Input paper: [[1]](#footnote-2) ENAV28-5.1.1.3

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-3)

Technical Domain / Task Number 2 …………………………………

Author(s) / Submitter(s) GLA, SMA, OFFIS

Change Proposal for IALA G1128: THE SPECIFICATION OF e-NAVIGATION

TECHNICAL SERVICES

# Summary

IALA G1128 is a guideline that provides information on how to develop specifications of e-Navigation Technical Services. In the past years, efforts have been made to use the guideline and its templates to successfully develop service specifications for maritime services, especially in the context of the Maritime Connectivity Platform. However, while applying the principles and templates as well as the supplied xsd-Schemas some smaller problems, or open points in the guideline could be identified.

## Purpose of the document

This document provides input to a revision of G1128 and proposes corrections to the supplied xsd-Schemas.

# Discussion

The following points relate to the contents of the guideline:

* The guideline includes explicit references to the Maritime Connectivity Platform’s service registry. These should be rephrased or removed, as the guideline should be service registry implementation agnostic. Se section 3.2.1 below.
* The orchestration within a service not described in the guideline.
* Service accepted payloads not explicitly expressed in formal representation for respective documentation level. Hint, this could be included in Service Design ServicePhysicalDataModel and possibly described in the Service instance as well.
* It is not discussed how to deal with the composition of services in the guideline. It could be beneficial to mention service composition at least on an informative level.
* Operational Service Instance descriptions and technical documentation for the service instance, should support more than one document each for a service instance in such a way, that an operator could benefit from a shorter version of the specification.
  + Suggest to expand on definitions of the different actors in chapter 2.1, see proposed change in section 3.2.1 below.
* Extend the support for searching of services based on same technical design, STM have extended the Service Registry, but enumerations is not added in XML Schema. This requires today separate governance and monitoring of these fields. (ServiceType, Keywords)
  + Navelink has adopted the STM guidelines and included these in the monitoring tools for Service Registry. But there still need to be an agreement of the maintenance procedures for e.g. ServiceType enumeration, and other guidelines for keywords, naming of ships and shore centers, service coverage, descriptions etc.
* Guideline for service registration implicates further specifying of how elements in specification, design and instances should be filled out i.e. UnLocode - 5 characters, no space, capital letters or empty (blank).
  + Suggest to further specify elements in section 3.3 Service Specification XSD Structure Table 2 Information elements of the Service Specification, see section 3.3.3 below.
  + Suggest to further specify elements in section 4.3 Service Design Description XSD Structure Table 3 Information Elements of the Service Design Description, see section 3.4.3 below.
  + Suggest to further specify elements in section 5.3 Service Instance Description XSD Structure Table 4 Information Elements of the Service Instance Description, see section 3.5.3 below.

The following points relate to issues/errors in the XSD Schemas provided in Annex of the guideline:

**ServiceSpecificationSchema.xsd**

* The “ValueTypeDataModelMapping” type of the specification contains a parameter element with type “S100Base:S100\_Parameter” referencing the S100 standard. This is a problem however, since the “S100\_Parameter” is not actually part of the “S100Base” XML schema definition.
* The “ServiceDataModel” type of the specification contains an element with name “FeatureCatalogue” and type “S100Base:S100\_FC\_Catalogue”, referencing the “S100FC” XML schema definition. “S100\_FC\_Catalogue” however, is actually an element in that XSD file, not a type, hence the type reference is invalid.

**ServiceInstanceSchema.xsd**

* The defined “ServiceInstance” type contains an element with name “coversAreas” which is defined with the attributes “minOccurs=1” and “maxOccurs=unbounded”. This is not valid however inside an “<all></all>” statement, where “maxOccurs” can only have values up to one (1). To resolve that, we could change the referenced “ServiceInstanceSchema:CoverageInfo” type to have an unbounded number of cover area elements.

