusCurrent	The maximum charging current of the battery system
maxChargingTemp	Temperature	The maximum charging temperature
maxCrankingAmps	PowerBusCurrent	The maximum current output while engine starter is cranking
maxOutputCurrent	PowerBusCurrent	The maximum output current of the battery system
maxStorageTemp	Temperature	The maximum storage temperature
maxTemperature	Temperature	The maximum operating temperature
maxVoltage	PowerBusVoltage	The maximum voltage of the battery system
minChargingTemp	Temperature	The minimum charging temperature
minStorageTemp	Temperature	The minimum storage temperature

Attribute Name	Attribute Type	Attribute Description
minTemperature	Temperature	The minimum operating temperature
minVoltage	PowerBusVoltage	The minimum voltage of the battery system
name	StringShortDescription	The description of the battery system
nominalVoltage	PowerBusVoltage	The nominal voltage of the battery system

6.1.6 BatteryStatus

The purpose of this service is to provide the current battery status.

Table 27: BatteryStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryBattery⊕	reportBattery

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

6.1.6.1 reportBattery

Description: This operation is used to report the current status of battery of the unmanned platform.

Namespace: UMAA::EO::BatteryStatus

Topic: BatteryReport

Data Type: BatteryReportType

Table 28: BatteryReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
cells	sequence<BatteryCellDataType>	The charge data for each cell in the battery.
chargeRemaining	Energy_Percent	The amount of charge remaining
current	PowerBusCurrent	The runtime current of the battery
energyUsageRate	ElectricalPower	The rates of power at a moment in time
hours	Duration_Hours	The total runtime of the battery in its lifetime
state	PowerPlantStateEnumType	the current power plant state
temp	Temperature	The current temperature of the battery
voltage	PowerBusVoltage	The runtime voltage of the battery

6.1.7 BellControl

The purpose of this service is to provide the operations and interfaces to control and monitor the vehicle's bell.

Table 29: BellControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setBellControl	reportBellControlCommandStatus
queryBellControlCommand⊕	reportBellControlCommand
cancelBellControlCommand⊕	reportBellControlCancelCommandStatus⊕
queryBellControl⊕	reportBellControl

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

6.1.7.1 reportBellControl

Description: This operation is used to report the current state of the bell sounding device on the unmanned platform.

Namespace: UMAA::EO::BellControl

Topic: BellControlReport

Data Type: BellControlReportType

Table 30: BellControlReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
blastCondition†	BlastConditionEnumType	The blast state of the annunciator
blastNumber†	PositiveCount	The number of times blasted within the current set; includes current blast if in progress
status	OnOffStatusEnumType	The current on/off status of the bell switch

6.1.7.2 reportBellControlCommand

Description: This operation is used to report the current bell sounding device command on the unmanned platform.

Namespace: UMAA::EO::BellControl

Topic: BellControlCommandReport

Data Type: BellControlCommandReportType

Table 31: BellControlCommandReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
blastProperties†	BlastPropertiesType	Blast properties. If the attribute is not provided, it means turn off the bell.

Attribute Name	Attribute Type	Attribute Description
status	OnOffStatusEnumType	Bell active status.

6.1.7.3 reportBellControlCommandStatus

Description: This operation is used to report the state of the bell sounding device command on the unmanned platform.

Namespace: UMAA::EO::BellControl

Topic: BellControlCommandStatus

Data Type: BellControlCommandStatusType

Table 32: BellControlCommandStatusType Message Definition

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

6.1.7.4 setBellControl

Description: This operation is used to control the bell sounding device on 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::EO::BellControl

Topic: BellControlCommand

Data Type: BellControlCommandType

Table 33: BellControlCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
blastProperties†	BlastPropertiesType	Blast properties. If the attribute is not provided, it means turn off the bell.
status	OnOffStatusEnumType	Bell active status.

6.1.8 EngineControl

The purpose of this service is to provide the control of the engine on the unmanned platform.

Table 34: EngineControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setEngine	reportEngineCommandStatus
queryEngineCommandAck⊕	reportEngineCommandAck
cancelEngineCommand⊕	reportEngineCancelCommandStatus⊕

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

6.1.8.1 reportEngineCommandAck

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

Namespace: UMAA::EO::EngineControl

Topic: EngineCommandAckReport

Data Type: EngineCommandAckReportType

Table 35: EngineCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
plugState†	OnOffStatusEnumType	The desired glow plug state
propulsion	PropulsionType	The desired propulsive value of the engine.
state	IgnitionStateEnumType	The desired power state of the subsystem

6.1.8.2 reportEngineCommandStatus

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

Namespace: UMAA::EO::EngineControl

Topic: EngineCommandStatus

Data Type: EngineCommandStatusType

Table 36: EngineCommandStatusType Message Definition

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

6.1.8.3 setEngine

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

Namespace: UMAA::EO::EngineControl

Topic: EngineCommand

Data Type: EngineCommandType

Table 37: EngineCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
plugState†	OnOffStatusEnumType	The desired glow plug state
propulsion	PropulsionType	The desired propulsive value of the engine.
state	IgnitionStateEnumType	The desired power state of the subsystem

6.1.9 EngineSpecs

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

Table 38: EngineSpecs Operations

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

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

6.1.9.1 reportEngineSpecs

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

Namespace: UMAA::EO::EngineSpecs

Topic: EngineSpecsReport

Data Type: EngineSpecsReportType

Table 39: EngineSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
engineKind	EngineKindEnumType	The type of engine
glowPlugTime†	Duration_Seconds	The glow plug preset heating time

Attribute Name	Attribute Type	Attribute Description
maxCoolantLevel	Volume_CubicMeter	The maximum coolant level limit
maxCoolantPressure	Pressure_KiloPascals	The maximum coolant pressure limit
maxCoolantTemp	Temperature	The maximum coolant temperature limit
maxEngineTemp	Temperature	The maximum engine temperature limit
maxGlowPlugTemp†	Temperature	The maximum glow plug temperature limit
maxManifoldAirTemp	Temperature	The maximum engine manifold air temperature limit
maxManifoldPressure	Pressure_KiloPascals	The maximum engine manifold pressure limit
maxOilPressure	Pressure_KiloPascals	The maximum engine oil pressure limit to signal an alarm (0 to 512)
maxOilTemp	Temperature	The maximum engine oil temperature limit
minCoolantLevel	Volume_CubicMeter	The minimum coolant level limit
minOilLevel	Volume_CubicMeter	The minimum engine oil level limit
name	StringShortDescription	The name of the engine
oilCapacity	Volume_CubicMeter	The oil capacity of the engine
reverseRPMLowerLimit	EngineSpeed	the lower limit of reverse RPM
reverseRPMMaxLimit	EngineSpeed	the maximum limit of reverse RPM
reverseRPMUpperLimit	EngineSpeed	the upper limit of reverse RPM
reversible	BooleanEnumType	The reversibility of the engine rotation
RPMLowerLimit	EngineSpeed	The lower RPM limit to operate the engine
RPMMaxLimit	EngineSpeed	The physical maximum RPM limit to operate the engine
RPMUpperLimit	EngineSpeed	The upper RPM limit to operate the engine

6.1.10 EngineStatus

The purpose of this service is to report the current status of the engine on the unmanned platform.

Table 40: EngineStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryEngine⊕	reportEngine

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

6.1.10.1 reportEngine

Description: This operation is used to report the current status of engines of the unmanned platform.

Namespace: UMAA::EO::EngineStatus

Topic: EngineReport

Data Type: EngineReportType

Table 41: EngineReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
coolantLevel	Volume_Percent	The engine coolant level
coolantPressure	Pressure_KiloPascals	The engine coolant pressure
coolantTemp	Temperature	The temperature of the engine coolant
engineTemp	Temperature	The temperature of the engine
exhaustTemp	Temperature	The engine exhaust temperature
glowPlugIndicator†	BooleanEnumType	The glow plug indicator state
glowPlugState†	OnOffStatusEnumType	The glow plug state
glowPlugTemp†	Temperature	The glow plug temperature
glowPlugTimeRemaining†	Duration_Seconds	The time remaining for the glow plug to be on
hours	Duration_Hours	The total runtime of the power plant in its lifetime
manifoldAirTemp	Temperature	The engine manifold air temperature
manifoldPressure	Pressure_KiloPascals	The engine manifold pressure
oilLevel	Volume_Percent	The engine oil level
oilPressure	Pressure_KiloPascals	The engine oil pressure
oilTemp	Temperature	The engine oil temperature
percentOilPressure	Pressure_Percent	The engine oil pressure
RPM	EngineSpeed	The engine RPM
state	IgnitionStateEnumType	The current ignition state.
throttle	Effort	The engine throttle. Negative values will be regarded as zero for non-reversible engines

6.1.11 FuelTankSpecs

The purpose of this service is to provide the current fuel tank specifications.

