GUIDANCE NOTES
GD21-2023

INTERNATIONAL SHIP CLASSIFICATION

TECHNICAL GUIDELINES FOR DATA EXCHANGE OF SHIP DIGITAL SURVEY

2023

Effective from 1 May 2023

CONTENTS

CHAPTER 1 GENERAL

Section 1 General Provisions

1.1.1 Scope of application

1.1.1.1 The Guidelines stipulate the data standards and interface rules for data exchange of ship digital survey. They are applicable to ships applying to digital survey by International Ship Classification (hereinafter referred to as "ISC") and can also be used as a reference for other businesses involving data exchange.

1.1.1.2 In terms of ships applying for digital survey, the technical requirements of the Guidelines shall be followed when the data is sent to ISC or data access is authorized to ISC, including data naming format and data structure, and data should be provided as per standard interface protocols.

1.1.2 Definition

1.1.2.1 Unless otherwise specified, relevant definitions in the Guidelines are as follows:

(1) **Digitization**: It refers to information system, various sensors, machine vision and other technologies that are used to obtain the original data, various information and related knowledge of the survey object to generate identifiable, storable, and computable data, so as to build relevant data models for processing, analysis and application.

(2) **Survey object**: It refers to the ship documents/materials, hull structure, mechanical/electronic equipment (system) and other ship survey targets.

(3) **Data channel**: It refers to virtual channels that are used for data transmission from ship machinery/electronic equipment to ship-board data server or ship digital system, defining the data static characteristics of equipment operating status, such as expressing the exhaust temperature of the main engine cylinder and the fuel inlet pressure through the data channels.

(4) **Data channel ID**: It refers to the data channel identifier that is used to uniquely identify the data channel of the ship-board digital equipment. For its usage, see 2.3.2 of Chapter 2.

(5) **Data channel list**: It refers to a list that defines the data channel IDs and data channel attributes and is shared by the data owner.

(6) **Data channel attribute**: It refers to the attribute of the data channel, such as the unit and range.

(7) **Namespace**: It refers to the set of names to avoid name conflicts.

(8) **Time series data**: It refers to a dataset collected in chronological order.

Note: The definitions of terms related to data channels above apply to the expression of time series data.

1.1.3 References

1.1.3.1 Clauses in relevant documents referenced herein will become an integral part of the Guidelines. For dated reference documents, only the dated version is applicable to the Guidelines. For undated reference documents, the latest version (including all revision notices and change notices) applies to the Guidelines.

(1) ISO 19847:2018 Ships and marine technology - Shipboard data servers to share field data at sea

Note: GB/T 42054 Ships and marine technology - Shipboard data servers to share field data at sea (ISO 19847:2018, IDT)

(2) ISO 19848:2018 Ships and marine technology - Standard data for shipboard machinery and equipment

Note: GB/T 42055-2022 Ships and marine technology - Standard data for shipboard machinery and equipment (ISO 19848:2018, IDT)

(3) ISO 80000 Series Quantities and units

(4) ISO 8601 Data elements and interchange formats - Information interchange - Representation of dates and times

Note: GB/T 7408-2005 Data elements and interchange formats - Information interchange - Representation of dates and times (ISO 8601:2000, IDT)

(5) W3C XML Extensible Markup Language (XML) 1.0, W3C Recommendation

(6) W3C XML Schema Part 1: W3C Recommendation

(7) W3C XML Schema Part 2: Datatypes, W3C Recommendation

(8) RFC 3339 Date and Time on the Internet: Timestamps

(9) RFC 4180 Common Format and MIME Type for Comma-Separated Values (CSV) Files

(10) RFC 5234 Augmented BNF for Syntax Specifications: ABNF

(11) RFC 3986 Uniform Resource Identifier URI: Generic Syntax

1.1.4 Abbreviations

1.1.4.1 The following abbreviations are applicable to the Guidelines:

(1) HTTP: Hypertext Transfer Protocol

(2) RESTFUL: also known as REST (Representational State Transfer), a design style and development method for web applications

