Input paper: [[1]](#footnote-1) ENAV22-8.2.9

Input paper for the following Committee(s): check as appropriate Purpose of paper:

**□** ARM **□** ENG **□** PAP **x** Input

**x** ENAV **□** VTS **□** Information

Agenda item [[2]](#footnote-2) 8

Technical Domain / Task Number 2 1.2

Author(s) / Submitter(s) China Maritime Safety Administration

Proposal to Support Instant Messaging Data Exchange Pattern in The   
e-Navigation Technical Service Specification

# Summary

This document proposes to introduce Instant Messaging (IM) into the data exchange patterns of the E-Navigation Technical Service Specification. Some information is given about the development and testing of IM applied to navigational assistance services in the Pearl River Delta (PRD) e-Navigation testbed. Through the technology of instant messaging, different navigational service information could be sent to different ships, depending on their different navigational situation.

## Purpose of the document

The Committee is invited to consider this proposal.

## Related documents

ENAV21-9.2 Draft Guideline on Specification of e-Navigation Technical Services.

# Background

With reference to the "Draft Guideline on e-Navigation Technical Service Specification" (ENAV21-9.2), a series of service specifications were developed in the PRD e-Navigation Project, in which a Navigational Assistance Instant Messaging Service(NAIMS) was developed. The NAIMS adopts the Instant Messaging(IM) data exchange pattern to provide customized and efficient navigation assistance service for users.

Instant Messaging (IM) technology is a type of online chat that offers real-time text transmission over the Internet. Short messages are typically transmitted between two parties, when each user select "send". Some IM applications can use push technology to provide real-time text, which transmits messages character by character, as they are composed. More advanced instant messaging can add file transfer, clickable hyperlinks, Voice over IP, or video chat, as well as encoded text.

Depending on the IM protocol, the technical architecture can be peer-to-peer (direct point-to-point transmission) or client-server (an Instant Message service centre retransmits messages from the sender to the communication device). At present, IM technology is not only used in the scenario of chat, it has been widely used in news/entertainments/search/ e-commerce/business cooperation/customer service etc. For example, after online shopping, the user’s IM app will continuously receive the logistics information of purchased products.

Currently, Instant Messaging technology is very mature, and the public mobile communications are already available in coastal waters, while the distant seas can use satellite communications networks, which can meet the technical requirements of instant messaging.

MQTT is a connectivity protocol developed by IBM which will be an important part of "IoT" technology. This protocol, which supports all platforms and connects almost all internet-connected objects to the outside world, is used as a communications protocol for sensors and actuators, such as connecting homes via Twitter. It can be used as one of the technology to support Instant Messaging. It was designed as an extremely lightweight publish/subscribe messaging transport. It is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium. For example, it has been used in sensors communicating to a broker via satellite link, over occasional dial-up connections with healthcare providers, and in a range of home automation and small device scenarios. It is also ideal for mobile applications because of its small size, low power usage, minimised data packets, and efficient distribution of information to one or many receivers.

Based on the MQTT protocol, the PRD project developed a service instance of NAIMS that can send navigational assistance messages, such as navigation guides, collision avoidance warnings, bridge operation support, etc., according to ship parameters, especially real-time position, speed and heading.

For example, when a ship which has already connected the service, is sailing on the main channel, the shore-based system will conduct a comprehensive analysis, based on the ship’s information, especially real-time dynamic information, such as location, speed, direction, in conjunction with the situation of the area in which the ship is sailing. Relevant navigational assistance instant messages will be sent to the ship, based on the results of the analysis, to help mariners in decision-making.

Proved by facts, this data exchange pattern has the characteristics of good timeliness and strong pertinence, it can effectively reduce the burden of information filtering in the process of obtaining navigation decision-making assistant information for ship users.

The NAIMS was tested in the PRD project, which proved that this data exchange pattern is effective and efficient, and can reduce the burden on the mariners to identify navigation decision aid information.

# Discussion

Difference between IM and Publish/Subscribe：

* The Instant Messaging we proposed mainly refers to point-to-point transmission, that is, the messages sent by the server is only for a specific user. Even though the server send messages to multiple receivers at the same time, the messages is different to different receivers, that means the content of the message is determined by the behavior of the receiver. This data exchange pattern is based on the Message Queue. It may also be used for point to point data exchange between ships in the future.
* While pattern of Publish / Subscribe usually refers to one-to-many transmission, in which multiple clients may receive the same information, content is determined by the server side. It is not applicable to exchange data between ships.

So far, the Draft Guideline of e-Navigation Technical Service Specification has provided five data exchange modes: one-way, request-feedback, request-callback, publish-subscribe and broadcast.

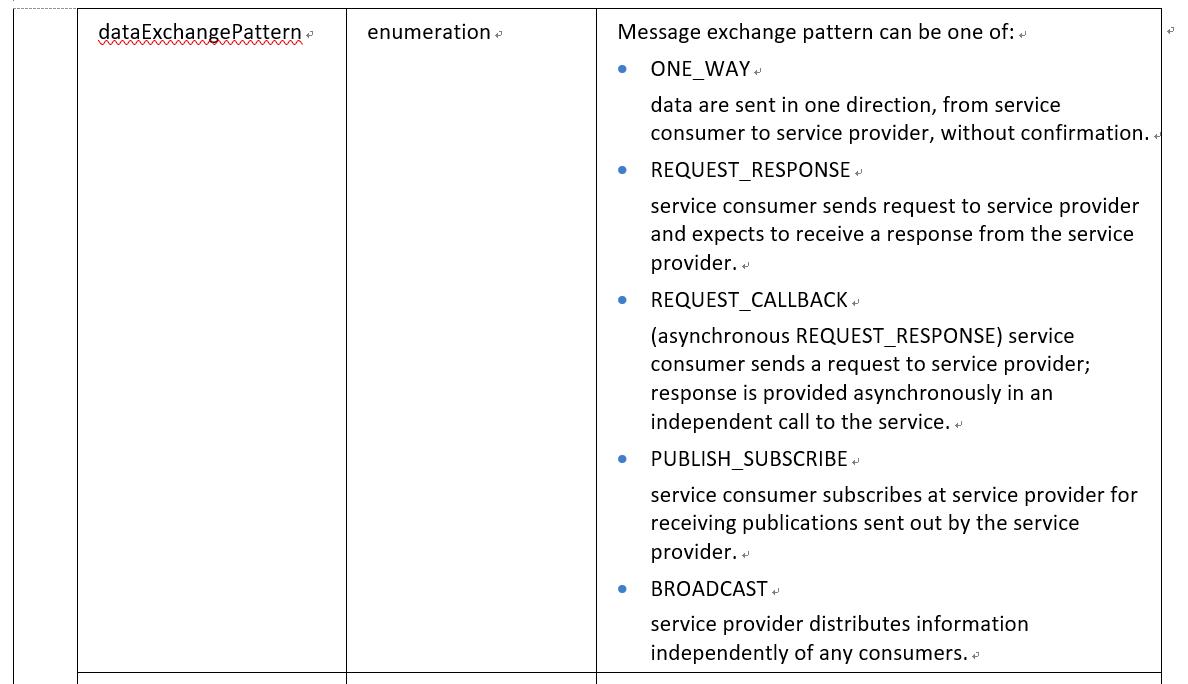


Figure 1 ENAV21-9.2 ‘The Specification on e-Navigation Technical Services, 3.3 Service Specification XSD Structure table 1’

This document will give the Service Specification for Navigation Assistant Instant Messaging Services in the ANNEXs, and recommended ENAV Committee add IM as a new data exchange pattern to ‘The Specification on e-Navigation Technical Services, 3.3 Service Specification XSD Structure table 1’.

# References

1. https://en.wikipedia.org/wiki/Instant\_messaging..........

# Action requested of the Committee

The Committee is requested to consider the proposals and take action as appropriate.

**ANNEX 1 NAIMS Service Specification V0.1**

**ANNEX 2 NAIMS REST Service Technical Design v0.1**

**ANNEX 3 NAIMS Service Instance Description v0.1**

ANNEX1

**NAIMS Service Instance Description v0.1**

CONTENTS

[1 Introduction 4](#_Toc523059477)

[1.1 Purpose of the Document 4](#_Toc523059478)

[1.2 Intended Readership 4](#_Toc523059479)

[1.3 Inputs from Other Projects 4](#_Toc523059480)

[2 Service Identification 5](#_Toc523059481)

[3 Operational Context 6](#_Toc523059482)

[3.1 Functional and Non-functional Requirements 6](#_Toc523059483)

[3.2 Other Constraints 8](#_Toc523059484)

[3.2.1 Relevant Industrial Standards 8](#_Toc523059485)

[3.2.2 Operational Nodes 8](#_Toc523059486)

[3.2.3 Operational Activities 9](#_Toc523059487)

[4 Service Overview 10](#_Toc523059488)

[4.1 Service Interfaces 10](#_Toc523059489)

[5 Service Data Model 11](#_Toc523059490)

[5.1 Message 11](#_Toc523059491)

[5.2 MessageDesc 12](#_Toc523059492)

[5.3 ShipInfo 12](#_Toc523059493)

[5.4 Content 12](#_Toc523059494)

[5.5 SpeedLimited 13](#_Toc523059495)

[5.6 TSS 13](#_Toc523059496)

[5.7 NoEntry 13](#_Toc523059497)

[5.8 NoAnchor 14](#_Toc523059498)

[5.9 ThroughBridge 14](#_Toc523059499)

[5.10 InsufficientDraft 14](#_Toc523059500)

[5.11 AnchorInformation 14](#_Toc523059501)

[5.12 GuideToPortEntry 15](#_Toc523059502)

[5.13 CloseToDanger 15](#_Toc523059503)

[6 Service Interface Specifications 16](#_Toc523059504)

[6.1 SetMessage Interface 16](#_Toc523059505)

[6.1.1 Operation setOptions() 16](#_Toc523059506)

[6.1.2 Operation swtich() 16](#_Toc523059507)

[6.2 InstantMessaging Interface 16](#_Toc523059508)

[6.2.1 Operation instantMessaging () 16](#_Toc523059509)

[7 Service Dynamic Behaviour 18](#_Toc523059510)

[8 References 19](#_Toc523059511)

[9 Acronyms and Terminology 20](#_Toc523059512)

[9.1 Acronyms 20](#_Toc523059513)

[9.2 Terminology 20](#_Toc523059514)

# Introduction

* 1. Purpose of the Document

The purpose of this service specification document is to provide a holistic overview of the Navigational Assistance Intsant Messaging Services (NAIMS) and its building blocks in a technology-independent way, according to the guidelines. It describes a well-defined baseline of the service by clearly identifying the service version.

The aim is to document the key aspects of the NAIMS at the logical level:

* the operational and business context of the service
  + requirements for the service (e.g., information exchange requirements)
  + involved nodes: which operational components provide/consume the service
  + operational activities supported by the service
  + relation of the service to other services
* the service description
  + service interface definitions
  + service interface operations
  + service payload definition
  + service dynamic behaviour description
* service provision and validation aspects
  1. Intended Readership

This service specification is intended to be read by service architects, system engineers and developers in charge of designing and developing an instance of the NAIMS service.

Furthermore, this service specification is intended to be read by enterprise architects, service architects, information architects, system engineers and developers in pursuing architecting, design and development activities of other related services.