Table 42: FuelTankSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryFuelTankSpecs⊕	reportFuelTankSpecs

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

6.1.11.1 reportFuelTankSpecs

Description: This operation is used to report the system specifications of the fuel tanks of the unmanned platform.

Namespace: UMAA::EO::FuelTankSpecs

Topic: FuelTankSpecsReport

Data Type: FuelTankSpecsReportType

Table 43: FuelTankSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
capacity	Volume_CubicMeter	The maximum capacity of the fuel tank
name	StringShortDescription	The name of the fuel tank

6.1.12 FuelTankStatus

The purpose of this service is to provide the current fuel tank status.

Table 44: FuelTankStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryFuelTank⊕	reportFuelTank

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

6.1.12.1 reportFuelTank

Description: This operation is used to report the current status of fuel tanks of the unmanned platform.

Namespace: UMAA::EO::FuelTankStatus

Topic: FuelTankReport

Data Type: FuelTankReportType

Table 45: FuelTankReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
fuelLevel	Volume_Percent	The amount of fuel remaining in the tank
waterInFuel	BooleanEnumType	The detection of water in the fuel

6.1.13 GeneratorSpecs

The purpose of this service is to provide the current generator specifications.

Table 46: GeneratorSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryGeneratorSpecs⊕	reportGeneratorSpecs

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

6.1.13.1 reportGeneratorSpecs

Description: This operation is used to report the system specifications of the generators of the unmanned platform.

Namespace: UMAA::EO::GeneratorSpecs

Topic: GeneratorSpecsReport

Data Type: GeneratorSpecsReportType

Table 47: GeneratorSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
maxCurrent	PowerBusCurrent	The maximum current supported by the generator
maxPower	ElectricalPower	The maximum power this generator can produce
name	StringShortDescription	The name of the generator unit
ratedPower	ElectricalPower	The amount of generated power over a long period of operation
ratedVoltage	PowerBusVoltage	The rated voltage for generator operation

6.1.14 GeneratorStatus

The purpose of this service is to provide the current generator status.

Table 48: GeneratorStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryGenerator⊕	reportGenerator

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

6.1.14.1 reportGenerator

Description: This operation is used to report the current status of generators of the unmanned platform.

Namespace: UMAA::EO::GeneratorStatus

Topic: GeneratorReport

Data Type: GeneratorReportType

Table 49: GeneratorReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
current	PowerBusCurrent	The current of the generator. A negative current indicates that the generator is sourcing, a positive current indicates that it is sinking
state	PowerPlantStateEnumType	The state of the generator unit
voltage	PowerBusVoltage	The actual voltage of the generator

6.1.15 GongControl

The purpose of this service is to provide the operations and interfaces to control and monitor the vehicle's gong.

Table 50: GongControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setGongControl	reportGongControlCommandStatus
queryGongControlCommand⊕	reportGongControlCommand
cancelGongControlCommand⊕	reportGongControlCancelCommandStatus⊕
queryGongControl⊕	reportGongControl

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

6.1.15.1 reportGongControl

Description: This operation is used to report the current state of the gong sounding device on the unmanned platform.

Namespace: UMAA::EO::GongControl

Topic: GongControlReport

Data Type: GongControlReportType

Table 51: GongControlReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
blastCondition†	BlastConditionEnumType	The blast state of the annunciator
blastNumber†	PositiveCount	The number of times blasted within the current set; includes current blast if in progress

Attribute Name	Attribute Type	Attribute Description
status	OnOffStatusEnumType	The current on/off status of the gong switch

6.1.15.2 reportGongControlCommand

Description: This operation is used to retrieve the current state of the gong sounding device on the unmanned platform.

Namespace: UMAA::EO::GongControl

Topic: GongControlCommandReport

Data Type: GongControlCommandReportType

Table 52: GongControlCommandReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
blastProperties†	BlastPropertiesType	Blast properties. If the attribute is not provided, it means turn off the gong.
status	OnOffStatusEnumType	Gong active status.

6.1.15.3 reportGongControlCommandStatus

Description: This operation is used to retrieve the current state of the gong sounding device on the unmanned platform.

Namespace: UMAA::EO::GongControl

Topic: GongControlCommandStatus

Data Type: GongControlCommandStatusType

Table 53: GongControlCommandStatusType Message Definition

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

6.1.15.4 setGongControl

Description: This operation is used to control the gong sounding device on 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::EO::GongControl

Topic: GongControlCommand

Data Type: GongControlCommandType

Table 54: GongControlCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
blastProperties†	BlastPropertiesType	Blast properties. If the attribute is not provided, it means turn off the gong.
status	OnOffStatusEnumType	Gong active status.

6.1.16 LightControl

The purpose of this service is to provide the operations and interfaces to control the vehicle's lights.

Table 55: LightControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setLight	reportLightCommandStatus
queryLightCommandAck⊕	reportLightCommandAck
cancelLightCommand⊕	reportLightCancelCommandStatus⊕

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

6.1.16.1 reportLightCommandAck

Description: This operation is used to retrieve the current light switch command on the unmanned platform.

Namespace: UMAA::EO::LightControl

Topic: LightCommandAckReport

Data Type: LightCommandAckReportType

Table 56: LightCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
allroundLight	BooleanEnumType	State of all-around light.
flashingLight	BooleanEnumType	State of flashing light.
mastheadLight	BooleanEnumType	State of mast light.

Attribute Name	Attribute Type	Attribute Description
portSideLight	BooleanEnumType	State of port light.
starboardSideLight	BooleanEnumType	State of starboard light.
sternLight	BooleanEnumType	State of stern light.
towingLight	BooleanEnumType	State of towing light.

6.1.16.2 reportLightCommandStatus

Description: This operation is used to retrieve the status of the light switch command on the unmanned platform.

Namespace: UMAA::EO::LightControl

Topic: LightCommandStatus

Data Type: LightCommandStatusType

Table 57: LightCommandStatusType Message Definition

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

6.1.16.3 setLight

Description: This operation is used to control the light switch on the unmanned platform.

Namespace: UMAA::EO::LightControl

Topic: LightCommand

Data Type: LightCommandType

Table 58: LightCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
allroundLight	BooleanEnumType	State of all-around light.
flashingLight	BooleanEnumType	State of flashing light.
mastheadLight	BooleanEnumType	State of mast light.
portSideLight	BooleanEnumType	State of port light.
starboardSideLight	BooleanEnumType	State of starboard light.
sternLight	BooleanEnumType	State of stern light.
towingLight	BooleanEnumType	State of towing light.

6.1.17 LightSpecs

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

Table 59: LightSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryLightSpecs⊕	reportLightSpecs

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

6.1.17.1 reportLightSpecs

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

Namespace: UMAA::EO::LightSpecs

Topic: LightSpecsReport

Data Type: LightSpecsReportType

Table 60: LightSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
allroundLight	BooleanEnumType	All around light supported flag.
flashingLight	BooleanEnumType	Flashing light supported flag.
mastheadLight	BooleanEnumType	Mast light supported flag.
portSideLight	BooleanEnumType	Port light supported flag.
starboardSideLight	BooleanEnumType	Starboard light supported flag.
sternLight	BooleanEnumType	Stern light supported flag.
towingLight	BooleanEnumType	Towing light supported flag.

6.1.18 LightStatus

The purpose of this service is to provide the operations and interfaces to monitor the vehicle's lights.

Table 61: LightStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryLight⊕	reportLight

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

6.1.18.1 reportLight

Description: This operation is used to retrieve the current state of the light switch on the unmanned platform.

Namespace: UMAA::EO::LightStatus

Topic: LightReport

Data Type: LightReportType

Table 62: LightReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
allroundLight	BooleanEnumType	All-around light power status.
flashingLight	BooleanEnumType	Flashing light power status.
mastheadLight	BooleanEnumType	Mast light power status.
portSideLight	BooleanEnumType	Port light power status.
starboardSideLight	BooleanEnumType	Starboard light power status.
sternLight	BooleanEnumType	Stern light power status.
towingLight	BooleanEnumType	Towing light power status.

6.1.19 MastControl

The purpose of this service is to provide the operations and interfaces to control the mast position of the unmanned platform.