(3) JSON: JavaScript Object Notation

(4) URI: Uniform Resource Identifier

(5) URL: Uniform Resource Locator

(6) UTC: Universal Time Coordinated

(7) UTF-8: UCS Transformation Format 8

(8) XML: Extensible Markup Language

(9) CSV: Comma Separated Values

Section 2 Data Scope and Type

1.2.1 Scope of data

1.2.1.1 Relevant data used for ship digital survey specified in the Guidelines includes but not limited to ship documents/data, hull structure, shipboard machinery/electronic equipment (system) and other survey objects that produce data in various structural forms.

1.2.2 Type of data

1.2.2.1 According to data range involved in digital survey and the data type, the data are classified into structured data, semi-structured data and unstructured data, as shown in Figure 1.2.2.1 below.

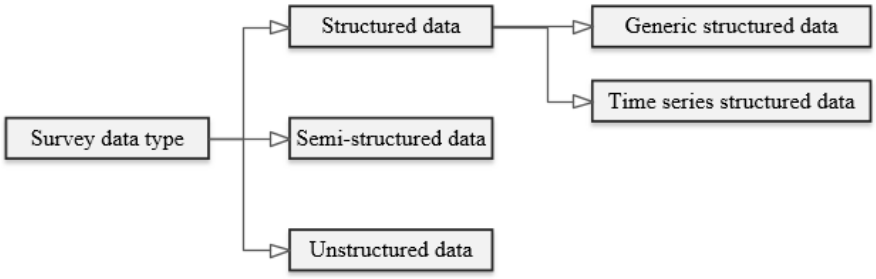

Figure 1.2.2.1 Survey Data Type

1.2.2.2 Structured data refers to the data that can be expressed and stored using a relational database, with characteristics of two-dimensional data, for example, the structured query language (SQL) database. According to the features of data composition involved in digital survey, the structured data are classified into generic structured data (hereinafter referred to as "generic data") and time series structured data (hereinafter referred to as "time series data").

1.2.2.3 Semi-structured data has a basic fixed structure pattern, which is an expression form of the structured data. For example, system logs and electronic record books represented by XML or JSON, etc., are processed as generic data during the data organization and data exchange in the Guidelines.

1.2.2.4 Unstructured data, without fixed structure such as documents, pictures, audio and video, etc., can be described in the form of structured metadata. Data entities are described and stored as elements of structured metadata, and can be transmitted in accordance with the generic data structure principles in the Guidelines. The unstructured data are processed as generic data during the data organization and data exchange in the Guidelines.

1.2.2.5 Based on the sorting out and classification of survey data types, the naming, data structure and data interface of generic data and time series data are specified in Chapter 2 and Chapter 3 of the Guideline, respectively.

CHAPTER 2 DATA IDENTIFICATION AND STRUCTURE

Section 1 General Provisions

2.1.1 General requirements

2.1.1.1 This chapter stipulates the identification methods and data structure requirements for generic data and time series data.

2.1.1.2 Data identification is to identify and distinguish different survey objects, and ensure the consistency and uniqueness of survey objects at the level of identification and naming.

2.1.1.3 Identification coding is the process of assigning a specific code to the survey object data.

2.1.1.4 The principle of identification coding should comply with the requirements of 2.2.2.1 in the Guidelines for Verification of Digital Systems of Ships and Offshore Installations of ISC.

2.1.1.5 The time series data identification method is adopted for the point objects collected by ship equipment sensors.

2.1.1.6 Other survey objects, such as ship documents, electronic record books, etc., are collected with the generic data identification method.

2.1.1.7 Data structure is defined as a logical structure independent of data format (such as XML, JSON, CSV, etc.).

Section 2 Generic Data Identification

2.2.1 General requirements

2.2.1.1 This section stipulates the identification and attribute requirements for generic data.

2.2.1.2 The identification of generic data is based on ISC ship digital model (referred to as: emd).

2.2.2 Identification of generic data type

2.2.2.1 The English code for the identification of generic data type is TypeID.

2.2.2.2 The type identification code for generic data complies with the URI format definition, which supports many different combinations. The type identification methods are a subset of these combinations, and the specific format requirements are as shown in Table 2.2.2.2 (1).