* 1. Inputs from Other Projects

N/A

1. **Service Identification**

The purpose of this chapter is to provide a unique identification of the service and describe where the service is in terms of the engineering lifecycle.

|  |  |
| --- | --- |
| Name | * Navigational Assistance Intsant Messaging Services |
| ID | * urn:mrn:msa:enav:service:specification:NAIMS |
| Version | * 0.1 |
| Description | * This service provides customised and Navigational Assistance Intsant Messaging Services |
| Keywords | * Navigational Assistance Intsant Messaging Services, NAIMS, Instant-Messaging, instant service, specific services, guide to port, danger |
| Architect(s) | * PRD e-Navigation project: * CMSA(China Maritime Safety Administration) |
| Status | * Provisional |

1. **Operational Context**

This section describes the context of the service from an operational perspective.

The main purpose of NAIMS (Navigational Assistance Instant Messaging Services) is to provide instant messaging for navigational assistance to vessels with both safety and efficiency considered.

The service takes the mission of the voyage including starting and arrival point, provides Sailing Direction/Collision Avoidance/Bridge Operation Support information etc. according to the ship users’ status. The service considers weather forecast (such as wave, swell, current etc.), the characteristics of the vessel (such as propulsion, manoeuvring, stability), the navigation status of the vessel and maritime traffics to send the assistant message instantly to a certain ship.

The NAIMS has a main use case is for those ships which want to obtain the guidance of instant navigation instruction under their own specific navigation status (ship characteristics, current speed, current position, etc.), so as to reduce the risk of navigation in unfamiliar environment and improve navigation efficiency. For example, a ship sailing to the main channel of Guangzhou Port will be provided with MSI, traffic flow information and Sailing Directions, such as speed/ course/ isolated dangers and collision avoidance suggestions.

* 1. Functional and Non-functional Requirements

The table below defines requirements for the Navigational Assistance Intsant Messaging Services.

Table 2: Requirements Definition

|  |  |
| --- | --- |
| Requirement Id | urn:mrn:msa:enav:service:requirement:NAIMS:1 |
| * Requirement Name | * Sailing Directions Information |
| * Requirement Text | * A Sailing Direction information is provided instantly for the ships sailing in the certain area |
| * Rationale |  |
| * Author |  |

|  |  |
| --- | --- |
| Requirement Id | urn:mrn:msa:enav:service:requirement:NAIMS:2 |
| * Requirement Name | * Collision Avoidance |
| * Requirement Text | * An anti-collision aids information is provided instantly according to the navigation situation |
| * Rationale |  |
| * Author |  |

|  |  |
| --- | --- |
| Requirement Id | urn:mrn:msa:enav:service:requirement:NAIMS:3 |
| * Requirement Name | * Safety Sailing Reminder |
| * Requirement Text | * A ship sailing status will be benchmarked with MSI/Sailing Direction etc. to provide a safety sailing suggestion for navigators. |
| * Rationale |  |
| * Author |  |

|  |  |
| --- | --- |
| Requirement Id | urn:mrn:msa:enav:service:requirement:NAIMS:4 |
| * Requirement Name | * Maritime Domain Awareness |
| * Requirement Text | * Provide harmonized presentation of domain awareness to improve situational awareness for allied and other support services |
| * Rationale |  |
| * Author |  |

|  |  |
| --- | --- |
| Requirement Id | urn:mrn:msa:enav:service:requirement:NAIMS:5 |
| * Requirement Name | * Real-time and Integrated Navigation Display. |
| * Requirement Text | * MSI and other navigational warnings/broadcast, etc. to be received in real-time mode and be integrated or in conjunction with the navigation display. |
| * Rationale |  |
| * Author |  |

|  |  |
| --- | --- |
| Requirement Id | urn:mrn:msa:enav:service:requirement:NAIMS:6 |
| * Requirement Name | * Berthing Requirements |
| * Requirement Text | * Provide information about special berthing requirements on navigation systems especially for pilot. |
| * Rationale |  |
| * Author |  |

|  |  |
| --- | --- |
| Requirement Id | urn:mrn:msa:enav:service:requirement:NAIMS:7 |
| * Requirement Name | * Bridge Operation Support |
| * Requirement Text | * Provide information in real-time with possible presentation on the navigational display to support bridge operation |
| * Rationale |  |
| * Author |  |

* 1. Other Constraints
     1. **Relevant Industrial Standards**

N/A

* + 1. **Operational Nodes**

**Table 3: Operational Nodes providing the NAIMS service**

|  |  |
| --- | --- |
| Operational Node | **Remarks** |
| * Shore-based Centre | * The Shore-based Centre obtains vessel dynamic status and provide special navigation assistant message instantly to a certain vessel. |

**Table 4: Operational Nodes consuming the NAIMS service**

|  |  |
| --- | --- |
| Operational Node | **Remarks** |
| * Ships/Websites/APPs (on-board device) | * The on-board devices receive navigation assistant message instantly which sent by Shore-based Centre. |

* + 1. **Operational Activities**

N/A

1. **Service Overview**

The NAIMS is a type of location-based and environment-aware service. Ships can subscribe the assistant information depending on their own requirements, such as speed limited area, insufficient draft and etc. Then, when a ship drive into a speed limited area or an area of which the depth is not sufficient for the ship, the information would be sent to the ship instantly.

* 1. Service Interfaces

The Instant Assistant Navigation Service consists of two service interfaces and exposes three operations, as the following figure and table show.

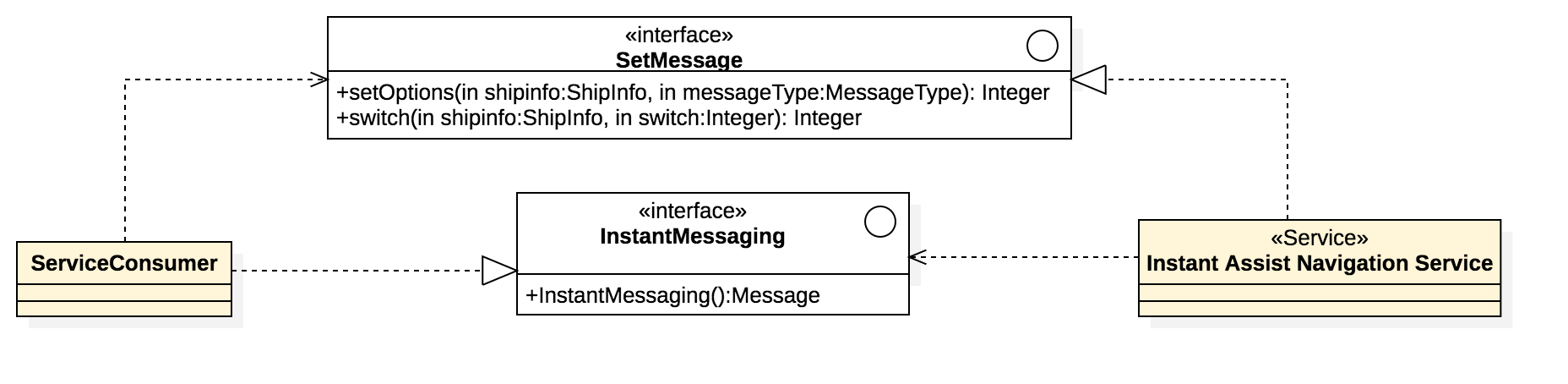


Figure 4.1 NAIMS Interface Definition diagram

Table 4.1 NAIMS Service Interfaces

|  |  |  |
| --- | --- | --- |
| **ServiceInterface** | **Role (from service provider point of view)** | **ServiceOperation** |
| SetMessage | Required | setOptions  switch |
| InstantMessaging | Provided | instantMessaging |

1. **Service Data Model**

This section describes the logical data structures to be exchanged between providers and consumers of the service.

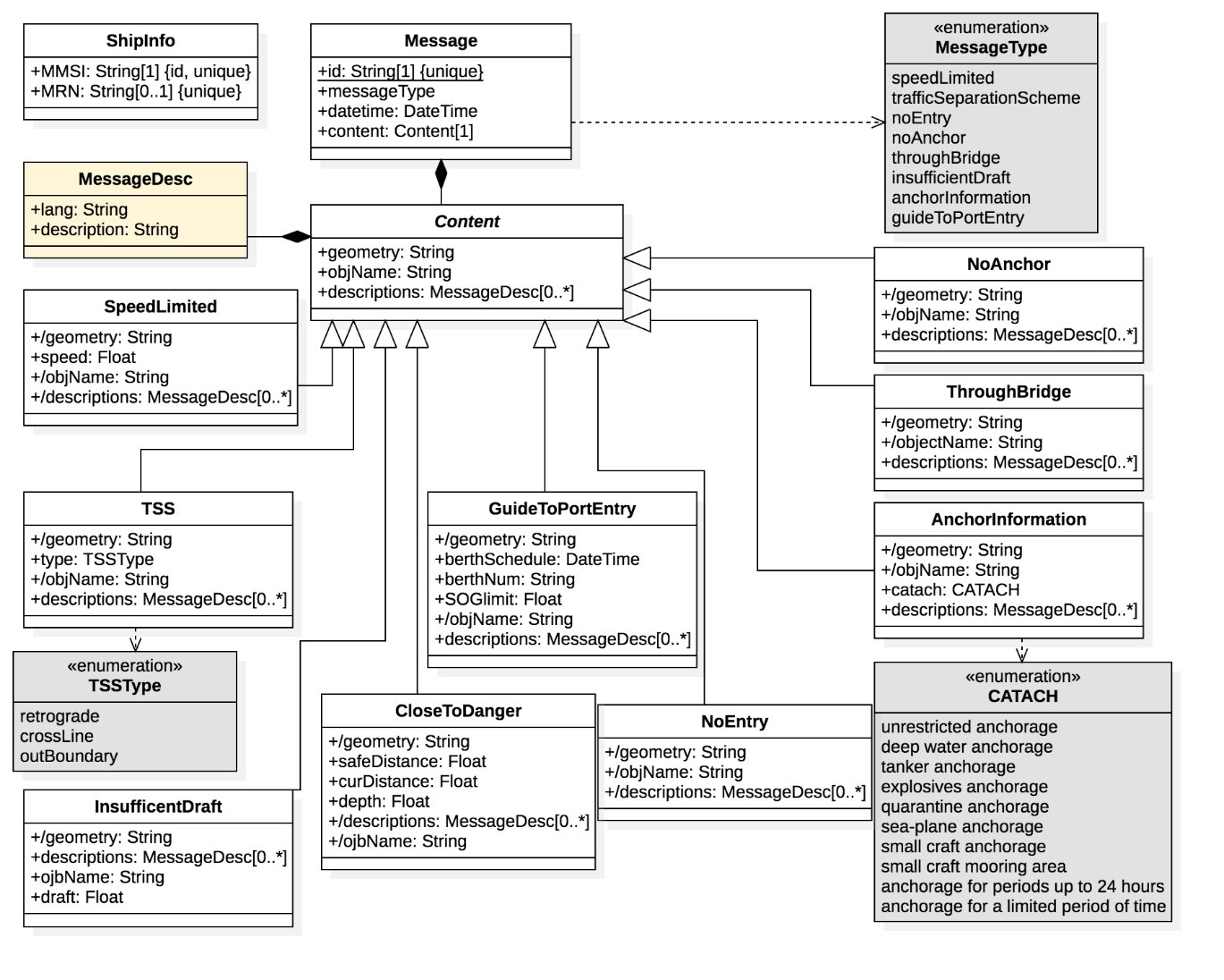


Figure 5.1 Service Data Model diagram

The diagram uses the following color codes:

* Light-white: Used for classes.
* Light-gray background: Used for enumerations.
* Light-yellow background: Used for localized description entities - see Design Pattern section below.