# G1128 proposed changes

## Introduction

## Overview

### Service Management Overview

A service management concept can be visualised as shown in Figure 1. Both, service specifications as well as information about service instances can be published in a service registry. A service registry can be a collection of documents, or could be realised as a service itself that would have an Application Programming Interface (API) for automatic interfacing to the registry (lookup, updating, deleting etc.). ~~This concept is implemented as the Maritime Service Registry within the Maritime Connectivity Platform (MCP, formerly called the Maritime Cloud), see~~ [~~http://www.maritimecloud.net~~](http://www.maritimecloud.net)~~.~~

In the following figure 1, four different roles are envisioned in managing the contents of a service registry following this guideline.

* Service specification producers might represent organizations governing a certain type of service type like Navigational Warning Service, Route Optimization Service, Search and Rescue Service etc.
* Service implementers typically develop services in accordance with published specifications resulting in service technical designs. I.e. realizing a service specification by use of a certain technology like REST, SOAP, FTP etc.
* Service providers publishes implemented technical services following a service technical design. Service providers are responsible for publishing service instances ready for consumption in a service registry. Examples of service providers are, Shipping companies, VTS, National Maritime Administrations etc.
* Service consumers discovers service instances for consumption. In this context the service consumer considered is could be an actual person or an application like an ECDIS software or perhaps a VTS software. In discovering service instances a service consumer might need to make use of all three service description artefacts (specification, design and instance) in order to determine the full specification of the published service instance.