Identification of Generic Data Type **Table 2.2.2.2 (1)**

Identification of generic data type	=[Protocol]: //Named entity/ship ID/naming rules/version/classification code of generic data type
Named element	**Description**

Named entity	Entity
Ship ID	Ship identification
/	Path retention character
Naming rules	Identification scheme rules, specifically referring to the emd in the Guidelines
Version	Naming rule version, specifically referring to the emd version number in the Guidelines
Classification code of generic data type	Specifically referring to the classification of the emd generic data in the Guidelines

(1) Identification combinations of generic data type are defined by the augmented BNF (ABNF) in RFC5234, the "protocol" element is optional in URI, and the slash ("/") is a reserved character for describing the hierarchy.

(2) Identification combinations of generic data type are explained in details.

① Named entity

Named entity elements shall be domains owned or controlled by the entity generating the universal ID.

The named entity of ISC: digitalship.ISC.org.cn

② Ship ID

(a) Ship ID is the universal identification of the ship.

(b) Generally, the ship ID should be the IMO number (international navigation ship) or CMSA ID (Chinese ship).

Example:

——IMO1234567

——/CN202212345672

③ Naming rules

(a) The naming rule shall cover the specified name of the data channel naming rule.

(b) The name can be set freely under the supervision of the named entity and shall be preceded by a symbol denoting the named entity, to avoid duplication.

(c) The naming rule refers to a set of requirements defining a naming scheme (or identification scheme) for shipboard components and systems. The naming rule should define the composition of the identification string and the method for developing the identification string.

(d) The naming rule in the Guidelines specifically refers to emd. See Appendix 1 for the model acquisition method.

④ Version

The version refers to the version number of naming rule.

⑤ Classification code of generic data

(a) Classification code of generic data specifically refers to the classification code in emd. Refer to Appendix 1 for the classification codes.

(b) The classification code of generic data should be selected according to emd.

Example: P1.003.006

It represents a ship certificate classification code.

2.2.2.3 The following are naming examples based on ISC named entities.

Example: ship certificate

Identification of generic data type	http://digitalship.ISC.org.cn/imo1234567/emd/v1.0/P1.003.006
Ship ID	imo1234567
Named entity	digitalship.ISC.org.cn
Naming rules	emd
Version	v1.0
Classification code of generic data	P1.003.006

Note: The generic data type is not required to be identified as a resolvable URI, i.e., the URI is not necessarily a valid URL.

2.2.3 Attribute of generic data type

2.2.3.1 The set of generic data type attributes specified in the Guidelines is as shown in Table 2.2.3.1.

Type Attribute of Generic Data **Table 2.2.3.1**

Attribute code	Description
Data Object Type	To identify data class, such as the normal files, media files, structured data, etc.
Description Language	To describe the language or syntax used by the data object, supporting XML and JSON
Object Metadata	To describe the metadata structure of data object
Name	To identify name of generic data type
Remarks	

(1) In practical application, the data should be described according to the attribute regulations above.

(2) If the attribute types not listed above are obviously different from those stipulated in the Guidelines, they can be extended on this basis with the consent of ISC.

2.2.3.2 Regulations on attribute of generic data type

(1) Data object type

The attribute values of the data object type conform to Table 2.2.3.2 (1).

Type Attribute of Data Object **Table 2.2.3.2 (1)**

Type code	Description
Normal File	Common format documents, such as certificates, reports, drawings, and materials.
Media File	Audio, video and other media files.
Structured Data	Simple or structured relational data with certain relationships.

The attribute of data object type is mandatory. For special data uncovered by the type code range, a new type can be added with the consent of ISC.