The remainder of the chapter will detail the individual classes.

* 1. Message

Message contains the specific content of the instant message sent to users by the service provider, including message ID, message type, timestamp and description, as the following table shows. The description varies depending on the type of the message. The MessageType is an enumeration type, where 9 items were defined in this specification, details as follows:

|  |  |  |
| --- | --- | --- |
| Attribute Name | Type | Description |
| id | String | Internal system ID of the message |
| messageType | MessageType | The type of the message. One of :   * SpeedLimited * TrafficSeparationScheme * NoEntry * NoAnchor * ThroughBridge * InsufficientDraft * AnchorInformation * GuideToPortEntry * CloseToDanger   See Figure 5.1. |
| dateTime | DateTime | The timestamp the message was sent to users. |
| content | Content | The specific content of the message. The content varies according to the MessageType. See 5.4. |

* 1. MessageDesc

The MessageDesc class contains the list of localizable attributes for a Message.

|  |  |  |
| --- | --- | --- |
| Attribute Name | Type | Description |
| lang | String | The ISO 639-1 language code. |
| description | String | The localized name of a reference. |

* 1. ShipInfo

The Shipinfo class contains the identification information of a ship. The class is also a parameter used for subscribing services. Service providers should get all the static and dynamic information of the ship depending on the MMSI or MRN.

|  |  |  |
| --- | --- | --- |
| Attribute Name | Type | Description |
| MMSI | String | The MMSI of the ship. |
| MRN | String | The MRN of the ship. MRN refers to Maritime Resource Name. This attribute is optional. |

* 1. Content

The specific content of the message. The content varies according to the MessageType. Content is a parent class and has 8 subclasses (See 5.5-5.12). Class which inherits from Content should realize the two attributes defined in the Content class.

|  |  |  |
| --- | --- | --- |
| Attribute Name | Type | Description |
| geometry | String | The shape of the object in the ENC. It should be Well-known text (WKT) format, one of:   * Point * Line * Surface   Format example:   * POINT(6 10) * LINESTRING(3 4,10 50,20 25) * POLYGON((1 1,5 1,5 5,1 5,1 1),(2 2,2 3,3 3,3 2,2 2)) |
| objName | String | The name of the object in the ENC. |
| descriptions | MessageDesc[] | The list of localized message descriptions. A message may have more than 1 descriptions. See 5.2. |

* 1. SpeedLimited

This class contains the information about the max speed in the area. If the ship exceeds the limited speed, an instant message would be sent to the ship.

|  |  |  |
| --- | --- | --- |
| Attribute Name | Type | Description |
| geometry | String | The shape of the area in the ENC. See 5.4. |
| objName | String | The name of the area in the ENC. |
| speed | Float | The max speed allowed in the area. |
| descriptions | MessageDesc[] | The list of localized message descriptions. A message may have more than 1 descriptions. See 5.2. |

* 1. TSS

This class contains the information about the traffic segregate scheme in the area. If the ship retrogrades or exceeds the boundary of the TSS part, an instant message would be sent to the ship.

|  |  |  |
| --- | --- | --- |
| Attribute Name | Type | Description |
| geometry | String | The shape of the TSS in the ENC. See 5.4. |
| objName | String | The name of the TSS in the ENC. |
| type | TSSType | One of:   * retrograde * crossline * outBoundary |
| descriptions | MessageDesc[] | The list of localized message descriptions. A message may have more than 1 descriptions. See 5.2. |

* 1. NoEntry

This class contains the restricted area information. If the ship approaches the area, an instant message would be sent to the ship.

|  |  |  |
| --- | --- | --- |
| Attribute Name | Type | Description |
| geometry | String | The shape of the area in the ENC. See 5.4. |
| objName | String | The name of the area in the ENC. |
| descriptions | MessageDesc[] | The list of localized message descriptions. A message may have more than 1 descriptions. See 5.2. |

* 1. NoAnchor

This class contains the information about an area where anchorage is no allowed. If the speed of the ship stays 0 n/s for a period of time in this area, an instant message would be sent to the ship.

|  |  |  |
| --- | --- | --- |
| Attribute Name | Type | Description |
| geometry | String | The shape of the anchor area in the ENC. See 5.4. |
| objName | String | The name of the area in the ENC. |
| descriptions | MessageDesc[] | The list of localized message descriptions. A message may have more than 1 descriptions. See 5.2. |

* 1. ThroughBridge

This class contains the information about the bridge that the ship would get through later.

|  |  |  |
| --- | --- | --- |
| Attribute Name | Type | Description |
| geometry | String | The shape of the bridge in the ENC. See 5.4. |
| objName | String | The name of the bridge in the ENC. |
| descriptions | MessageDesc[] | The list of localized message descriptions. A message may have more than 1 descriptions. See 5.2. |

* 1. InsufficientDraft

This class contains the draft information for the ship. If the ship drives into an area of which the depth is not sufficient for the ship, an instant message would be sent to the ship. The safe draft of the ship may be set by users depending on their own requirements.

|  |  |  |
| --- | --- | --- |
| Attribute Name | Type | Description |
| geometry | String | The shape of the area in the ENC. See 5.4. |
| objName | String | The name of the area in the ENC. Null allowed. |
| descriptions | MessageDesc[] | The list of localized message descriptions. A message may have more than 1 descriptions. See 5.2. |
| draft | Float | Current draft. |

* 1. AnchorInformation

This class contains the anchor information. If the ship approaches an anchor area, ,an instant message would be sent to the ship.

|  |  |  |
| --- | --- | --- |
| Attribute Name | Type | Description |
| geometry | String | The shape of the area in the ENC. See 5.4. |
| geoName | String | The name of the area in the ENC. |
| descriptions | MessageDesc[] | The list of localized message descriptions. A message may have more than 1 descriptions. See 5.2. |
| catach | CATACH | The category of the anchor. This attribute comes from the ENC Object Catalogue. One of:   * unrestricted anchorage * deep water anchorage * tanker anchorage * explosives anchorage * quarantine anchorage * sea-plane anchorage * small craft anchorage * small craft mooring area * anchorage for periods up to 24 hours * anchorage for a limited period of time |

* 1. GuideToPortEntry

This class contains the information about the time and speed specified by the port manager. When the ship approaches the port (by voyage plan monitoring), an instant message would be sent to the ship to guide the ship enter into the port.

|  |  |  |
| --- | --- | --- |
| Attribute Name | Type | Description |
| geometry | String | The shape of the port in the ENC. See 5.4. |
| geoName | String | The name of the port in the ENC. |
| descriptions | MessageDesc[] | The list of localized message descriptions. A message may have more than 1 descriptions. See 5.2. |
| berthSchedule | DateTime | Information about a berth’s schedule, that is the time when the ship shall drive to the port. |
| berthNum | String | Number of the berth. |
| SOGlimit | Float | An over ground speed limitation for ship users |

* 1. CloseToDanger

This class contains the distance and depth about the danger point that the ship is approaching. If the distance from the danger point to the ship is short than the value set by the users, an instant message would be sent to the ship.

|  |  |  |
| --- | --- | --- |
| Attribute Name | Type | Description |
| geometry | String | The shape of the danger points in the ENC. See 5.4. |
| geoName | String | The name of the object in the ENC. Null allowed. |
| descriptions | MessageDesc[] | The list of localized message descriptions. A message may have more than 1 descriptions. See 5.2. |
| safeDistance | Float | The distance that the ship shall keep from the danger point. |
| curDistance | Float | The distance that from the ship to the danger point. |
| depth | Float | The depth of the danger point. |

An XML schema for this data model is included in the formal service specification xml file attached in Appendix A.

1. **Service Interface Specifications**

This chapter describes the details of each service interface. One sub-chapter is provided for each Service Interface.

The Service Interface specification covers only the static design description while the dynamic design (behaviour) is described in chapter 7.

* 1. SetMessage Interface

The interface is used to set the type of information received and whether to receive the information.

* + 1. **Operation setOptions()**

This operation is used to set the services which the ships want to get.

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Data Type** | **Description** |
| shipinfo | ShipInfo | The MMSI and MRN of the ship. |
| messageType | MessageType[] | The types of assist navigation information. This parameter is an array, consist of 1 or more MessageType (see Figure 5.1). |
| lang | String | An optional ISO 639-1 language code.  If specified, only this language variant is returned for localized entity. However, if, say, “en” is requested and an entity only has a “zh” language description entity, then this is returned instead. A client may want to flag this to the end user. |
| **Return** | **Data Type** | **Description** |
| STATUS | Integer | Status Success or Failure. 1 = true, 0 = false. |

* + 1. **Operation swtich()**

This operation is used to set whether to receive the service.

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Data Type** | **Description** |
| ShipInfo | ShipInfor | The MMSI and MRN of the ship. |
| Switch | Integer | 0 = off , 1 = on |
| **Return** | **Data Type** | **Description** |
| STATUS | Integer | Status Success or Failure. 1 = true, 0 = false. |

* 1. InstantMessaging Interface
     1. **Operation instantMessaging ()**

This operation is automatic triggered when the ships meet the conditions set in the MessageType.

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Direction** | **Data Type** | **Description** |
| Message | Return | Message | The message that the ship get. See 5.1. |

1. **Service Dynamic Behaviour**
   1. Service Interface <SetMessage>

The service operations sequence diagram for setOptions and switch is shown below.



Figure 7.1 < SetMessage > setOptions Operation Sequence Diagram



Figure 7.2 < SetMessage > switch Operation Sequence Diagram

* 1. Service Interface <InstantMessaging>

The service operations sequence diagram for instantMessaging is shown below.

This service operation works automatically.

Service provider should keep the subscription that the consumers have made and monitor the position of ships in real-time.