Table 63: MastControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setMast	reportMastCommandStatus
queryMastCommandAck⊕	reportMastCommandAck
cancelMastCommand⊕	reportMastCancelCommandStatus⊕

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

6.1.19.1 reportMastCommandAck

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

Namespace: UMAA::EO::MastControl

Topic: MastCommandAckReport

Data Type: MastCommandAckReportType

Table 64: MastCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
action	MastActionEnumType	The desired mast action.

6.1.19.2 reportMastCommandStatus

Description: This operation is used to report the current commanded mast action.

Namespace: UMAA::EO::MastControl

Topic: MastCommandStatus

Data Type: MastCommandStatusType

Table 65: MastCommandStatusType Message Definition

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

6.1.19.3 setMast

Description: This operation is used to set the control parameters for the Mast service.

Namespace: UMAA::EO::MastControl

Topic: MastCommand

Data Type: MastCommandType

Table 66: MastCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
action	MastActionEnumType	The desired mast action.

6.1.20 MastStatus

The purpose of this service is to provide the operations and interfaces to provide current status of the mast on the unmanned platform.

Table 67: MastStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryMast⊕	reportMast

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

6.1.20.1 reportMast

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

Namespace: UMAA::EO::MastStatus

Topic: MastReport

Data Type: MastReportType

Table 68: MastReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
state	MastStateEnumType	The state of the mast.

6.1.21 PowerControl

The purpose of this service is to provide the power control of all subsystems on the unmanned platforms.

Table 69: PowerControl Operations

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

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

6.1.21.1 reportPowerCommandAck

Description: This operation is used to report the commanded power state of the subsystems on the unmanned platform.

Namespace: UMAA::EO::PowerControl

Topic: PowerCommandAckReport

Data Type: PowerCommandAckReportType

Table 70: PowerCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
identification	StringShortDescription	The name or description of the subsystem.
state	IgnitionStateEnumType	The desired power state of the subsystem

6.1.21.2 reportPowerCommandStatus

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

Namespace: UMAA::EO::PowerControl

Topic: PowerCommandStatus

Data Type: PowerCommandStatusType

Table 71: PowerCommandStatusType Message Definition

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

6.1.21.3 setPower

Description: This operation is used to control the power state of the specified subsystems using the subsystem identifiers on the unmanned platform.

Namespace: UMAA::EO::PowerControl

Topic: PowerCommand

Data Type: PowerCommandType

Table 72: PowerCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
identification	StringShortDescription	The name or description of the subsystem.
state	IgnitionStateEnumType	The desired power state of the subsystem

6.1.22 PowerStatus

The purpose of this service is to provide the current power status of all subsystems on the unmanned platforms.

Table 73: PowerStatus Operations

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

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

6.1.22.1 reportPower

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

Namespace: UMAA::EO::PowerStatus

Topic: PowerReport

Data Type: PowerReportType

Table 74: PowerReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
name	StringShortDescription	The description of the subsystem.
powerPlantID	NumericGUID	The identification of the subsystem.
state	PowerPlantStateEnumType	Describes the power state

6.1.23 PropulsorConfig

The purpose of this service is to provide the operations and interfaces report the current configuration of the UMV propulsors.

Table 75: PropulsorConfig Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryPropulsorConfig⊕	reportPropulsorConfig

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

6.1.23.1 reportPropulsorConfig

Description: This operation is used to report the configuration of the propulsor on the unmanned platform.

Namespace: UMAA::EO::PropulsorConfig

Topic: PropulsorConfigReport

Data Type: PropulsorConfigReportType

Table 76: PropulsorConfigReportType Message Definition

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

6.1.24 PropulsorControl

The purpose of this service is to provide the operations and interfaces to control the UMV propulsors. The coordinate system is defined as attached to the vehicle frame. The X-axis points in the forward direction and the Z-axis points downward. The Y-axis is defined so as to have a right-handed coordinate system, i.e. i x j = k where i, j, and k are unit vectors along the X, Y, and Z coordinate axes. The origin of the vehicle frame must lie within a three-dimensional bounding box of the vehicle dimensions; however, it is not necessarily tied to the vehicle's center of gravity or its geometric center.

Table 77: PropulsorControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setPropulsor	reportPropulsorCommandStatus
queryPropulsorCommandAck⊕	reportPropulsorCommandAck
cancelPropulsorCommand⊕	reportPropulsorCancelCommandStatus⊕

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

6.1.24.1 reportPropulsorCommandAck

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

Namespace: UMAA::EO::PropulsorControl

Topic: PropulsorCommandAckReport

Data Type: PropulsorCommandAckReportType

Table 78: PropulsorCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
propulsion	PropulsionType	Specifies the propulsion value of the vehicle.
rotation	PropulsionRotationType	Specifies the rotation value of the vehicle.
thrustPitch	Angle	The desired propulsor thrust pitch

6.1.24.2 reportPropulsorCommandStatus

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

Namespace: UMAA::EO::PropulsorControl

Topic: PropulsorCommandStatus

Data Type: PropulsorCommandStatusType

Table 79: PropulsorCommandStatusType Message Definition

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

6.1.24.3 setPropulsor

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

Namespace: UMAA::EO::PropulsorControl

Topic: PropulsorCommand

Data Type: PropulsorCommandType

Table 80: PropulsorCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
propulsion	PropulsionType	Specifies the propulsion value of the vehicle.
rotation	PropulsionRotationType	Specifies the rotation value of the vehicle.
thrustPitch	Angle	The desired propulsor thrust pitch

6.1.25 PropulsorSpecs

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

Table 81: PropulsorSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryPropulsorSpecs⊕	reportPropulsorSpecs

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

6.1.25.1 reportPropulsorSpecs

Description: This operation is used to report the specifications of the propulsor of the unmanned platform.

Namespace: UMAA::EO::PropulsorSpecs

Topic: PropulsorSpecsReport

Data Type: PropulsorSpecsReportType

Table 82: PropulsorSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
bearingMax	RelativeAngle	The maximum bearing offset of the propulsor
bearingMin	RelativeAngle	The minimum bearing offset of the propulsor
bearingOffset	RelativeAngle	The bearing offset of the propulsor
maxReverseRPM	FrequencyRPM	The maximum reverse RPM
maxRPM	FrequencyRPM	The maximum RPM of the propulsor
minReverseRPM	FrequencyRPM	The minimum reverse RPM
minRPM	FrequencyRPM	The minimum RPM of the propulsor
mountType	MountModeEnumType	The mount type of the propulsor.
name	StringShortDescription	The name of the propulsor unit
OnOffSwitchable	BooleanEnumType	The motor on/off control
pitchMax	RelativeAngle	The maximum pitch offset of the propulsor
pitchMin	RelativeAngle	The minimum pitch offset of the propulsor
pitchOffset	RelativeAngle	The pitch offset of the propulsor
positionOffset	Position3D_PlatformXYZ	The position offset of the unit
reversible	BooleanEnumType	The motor rotation reversibility
thrustControl	BooleanEnumType	The thrust control indicator

6.1.26 PropulsorStatus

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

Table 83: PropulsorStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryPropulsor⊕	reportPropulsor

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

6.1.26.1 reportPropulsor

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

Namespace: UMAA::EO::PropulsorStatus

Topic: PropulsorReport

Data Type: PropulsorReportType

Table 84: PropulsorReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
motorFault	BooleanEnumType	The current configured motor fault
pitch	Angle	The current configured pitch rotation angle
pitchEffort	Effort	The current configured pitch rotation effort of the unit
propulsiveEffort†	Effort	The current configured propulsive effort of the propulsor. Negative values will be regarded as zero for non-reversible propulsors
RPM†	FrequencyRPM	The current configured RPM of the propulsor
state	PowerPlantStateEnumType	The current configured power state of the propulsion unit
yaw†	Angle	The current configured yaw rotation angle
yawEffort†	Effort	The current configured yaw rotation effort of the unit

6.1.27 RudderControl

The purpose of this service is to provide the operations and interfaces to control the rudder position.

Table 85: RudderControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setRudder	reportRudderCommandStatus
queryRudderCommandAck⊕	reportRudderCommandAck
cancelRudderCommand⊕	reportRudderCancelCommandStatus⊕

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

6.1.27.1 reportRudderCommandAck

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

Namespace: UMAA::EO::RudderControl

Topic: RudderCommandAckReport

Data Type: RudderCommandAckReportType

Table 86: RudderCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
angle	Angle	The desired rudder angle.

6.1.27.2 reportRudderCommandStatus

Description: This operation is used to report the status of the associated Rudder command message.