Example:

GeneralObject.TypeID	…	GeneralObject.ObjectType	…
……	…	Normal File	…

(2) Description language

The description language attribute is used to specify the language or syntax used in the description of generic data type objects. The languages or syntax supported by the Guidelines comply with Table 2.2.3.2 (2).

Description Language Attribute **Table 2.2.3.2 (2)**

Type code	Description
XML	To describe the generic data object with XML language.
JSON	To describe the generic data object with JSON syntax.

The description language attribute is mandatory, for which the language or syntax used to describe the generic data type object should be specified.

Example:

GeneralObject.TypeID	…	GeneralObject.Language	…
……	…	JSON	…

(3) Object metadata

Object metadata entities shall be encoded with the Base64 algorithm. For plaintext examples of specific object metadata structures, see Appendix 4.

(4) Name

The name attribute represents the name of the object of generic data type.

Example:

GeneralObject.TypeID	…	GeneralObject.Name	…
……	…	Electronic record book of ship	…

(5) Remarks

Supplementary information about the object of the target generic data type can be described in the remarks.

Section 3 Time Series Data Identification

2.3.1 General requirements

2.3.1.1 Time series data are identified in the form of a data channel, which consists of the data channel ID and data channel attributes.

2.3.1.2 The data channel naming specified in this section is based on data channel ID and attribute identification method in emd.

2.3.2 Data channel ID

2.3.2.1 The data channel ID includes the following three expressions:

—— Universal ID

—— Local ID

—— Short ID

(1) Universal ID is a universal identification shipboard data channel, which is a universal identification method aiming at data exchange, and it is a globally unique identifier.

(2) Local ID is used to identify the shipboard data channel in the ship. For example, the shipboard computer system, integrated automation system (IAS) and alarm monitoring system (AMS) all have their own data channel lists, which are composed of unique channel IDs corresponding to local IDs.

(3) Short ID is an optional short alternative identifier for the local ID.

(4) These IDs should be case-insensitive to avoid accidental typos.

(5) For the same ship, local ID and short ID shall be unique. For different ships, the data channels of the same type of sensors have the same local ID. On the contrary, the short ID may be different for each ship even if the data channel has the same meaning.

2.3.2.2 Universal ID

(1) The universal ID identifier complies with the URI format, which supports many different combinations. The ID is a subset of these combinations, and the specific format requirements are as shown in Table 2.3.2.2 (1).

Universal ID naming format **Table 2.3.2.2 (1)**

Universal ID	=[Protocol]: //Named entity/ship ID/local ID
Named element	**Description**
Named entity	Entity
Ship ID	Ship identification
/	Reserved character
Local ID	Path element | path element...

(2) Universal ID combinations are defined by the augmented BNF (ABNF) in RFC5234, the "protocol" element is optional in URI, and the slash ("/") is a reserved character for describing the hierarchy.

(3) Universal ID combinations are explained in details.

① Named entity

(a) Named entity elements shall be domains owned or controlled by the entity generating the universal ID.

(b) The named entities of ISC: digitalship.ISC.org.cn

② Ship ID

(a) Ship ID is the universal identification of the ship.

(b) Generally, the ship ID should be the IMO number (international navigation ship) or CMSA ID (Chinese ship).

Example:

——/IMO1234567

——/CN202212345672

2.3.2.3 Local ID

(1) The local ID is defined by ABNF, and the format is as shown in Table 2.3.2.3 (1).

Local ID naming format **Table 2.3.2.3 (1)**

Local ID	= Naming rules/version/ship equipment classification code/equipment monitoring object/position/sensor signal type
Named element	**Description**
Naming rules	Identification scheme rules, specifically referring to emd in the Guidelines
/	Path identifier
Version	emd version number
Classification code of ship equipment	Equipment classification identification based on emd
Equipment monitoring object	Combinations of equipment monitoring objects and object numbers
Location	Location of equipment monitoring object
Sensor signal type	Signal type identification of equipment sensor

(2) Local ID combinations are explained in details.

① Naming rules

(a) The naming rule shall by the specified name as per the data channel naming rule.