Figure 7.3 < InstantMessaging > InstantMessaging Operation Sequence Diagram

1. **References**

| **Nr.** | **Version** | **Reference** |
| --- | --- | --- |
| 1. Service Documentation Guidelines | 01.00 | SG\_Annex\_A\_Service\_Documentation\_Guidelines |
| 1. Maritime Resource Name |  | Maritime Resource Name, ENAV17-n.n.n |
| 1. S-100 Universal Hydrographic Data Model | 4.0.0 | S-100 –  UNIVERSAL HYDROGRAPHIC DATA MODEL |

1. **Acronyms and Terminology**
   1. Acronyms

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **API** | Application Programming Interface |
| **MC** | Maritime Cloud |
| **MEP** | Message Exchange Pattern |
| **MRN** | Maritime Resource Name |
| **NAF** | NATO Architectural Framework |
| **REST** | Representational State Transfer |
| **SOA** | Service Oriented Architecture |
| **SOAP** | Simple Object Access Protocol |
| **SSD** | Service Specification Document |
| **UML** | Unified Modelling Language |
| **URL** | Uniform Resource Locator |
| **VTS** | Vessel Traffic Service |
| **WSDL** | Web Service Definition Language |
| **XML** | Extendible Mark-up Language |
| **XSD** | XML Schema Definition |

* 1. Terminology

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **External Data Model** | Describes the semantics of the “maritime world” (or a significant part thereof) by defining data structures and their relations. This could be at logical level (e.g., in UML) or at physical level (e.g., in XSD schema definitions), as for example standard data models, or S-100 based data produce specifications. |
| **Message Exchange Pattern** | Describes the principles how two different parts of a message passing system (in our case: the service provider and the service consumer) interact and communicate with each other. Examples:  In the Request/Response MEP, the service consumer sends a request to the service provider in order to obtain certain information; the service provider provides the requested information in a dedicated response.  In the Publish/Subscribe MEP, the service consumer establishes a subscription with the service provider in order to obtain certain information; the service provider publishes information (either in regular intervals or upon change) to all subscribed service consumers. |
| **Operational Activity** | An activity performed by an operational node. Examples of operational activities in the maritime context are: Route Planning, Route Optimization, Logistics, Safety, Weather Forecast Provision, … |
| **Operational Model** | A structure of operational nodes and associated operational activities and their inter-relations in a process model. |
| **Operational Node** | A logical entity that performs activities. Note: nodes are specified independently of any physical realisation.  Examples of operational nodes in the maritime context are: Maritime Control Center, Maritime Authority, Ship, Port, Weather Information Provider, … |
| **Service** | The provision of something (a non-physical object), by one, for the use of one or more others, regulated by formal definitions and mutual agreements. Services involve interactions between providers and consumers, which may be performed in a digital form (data exchanges) or through voice communication or written processes and procedures. |
| **Service Consumer** | A service consumer uses service instances provided by service providers. All users within the maritime domain can be service customers, e.g., ships and their crew, authorities, VTS stations, organizations (e.g., meteorological), commercial service providers, etc. |
| **Service Data Model** | Formal description of one dedicated service at logical level. The service data model is part of the service specification. Is typically defined in UML and/or XSD. If an external data model exists (e.g., a standard data model), then the service data model shall refer to it: each data item of the service data model shall be mapped to a data item defined in the external data model. |
| **Service Design Description** | Documents the details of a service technical design (most likely documented by the service implementer). The service design description includes (but is not limited to) a service physical data model and describes the used technology, transport mechanism, quality of service, etc. |
| **Service Implementation** | The provider side implementation of a dedicated service technical design (i.e., implementation of a dedicated service in a dedicated technology). |
| **Service Implementer** | Implementers of services from the service provider side and/or the service consumer side. Anybody can be a service implementer but mainly this will be commercial companies implementing solutions for shore and ship. |
| **Service Instance** | One service implementation may be deployed at several places by same or different service providers; each such deployment represents a different service instance, being accessible via different URLs. |
| **Service Instance Description** | Documents the details of a service implementation (most likely documented by the service implementer) and deployment (most likely documented by the service provider). The service instance description includes (but is not limited to) service technical design reference, service provider reference, service access information, service coverage information, etc. |
| **Service Interface** | The communication mechanism of the service, i.e., interaction mechanism between service provider and service consumer. A service interface is characterised by a message exchange pattern and consists of service operations that are either allocated to the provider or the consumer of the service. |
| **Service Operation** | Functions or procedure which enables programmatic communication with a service via a service interface. |
| **Service Physical Data Model** | Describes the realisation of a dedicated service data model in a dedicated technology. This includes a detailed description of the data payload to be exchanged using the chosen technology. The actual format of the service physical data model depends on the chosen technology. Examples may be WSDL and XSD files (e.g., for SOAP services) or swagger (Open API) specifications (e.g., for REST services). If an external data model exists (e.g., a standard data model), then the service physical data model shall refer to it: each data item of the service physical data model shall be mapped to a data item defined in the external data model.  In order to prove correct implementation of the service specification, there shall exist a mapping between the service physical data model and the service data model. This means, each data item used in the service physical data model shall be mapped to a corresponding data item of the service data model. (In case of existing mappings to a common external (standard) data model from both the service data model and the service physical data model, such a mapping is implicitly given.) |
| **Service Provider** | A service provider provides instances of services according to a service specification and service instance description. All users within the maritime domain can be service providers, e.g., authorities, VTS stations, organizations (e.g., meteorological), commercial service providers, etc. |
| **Service Specification** | Describes one dedicated service at logical level. The Service Specification is technology-agnostic. The Service Specification includes (but is not limited to) a description of the Service Interfaces and Service Operations with their data payload. The data payload description may be formally defined by a Service Data Model. |
| **Service Specification Producer** | Producers of service specifications in accordance with the service documentation guidelines. |
| **Service Technical Design** | The technical design of a dedicated service in a dedicated technology. One service specification may result in several technical service designs, realising the service with different or same technologies. |
| **Service Technology Catalogue** | List and specifications of allowed technologies for service implementations. Currently, SOAP and REST are envisaged to be allowed service technologies. The service technology catalogue shall describe in detail the allowed service profiles, e.g., by listing communication standards, security standards, stacks, bindings, etc. |
| **Spatial Exclusiveness** | A service specification is characterised as “spatially exclusive”, if in any geographical region just one service instance of that specification is allowed to be registered per technology.  The decision, which service instance (out of a number of available spatially exclusive services) shall be registered for a certain geographical region, is a governance issue. |

**Appendix A XML schema for the Data Model**

<?xml version="1.0" encoding="UTF-8"?>

<serviceSpecification

xmlns="http://msp.msa.com/v1/ServiceSpecificationSchema.xsd"

xmlns:ServiceSpecificationSchema="http://msp.msa.com//v1/ServiceSpecificationSchema.xsd"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://msp.msa.com//v1/ServiceSpecificationSchema.xsd ServiceSpecificationSchema.xsd "

xmlns:xs="http://www.w3.org/2001/XMLSchema" >

<name>Navigational Assistance Instant Messaging Service - MAIMS</name>

<status>provisional</status>

<id>urn:mrn:msa:enav:service:specification:NAIMS</id>

<version>0.1</version>

<description>This service provides customised and instantly assistant navigation services.</description>

<keywords>Navigational Assistance Intsant Messaging Services, NAIMS, Instant-Messaging, instant service, specific services, guide to port, danger</keywords>

<isSpatialExclusive>false</isSpatialExclusive>

<authorInfos>

<authorInfo>

<id>urn:mrn: msa:ngscs</id>

<name>Yi Yang</name>

<description>Responsible for the NAIMS service</description>

<contactInfo>croon\_yy@qq.com</contactInfo>

</authorInfo>

</authorInfos>

<requirements>

<requirement>

<id>urn:mrn:msa:enav:service:requirement:NAIMS:1</id>

<name>Sailing Directions Information</name>

<text>A Sailing Direction information is provided instantly for the ships sailing in the certain area.</text>

</requirement>

<requirement>

<id>urn:mrn:msa:enav:service:requirement:NAIMS:2</id>

<name>Collision Avoidance</name>

<text>An anti-collision aids information is provided instantly according to the navigation situation.</text>

</requirement>

<requirement>

<id>urn:mrn:msa:enav:service:requirement:NAIMS:3</id>

<name>Safety Sailing Reminder</name>

<text>A ship sailing status will be benchmarked with MSI/Sailing Direction etc. to provide a safety sailing suggestion for navigators. </text>

</requirement>

<requirement>

<id> urn:mrn:msa:enav:service:requirement:NAIMS:4</id>

<name>Maritime Domain Awareness</name>

<text>Provide harmonized presentation of domain awareness to improve situational awareness for allied and other support services.</text>

</requirement>

<requirement>

<id> urn:mrn:msa:enav:service:requirement:NAIMS:5</id>

<name>Real-time and Integrated Navigation Display</name>

<text>MSI and other navigational warnings/broadcast, etc. to be received in real-time mode and be integrated or in conjunction with the navigation display.</text>

</requirement>

<requirement>

<id> urn:mrn:msa:enav:service:requirement:NAIMS:6</id>

<name>Berthing Requirements</name>

<text>Provide information about special berthing requirements on navigation systems especially for pilot.</text>

</requirement>

<requirement>

<id> urn:mrn:msa:enav:service:requirement:NAIMS:7</id>

<name>Bridge Operation Support</name>

<text>Provide information in real-time with possible presentation on the navigational display to support bridge operation.</text>

</requirement>

</requirements>

<serviceDataModel>

<definitionAsXSD>

<?xml version="1.0" encoding="utf-8"?>

<xs:schema targetNamespace="http://msp.msa.com/XMLSchema.xsd"

elementFormDefault="qualified"

xmlns="http://msp.msa.com/XMLSchema.xsd"

xmlns:mspns="http://msp.msa.com/XMLSchema.xsd"

xmlns:xs="http://www.w3.org/2001/XMLSchema"

>

<!--Entities-->

<xs:element name="Message" type="MessageT"/>

<xs:element name="MessageDesc" type="MessageDescT"/>

<xs:element name="ShipInfo" type="ShipInfoT"/>

<xs:element name="Content" type="contentT"/>

<xs:element name="SpeedLimited" type="SpeedLimitedT"/>

<xs:element name="TSS" type="TSST"/>

<xs:element name="NoEntry" type="NoEntryT"/>

<xs:element name="NoAnchor" type="NoAnchorT"/>

<xs:element name="ThroughBridge" type="ThroughBridgeT"/>

<xs:element name="InsufficientDraft" type="InsufficientDraftT"/>

<xs:element name="AnchorInformation" type="AnchorInformationT"/>

<xs:element name="GuideToPortEntry" type="GuideToPortEntryT"/>

<xs:element name="CloseToDanger" type="CloseToDangerT"/>

<!--Entities Type-->

<xs:complexType name="MessageT">

<xs:sequence>

<xs:element name="id" minOccurs="1" maxOccurs="1" type="xs:string" nillable="false" />

<xs:element name="messageType" minOccurs="1" maxOccurs="1" type="messageTypeT" nillable="false" />

<xs:element name="dateTime" type="xs:dateTime" nillable="false" />

<xs:element name="content" minOccurs="1" maxOccurs="1" type="contentT" nillable="false" />

</xs:sequence>

</xs:complexType>

<xs:simpleType name="messageTypeT">

<xs:restriction base="xs:string">

<xs:enumeration value="SpeedLimited" />

<xs:enumeration value="TrafficSeparationScheme" />

<xs:enumeration value="NoEntry" />

<xs:enumeration value="NoAnchor" />

<xs:enumeration value="ThroughBridge" />

<xs:enumeration value="InsufficientDraft" />

<xs:enumeration value="AnchorInformation" />

<xs:enumeration value="GuideToPortEntry" />

<xs:enumeration value="CloseToDanger" />

</xs:restriction>

</xs:simpleType>

<xs:complexType name="contentT" abstract="true">

<xs:sequence>

<!--he shape of the object in the ENC. It should be Well-known text (WKT) format, one of:

• Point

• Line

• Surface

Format example:

• POINT(6 10)

• LINESTRING(3 4,10 50,20 25)