Namespace: UMAA::EO::RudderControl

Topic: RudderCommandStatus

Data Type: RudderCommandStatusType

Table 87: RudderCommandStatusType Message Definition

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

6.1.27.3 setRudder

Description: This operation is used to set the control parameters for the Rudder service. If a commanded rudder angle exceeds the maximum or minimum allowed angle, the rudder angle will be set to its maximum or minimum angle, respectively. If a commanded angle rate exceeds the maximum or minimum allowed rate, the rudder angle will be set to its maximum or minimum rate, respectively.

Namespace: UMAA::EO::RudderControl

Topic: RudderCommand

Data Type: RudderCommandType

Table 88: RudderCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
angle	Angle	The desired rudder angle.

6.1.28 RudderSpecs

The purpose of this service is to provide the operations and interfaces to provide the specifications of the rudder.

Table 89: RudderSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryRudderSpecs⊕	reportRudderSpecs

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

6.1.28.1 reportRudderSpecs

Description: This operation is used to report the specification parameters for the Rudder service.

Namespace: UMAA::EO::RudderSpecs

Topic: RudderSpecsReport

Data Type: RudderSpecsReportType

Table 90: RudderSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
maxPortAngle	Angle	The minimum amount of rudder deflection.
maxStarboardAngle	Angle	The maximum amount of rudder deflection.
name	StringShortDescription	The name of the rudder.

6.1.29 RudderStatus

The purpose of this service is to provide the operations and interfaces to provide the current status of the rudder.

Table 91: RudderStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryRudder⊕	reportRudder

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

6.1.29.1 reportRudder

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

Namespace: UMAA::EO::RudderStatus

Topic: RudderReport

Data Type: RudderReportType

Table 92: RudderReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
angle	Angle	The actual angle of the vehicle rudder.

6.1.30 SituationalSignalControl

This service provides the ability to control the active situation(s) in order to generate the corresponding audible and/or visual signals.

Table 93: SituationalSignalControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setSituationalSignal	reportSituationalSignalCommandStatus
querySituationalSignalCommandAck⊕	reportSituationalSignalCommandAck
cancelSituationalSignalCommand⊕	reportSituationalSignalCancelCommandStatus⊕

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

6.1.30.1 reportSituationalSignalCommandAck

Description: This operation is used to report the current commanded situational signals, which emits one or a combination of signals.

Namespace: UMAA::EO::SituationalSignalControl

Topic: SituationalSignalCommandAckReport

Data Type: SituationalSignalCommandAckReportType

Table 94: SituationalSignalCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
agreeToBeOvertaken	BooleanEnumType	Set vessel agrees to be overtaken signal.
alteringCourseToPort	BooleanEnumType	Set vessel turning to port signal.
alteringCourseToStarboard	BooleanEnumType	Set vessel turning to starboard signal.
engagedInMineClearanceOperations	BooleanEnumType	Set engaged in mine clearance operations signal.
inDistressNeedAssistance	BooleanEnumType	Set in distress need assistance signal.
inSightCollisionWarning	BooleanEnumType	Set vessel in sight collision warning signal.
maneuverabilityRestricted	BooleanEnumType	Set vessel maneuverability restricted signal.

Attribute Name	Attribute Type	Attribute Description
operatingAsternPropulsion	BooleanEnumType	Set vessel operating astern propulsion signal.
outOfSightCollisionWarning	BooleanEnumType	Set vessel out of sight collision warning signal.
toOvertakeOnPortSide	BooleanEnumType	Set vessel overtaking on port signal.
toOvertakeOnstarboardSide	BooleanEnumType	Set vessel overtaking on starboard signal.
towingAndPushingAlongsideOrAhead	BooleanEnumType	Set vessel towing and pushing alongside or ahead signal.
towingAndPushingBehind	BooleanEnumType	Set vessel towing and pushing behind signal.
vesselAground	BooleanEnumType	Set vessel aground signal.
vesselAnchored	BooleanEnumType	Set vessel anchored signal.
vesselBeingPushed	BooleanEnumType	Set vessel being pushed signal.
vesselBeingTowedAlongside	BooleanEnumType	Set vessel being towed alongside signal.
vesselBeingTowedBehind	BooleanEnumType	Set vessel being towed behind signal.
vesselConstrainedByHerDraft	BooleanEnumType	Set vessel constrained by her draft signal.
vesselLeavingDock	BooleanEnumType	Set vessel leaving dock signal.
vesselNotUnderControl	BooleanEnumType	Set vessel not under control signal.
vesselStopped	BooleanEnumType	Set vessel stopped signal.
vesselUnderWay	BooleanEnumType	Set vessel underway signal.
visibilityRestricted	BooleanEnumType	Set vessel operating visibility restricted signal. This signal needs to combine with signal vesselUnderWay, vesselNotUnderControl, or maneuverabilityRestricted
visibilityRestrictedStopped	BooleanEnumType	Set visibility restricted stop signal.

6.1.30.2 reportSituationalSignalCommandStatus

Description: This operation is used to retrieve the current status of the command to emit one or a combination of signals as recommended in COLREGs.

Namespace: UMAA::EO::SituationalSignalControl

Topic: SituationalSignalCommandStatus

Data Type: SituationalSignalCommandStatusType

Table 95: SituationalSignalCommandStatusType Message Definition

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

6.1.30.3 setSituationalSignal

Description: This operation is used to command the component to emit one or a combination of signals as recommended in COLREGs.

Namespace: UMAA::EO::SituationalSignalControl

Topic: SituationalSignalCommand

Data Type: SituationalSignalCommandType

Table 96: SituationalSignalCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
	Additional fields included from UMAA::UMAACommand	
agreeToBeOvertaken	BooleanEnumType	Set vessel agrees to be overtaken signal.
alteringCourseToPort	BooleanEnumType	Set vessel turning to port signal.
alteringCourseToStarboard	BooleanEnumType	Set vessel turning to starboard signal.
engagedInMineClearanceOperations	BooleanEnumType	Set engaged in mine clearance operations signal.
inDistressNeedAssistance	BooleanEnumType	Set in distress need assistance signal.
inSightCollisionWarning	BooleanEnumType	Set vessel in sight collision warning signal.
maneuverabilityRestricted	BooleanEnumType	Set vessel maneuverability restricted signal.
operatingAsternPropulsion	BooleanEnumType	Set vessel operating astern propulsion signal.
outOfSightCollisionWarning	BooleanEnumType	Set vessel out of sight collision warning signal.
toOvertakeOnPortSide	BooleanEnumType	Set vessel overtaking on port signal.
toOvertakeOnstarboardSide	BooleanEnumType	Set vessel overtaking on starboard signal.
towingAndPushingAlongsideOrAhead	BooleanEnumType	Set vessel towing and pushing alongside or ahead signal.
towingAndPushingBehind	BooleanEnumType	Set vessel towing and pushing behind signal.
vesselAground	BooleanEnumType	Set vessel aground signal.
vesselAnchored	BooleanEnumType	Set vessel anchored signal.
vesselBeingPushed	BooleanEnumType	Set vessel being pushed signal.
vesselBeingTowedAlongside	BooleanEnumType	Set vessel being towed alongside signal.
vesselBeingTowedBehind	BooleanEnumType	Set vessel being towed behind signal.
vesselConstrainedByHerDraft	BooleanEnumType	Set vessel constrained by her draft signal.
vesselLeavingDock	BooleanEnumType	Set vessel leaving dock signal.
vesselNotUnderControl	BooleanEnumType	Set vessel not under control signal.
vesselStopped	BooleanEnumType	Set vessel stopped signal.
vesselUnderWay	BooleanEnumType	Set vessel underway signal.
visibilityRestricted	BooleanEnumType	Set vessel operating visibility restricted signal. This signal needs to combine with signal vesselUnderWay, vesselNotUnderControl, or maneuverabilityRestricted
visibilityRestrictedStopped	BooleanEnumType	Set visibility restricted stop signal.