(b) The name can be set freely under the supervision of the named entity and should be preceded by a symbol denoting the named entity, to avoid duplication.

(c) The naming rule refers to a set of requirements defining a naming scheme (or identification scheme) for shipboard components and systems. The naming rule shall specify the composition of the identification string and the method for developing the identification string.

(d) The naming rule in the Guidelines specifically refers to emd. See Appendix 1 for the model acquisition method.

② Version

The version refers to the ship. For version number of naming rule.

③ Classification code of ship equipment

(a) Classification code of ship equipment specifically refers to the classification code in emd. Refer to Appendix 1 for the acquisition of classification codes.

(b) The classification code of ship equipment should be selected according to emd.

Format: Main device code. sub-device code + number

Example: P4.003.003+1

The example represents No. 1 diesel engine, where "+number" is an optional item, which is applicable to the multiple similar installations.

④ Equipment monitoring object

(a) The equipment monitoring object is suitable for situations where the data channel cannot be clearly expressed through the classification code of ship equipment.

(b) If the device monitoring object is expressed as a combination of multi-level objects, "." shall be used to distinguish objects.

(c) This item is optional. In principle, the object name shall be expressed in English words or phrases, including no spaces between words and no reserved characters in URI such as /, ?, #, [ ], and @.

Format: Equipment monitoring object.equipment monitoring object+number

Example: CYLINDER.EXHAUST_GAS + 2

The example represents the exhaust gas from No. 2 cylinder, where "+number" is an optional item, which is applicable to the multiple similar monitoring objects.

⑤ Location

Location identification represents the position of the equipment monitoring object. Refer to Appendix 2 for the location code. The location code in Appendix 2 is preferred. In case of code unlisted in Appendix 2, self-defining is allowed. Meaningful English words are recommended for the location, and can be used after the evaluation and confirmation by ISC.

Example: Inlet

⑥ Sensor signal type

The sensor signal type represents the type code of the monitored object sensor. Refer to Appendix 3 for the sensor signal. The signal type in Appendix 3 is preferred. In case of type unlisted in Appendix 3, self-defining is allowed, and meaningful English words are recommended.

Example: Speed

2.3.2.4 Short ID

(1) Short ID is an optional short alternative identifier for the local ID. There should be a one-to-one correspondence between the data channel and the short ID, in other words, the short ID is unique to a ship.

(2) The short ID rules are as follows:

Short ID = Unreserved character

(3) Short ID should be as short as possible and expressed using machine-friendly symbols, user-friendly phrases, or a combination of these symbols and phrases.

Example:

——0001

——TAH001

——ME001_RPM

2.3.2.5 The following are naming examples based on ISC named entities.

Example: exhaust temperature of No. 2 cylinder in No. 1 diesel generator set

Universal ID	http://digitalship.ISC.org.cn/IMO1234567/emd/v1.0/P8.003.006 +1/Cylinder.Exhaust_Gas+2/Outlet/Temperature
Local ID	emd/v1.0/P8.003.006 +1/Cylinder.Exhaust_Gas+2/Outlet/Temperature
Short ID	0001
Ship ID	IMO1234567
Named entity	digitalship.ISC.org.cn
Naming rules	emd
Local name	G/E EXHAUST GAS TEMP. CYL.2

Note: The universal ID is not required to be identified as a resolvable URI, i.e., the URI is not necessarily a valid URL.

2.3.3 Data channel attribute

2.3.3.1 The classes of data channel attributes specified in Guidelines is as shown in Table 2.2.3.1.

Data channel attribute Table 2.3.3.1

Attribute code	Description
Data Channel Type	To identify the type of data channel
Format	To describe data format
Range	To describe the data range, such as the upper and lower limits of the value
Unit	Units and quantities used for measured values
Quality Coding	To represent the name of the data quality evaluation model
Name	To describe the designation assigned to shipboard control systems and other instruments
Remarks	

(1) In practical application, the data shall be described according to the above attribute regulations.