• POLYGON((1 1,5 1,5 5,1 5,1 1),(2 2,2 3,3 3,3 2,2 2))-->

<xs:element name="geomery" type="xs:string" />

<xs:element name="objName" type="xs:string" />

<xs:element name="description" type="MessageDescT" maxOccurs="unbounded" />

</xs:sequence>

</xs:complexType>

<xs:complexType name="MessageDescT" abstract="true">

<xs:sequence>

<xs:element name="lang" type="xs:string" />

<xs:element name="description" type="xs:string" />

</xs:sequence>

</xs:complexType>

<xs:complexType name="ShipInfoT">

<xs:sequence>

<xs:element name="MMSI" type="xs:string" minOccurs="1" maxOccurs="1" />

<xs:element name="MRN" type="xs:string" minOccurs="0" maxOccurs="1" />

</xs:sequence>

</xs:complexType>

<xs:complexType name="SpeedLimitedT">

<xs:complexContent>

<xs:extension base="contentT">

<xs:sequence>

<xs:element name="speed" type="xs:float"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

<xs:complexType name="TSST">

<xs:complexContent>

<xs:extension base="contentT">

<xs:sequence>

<xs:element name="type" type="tssTypeT"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

<xs:simpleType name="tssTypeT">

<xs:restriction base="xs:string">

<xs:enumeration value="retrograde" />

<xs:enumeration value="crossLine" />

<xs:enumeration value="outBoundary" />

</xs:restriction>

</xs:simpleType>

<xs:complexType name="NoEntryT">

<xs:complexContent>

<xs:extension base="contentT">

</xs:extension>

</xs:complexContent>

</xs:complexType>

<xs:complexType name="NoAnchorT">

<xs:complexContent>

<xs:extension base="contentT">

</xs:extension>

</xs:complexContent>

</xs:complexType>

<xs:complexType name="ThroughBridgeT">

<xs:complexContent>

<xs:extension base="contentT">

</xs:extension>

</xs:complexContent>

</xs:complexType>

<xs:complexType name="InsufficientDraftT">

<xs:complexContent>

<xs:extension base="contentT">

<xs:sequence>

<xs:element name="draft" type ="xs:float" />

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

<xs:complexType name="AnchorInformationT">

<xs:complexContent>

<xs:extension base="contentT">

<xs:sequence>

<xs:element name="catach" type ="CATACH" />

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

<xs:simpleType name="CATACH">

<xs:restriction base="xs:string">

<xs:enumeration value="unrestricted anchorage" />

<xs:enumeration value="deep water anchorage" />

<xs:enumeration value="tanker anchorage" />

<xs:enumeration value="explosives anchorage" />

<xs:enumeration value="quarantine anchorage" />

<xs:enumeration value="sea-plane anchorage" />

<xs:enumeration value="small craft anchorage" />

<xs:enumeration value="small craft mooring area" />

<xs:enumeration value="anchorage for periods up to 24 hours" />

<xs:enumeration value="anchorage for a limited period of time" />

</xs:restriction>

</xs:simpleType>

<xs:complexType name="GuideToPortEntryT">

<xs:complexContent>

<xs:extension base="contentT">

<xs:sequence>

<xs:element name="berthSchedule" type ="xs:dateTime" />

<xs:element name="berthNum" type ="xs:string" />

<xs:element name="SOGlimit" type ="xs:float" />

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

<xs:complexType name="CloseToDangerT">

<xs:complexContent>

<xs:extension base="contentT">

<xs:sequence>

<xs:element name="safeDistance" type ="xs:float" />

<xs:element name="curDistance" type ="xs:float" />

<xs:element name="depth" type ="xs:float" />

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

</xs:schema>

</definitionAsXSD>

</serviceDataModel>

</serviceSpecification>

ANNEX2

**NAIMS REST/MQTT Service Technical Design v0.1**

Contents

[1 Introduction 4](#_Toc523126211)

[1.1 Purpose of the Document 4](#_Toc523126212)

[1.2 Intended Readership 4](#_Toc523126213)

[1.3 Inputs from Other Projects 4](#_Toc523126214)

[2 Service Design Identification 5](#_Toc523126215)

[3 Technology Introduction 6](#_Toc523126216)

[3.1 REST 6](#_Toc523126217)

[3.2 IM (Instant-Messaging) 6](#_Toc523126218)

[4 Service Design Overview 7](#_Toc523126219)

[4.1 SetMessage Interface 7](#_Toc523126220)

[4.1.1 Operation setOptions() 7](#_Toc523126221)

[4.1.2 Operation swtich() 7](#_Toc523126222)

[4.2 InstantMessaging Interface 8](#_Toc523126223)

[4.2.1 Operation instantMessaging () 8](#_Toc523126224)

[5 Physical Data Model 9](#_Toc523126225)

[5.1 Data Model 9](#_Toc523126226)

[5.1.1 Message 9](#_Toc523126227)

[5.1.2 ShipInfo 10](#_Toc523126228)

[6 Service Interface Design 11](#_Toc523126229)

[6.1 NAIMS setMessage REST interface 11](#_Toc523126230)

[6.1.1 POST v1/setMessage/setOptions 11](#_Toc523126231)

[6.1.2 v1/setMessage/switch 11](#_Toc523126232)

[6.2 NAIMS instantMessaging IM interface 12](#_Toc523126233)

[6.2.1 INSTANT\_MESSAGING 12](#_Toc523126234)

[7 Service Dynamic Behaviour 13](#_Toc523126235)

[8 References 15](#_Toc523126236)

[9 Acronyms and Terminology 16](#_Toc523126237)

[9.1 Acronyms 16](#_Toc523126238)

[9.2 Terminology 16](#_Toc523126239)

[Appendix A Service Design Description XML 19](#_Toc523126240)

# Introduction

* 1. Purpose of the Document

The purpose of this service design description document is to provide a detailed description of the Navigational Assistance Intsant Messaging Services (NAIMS), realized by using the REST and MQTT-IM technology, according to the guidelines. It describes a well-defined baseline of the service design by clearly identifying the service design version.

The aim is to document the key aspects of the service technical design. This includes:

* identification and summary of the service design
  + reference to the service specification
  + identification of the service design
* identification and summary of chosen technology
* detailed description about the realization of each service interface and service operation
  + mapping of interfaces to the chosen technology
  + mapping of operations to the chosen technology
  + mapping of the message exchange patterns to the chosen technology
* detailed description of the physical data model
  + mapping to the service data model of the service specification.
  1. Intended Readership

This service design description document is intended to be read by service architects, designers, system engineers and developers in charge of designing and developing an instance of the *NAIMS.*

Furthermore, this service design description is intended to be read by service architects, information architects, system engineers and developers in pursuing architecting, design and development activities of other related services.

* 1. Inputs from Other Projects

N/A

1. **Service Design Identification**

The purpose of this chapter is to provide a unique identification of the service design and describe where the service is in terms of the engineering lifecycle.

|  |  |
| --- | --- |
| Name | * Service Design Description for NAIMS REST and IM Technology |
| ID | * urn:mrn:msa:enav:service:design:NAIMS:MQTT-IM&REST |
| Version | * 0.1 |
| Technology | * REST and MQTT-IM |
| Service Specification ID | * urn:mrn:msa:enav:service:specification:NAIMS |
| Service Specification Version | * 0.1 |
| Description | * Technology design for NAIMS service. |
| Keywords | * Navigational Assistance Instant Messaging Services, NAIMS, Instant-Messaging, instant service, specific services |
| Architect(s) | * PRD e-Navigation project: * CMSA(China Maritime Safety Administration) |
| Status | * Provisional |

1. **Technology Introduction**
   1. REST

From the Tide Level Information Technical Design:

REST (REpresentational State Transfer) is one way of providing interoperabilitybetween system on the internet. It allows requesting systems to access and manipulate textual representations of web resources using a uniform and predefined set of stateless operations: more than efficiently WSDL and SOAP.

In a web service which using REST, requests made to a resource’s URI will elicit a response that maybe in XML, HTML, JSON or some other defined format. The response may confirm that some alteration has been made to the stored resource, and it may provide hypertext links to other related resources or collections of resources.

Using HTTP, as is most common, the kind of operations available include those predefined by the HTTP verbs GET, POST, PUT, DELETE and so on. By making use of a stateless protocol and standard operations, REST aim for fast performance, reliability, and the ability to grow, by re-using components that can be managed and updated without affecting the system as a whole, even while it is running.

For more details, please refer to <https://en.wikipedia.org/wiki/Representational_state_transfer>

* 1. IM (Instant-Messaging - MQTT)

Instant messaging (IM) technology is a type of online chat that offers real-time text transmission over the Internet. Short messages are typically transmitted between two parties, when each user select "send". Some IM applications can use push technology to provide real-time text, which transmits messages character by character, as they are composed. More advanced instant messaging can add file transfer, clickable hyperlinks, Voice over IP, or video chat.

MQTT is a machine-to-machine (M2M)/"Internet of Things" connectivity protocol. It was designed as an extremely lightweight publish/subscribe messaging transport. It is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium. For example, it has been used in sensors communicating to a broker via satellite link, over occasional dial-up connections with healthcare providers, and in a range of home automation and small device scenarios. It is also ideal for mobile applications because of its small size, low power usage, minimised data packets, and efficient distribution of information to one or many receivers.

1. **Service Design Overview**

This chapter will outline the REST implementation of the services described in the Service

Specification [3].

The Instant Assist Navigation Information Service consists of two service interfaces and exposes three operations, as the following figure and table show.

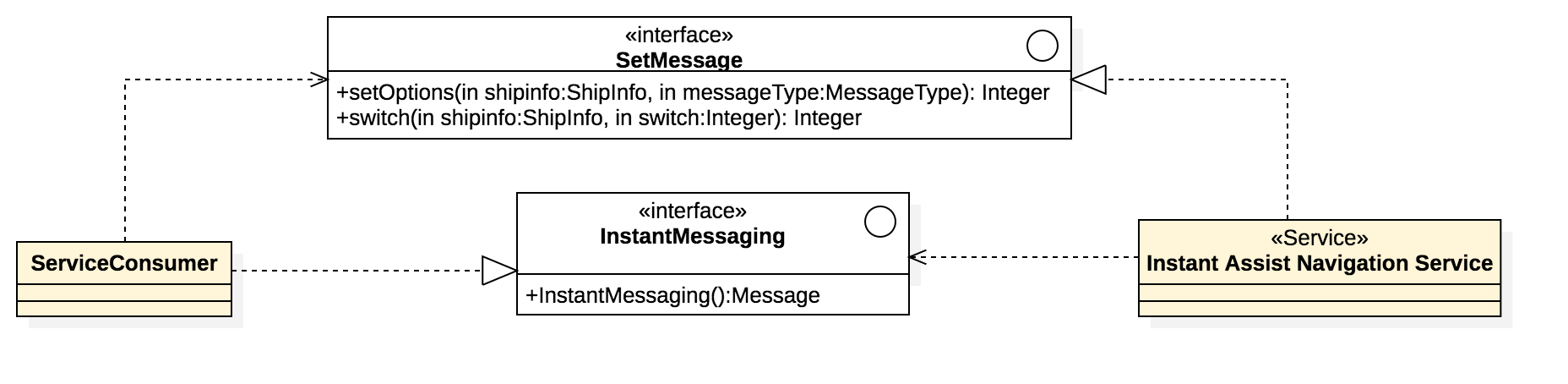


Figure 4.1 IANIS Interface Definition diagram

* 1. SetMessage Interface

The interface is used to set the type of information received and whether to receive the information.