6.1.31 SituationalSignalSpecs

This function provides the ability to report the signal capabilities of the unmanned platform, such as what situational signals are supported

Table 97: SituationalSignalSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
querySituationalSignalSpecs⊕	reportSituationalSignalSpecs

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

6.1.31.1 reportSituationalSignalSpecs

Description: This operation is used to report the situational signals (as recommended in COLREGs) that the unmanned platform is capable of emitting

Namespace: UMAA::EO::SituationalSignalSpecs

Topic: SituationalSignalSpecsReport

Data Type: SituationalSignalSpecsReportType

Table 98: SituationalSignalSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
agreeToBeOvertaken	BooleanEnumType	Agree to be overtaken signal supported flag.
alteringCourseToPort	BooleanEnumType	Turning to port signal supported flag.
alteringCourseToStarboard	BooleanEnumType	Turning to starboard signal supported flag.
engagedInMineClearanceOperations	BooleanEnumType	Engaged in mine clearance operations signal supported flag.
inDistressNeedAssistance	BooleanEnumType	In distress need assistance signal supported flag.
inSightCollisionWarning	BooleanEnumType	In sight collision warning signal supported flag.
maneuverabilityRestricted	BooleanEnumType	Maneuverability restricted signal supported flag.
operatingAsternPropulsion	BooleanEnumType	Operating astern propulsion signal supported flag.
outOfSightCollisionWarning	BooleanEnumType	Out of sight collision warning signal supported flag.
toOvertakeOnPortSide	BooleanEnumType	Overtaking on port signal supported flag.
toOvertakeOnstarboardSide	BooleanEnumType	Overtaking on starboard signal supported flag.
towingAndPushingAlongsideOrAhead	BooleanEnumType	Towing and pushing alongside or ahead signal supported flag.
towingAndPushingBehind	BooleanEnumType	Towing and pushing behind signal supported flag.
vesselAground	BooleanEnumType	Aground signal supported flag.
vesselAnchored	BooleanEnumType	Anchored signal supported flag.
vesselBeingPushed	BooleanEnumType	Being pushed signal supported flag.
vesselBeingTowedAlongside	BooleanEnumType	Being towed alongside signal supported flag.
vesselBeingTowedBehind	BooleanEnumType	Being towed behind signal supported flag.
vesselConstrainedByHerDraft	BooleanEnumType	Constrained by her draft signal supported flag.
vesselLeavingDock	BooleanEnumType	Leaving dock signal supported flag.
vesselNotUnderControl	BooleanEnumType	Not under control signal supported flag.

Attribute Name	Attribute Type	Attribute Description
vesselStopped	BooleanEnumType	Stopped signal supported flag.
vesselUnderWay	BooleanEnumType	Underway signal supported flag.
visibilityRestricted	BooleanEnumType	Visibility restricted signal supported flag.
visibilityRestrictedStopped	BooleanEnumType	Visibility restricted stopped signal supported flag.

6.1.32 SituationalSignalStatus

This function provides the ability to report the status of the various situational signals on the unmanned platform.

Table 99: SituationalSignalStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
querySituationalSignal⊕	reportSituationalSignal

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

6.1.32.1 reportSituationalSignal

Description: This operation is used to report the current active situational signals, which emits one or a combination of signals.

Namespace: UMAA::EO::SituationalSignalStatus

Topic: SituationalSignalReport

Data Type: SituationalSignalReportType

Table 100: SituationalSignalReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
agreeToBeOvertaken	BooleanEnumType	Agree to be overtaken signal active status.
alteringCourseToPort	BooleanEnumType	Turning to port signal active status.
alteringCourseToStarboard	BooleanEnumType	Turning to starboard signal active status.
engagedInMineClearanceOperations	BooleanEnumType	Engaged in mine clearance operations signal active status.
inDistressNeedAssistance	BooleanEnumType	In distress need assistance signal active status.
inSightCollisionWarning	BooleanEnumType	In sight collision warning signal active status.
maneuverabilityRestricted	BooleanEnumType	Maneuverability restricted signal active status.
operatingAsternPropulsion	BooleanEnumType	Operating astern propulsion signal active status.
outOfSightCollisionWarning	BooleanEnumType	Out of sight collision warning signal active status.
toOvertakeOnPortSide	BooleanEnumType	Overtaking on port signal active status.
toOvertakeOnstarboardSide	BooleanEnumType	Overtaking on starboard signal active status.

Attribute Name	Attribute Type	Attribute Description
towingAndPushingAlongsideOrAhead	BooleanEnumType	Towing and pushing alongside or ahead signal active status.
towingAndPushingBehind	BooleanEnumType	Towing and pushing behind signal active status.
vesselAground	BooleanEnumType	Aground signal active status.
vesselAnchored	BooleanEnumType	Anchored signal active status.
vesselBeingPushed	BooleanEnumType	Being pushed signal active status.
vesselBeingTowedAlongside	BooleanEnumType	Being towed alongside signal active status.
vesselBeingTowedBehind	BooleanEnumType	Being towed behind signal active status.
vesselConstrainedByHerDraft	BooleanEnumType	Constrained by her draft signal active status.
vesselLeavingDock	BooleanEnumType	Leaving dock signal active status.
vesselNotUnderControl	BooleanEnumType	Not under control signal active status.
vesselStopped	BooleanEnumType	Stopped signal active status.
vesselUnderWay	BooleanEnumType	Underway signal active status.
visibilityRestricted	BooleanEnumType	Visibility restricted signal active status. This signal needs to combine with signal vesselUnderWay, vesselNotUnderControl, or maneuverabilityRestricted.
visibilityRestrictedStopped	BooleanEnumType	Visibility restricted stopped signal active status.

6.1.33 ThrusterControl

The purpose of this service is to provide the operations and interfaces to control the vehicle bow and stern thrusters.

Table 101: ThrusterControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setThruster	reportThrusterCommandStatus
queryThrusterCommandAck⊕	reportThrusterCommandAck
cancelThrusterCommand⊕	reportThrusterCancelCommandStatus⊕

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

6.1.33.1 reportThrusterCommandAck

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

Namespace: UMAA::EO::ThrusterControl

Topic: ThrusterCommandAckReport

Data Type: ThrusterCommandAckReportType

Table 102: ThrusterCommandAckReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommandStatusBase		
thruster	ThrusterControlType	Defines the attributes needed to control the thruster's desired speed/power.

6.1.33.2 reportThrusterCommandStatus

Description: This operation is used to report the status of the associated Thruster command message.

Namespace: UMAA::EO::ThrusterControl

Topic: ThrusterCommandStatus

Data Type: ThrusterCommandStatusType

Table 103: ThrusterCommandStatusType Message Definition

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

6.1.33.3 setThruster

Description: This operation is used to set the control parameters for the Thruster service. The consumer must perform a "cancel" of the command to initiate the end of command execution as this command has no determinate end of execution.

Namespace: UMAA::EO::ThrusterControl

Topic: ThrusterCommand

Data Type: ThrusterCommandType

Table 104: ThrusterCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAACommand		
thruster	ThrusterControlType	Defines the attributes needed to control the thruster's desired speed/power.

6.1.34 ThrusterSpecs

The purpose of this service is to provide the operations and interfaces to retrieve the specifications of the vehicle bow and stern thrusters.

Table 105: ThrusterSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryThrusterSpecs⊕	reportThrusterSpecs

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

6.1.34.1 reportThrusterSpecs

Description: This operation is used to report the specification parameters for the Thruster service.

Namespace: UMAA::EO::ThrusterSpecs

Topic: ThrusterSpecsReport

Data Type: ThrusterSpecsReportType

Table 106: ThrusterSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
thruster	ThrusterSpecsType	Defines the attributes used to report the current status of the thruster.

6.1.35 ThrusterStatus

The purpose of this service is to provide the operations and interfaces to monitor the vehicle bow and stern thrusters.

Table 107: ThrusterStatus Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryThruster⊕	reportThruster

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

6.1.35.1 reportThruster

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

Namespace: UMAA::EO::ThrusterStatus

Topic: ThrusterReport

Data Type: ThrusterReportType

Table 108: ThrusterReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
effort†	Effort	The current effort of the thruster. Negative effort causes motion in the port direction.
motorFault	BooleanEnumType	The current motor fault.
RPM†	FrequencyRPM	The current RPM of the thruster.
state	PowerPlantStateEnumType	The current power state of the thruster

6.1.36 UVPlatformSpecs

The purpose of this service is to report the physical and operational capabilities of the unmanned platform. For example, this information should be used in UMAA services to validate and reject commands received that are not able to be performed by the unmanned vehicle.

Table 109: UVPlatformSpecs Operations

Service Requests (Inputs)	Service Responses (Outputs)
queryUVPlatformCapabilities⊕	reportUVPlatformCapabilities
queryUVPlatformSpecs⊕	reportUVPlatformSpecs

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

6.1.36.1 reportUVPlatformCapabilities

Description: This operation is used to report the current operational capabilities of the unmanned platform.