(2) If the attribute types not listed above are obviously different from those stipulated in the Guidelines, they can be extended on this basis with the consent of ISC.

2.3.3.2 Regulations on channel attribute

(1) Data channel type

① The data channel type is used to identify the channel type, such as the measured value, average value, alarm and status. The data channel type consists of the following sub-attributes:

——Type

——Update cycle

——Calculation period

② The type sub-attribute specifies the type of data channel, and the value of the attribute conforms to Table 2.3.3.2 (1).

Type Sub-attributes **Table 2.3.3.2 (1)**

Type code	Description
Inst	A measured value at certain moment.
Average	Average value over a certain period of time. "Average" does not mean the average value of multiple sensors at the same moment, but that of the time series values of a single sensor.
Max	The maximum value over a certain period of time. "Max" does not mean the maximum value of multiple sensors at same moment, but that of the time series values of a single sensor.
Min	The minimum value over a certain period of time. "Min" does not mean the minimum value of multiple sensors at same moment, but that of the time series values of a single sensor.
StandardDeviation	The standard deviation of values over a period of time. "StandardDeviation" does not mean the standard deviation of multiple sensors at the same moment, but that of the time series values of a single sensor.
Calculated	A value obtained by calculation rather than measurement.
SetPoint	Automatically-controlled target value.
ControlOutput	Automatically-controlled manipulate value.
Alert	Available alarm value.
Status	Available status value
ManuallyInput	The value entered by the crew. The assumed value herein is the reading of the indicator.

③ Update frequency, representing how often the measurement is updated, shall be used when the measured value is updated on a regular basis.

④ If the value of a data channel is the calculated result based on measurements for a specific time period, the calculation period shall be used to represent the time period.

⑤ The update frequency and calculation period shall be described by decimal numbers greater than 0

(in "seconds").

⑥ The type sub-attribute is mandatory, and other attributes are optional.

Example:

DataChannelID.LocalID	…	DataChannelProperty. DataChannelType. Type	DataChannelProperty. DataChannelType. UpdateCycle	DataChannelProperty. DataChannelType. CalculationPeriod
emd/v1.0/ P8.003.006+1/ALTERNATO R_BEARING// TEMPERATURE	…	Inst	5	60

(2) Format

① Format, which is used to describe the data format, consists of the following sub-attributes.

——Type

——Restriction

② The type sub-attribute is mandatory while the restriction sub-attribute is optional, and the format attribute may include multiple restriction sub-attributes.

③ Available types, as shown in 2.3.3.2 (2), comply with Part 2: Data Type of W3C XML Schema Definition Language (XSD) 1.1.

Type Sub-attributes **Table 2.3.3.2 (2)**

Type code	Description
Decimal	Numbers with decimal places represent a subset of the real numbers, which can be expressed as decimal numbers. The value of a decimal is a set of numbers obtained by dividing an integer by a non-negative power of 10, which can be expressed as $i/10^n$, where i and n are integers ($n \geq 0$). Precision is not reflected in this value space, for example, 2.0 is indistinguishable from 2.00. The ordinal relation of the decimal system is that of the real numbers, limited to this subset.
Integer	Integers are derived from decimal by fixing the value of the decimal place to 0 without the decimal point. This leads to the standard mathematical concept of the integer. The value space of integers is an infinite set{....,-2, -1, 0, 1, 2, ... }. The base type for integers is decimal.
Boolean	The Boolean value represents the value of binary logic.
String	The string data type represents strings in XML.
DateTime	Date and time data types are used for values that contain dates and times. The format should follow ISO 8601 "YYYY-MM-DDThh:mm:ssZ", where: YYYY represents the year; MM represents the month; DD represents date; T represents the start of the desired time period; hh represents hour; mm represents minutes; ss represents seconds; Z represents UTC.

④ The restrictions defined in W3C XML schema are as shown in Table 2.3.3.2 (3), and can be used to specify acceptable values. Verification rules and available constraints for each data type shall be subject to W3C XML schema.