* + 1. **Operation setOptions()**

This operation is used to set the services which the ships want to get.

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Data Type** | **Description** |
| shipinfo | ShipInfo | The MMSI and MRN of the ship. |
| messageType | MessageType[] | The types of assist navigation information. This parameter is an array, consist of 1 or more MessageType. |
| lang | String | An optional ISO 639-1 language code.  If specified, only this language variant is returned for localized entity. However, if, say, “en” is requested and an entity only has a “zh” language description entity, then this is returned instead. A client may want to flag this to the end user. |
| **Return** | **Data Type** | **Description** |
| STATUS | Integer | Status Success or Failure. 1 = true, 0 = false. |

* + 1. **Operation swtich()**

This operation is used to set whether to receive the service.

|  |  |  |
| --- | --- | --- |
| **Parameter Name** | **Data Type** | **Description** |
| ShipInfo | ShipInfor | The MMSI and MRN of the ship. |
| Switch | Integer | 0 = off , 1 = on |
| **Return** | **Data Type** | **Description** |
| STATUS | Integer | Status Success or Failure. 1 = true, 0 = false. |

* 1. InstantMessaging Interface
     1. **Operation instantMessaging ()**

This operation is automatic triggered when the ships meet the conditions set in the MessageType.

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Direction** | **Data Type** | **Description** |
| Message | Return | Message | The message that the ship get. See 5.1. |

The returned data model is detailed in chapter 5 and the service operation in chapter 6.

1. **Physical Data Model**

This chapter details the concrete XML data model implementation of the data model described in the Service Specification [1].

* 1. Data Model

The physical data model is the same as the Service Data Model defined in the Service Specification [1]. This section gives the XML encoding example for each entities in the data model.

* + 1. **Message**

XML Encoding:

<Message>

<id></id>

<messageType></messageType>

<dateTime></dateTime>

<content>There are 9 subclasses inherits the Content class, the content varies depending on the messageType. </content>

</Message>

An example for SpeedLimited:

<Message>

<id> urn:mrnx:prd:service:msa:naims:message:2018082700001</id>

<messageType>speedLimited</messageType>

<dateTime>2018-08-27T09:12:43.083Z</dateTime>

<content>

<geometry> PLOYGON ((113.53820555555555 21.851300000000002, 113.724975 21.841102777777778, 113.6151111111111 21.73398611111111, 113.47778055555555 21.774802777777776, 113.53820555555555 21.851300000000002))</geometry>

<objName>Wuyuanwan Harbour</objName>

<speed>8.0</speed>

<descriptions>

<description>

<lang>en</lang>

<description>This area limits the speed at 8 n/s. </description>

</description>

<description>

<lang>zh</lang>

<description>该区域限速 8 海里/秒. </description>

</description>

</descriptions>

</content>

</Message>

* + 1. **ShipInfo**

XML Encoding:

<shipInfo>

<MMSI></MMSI >

<MRN></MRN >

</ shipInfo >

1. **Service Interface Design**

This chapter describes the details of the NAIMS REST service interface, which is the REST

based implementation of the interfaces specified in the Service Specification [1].

* 1. NAIMS setMessage REST interface

Message exchange pattern: REQUEST\_RESPONSE.

Subscribe the service and switch on/off the service.

* + 1. **POST v1/setMessage/setOptions**

Operation id from specification: setOptions()

Operation for subscribe NAIMS.

Request type POST

Endpoint path: v1/setMessage/setOptions

In Parameters

shipinfo is required, eg. “shipinfo = <shipInfo><MMSI>467895437</MMSI><MRN>urn:mrn:msa:ship:00-1</MRN >

</ shipInfo >”

messageType is required, eg. “messageType = [speedLimited, throughBridge]”

lang is optional, eg. “lang = en”.

Return

http code

http message

XML object with

status = 0 or 1 (1 for success, 0 for fail)

Returns the following HTTP response codes:

200=Successful

400=Bad Request

401=Unauthorized

403=Forbidden

404=Not Found500=Internal Server Error

**Operation functionality**

Subscribe the service. The return parameter “status” indicate whether the operation is successful or unsuccessful.

* + 1. **v1/setMessage/switch**

Operation id from specification: setOptions()

Operation for subscribing NAIMS.

Request type POST

Endpoint path: v1/setMessage/switch

In Parameters

switch isrequired, eg. “switch = 0”

Return

http code

http message

XML object with

status = 0 or 1 (1 for success, 0 for fail)

Returns the following HTTP response codes:

200=Successful

400=Bad Request

401=Unauthorized

403=Forbidden

404=Not Found500=Internal Server Error

**Operation functionality**

Switch on/off the service. The return parameter “status” indicate whether the operation is successful or unsuccessful.

* 1. NAIMS instantMessaging IM interface

Message exchange pattern: instant-Messaging.

Send the assistant navigation message to the ship instantly when the ship meet the conditions set in the service.

* + 1. **MQTT**

All the Message is sent through MQTT protocol.

Operation id from specification: instantMessaging()

Operation for sending instant messages to the users who have subscribed the NAIMS service.

MQTT Topics in:

[SpeedLimited, TrafficSeparationScheme, NoEntry, NoAnchor, ThroughBridge, InsufficientDraft, AnchorInformation, GuideToPortEntry, CloseToDanger];

MQTT listener port:

9001

Return:

XML object with Message (see 5.1.1)

**Operation functionality**

The ship-side system receive the NAIMS service directly when the ship meet the condition in the MessageType (see the Service Specification [1]).

1. **Service Dynamic Behaviour**

The service dynamic behaviour is the same as the behaviour defined in the Service Specification [1]. The technical implement diagram is shown below.



Figure 7.1 setMessage-setOptions sequence diagram



Figure 7.2 setMessage-switch sequence diagram



Figure 7.3 instantMessaging-instantMessaging sequence diagram

1. **References**

| **Nr.** | **Version** | **Reference** |
| --- | --- | --- |
| 1. Service Specification for the Navigational Assistance Intsant Messaging Services | 0.1 |  |

1. **Acronyms and Terminology**
   1. Acronyms

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **API** | Application Programming Interface |
| **MC** | Maritime Cloud |
| **MEP** | Message Exchange Pattern |
| **NAF** | NATO Architectural Framework |
| **REST** | Representational State Transfer |
| **SOAP** | Simple Object Access Protocol |
| **SSD** | Service Specification Document |
| **UML** | Unified Modelling Language |
| **URL** | Uniform Resource Locator |
| **VTS** | Vessel Traffic Service |
| **WSDL** | Web Service Definition Language |
| **XML** | Extendible Mark-up Language |
| **XSD** | XML Schema Definition |

* 1. Terminology

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **External Data Model** | Describes the semantics of the “maritime world” (or a significant part thereof) by defining data structures and their relations. This could be at logical level (e.g., in UML) or at physical level (e.g., in XSD schema definitions), as for example standard data models, or S-100 based data produce specifications. |
| **Message Exchange Pattern** | Describes the principles two different parts of a message passing system (in our case: the service provider and the service consumer) interact and communicate with each other. Examples:  In the Request/Response MEP, the service consumer sends a request to the service provider in order to obtain certain information; the service provider provides the requested information in a dedicated response.  In the Publish/Subscribe MEP, the service consumer establishes a subscription with the service provider in order to obtain certain information; the service provider publishes information (either in regular intervals or upon change) to all subscribed service consumers. |
| **Operational Activity** | An activity performed by an operational node. Examples of operational activities in the maritime context are: Route Planning, Route Optimization, Logistics, Safety, Weather Forecast Provision, … |
| **Operational Model** | A structure of operational nodes and associated operational activities and their inter-relations in a process model. |
| **Operational Node** | A logical entity that performs activities. Note: nodes are specified independently of any physical realisation.  Examples of operational nodes in the maritime context are: Maritime Control Center, Maritime Authority, Ship, Port, Weather Information Provider, … |
| **Service** | The provision of something (a non-physical object), by one, for the use of one or more others, regulated by formal definitions and mutual agreements. Services involve interactions between providers and consumers, which may be performed in a digital form (data exchanges) or through voice communication or written processes and procedures. |
| **Service Consumer** | A service consumer uses service instances provided by service providers. All users within the maritime domain can be service customers, e.g., ships and their crew, authorities, VTS stations, organizations (e.g., meteorological), commercial service providers, etc. |
| **Service Data Model** | Formal description of one dedicated service at logical level. The service data model is part of the service specification. Is typically defined in UML and/or XSD. If an external data model exists (e.g., a standard data model), then the service data model shall refer to it: each data item of the service data model shall be mapped to a data item defined in the external data model. |
| **Service Design Description** | Documents the details of a service technical design (most likely documented by the service implementer). The service design description includes (but is not limited to) a service physical data model and describes the used technology, transport mechanism, quality of service, etc. |
| **Service Implementation** | The provider side implementation of a dedicated service technical design (i.e., implementation of a dedicated service in a dedicated technology). |
| **Service Implementer** | Implementers of services from the service provider side and/or the service consumer side. Anybody can be a service implementer but mainly this will be commercial companies implementing solutions for shore and ship. |
| **Service Instance** | One service implementation may be deployed at several places by same or different service providers; each such deployment represents a different service instance, being accessible via different URLs. |
| **Service Instance Description** | Documents the details of a service implementation (most likely documented by the service implementer) and deployment (most likely documented by the service provider). The service instance description includes (but is not limited to) service technical design reference, service provider reference, service access information, service coverage information, etc. |
| **Service Interface** | The communication mechanism of the service, i.e., interaction mechanism between service provider and service consumer. A service interface is characterised by a message exchange pattern and consists of service operations that are either allocated to the provider or the consumer of the service. |
| **Service Operation** | Functions or procedure which enables programmatic communication with a service via a service interface. |
| **Service Physical Data Model** | Describes the realisation of a dedicated service data model in a dedicated technology. This includes a detailed description of the data payload to be exchanged using the chosen technology. The actual format of the service physical data model depends on the chosen technology. Examples may be WSDL and XSD files (e.g., for SOAP services) or swagger (Open API) specifications (e.g., for REST services). If an external data model exists (e.g., a standard data model), then the service physical data model shall refer to it: each data item of the service physical data model shall be mapped to a data item defined in the external data model.  In order to prove correct implementation of the service specification, there shall exist a mapping between the service physical data model and the service data model. This means, each data item used in the service physical data model shall be mapped to a corresponding data item of the service data model. (In case of existing mappings to a common external (standard) data model from both the service data model and the service physical data model, such a mapping is implicitly given.) |
| **Service Provider** | A service provider provides instances of services according to a service specification and service instance description. All users within the maritime domain can be service providers, e.g., authorities, VTS stations, organizations (e.g., meteorological), commercial service providers, etc. |
| **Service Specification** | Describes one dedicated service at logical level. The Service Specification is technology-agnostic. The Service Specification includes (but is not limited to) a description of the Service Interfaces and Service Operations with their data payload. The data payload description may be formally defined by a Service Data Model. |
| **Service Specification Producer** | Producers of service specifications in accordance with the service documentation guidelines. |
| **Service Technical Design** | The technical design of a dedicated service in a dedicated technology. One service specification may result in several technical service designs, realising the service with different or same technologies. |
| **Service Technology Catalogue** | List and specifications of allowed technologies for service implementations. Currently, SOAP and REST are envisaged to be allowed service technologies. The service technology catalogue shall describe in detail the allowed service profiles, e.g., by listing communication standards, security standards, stacks, bindings, etc. |
| **Spatial Exclusiveness** | A service specification is characterised as “spatially exclusive”, if in any geographical region just one service instance of that specification is allowed to be registered per technology.  The decision, which service instance (out of a number of available spatially exclusive services) shall be registered for a certain geographical region, is a governance issue. |