Namespace: UMAA::EO::UVPlatformSpecs

Topic: UVPlatformCapabilitiesReport

Data Type: UVPlatformCapabilitiesReportType

Table 110: UVPlatformCapabilitiesReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
endurance	Duration_Seconds	The expected endurance when fully fueled in seconds.
maxDepth	Distance_BSL	The maximum diving distance below water surface
range	Distance	The expected range when fully fueled.

Attribute Name	Attribute Type	Attribute Description
surfaceCapabilities	UVPlatformCapabilitiesType	The capabilities of the unmanned vehicle while operating on the surface.
towingCapacity	Mass_MetricTon	The maximum towing capacity
underwaterCapabilities†	UVPlatformCapabilitiesType	The capabilities of the unmanned vehicle while operating submerged.

6.1.36.2 reportUVPlatformSpecs

Description: This operation is used to report the physical specifications of the unmanned platform.

Namespace: UMAA::EO::UVPlatformSpecs

Topic: UVPlatformSpecsReport

Data Type: UVPlatformSpecsReportType

Table 111: UVPlatformSpecsReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
back	Distance	The length of the back of the vehicle, measured along the negative X axis of the vehicle coordinate from
baselineBuoyancy	Mass	The additional mass required to either achieve neutral buoyancy or sink the submergible vehicle in fresh water
beamAtWaterline	Distance	The beam at waterline (BWL)
bottom	Distance	The bottom of the vehicle, measured along the positive Z axis of the vehicle coordinate frame
centerOfBuoyancy	Position3D_PlatformXYZ	The measurements on the vehicle coordinate frame at which the center of buoyancy of the vehicle is located
centerOfGravity	Position3D_PlatformXYZ	The measurements on the vehicle coordinate frame at which the center of gravity of the vehicle is located
displacement	Mass	The equivalent mass of the fluid displaced by the unmanned maritime platform
draft	Distance	The draft of the unmanned platform
front	Distance	The length of the front of the vehicle, measured along the positive X axis of the vehicle coordinate frame
heightOverall	Distance	The overall height of the unmanned platform
hullDepth	Distance	The depth of the unmanned platform
left	Distance	The side of the vehicle, measured along the negative Y axis of the vehicle coordinate frame
lengthAtWaterline	Distance	The length at waterline (LWL)
lengthOverall	Distance	The length over all (LOA)
massOnLand	Mass	The mass of a dry vehicle on land with empty ballast tanks and suspend from a crane
name	StringShortDescription	The name of the unmanned platform

Attribute Name	Attribute Type	Attribute Description
outerWidth	Distance	The outer diameter as a clean, cylindrical pressure hull of the unmanned platform
right	Distance	The side of the vehicle, measured along the positive Y axis of the vehicle coordinate frame
top	Distance	The top of the vehicle, measured along the negative Z axis of the vehicle coordinate frame
widthOverall	Distance	The overall distance of the body

6.1.37 WhistleControl

The purpose of this service is to provide the whistle control and the current status of a specific whistle on the unmanned surface platform.

Table 112: WhistleControl Operations

Service Requests (Inputs)	Service Responses (Outputs)
setWhistleControl	reportWhistleControlCommandStatus
queryWhistleControlCommand⊕	reportWhistleControlCommand
cancelWhistleControlCommand⊕	reportWhistleControlCancelCommandStatus⊕
queryWhistleControl⊕	reportWhistleControl

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

6.1.37.1 reportWhistleControl

Description: This operation is used to retrieve the current state of the whistle sounding device on the unmanned platform.

Namespace: UMAA::EO::WhistleControl

Topic: WhistleControlReport

Data Type: WhistleControlReportType

Table 113: WhistleControlReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
blastCondition†	BlastConditionEnumType	The blast state of the annunciator
blastNumber†	PositiveCount	The number of times blasted within the current set; includes current blast if in progress
status	OnOffStatusEnumType	The current on/off status of the whistle switch

6.1.37.2 reportWhistleControlCommand

Description: This operation is used to retrieve the current whistle sounding device command on the unmanned platform.

Namespace: UMAA::EO::WhistleControl

Topic: WhistleControlCommandReport

Data Type: WhistleControlCommandReportType

Table 114: WhistleControlCommandReportType Message Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::UMAAStatus		
status	OnOffStatusEnumType	Whistle active state.
whistleProperties†	WhistlePropertiesType	Whistle properties. If the attribute is not provided, it means turn off the whistle.

6.1.37.3 reportWhistleControlCommandStatus

Description: This operation is used to retrieve the state of the whistle sounding device command on the unmanned platform.

Namespace: UMAA::EO::WhistleControl

Topic: WhistleControlCommandStatus

Data Type: WhistleControlCommandStatusType

Table 115: WhistleControlCommandStatusType Message Definition

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

6.1.37.4 setWhistleControl

Description: This operation is used to control the whistle sounding device on 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::EO::WhistleControl

Topic: WhistleControlCommand

Data Type: WhistleControlCommandType

Table 116: WhistleControlCommandType Message Definition

Attribute Name	Attribute Type	Attribute Description
	Additional fields included from UMAA::UMAACommand	
status	OnOffStatusEnumType	Whistle active state.
whistleProperties†	WhistlePropertiesType	Whistle properties. If the attribute is not provided, it means turn off the whistle.

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 117: 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 118: 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 119: 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 120: 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 121: 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 122: 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 BallastFillType

Namespace: UMAA::EO::BallastTank::BallastFillType

Description: Union Type. The desired fill amount of the ballast tank

Table 123: BallastFillType Union(s)

Type Name	Type Description
BallastMassType	The desired filled or empty level the ballast tank measured by mass
LevelType	Defines the desired percentage filled or empty level of the ballast tank

6.2.8 BallastMassType

Namespace: UMAA::Common::Measurement::BallastMassType

Description: The desired filled or empty level the ballast tank measured by mass

Table 124: BallastMassType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::EO::BallastTank::BallastFillType		
mass	Mass	specifies the desired filled or empty level the ballast tank measured by mass

6.2.9 BallastPumpFlowRateType

Namespace: UMAA::EO::BallastTank::BallastPumpFlowRateType

Description: Union Type. The desired flow rate to fill or empty the ballast pump

Table 125: BallastPumpFlowRateType Union(s)

Type Name	Type Description
MassBallastFlowRateType	The desired flow rate to fill or empty the ballast pump measured by mass
VolumeBallastFlowRateType	The desired flow rate to fill or empty the ballast pump measured by volume

6.2.10 BatteryCellDataType

Namespace: UMAA::EO::BatteryStatus::BatteryCellDataType

Description: This structure is used to report the current status of a cell in the battery system.

Table 126: BatteryCellDataType Structure Definition

Attribute Name	Attribute Type	Attribute Description
current	PowerBusCurrent	The runtime current of the battery cell.
temperature	Temperature	The temperature of the battery cell.
voltage	PowerBusVoltage	The voltage of the battery cell.

6.2.11 BlastPropertiesType

Namespace: UMAA::Common::Propulsion::BlastPropertiesType

Description: This structure is used to describe the properties of a signal blast.

Table 127: BlastPropertiesType Structure Definition

Attribute Name	Attribute Type	Attribute Description
blastNumber	PositiveCount	The number of times commanded to blast, if attribute is not included then it is continuous
restTime	Duration_Seconds	1 to 255: The time in seconds between blasts

6.2.12 LevelType

Namespace: UMAA::Common::Measurement::LevelType

Description: Defines the desired percentage filled or empty level of the ballast tank

Table 128: LevelType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::EO::BallastTank::BallastFillType		
level	Volume_Percent	specifies The desired percentage filled or empty level of the ballast tank

6.2.13 MassBallastFlowRateType

Namespace: UMAA::EO::BallastTank::MassBallastFlowRateType

Description: The desired flow rate to fill or empty the ballast pump measured by mass

Table 129: MassBallastFlowRateType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::EO::BallastTank::BallastPumpFlowRateType		
massBallastFlowRate	MassFlowRate	specifies the desired flow rate to fill or empty the ballast pump measured by mass

6.2.14 Position3D_PlatformXYZ

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

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

Table 130: Position3D_PlatformXYZ Structure Definition

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

6.2.15 PropulsionRotationType

Namespace: UMAA::Common::Propulsion::PropulsionRotationType

Description: Union Type. Propulsion value in either effort or RPM.

Table 131: PropulsionRotationType Union(s)

Type Name	Type Description
PropulsiveRotationAngleType	Defines the rotational value as an angle.
PropulsiveRotationEffortType	Defines the rotational value as an effort.