Figure 1 Service Management Concept

## SERVICE SPECIFICATION

### Service Specification Document

…

### Service Specification Template

…

### Service Specification XSD Structure

***Table 2 Information elements of the Service Specification***

| Type Name | | Description | |
| --- | --- | --- | --- |
| ServiceSpecification | | A service specification describes one dedicated service at logical level in a technology-independent way. The service specification identifies a service by its id and version. The service specification refers to requirements for the service, defines a service data model at logical level, defines the service interface(s) and provides information about the author(s). | |
|  | Element Name | Type | Description |
| Name | CharacterString | The human readable service name. The service name shall be at maximum a one-line brief label for the service. Newer versions of the same service specification shall not change the name. |
| Id | ServiceIdentifier | Service identifier type to be used by service specifications, designs, instances. Currently, the identifier is defined as a string.  Globally unique identification of the service. Newer versions of the same service specification shall not change the id.  The identifier should conform to an MRN identity. |
| version | ServiceVersion | Service version indicator type to be used by service specifications, designs, instances. Currently, the version indicator is defined as a string.  Version of the service specification. A service specification is uniquely identified by its id and version. Any change in the service data model or in the service interface definition requires a new version of the service specification |
| status | StatusType <<enumeration>> | Status of the service specification. The status field has one of the following values:   * provisional; * released; * deprecated; * deleted. |
| description | CharacterString | A human readable short description of the service. The description shall contain an abstract of what a service implementing this specification would do. |
| keywords | CharacterString | A list of keywords associated with the service. |
| isSpatialExclusive | Boolean | Flag to indicate whether the service shall be ‘spatial exclusive’. ‘Spatial exclusiveness’ means that just one service instance of the same service specification providing the same technical design is allowed to be registered for a certain geographical area. |
| requirements | Requirements | Refers to requirements specifications for the service. Business requirements, functional and non-functional requirements shall be listed here. At least one requirement is mandatory. |
| authorInfos | AuthorInfo | Refers to administrative information about the authors of the service. It is mandatory to provide at least one author information. |
| serviceInterfaces | ServiceInterface | Refers to the definition of service interfaces. At least one service interface shall be defined. |
| serviceDataModel | ServiceDataModel | Mandatory reference to the definition of the logical service data model. |
| Type Name | | Description | |
| Requirements | | A requirement that the service shall fulfil. | |
|  | Element Name | Type | Description |
| id | CharacterString | ~~Globally unique requirement identification~~  Unique identification of the requirement |
| name | CharacterString | Human readable requirement name/summary. Shall not be longer than one line. |
| text | CharacterString | The human readable requirement text. Usually formulated in form of a ‘shall’-statement. |
| rationale | CharacterString | Rationale for this requirement. Textual explanation of why this requirement exists. Provides background information about the need of the service. |
| reference | CharacterString | Optional information about where the requirement was originally stated. If the requirement comes from external documents, this attribute shall refer to this source. |
| author | AuthorInfo | Optional reference(s) to administrative information about the author(s) of the requirement. |
| Type Name | | Description | |
| AuthorInfo | | Describes an author of a service specification or requirement. | |
|  | Element Name | Type | Description |
| id | CharacterString | Unique identifier of the author. |
| name | CharacterString | Human readable name of the author. |
| description | CharacterString | Human readable description of the author. |
| contactInfo | CharacterString | Human readable contact information of the author. |
| organizationId | CharacterString | Unique identifier of the organization, the author belongs to i.e. the MRN identifier of the organization. |
| Type Name | | Description | |
| ServiceInterface | | Specification of a service interface. One service can offer several interfaces, e.g. both a request/response interface and a publish/subscribe interface at the same time. Different interfaces will usually provide different service operations. | |
|  | Element Name | Type | Description |
| name | CharacterString | Human readable service interface name. The name shall be no longer than one line. |
| description | CharacterString | Human readable description of the service interface. |