**Service Design Description XML**

<?xml version="1.0" encoding="UTF-8"?>

<serviceDesign

xmlns:xs="http://www.w3.org/2001/XMLSchema"

xsi:schemaLocation="http://msp.china.com/v1/ServiceDesignSchema.xsd ServiceDesignSchema.xsd "

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:ServiceDesignSchema="http://msp.china.com/maritime-cloud/service-registry/v1/ServiceDesignSchema.xsd"

xmlns:ServiceSpecificationSchema="http://msp.china.com/v1/ServiceSpecificationSchema.xsd"

xmlns="http://msp.china.com/v1/ServiceDesignSchema.xsd">

<id>urn:mrn:prd:service:msa:naims:rest&amp;</id>

<version>0.1</version>

<name>Service Design Description for NAIMS REST and IM Technology</name>

<status>provisional</status>

<description>A REST-based and IM-based implementation of the NAIMS service specification.</description>

<offersTransport>

<offersTransport>

<name>REST</name>

<description>This service implementation is available as REST over HTTPS</description>

<protocol>HTTPS</protocol>

</offersTransport>

<offersTransport>

<name>IM</name>

<description>This service implementation is available as IM over MQTT </description>

<protocol>MQTT</protocol>

</offersTransport>

</offersTransport>

<designsServiceSpecifications>

<designsServiceSpecifications>

<id>urn:mrn:mcl:service:specification:msa:naims</id>

<version>0.1</version>

</designsServiceSpecifications>

</designsServiceSpecifications>

<designedBy>

<ServiceSpecificationSchema:id>urn:mrn:mcl:user:msa:ngscs</ServiceSpecificationSchema:id>

<ServiceSpecificationSchema:name>YI YANG</ServiceSpecificationSchema:name>

<ServiceSpecificationSchema:description>Responsible for the NAIMS service</ServiceSpecificationSchema:description>

<ServiceSpecificationSchema:contactInfo>croon\_yy@qq.com</ServiceSpecificationSchema:contactInfo>

<ServiceSpecificationSchema:isCommercial>false</ServiceSpecificationSchema:isCommercial>

</designedBy>

<servicePhysicalDataModel>

<name>NAIMS PDM</name>

<description>The service physical data model for NAIMS</description>

<modelType>XML</modelType>

<model>

<datamodel>

<definitionAsXSD>

<?xml version="1.0" encoding="utf-8"?>

<xs:schema targetNamespace="http://msp.msa.com/XMLSchema.xsd"

elementFormDefault="qualified"

xmlns="http://msp.msa.com/XMLSchema.xsd"

xmlns:mspns="http://msp.msa.com/XMLSchema.xsd"

xmlns:xs="http://www.w3.org/2001/XMLSchema"

>

<!--Entities-->

<xs:element name="Message" type="MessageT"/>

<xs:element name="MessageDesc" type="MessageDescT"/>

<xs:element name="ShipInfo" type="ShipInfoT"/>

<xs:element name="Content" type="contentT"/>

<xs:element name="SpeedLimited" type="SpeedLimitedT"/>

<xs:element name="TSS" type="TSST"/>

<xs:element name="NoEntry" type="NoEntryT"/>

<xs:element name="NoAnchor" type="NoAnchorT"/>

<xs:element name="ThroughBridge" type="ThroughBridgeT"/>

<xs:element name="InsufficientDraft" type="InsufficientDraftT"/>

<xs:element name="AnchorInformation" type="AnchorInformationT"/>

<xs:element name="GuideToPortEntry" type="GuideToPortEntryT"/>

<xs:element name="CloseToDanger" type="CloseToDangerT"/>

<!--Entities Type-->

<xs:complexType name="MessageT">

<xs:sequence>

<xs:element name="id" minOccurs="1" maxOccurs="1" type="xs:string" nillable="false" />

<xs:element name="messageType" minOccurs="1" maxOccurs="1" type="messageTypeT" nillable="false" />

<xs:element name="dateTime" type="xs:dateTime" nillable="false" />

<xs:element name="content" minOccurs="1" maxOccurs="1" type="contentT" nillable="false" />

</xs:sequence>

</xs:complexType>

<xs:simpleType name="messageTypeT">

<xs:restriction base="xs:string">

<xs:enumeration value="SpeedLimited" />

<xs:enumeration value="TrafficSeparationScheme" />

<xs:enumeration value="NoEntry" />

<xs:enumeration value="NoAnchor" />

<xs:enumeration value="ThroughBridge" />

<xs:enumeration value="InsufficientDraft" />

<xs:enumeration value="AnchorInformation" />

<xs:enumeration value="GuideToPortEntry" />

<xs:enumeration value="CloseToDanger" />

</xs:restriction>

</xs:simpleType>

<xs:complexType name="contentT" abstract="true">

<xs:sequence>

<!--he shape of the object in the ENC. It should be Well-known text (WKT) format, one of:

• Point

• Line

• Surface

Format example:

• POINT(6 10)

• LINESTRING(3 4,10 50,20 25)

• POLYGON((1 1,5 1,5 5,1 5,1 1),(2 2,2 3,3 3,3 2,2 2))-->

<xs:element name="geomery" type="xs:string" />

<xs:element name="objName" type="xs:string" />

<xs:element name="description" type="MessageDescT" maxOccurs="unbounded" />

</xs:sequence>

</xs:complexType>

<xs:complexType name="MessageDescT" abstract="true">

<xs:sequence>

<xs:element name="lang" type="xs:string" />

<xs:element name="description" type="xs:string" />

</xs:sequence>

</xs:complexType>

<xs:complexType name="ShipInfoT">

<xs:sequence>

<xs:element name="MMSI" type="xs:string" minOccurs="1" maxOccurs="1" />

<xs:element name="MRN" type="xs:string" minOccurs="0" maxOccurs="1" />

</xs:sequence>

</xs:complexType>

<xs:complexType name="SpeedLimitedT">

<xs:complexContent>

<xs:extension base="contentT">

<xs:sequence>

<xs:element name="speed" type="xs:float"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

<xs:complexType name="TSST">

<xs:complexContent>

<xs:extension base="contentT">

<xs:sequence>

<xs:element name="type" type="tssTypeT"/>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

<xs:simpleType name="tssTypeT">

<xs:restriction base="xs:string">

<xs:enumeration value="retrograde" />

<xs:enumeration value="crossLine" />

<xs:enumeration value="outBoundary" />

</xs:restriction>

</xs:simpleType>

<xs:complexType name="NoEntryT">

<xs:complexContent>

<xs:extension base="contentT">

</xs:extension>

</xs:complexContent>

</xs:complexType>

<xs:complexType name="NoAnchorT">

<xs:complexContent>

<xs:extension base="contentT">

</xs:extension>

</xs:complexContent>

</xs:complexType>

<xs:complexType name="ThroughBridgeT">

<xs:complexContent>

<xs:extension base="contentT">

</xs:extension>

</xs:complexContent>

</xs:complexType>

<xs:complexType name="InsufficientDraftT">

<xs:complexContent>

<xs:extension base="contentT">

<xs:sequence>

<xs:element name="draft" type ="xs:float" />

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

<xs:complexType name="AnchorInformationT">

<xs:complexContent>

<xs:extension base="contentT">

<xs:sequence>

<xs:element name="catach" type ="CATACH" />

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

<xs:simpleType name="CATACH">

<xs:restriction base="xs:string">

<xs:enumeration value="unrestricted anchorage" />

<xs:enumeration value="deep water anchorage" />

<xs:enumeration value="tanker anchorage" />

<xs:enumeration value="explosives anchorage" />

<xs:enumeration value="quarantine anchorage" />

<xs:enumeration value="sea-plane anchorage" />

<xs:enumeration value="small craft anchorage" />

<xs:enumeration value="small craft mooring area" />

<xs:enumeration value="anchorage for periods up to 24 hours" />

<xs:enumeration value="anchorage for a limited period of time" />

</xs:restriction>

</xs:simpleType>

<xs:complexType name="GuideToPortEntryT">

<xs:complexContent>

<xs:extension base="contentT">

<xs:sequence>

<xs:element name="berthSchedule" type ="xs:dateTime" />

<xs:element name="berthNum" type ="xs:string" />

<xs:element name="SOGlimit" type ="xs:float" />

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

<xs:complexType name="CloseToDangerT">

<xs:complexContent>

<xs:extension base="contentT">

<xs:sequence>

<xs:element name="safeDistance" type ="xs:float" />

<xs:element name="curDistance" type ="xs:float" />

<xs:element name="depth" type ="xs:float" />

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

</xs:schema>

</definitionAsXSD>

<example>

<Message>

<id>urn:mrnx:prd:service:msa:naims:message:2018082700001</id>

<messageType>speedLimited</messageType>

<dateTime>2018-08-27T09:12:43.083Z</dateTime>

<content>

<geometry> PLOYGON ((113.53820555555555 21.851300000000002, 113.724975 21.841102777777778, 113.6151111111111 21.73398611111111, 113.47778055555555 21.774802777777776, 113.53820555555555 21.851300000000002))</geometry>

<objName>Wuyuanwan Harbour</objName>

<speed>8.0</speed>

<descriptions>

<description>

<lang>en</lang>

<description>This area limits the speed at 8 n/s. </description>

</description>

<description>

<lang>zh</lang>

<description>该区域限速8海里/秒. </description>

</description>

</descriptions>

</content>

</Message>

<shipInfo>

<MMSI>576357861</MMSI >

<MRN>urn:mrnx:prd:service:msa:ship:100001</MRN >

</shipInfo>

</example>

</datamodel>

<serviceInterfaces>

<!--setMessage Interface-->

<serviceInterface>

<name>setMessage</name>

<oerations>

<operation>

<name>setOptions</name>

<endpoint>v1/setMessage/setOptions</endpoint>

<inputs>

<parameter>

<name>shipinfo</name>

<datetype>ShipInfo</datetype>

</parameter>

<parameter>

<name>messageType</name>

<datetype>MessageType</datetype>

</parameter>

</inputs>

<outputs>

<parameter>

<name>status</name>

<datetype>Integer</datetype>

</parameter>

</outputs>

</operation>

<operation>

<name>switch</name>

<endpoint>v1/setMessage/switch</endpoint>

<inputs>

<parameter>

<name>shipinfo</name>

<datetype>ShipInfo</datetype>

</parameter>

<parameter>

<name>switch</name>

<datetype>Integer</datetype>

</parameter>

</inputs>

<outputs>

<parameter>

<name>status</name>

<datetype>Integer</datetype>

</parameter>

</outputs>

</operation>

</oerations>

</serviceInterface>

<!--instantMessaging Interface-->

<serviceInterface>

<name>instantMessaging</name>

<oerations>

<operation>

<name>instantMessaging</name>

<endpoint>

<MQTTport>9001</MQTTport>

<Websockport>9002</Websockport>

</endpoint>

<inputs>

<parameter>

<name>shipinfo</name>

<datetype>ShipInfo</datetype>

</parameter>

<parameter>

<name>location</name>

<!--The real-time coordinate of the ship. It should be Well-known text (WKT) format, one of:

• Point

Format example:

• POINT(6 10) -->

<datetype>WKT</datetype>

</parameter>

</inputs>

<outputs>

<parameter>

<name>status</name>

<datetype>Integer</datetype>

</parameter>

</outputs>

</operation>

</oerations>

</serviceInterface>

</serviceInterfaces>

</model>

</servicePhysicalDataModel>

</serviceDesign>

ANNEX3

**NAIMS Service Instance Description v0.1**

Contents

[1 Introduction 3](#_Toc489020511)

[1.1 Purpose of the Document 3](#_Toc489020512)

[1.2 Intended Readership 5](#_Toc489020513)

[2 Service Instance Identification 6](#_Toc489020514)

[3 Service Implementation and Instantiation Details 8](#_Toc489020515)

[4 Release Notes 9](#_Toc489020516)

[5 References 10](#_Toc489020517)

[6 Acronyms and Terminology 11](#_Toc489020518)

[6.1 Acronyms 11](#_Toc489020519)

[6.2 Terminology 11](#_Toc489020520)

# Introduction

* 1. Purpose of the Document

The purpose of this service instance description document is to provide a documentation of the implementation and instantiation of the Navigational Assistance Intsant Messaging Services *(NAIMS)*, realized by using the IP, IM technology, according to the guidelines. It describes a well-defined baseline of the service implementation by clearly identifying the service implementation version.

The aim is to document the key aspects of the NAIMS instantiation. This includes:

* identification and summary of the service instance
  + reference to the service design description
  + reference to the service specification
  + identification of the service instance
* service implementation and instantiation details
  + internal design decisions
  + configuration data
  + deployment information
* release notes
  + feature list
  + bug list.
  1. Intended Readership

This service instance description document is intended to be read by service providers, system engineers and developers in charge of deploying and operating an instance of the *NAIMS* service.

1. **Service Instance Identification**

The purpose of this chapter is to provide a unique identification of the service instance and describe where the service is in terms of the engineering lifecycle.

|  |  |
| --- | --- |
| Name | * REST and IM Instance of Navigational Assistance Intsant Messaging Services |
| ID | * urn:mrn:msa:enav:service:instance:NAIMS:MQTT-IM&REST:1 |
| Version | * 0.1 |
| Technology | * IP, IM |
| Service Specification ID | * urn:mrn:msa:enav:service:specification:NAIMS |
| Service Specification Version | * 0.1 |
| Service Design ID | * urn:mrn:msa:enav:service:design:NAIMS:MQTT-IM&REST |
| Service Design Version | * 0.1 |
| Description | * This service provides customised and instantly navigation assistant services |
| Keywords | * Navigational Assistance Intsant Messaging Services, NAIMS, Instant-Messaging, instant service, specific services |
| Supplier | * PRD e-Navigation project Team * CMSA(China Maritime Safety Administration) |
| Status | * Provisional |

1. **Service Implementation and Instantiation Details**

This chapter will define the absolute URL of NAIMS service in the PRD e-navigation project.

* 1. NAIMS REST Service URL

The NAIMS REST service is located at:

• http:// [enav.ngscs.org/enavprd/services/naims/v1/](http://www.ngscs.org/enav/services/namis/v1/)

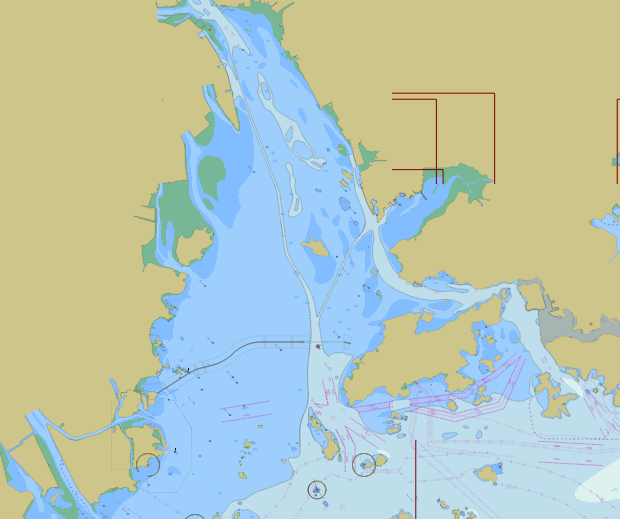
* 1. NAIMS MQTT Service

The url of NAIMS MQTT is:

• <http://enav.ngscs.org:9001>

1. **Coverage Area**

The coverage area of NAIMS service is the Pearl River Delta:



1. **Release Notes**
2. **References**

| **Nr.** | **Version** | **Reference** |
| --- | --- | --- |
| 1. Service Specification for the Navigational Assistance Intsant Messaging Services | 0.1 | Service Specification for the NAIMS service. |
| 1. Service Design Description for NAIMS REST and IM Technology | 0.1 | Service Design Description for the NAIMS service. |

1. **Acronyms and Terminology**
   1. Acronyms

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **API** | Application Programming Interface |
| **MC** | Maritime Cloud |
| **MEP** | Message Exchange Pattern |
| **NAF** | NATO Architectural Framework |
| **REST** | Representational State Transfer |
| **SOAP** | Simple Object Access Protocol |
| **SSD** | Service Specification Document |
| **UML** | Unified Modelling Language |
| **URL** | Uniform Resource Locator |
| **VTS** | Vessel Traffic Service |
| **WSDL** | Web Service Definition Language |
| **XML** | Extendible Mark-up Language |
| **XSD** | XML Schema Definition |

* 1. Terminology

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **External Data Model** | Describes the semantics of the “maritime world” (or a significant part thereof) by defining data structures and their relations. This could be at logical level (e.g., in UML) or at physical level (e.g., in XSD schema definitions), as for example standard data models, or S-100 based data produce specifications. |
| **Message Exchange Pattern** | Describes the principles two different parts of a message passing system (in our case: the service provider and the service consumer) interact and communicate with each other. Examples:  In the Request/Response MEP, the service consumer sends a request to the service provider in order to obtain certain information; the service provider provides the requested information in a dedicated response.  In the Publish/Subscribe MEP, the service consumer establishes a subscription with the service provider in order to obtain certain information; the service provider publishes information (either in regular intervals or upon change) to all subscribed service consumers. |
| **Operational Activity** | An activity performed by an operational node. Examples of operational activities in the maritime context are: Route Planning, Route Optimization, Logistics, Safety, Weather Forecast Provision, … |
| **Operational Model** | A structure of operational nodes and associated operational activities and their inter-relations in a process model. |
| **Operational Node** | A logical entity that performs activities. Note: nodes are specified independently of any physical realisation.  Examples of operational nodes in the maritime context are: Maritime Control Center, Maritime Authority, Ship, Port, Weather Information Provider, … |
| **Service** | The provision of something (a non-physical object), by one, for the use of one or more others, regulated by formal definitions and mutual agreements. Services involve interactions between providers and consumers, which may be performed in a digital form (data exchanges) or through voice communication or written processes and procedures. |
| **Service Consumer** | A service consumer uses service instances provided by service providers. All users within the maritime domain can be service customers, e.g., ships and their crew, authorities, VTS stations, organizations (e.g., meteorological), commercial service providers, etc. |
| **Service Data Model** | Formal description of one dedicated service at logical level. The service data model is part of the service specification. Is typically defined in UML and/or XSD. If an external data model exists (e.g., a standard data model), then the service data model shall refer to it: each data item of the service data model shall be mapped to a data item defined in the external data model. |
| **Service Design Description** | Documents the details of a service technical design (most likely documented by the service implementer). The service design description includes (but is not limited to) a service physical data model and describes the used technology, transport mechanism, quality of service, etc. |
| **Service Implementation** | The provider side implementation of a dedicated service technical design (i.e., implementation of a dedicated service in a dedicated technology). |
| **Service Implementer** | Implementers of services from the service provider side and/or the service consumer side. Anybody can be a service implementer but mainly this will be commercial companies implementing solutions for shore and ship. |
| **Service Instance** | One service implementation may be deployed at several places by same or different service providers; each such deployment represents a different service instance, being accessible via different URLs. |
| **Service Instance Description** | Documents the details of a service implementation (most likely documented by the service implementer) and deployment (most likely documented by the service provider). The service instance description includes (but is not limited to) service technical design reference, service provider reference, service access information, service coverage information, etc. |
| **Service Interface** | The communication mechanism of the service, i.e., interaction mechanism between service provider and service consumer. A service interface is characterised by a message exchange pattern and consists of service operations that are either allocated to the provider or the consumer of the service. |
| **Service Operation** | Functions or procedure which enables programmatic communication with a service via a service interface. |
| **Service Physical Data Model** | Describes the realisation of a dedicated service data model in a dedicated technology. This includes a detailed description of the data payload to be exchanged using the chosen technology. The actual format of the service physical data model depends on the chosen technology. Examples may be WSDL and XSD files (e.g., for SOAP services) or swagger (Open API) specifications (e.g., for REST services). If an external data model exists (e.g., a standard data model), then the service physical data model shall refer to it: each data item of the service physical data model shall be mapped to a data item defined in the external data model.  In order to prove correct implementation of the service specification, there shall exist a mapping between the service physical data model and the service data model. This means, each data item used in the service physical data model shall be mapped to a corresponding data item of the service data model. (In case of existing mappings to a common external (standard) data model from both the service data model and the service physical data model, such a mapping is implicitly given.) |
| **Service Provider** | A service provider provides instances of services according to a service specification and service instance description. All users within the maritime domain can be service providers, e.g., authorities, VTS stations, organizations (e.g., meteorological), commercial service providers, etc. |
| **Service Specification** | Describes one dedicated service at logical level. The Service Specification is technology-agnostic. The Service Specification includes (but is not limited to) a description of the Service Interfaces and Service Operations with their data payload. The data payload description may be formally defined by a Service Data Model. |
| **Service Specification Producer** | Producers of service specifications in accordance with the service documentation guidelines. |
| **Service Technical Design** | The technical design of a dedicated service in a dedicated technology. One service specification may result in several technical service designs, realising the service with different or same technologies. |
| **Service Technology Catalogue** | List and specifications of allowed technologies for service implementations. Currently, SOAP and REST are envisaged to be allowed service technologies. The service technology catalogue shall describe in detail the allowed service profiles, e.g., by listing communication standards, security standards, stacks, bindings, etc. |
| **Spatial Exclusiveness** | A service specification is characterised as “spatially exclusive”, if in any geographical region just one service instance of that specification is allowed to be registered per technology.  The decision, which service instance (out of a number of available spatially exclusive services) shall be registered for a certain geographical region, is a governance issue. |

1. Input document number, to be assigned by the Committee Secretary [↑](#footnote-ref-1)
2. Leave open if uncertain [↑](#footnote-ref-2)