6.2.16 PropulsionType

Namespace: UMAA::Common::Propulsion::PropulsionType

Description: Union Type. Propulsion value in either effort or RPM.

Table 132: PropulsionType Union(s)

Type Name	Type Description
PropulsiveEffortType	Defines the propulsive value as an effort.
PropulsiveRPMType	Defines the propulsive value as an RPM value.

6.2.17 PropulsiveEffortType

Namespace: UMAA::Common::Propulsion::PropulsiveEffortType

Description: Defines the propulsive value as an effort.

Table 133: PropulsiveEffortType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::Common::Propulsion::PropulsionType		
propulsiveEffort	Effort	The desired propulsive effort, as a percent value (-100 to 100). Negative values will cause motion in the port direction for fixed bow or stern thrusters.

6.2.18 PropulsiveRPMType

Namespace: UMAA::Common::Propulsion::PropulsiveRPMType

Description: Defines the propulsive value as an RPM value.

Table 134: PropulsiveRPMType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::Common::Propulsion::PropulsionType		
RPM	FrequencyRPM	The desired RPM of the propulsor thruster.

6.2.19 PropulsiveRotationAngleType

Namespace: UMAA::Common::Propulsion::PropulsiveRotationAngleType

Description: Defines the rotational value as an angle.

Table 135: PropulsiveRotationAngleType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::Common::Propulsion::PropulsionRotationType		
thrustYaw	Angle	The desired propulsor thrust yaw

6.2.20 PropulsiveRotationEffortType

Namespace: UMAA::Common::Propulsion::PropulsiveRotationEffortType

Description: Defines the rotational value as an effort.

Table 136: PropulsiveRotationEffortType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::Common::Propulsion::PropulsionRotationType		
yawEffort	RotationalEffort	The desired rotational effort, as a percent value.

6.2.21 Quaternion

Namespace: BasicTypes::Quaternion

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

Table 137: 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.

6.2.22 RotationalEffort

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

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

Table 138: 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.

6.2.23 ThrusterControlType

Namespace: UMAA::EO::ThrusterControl::ThrusterControlType

Description: This structure is used to set the thruster output of the unmanned platform.

Table 139: ThrusterControlType Structure Definition

Attribute Name	Attribute Type	Attribute Description
effort	Effort	The desired effort of the thruster. Negative values will cause motion in the port direction.
state	PowerPlantStateEnumType	The desired power state of the thruster

6.2.24 ThrusterSpecsType

Namespace: UMAA::EO::ThrusterSpecs::ThrusterSpecsType

Description: This structure is used to define the attributes used to report the current specification parameters of the thruster.

Table 140: ThrusterSpecsType Structure Definition

Attribute Name	Attribute Type	Attribute Description
maxRPM	FrequencyRPM	The maximum RPM of the thruster.
mountType	MountModeEnumType	The mount type of the thruster
name	StringShortDescription	The name of the thruster unit.

6.2.25 UVPlatformCapabilitiesType

Namespace: UMAA::EO::UVPlatformSpecs::UVPlatformCapabilitiesType

Description: This structure is used to describe the UV platform's capabilities.

Table 141: UVPlatformCapabilitiesType Structure Definition

Attribute Name	Attribute Type	Attribute Description
cruisingSpeed	GroundSpeed	The speed for maximum range of the unmanned platform
maxForwardSpeed	GroundSpeed	The maximum forward speed of the unmanned platform
maxPitchRate	PitchRate	The maximum pitch rate
maxReverseSpeed	GroundSpeed	The maximum reverse speed of the unmanned platform
maxTowingSpeed†	GroundSpeed	The maximum towing speed
maxTowingTurnRate†	TurnRate	The maximum towing turn rate
maxTurnRate	TurnRate	The maximum turning rate
minTowingSpeed†	GroundSpeed	The minimum towing speed
minTowingTurnRate†	TurnRate	The minimum towing turn rate

6.2.26 VolumeBallastFlowRateType

Namespace: UMAA::EO::BallastTank::VolumeBallastFlowRateType

Description: The desired flow rate to fill or empty the ballast pump measured by volume

Table 142: VolumeBallastFlowRateType Structure Definition

Attribute Name	Attribute Type	Attribute Description
Additional fields included from UMAA::EO::BallastTank::BallastPumpFlowRateType		
volumeBallastFlowRate	VolumetricFlowRate	specifies the desired flow rate to fill or empty the ballast pump measured by volume

6.2.27 WhistlePropertiesType

Namespace: UMAA::Common::Propulsion::WhistlePropertiesType

Description: This structure is used to describe the properties of a signal whistle.

Table 143: WhistlePropertiesType Structure Definition

Attribute Name	Attribute Type	Attribute Description
restTime	Duration_Seconds	1 to 255: The time in seconds between blasts
whistleNumber	PositiveCount	The number of times commanded to blast, if attribute is not included then it is continuous
whistleType	BlastKindEnumType	Whistle type

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 AnchorActionEnumType

Namespace: UMAA::Common::MaritimeEnumeration::AnchorActionEnumType

Description: Defines a mutually exclusive set of values for the anchor action.

Table 144: AnchorActionEnumType Enumeration

Enumeration Value	Description
LOWER	Lower the anchor.
RAISE	Raise the anchor.
STOP	Stop anchor from lowering or raising.

6.3.2 AnchorKindEnumType

Namespace: UMAA::Common::MaritimeEnumeration::AnchorKindEnumType

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

Table 145: AnchorKindEnumType Enumeration

Enumeration Value	Description
COMMERCIAL_STOCKLESS	Anchor type is commercial stockless.
DANFORTH	Anchor type is danforth.
FOUR_FLUKE	Anchor type is four-fluke.
GENERAL	Anchor type is general.
LIGHTWEIGHT	Anchor type is lightweight.
MARK_2_LWT	Anchor type is mark 2 lightweight.
MARK_2_STOCKLESS	Anchor type is mark 2 stockless.
MUSHROOM	Anchor type is mushroom.
NAVY_TYPE_STOCK	Anchor type is navy type stock.
NONMAGNETIC	Anchor type is nonmagnetic.
STANDARD_NAVY_STOCKLESS	Anchor type is standard navy stockless.
TWO_FLUKE_BALANCED_FLUKE	Anchor type is two-fluke balanced fluke.
WEDGE_BLOCK_LWT	Anchor type is wedge block lightweight.

6.3.3 AnchorLocationEnumType

Namespace: UMAA::Common::MaritimeEnumeration::AnchorLocationEnumType

Description: Defines the location of the anchor.

Table 146: AnchorLocationEnumType Enumeration

Enumeration Value	Description
BOWER	A bower anchor is carried on the bow.
KEEL	A keel anchor is housed within the hull neer the keel.
STERN	A stern anchor is carried on the stern.

6.3.4 AnchorStateEnumType

Namespace: UMAA::Common::MaritimeEnumeration::AnchorStateEnumType

Description: Defines a mutually exclusive set of values of the anchor state.

Table 147: AnchorStateEnumType Enumeration

Enumeration Value	Description
DEPLOYED	Anchor is deployed.
LOWERING	Anchor is lowering.
STOPPED	Anchor is neither DEPLOYED nor STOWED, but is not in the process of LOWERING or RAISING.
RAISING	Anchor is raising.
STOWED	Anchor is stowed.

6.3.5 BlastConditionEnumType

Namespace: UMAA::Common::MaritimeEnumeration::BlastConditionEnumType

Description: A mutually exclusive set of values that defines the blast condition of a whistle, bell, or gong on board of the unmanned vehicle.

Table 148: BlastConditionEnumType Enumeration

Enumeration Value	Description
BLASTING	Unmanned vehicle whistle, bell, or gong is currently blasting
RESTING	Unmanned vehicle whistle, bell, or gong is currently resting

6.3.6 BlastKindEnumType

Namespace: UMAA::Common::MaritimeEnumeration::BlastKindEnumType

Description: A mutually exclusive set of values that defines duration of the blast of the annunciator such as whistle, bell, gong, etc. on board of the unmanned vehicle.

Table 149: BlastKindEnumType Enumeration

Enumeration Value	Description
LONG_BLAST	Long blast, four to six seconds
SHORT_BLAST	Short blast, about one second duration

6.3.7 CommandStatusReasonEnumType

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

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

Table 150: 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.

6.3.8 EngineKindEnumType

Namespace: UMAA::Common::MaritimeEnumeration::EngineKindEnumType

Description: Defines a mutually exclusive set of values that defines the engine kind.

Table 151: EngineKindEnumType Enumeration

Enumeration Value	Description
DIESEL	A diesel engine.
GAS	A gas engine.