| dataExchangePattern | ExchangePattern <<enumeration>> | Message exchange pattern for the entire interface.  Can be one of:   * ONE\_WAY   data are sent in one direction, from service consumer to service provider, without confirmation.   * REQUEST\_RESPONSE   service consumer sends request to service provider and expects to receive a response from the service provider.   * REQUEST\_CALLBACK   (asynchronous REQUEST\_RESPONSE) service consumer sends a request to service provider; response is provided asynchronously in an independent call to the service.   * PUBLISH\_SUBSCRIBE   service consumer subscribes at service provider for receiving publications sent out by the service provider.   * BROADCAST   service provider distributes information independently of any consumers. |
| operations | Operation | Refers to the specification of service operations supported by the service interface. At least one operation shall be defined. |
| consumerInterfaces | ConsumerInterface | Optional reference to an interface definition that shall be provided by the service consumer to complement the service interface. Especially if a publish/subscribe service interface is designed, it is necessary to describe what the service expects to be available on the subscriber side. |
| Type Name | | Description | |
| ConsumerInterface | | Interface specification that is expected to be provided by the service consumer. For example, if a publish/subscribe service interface is designed, it is necessary to describe what the service expects to be available on the subscriber side. | |
|  | Element Name | Type | Description |
| name | CharacterString | Human readable interface name. The name shall be no longer than one line. |
| description | CharacterString | Human readable description of the interface. |
| operations | Operation | Refers to the specification of service operations supported by the consumer interface. At least one operation shall be defined. |
| Type Name | | Description | |
| Operation | | Definition of a service operation. Operations allow a service consumer to interact with the service. An operation describes a dedicated function of the service or the consumer. | |
|  | Element Name | Type | Description |
| name | CharacterString | Human readable operation name. The name shall be no longer than one line. |
| description | CharacterString | Human readable description of the operation. |
| returnValueType | ValueTypeData- ModelMapping | Optional definition of the return value for the operation. The return value could be a business object or a simple status code. The return value data type must be defined in the logical service data model. |
| parameterTypes | ValueTypeData- ModelMapping | Definition of one or more parameters for the operation. This could be business objects or simple types. Parameters must be defined in the logical service data model. |
| Type Name | | Description | |
| ValueTypeDataModel-  Mapping | | Definition of a data type by providing a reference into the logical service data model. A value type data model mapping is used either in a service operation parameter or return value. | |
|  | Element Name | Type | Description |
| typeReference | CharacterString | Only used if not S-100 compliant.  Reference to the logical service data model. It references a type (or element, though type is preferred) in the logical service model by the type's name attribute. |
| parameter | S100\_Parameter | Reference to the S-100 attribute. |
| multiplicity | S100\_Multiplicity | Minimum and maximum number of provided instances, where the maximum number may be infinitive. If no multiplicity is provided a multiplicity of 1 is assumed. |
| direction | DirectionKind <<enumeration>> | Determining the direction. Can be one of:   * in * out * inout |
| encoding | CharacterString | Encoding of the parameter |
| Type Name | | Description | |
| ServiceDataModel | | If the service is S-100 compliant it shall link to an existing Feature Catalogue in the featureCatalogue element. The other elements are then not used.  If the service is not S-100 compliant: The serviceDataModel is formally described in the sub-element modelDefinition to achieve interoperability and decouple it from implementing physical data models described in e.g. SOAP or REST. Encodings are formally described using sub-element encoding.  The model can either be described in-line, or existing schemata can be imported.  One service specification has one logical service model. Sub-element modelDefinition has to be provided. | |
|  | Element Name | Type | Description |
| featureCatalogue | S100\_FC\_FeatureCatalogue | Reference to an S-100 compliant logical data model description. To be used to be S-100 compliant, otherwise the modelDefinition element must be used. |
| modelDefinition | CharacterString | Only used if not S-100 compliant.  The definition of the service data model described. |
| encoding | CharacterString | Only used if not S-100 compliant.  The encoding of the attribute modelDefinition.  To be compliant to S100, refer to the S100 Data formats as described in the S100 Specification Appendix 4aD. |