6.3.9 IgnitionStateEnumType

Namespace: UMAA::Common::MaritimeEnumeration::IgnitionStateEnumType

Description: Defines a mutually exclusive set of values that defines the state of engine ignition.

Table 152: IgnitionStateEnumType Enumeration

Enumeration Value	Description
OFF	The engine is off.
RUN	The engine is running.
START	The engine is starting.

6.3.10 MastActionEnumType

Namespace: UMAA::Common::MaritimeEnumeration::MastActionEnumType

Description: A mutually exclusive set of values that defines the action of the mast.

Table 153: MastActionEnumType Enumeration

Enumeration Value	Description
LOWER	set to lower the mast down
RAISE	set to raise the mast up
STOP	set to stop the mast

6.3.11 MastStateEnumType

Namespace: UMAA::Common::MaritimeEnumeration::MastStateEnumType

Description: A mutually exclusive set of values that defines the state of the mast.

Table 154: MastStateEnumType Enumeration

Enumeration Value	Description
DOWN	set when the mast is down
MOVING_DOWN	set when the mast is moving down
MOVING_UP	set when the mast is moving up
UP	set when the mast is up

6.3.12 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 155: 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.

6.3.13 MountModeEnumType

Namespace: UMAA::Common::MaritimeEnumeration::MountModeEnumType

Description: A mutually exclusive set of values that defines the mounting type of the propulsor on the unmanned platform.

Table 156: MountModeEnumType Enumeration

Enumeration Value	Description
FIXED	Fixed
FIXED_BOW_THRUSTER	Fixed bow thruster
FIXED_PORT_PROPULSOR	Fixed port propulsor
FIXED_STERN_THRUSTER	Fixed stern thruster
VARIABLE_BEARING	Variable bearing
VARIABLE_PITCH	Variable pitch
VARIABLE_PITCH_AND_BEARING	Variable pitch and bearing

6.3.14 OnOffStatusEnumType

Namespace: UMAA::Common::Enumeration::OnOffStatusEnumType

Description: OnOffStatusEnumTypeLDM is a Realization of OnOffStatusEnumType which is a mutually exclusive set of values that defines the on/off status of a device or subsystem.

Table 157: OnOffStatusEnumType Enumeration

Enumeration Value	Description
OFF	The device or subsystem is off.
ON	The device or subsystem is on.

6.3.15 PowerPlantStateEnumType

Namespace: UMAA::Common::MaritimeEnumeration::PowerPlantStateEnumType

Description: A mutually exclusive set of values that defines the power state of each power plant on the unmanned platform.

Table 158: PowerPlantStateEnumType Enumeration

Enumeration Value	Description
FAULT	Faulted
OFF	Off
ON	On

6.3.16 PumpStateEnumType

Namespace: UMAA::Common::MaritimeEnumeration::PumpStateEnumType

Description: A mutually exclusive set of values that defines the mode of operation of each pump on the unmanned platform.

Table 159: PumpStateEnumType Enumeration

Enumeration Value	Description
FAULT	Faulted
OFF	Off
ON_FORWARD	Running forward direction
ON_REVERSE	Running reverse direction

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 160: Type Definitions

Type Name	Primitive Type	Range of Values	Description
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.
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.
Charge	double	units=Coulomb minInclusive=0 maxInclusive=3600000 fractionDigits=0	Represents physical property of matter that causes it to experience a force when placed in an electromagnetic field. Measured in Coulomb.
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.
Distance_BSL	double	units=Meter minInclusive=0 maxInclusive=10000 fractionDigits=3	The distance below sea level in meters.
Down	double	axisAbbrev=Z axisDirection=down axisUnit=Meter maximumValue=50000 minimumValue=-50000 rangeMeaning=exact resolution=0.001	The Down axis is used for measuring position and increases in magnitude as values extend toward the center of the Earth. Down measurements are expressed in meters.
Duration_Hours	double	units=Hour minInclusive=0 maxInclusive=10505 fractionDigits=3	Represents a time duration in hours.

Type Name	Primitive Type	Range of Values	Description
Duration_Seconds	double	units=Seconds minInclusive=0 maxInclusive=37817280 fractionDigits=6	Represents a time duration in seconds.
Effort	double	fractionDigits=3 maxInclusive=100 minInclusive=-100 units=Percent referenceFrame=PlatformXYZ	Represents the level of effort measured in percent.
ElectricalPower	double	units=Watt minInclusive=0 maxInclusive=100000000 fractionDigits=3	Represents the rate at which electric energy is transferred by an electric circuit measured in watts.
Energy_Percent	double	units=Percent minInclusive=0 maxInclusive=1000 fractionDigits=3	Defines a percentage 100% = 100.0. Values greater than 100% are allowed.
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.
Force	double	units=Newton minInclusive=0 maxInclusive=100000000 fractionDigits=3	Represents the degree of force measured in Newtons.
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.
FrequencyRPM	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.
GroundSpeed	double	units=MeterPerSecond minInclusive=0 maxInclusive=200 fractionDigits=6	This type stores speed in meters/s.
Mass	double	units=Kilogram minInclusive=0 maxInclusive=100000000 fractionDigits=3	This type stores mass in kilograms.
Mass_MetricTon	double	fractionDigits=3 maxInclusive=100,000 minInclusive=0 units=MetricTon referenceFrame=Counting	Represents the property of physical body measured in non-SI derived unit, metric ton.

Type Name	Primitive Type	Range of Values	Description
MassFlowRate	double	fractionDigits=6 maxInclusive=1,000,000 minInclusive=0 units=KilogramPerSecond referenceFrame=Counting	Represents the mass flow rate measured in kilogram per second.
NumericGUID	octet[16]	units=N/A minInclusive=0 maxInclusive=(2^128)-1 fractionDigits=0	Represents a 128-bit number according to RFC 4122 variant 2
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.
PositiveCount	double	units=N/A minInclusive=1 maxInclusive=2147483647 fractionDigits=0	Represents a whole (non-fractional) number that can be positive but not zero.
PowerBusCurrent	double	units=Ampere minInclusive=-100000 maxInclusive=100000 fractionDigits=3	Represents the time rate of flow of electric charge measured in amperes.
PowerBusVoltage	double	units=Volt minInclusive=-100000 maxInclusive=100000 fractionDigits=3	Represents the potential difference in charge between two points in an electrical field measured in volts.
Pressure_KiloPascals	double	units=KiloPascal minInclusive=0 maxInclusive=51200 fractionDigits=3	Represents barometric pressure and is stored in KiloPascals.
Pressure_Percent	double	minInclusive=0 maxInclusive=200 fractionDigits=3 referenceFrame=Counting	Represents the weight or force per unit area that is produced when something presses or pushes against something else.
RelativeAngle	double	fractionDigits=3 maxInclusive=3.1415926535897932384626433832795 minInclusive=-3.1415926535897932384626433832795 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.
Size_Numeral	double	units=N/A minInclusive=-1000000000000 maxInclusive=1000000000000 fractionDigits=3	Represents nonnegative integers.

Type Name	Primitive Type	Range of Values	Description
StringLongDescription	string	fractionDigits=N/A length=4095 maxExclusive=N/A maxInclusive=N/A minExclusive=N/A minInclusive=N/A units=N/A	Represents a long format description.
StringShortDescription	string	fractionDigits=N/A length=1023 maxExclusive=N/A maxInclusive=N/A minExclusive=N/A minInclusive=N/A units=N/A	Represents a short format description.
Temperature	double	units=Celsius minInclusive=-273 maxInclusive=1000 fractionDigits=3	Represents the degree or intensity of warmness or coldess presence in a substance. Measured in Celsius.
TurnRate	double	units=RadianPerSecondreference minInclusive=0 maxInclusive=32.767 fractionDigits=3 referenceFrame=Counting	TurnRate specifies the rate of change of the heading angle of a platform.
Volume_CubicMeter	double	units=VolumeCubicMeter minInclusive=0 maxInclusive=1000 fractionDigits=6 referenceFrame=Counting	Represents the quantity of three-dimensional space enclosed by some closed boundary
Volume_Percent	double	units=Percent minInclusive=0 maxInclusive=1000 fractionDigits=3	Defines a percentage 100% = 100.0. Values greater than 100% are allowed.
VolumetricFlowRate	double	fractionDigits=6 maxInclusive=100,000,000 minInclusive=0 units=CubicMeterPerSecond referenceFrame=Counting	Specifies the amount of fluid moving through a pipe or channel per unit time.

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