## SERVICE TECHNICAL DESIGN

### Service Design Description Document

…

### Service Design Description Template

…

### Service Design Description XSD Structure

***Table 3 Information Elements of the Service Design Description***

| **Type Name** | | **Description** | |
| --- | --- | --- | --- |
| ServiceDesign | | A service design description. | |
|  | **Element Name** | **Type** | **Description** |
| name | CharacterString | The human readable name of the service design. The service name shall be at maximum a one-line brief label. Newer versions of the same service design adopt the same name. |
| id | ServiceIdentifier | Service identifier type to be used by service specifications, designs, instances. Currently, the identifier is defined as a string.  Globally unique identification of the service design. Newer versions of the same service design shall adopt the same id.  The identifier should conform to an MRN identity. |
| version | ServiceVersion | Service version indicator type to be used by service specifications, designs, instances.  Currently, the version indicator is defined as a string.  Version of the service design. A service design is uniquely identified by its id and version. Any change in the service physical data model or in the service specification reference requires a new version of the service design. |
| status | ServiceStatus <<enumeration>> | Status of the service design. The status field has one of the following values:  provisional;  released;  deprecated;  deleted. |
| description | CharacterString | A human readable short description of the service design. The description shall contain an abstract of what a service implementation does. |
| designsService-Specifications | ServiceSpecificationReference | Refers to service specification(s) that is/are realised by this service design. As a minimum, one service specification shall be referenced. One service design may realise several service specifications (either different versions of one specification, or even different specifications). |
| offersTransport | Transport | Refers to transport technologies offered by the service design. At least one reference shall be provided. |
| designedBy | VendorInfo | Mandatory reference to information about the author of the service design. |
| servicePhysicalData-Model | ServicePhysical-DataModel | Mandatory reference to the service physical data model description. |
| Type Name | | Description | |
| ServiceSpecification-Reference | | A reference to the service specification that is realised by the service design. – It has the id and the version of the respective service specification. | |
|  | Element Name | Type | Description |
| id | ServiceIdentifier | Service identifier type to be used by service specifications, designs, instances. Currently, the identifier is defined as a string.  Identification of the service specification realised by the service design. |
| version | ServiceVersion | Service version indicator type to be used by service specifications, designs, instances. Currently, the version indicator is defined as a string.  Version of the service specification realised by the service design. |
| Type Name | | Description | |
| Transport | | Definition of the transport protocol used by the service design. | |
|  | Element Name | Type | Description |
| name | CharacterString | Human readable name. |
| description | CharacterString | Human readable description of the transport protocol used by the service design. |
| protocol | CharacterString | A non-formal string representation of the transport (e.g. http/rest, http/soap, … ) that provides enough information to a service consumer to be able to connect. |
| Type Name | | Description | |
| VendorInfo | | Describes the vendor providing the service design. | |
|  | Element Name | Type | Description |
| id | CharacterString | Unique identification of the vendor. |
| name | CharacterString | Human readable vendor name. The name shall be no longer than one line. |
| description | CharacterString | Human readable description of the vendor. |
| contactInfo | CharacterString | Human readable contact information of the vendor. |
| organizationId | CharacterString | Unique identifier of the organization, the vendor belongs to i.e. the MRN identifier of the organization. |
| isCommercial | Boolean | Optional indication on the commercial status of the vendor. |
| Type Name | | Description | |
| ServicePhysicalDataModel | | The ServicePhysicalDataModel describes the data model for the service design. The ServicePhysicalDataModel describes in detail all the data structures being exchanged when service consumers interact with a service instance that implements this design. | |
|  | Element Name | Type | Description |
| name | CharacterString | Human readable model name. The name shall be no longer than one line. |
| description | CharacterString | Human readable description of the model. |
| model | CharacterString | The model can e.g. be a WSDL file, a JSON API, or the like. It is recommended to wrap the model in a CDATA section, and provide enough information in the name and description to make clear how to deal with the content in model. |
| modelType | CharacterString | The modelType shall contain an abbreviation that indicates what technology is used to describe the mode, e.g. WSDL, JSON. |

## SERVICE INSTANCE

### Service Instance Description Document

…

### Service Instance Description Template

…

### Service Instance Description XSD Structure

***Table 4 Information Elements of the Service Instance Description***

| Type Name | | Description | |
| --- | --- | --- | --- |
| ServiceInstance | | A service instance description. One service implementation may be deployed at several places by same or different service providers; each such deployment represents a different service instance. | |
|  | Element Name | Type | Description |
| name | CharacterString | The human readable name of the service instance. The service name shall be at maximum a one-line brief label. Newer versions of the same service specification shall adopt the same name. |
| id | ServiceIdentifier | Service identifier type to be used by service specifications, designs, instances. Currently, the identifier is defined as a string.  Globally unique identification of the service instance. Newer versions of the same service instance shall adopt the same id.  The identifier should conform to an MRN identity. |
| version | ServiceVersion | Service version indicator type to be used by service specifications, designs, instances. Currently, the version indicator is defined as a string.  Version of the service instance. A service instance is uniquely identified by its id and version. Any change in the service design reference requires a new version of the service instance. |
| status | ServiceStatus <<enumeration>> | Status of the service instance. The status field has one of the following values:   * provisional; * released; * deprecated; * deleted. |
| description | CharacterString | A human readable short description of the service instance. The description shall contain an abstract of what a service implementation does and what the service consumer should know about how the service implementation works. |
| keywords | CharacterString | A list of keywords associated to the service.  Keyword shall contain searchable words characterizing the service provided.  The keyword list is recommended to reflect type of service but can also contain payload types possible to exchange by the service instance, i.e. an open enumeration. |
| endpoint | CharacterString | Endpoint that describes where the service endpoint is located (e.g. URL) |
| MMSI | CharacterString | Optional Maritime Mobile Service Identity.  MMSI should be registered with 9 digits or left empty (blank). |
| IMO | CharacterString | Optional International Maritime Organization (IMO) number.  IMO should be registered with 7 digits or left empty (blank). |
| serviceType | CharacterString | Optional field to categorize the service type. Example: “VIS”.  The service type shall reflect the operational service type provided according to defined types in paragraph Service Type. If type not yet defined, a proposed type will be reviewed and added when accepted, i.e. open enumeration. |
| requiresAuthorization | Boolean | Indicates whether authorization is required or not. |
| implementsService-Design | ServiceDesignReference | Refers to the service design that is implemented by this service instance. Exactly one service design shall be referenced. |
| coversAreas | CoverageInfo | Mandatory reference to the geographical area covered by the service instance. |
| offersServiceLevel | ServiceLevel | Refers to the definition of the service level fulfilled by the service instance. Exactly one service level definition shall be provided. |
| producedBy | VendorInfo | Optional reference to information about the producer of the service implementation. |
| providedBy | VendorInfo | Mandatory reference to information about the service provider of the service instance. |
| ServiceDesignReference | | A reference to the service design that is implemented by the service instance. – It has the id and the version of the respective service design. | |
|  | Element Name | Type | Description |
| id | ServiceIdentifier | Service identifier type to be used by service specifications, designs, instances. Currently, the identifier is defined as a string.  Service identifier type to be used by service specifications, designs, instances. Currently, the identifier is defined as a string.  Identification of the service design implemented by the service instance. |
| version | ServiceVersion | Service version indicator type to be used by service specifications, designs, instances. Currently, the version indicator is defined as a string.  Service version indicator type to be used by service specifications, designs, instances.  Version of the service design implemented by the service instance. |
| Type Name | | Description | |
| CoverageInfo | | Defines a geographical area from which the service instance is accessible. This is a choice between a geographical area defined by co-ordinates or the United Nations Code for Trade and Transport Locations (UN/LOCODE). One of the two options must be provided. Worldwide accessibility is indicated by a ‘coversArea’ element with a missing ‘geometryAsWKT’ element. | |
|  | Element Name | Type | Description |
| coversArea | CoverageArea | Identification of the service design implemented by the service instance. |
| unLoCode | CharacterString | Version of the service design implemented by the service instance.  UnLocode should be registered with 5 characters, no space, capital letters or left empty (blank) |
| Type Name | | Description | |
| CoverageArea | | Defines a geographical area from which the service instance is accessible. | |
|  | Element Name | Type | Description |
| name | CharacterString | Human readable name of the coverage area, e.g. a well-known name like ‘Bermuda Triangle’. The name shall be no longer than one line. |
| description | CharacterString | Human readable description of the coverage area. |
| geometryAsWKT | CharacterString | A polygon described in WKT (Well Known Text) with coordinates in co-ordinate reference system EPSG:4326, e.g. POLYGON(LON1 LAT1, LON2 LAT2, LON3, LAT3, LON1 LAT1).  If the element is empty, the default is the whole world. |
| ServiceLevel | | Defines the service availability level. | |
|  | Element Name | Type | Description |
| name | CharacterString | Human readable service level name. The name shall be no longer than one line. |
| description | CharacterString | Human readable description of the service level. |
| availability | Real | Indicates the guaranteed availability of the service in %, (e.g. 99.9). |
| VendorInfo | | Describes the vendor producing and/or providing the service instance. | |
|  | Element Name | Type | Description |
| id | CharacterString | Unique identification of the vendor. |
| name | CharacterString | Human readable vendor name. The name shall be no longer than one line. |
| description | CharacterString | Human readable description of the vendor. |
| contactInfo | CharacterString | Human readable contact information of the vendor. |
| organizationId | CharacterString | Unique identifier of the organization, the author belongs to. MRN |
| isCommercial | Boolean | Optional indication on the commercial status of the vendor. |

# Action requested of the Committee

The ENAV committee is asked to consider and adopt the presented proposals.

1. Input document number, to be assigned by the Committee Secretary [↑](#footnote-ref-2)
2. Leave open if uncertain [↑](#footnote-ref-3)