Constraint Sub-attributes **Table 2.3.3.2 (3)**

Restriction	Description	Data type
Enumeration	To define a list of acceptable enumeration values	String
FractionDigits	To specify the maximum decimal places allowed (equal to or greater than zero).	Non-negative integer
Length	To specify the exact number of characters or list items allowed (equal to or greater than zero).	Non-negative integer
MaxExclusive	To define the upper limit of the value (the allowed value shall be less than this value).	Value from the·value space·of {primitive type definition}
MaxInclusive	To define the upper limit of the value (allowed values shall be less than or equal to this value).	Value from the·value space·of {primitive type definition}
MaxLength	To define the maximum number of characters or list items allowed (greater than or equal to zero).	Non-negative integer
MinExclusive	To define the lower limit of the value. Allowed value shall be greater than this value.	Value from the·value space·of {primitive type definition}
Inclusive	To define the lower limit of the value. Allowed value shall be greater than or equal to this value.	Value from the·value space·of {primitive type definition}
MinLength	To define the minimum number of characters or list items allowed (greater than or equal to zero).	Non-negative integer
Pattern	To define the exact sequence of acceptable characters.	String
TotalDigits	To specify the exact digits allowed (greater than zero).	Positive integer
WhiteSpace	To specify how to process null values (line, tab, space, and enter).	

Example:

DataChannelID.LocalID	…	DataChannelProperty. Format.Type	DataChannelProperty. Format.Restriction
……	…	Integer	MaxExclusive 5000

(3) Range

① The range is used to describe the data range. The range attribute consists of the following sub-attributes.

——Low

——High

② The "low" sub-attribute represents the lower limit of the simulated data while the "high" sub-attribute represents the upper limit.

③ The data type of the sub-attribute value is “numeric” with precision or null. The value of a sub-attribute can only be “null” when the lower and/or upper limit cannot be specified.

④ The high value of the range must be greater than but not equal to the low value.

⑤ The range attribute is only mandatory for simulated data.

⑥ The range defines no upper and lower boundaries of a value. In the event of sensor failure or other abnormal conditions, this value may be out of range.

Example:

DataChannelID.LocalID	…	DataChannelProperty.Range.Low	DataChannelProperty.Range.High
……	…	0	120

(4) Unit

① Unit attribute applies to units and quantities of measured values. The unit attribute consists of the following sub-attributes.

——Unit symbol

——Quantity name

② The unit symbol and quantity name defined in ISO80000 apply to the unit symbol and quantity name sub-attributes.

The unit symbol sub-attribute represents the unit symbol for the measurement (for example, "m” represents the length, and "kg" represents the mass).

③ The quantity name sub-attribute applies to the variable value identified as the measured or calculated values for the physical or chemical item of the data channel.

④ The measures value shall be described in units specified in this attribute.

Example:

DataChannelID.LocalID	…	DataChannelProperty.Unit.UnitSymbol	DataChannelProperty.Unit.QuantityName
……	…	kW	Power

(5) Quality coding

The quality coding represents the name of the data quality evaluation model, which shall be able to distinguish valid measured values from the invalid of the data channel.

Example:

DataChannelID.LocalID	…	DataChannelProperty.OpcQuality	…
……	…	IEC 61162-STATUS	…

Note: Under "IEC 61162—STATUS", "A" (valid data) and "V" (invalid data) reflects the data quality.

(6) Name

It is used to describe the designation assigned to shipboard control systems and other instruments

Example:

DataChannelID.LocalID	…	DataChannelProperty.Name	…
……	…	Main engine speed	…

(7) Remarks

Supplementary information of data channel is described in the remarks.

Example:

DataChannelID.LocalID	…	DataChannelProperty.Remarks	…
……	…	Location: central control room, manufacturer: AAA Company, model: model - AAA	…

Section 4 Data Structure

2.4.1 General requirements

2.4.1.1 The data structure requirements for generic and time series data are specified in this section.

2.4.1.2 Data structures can be described in XML, JSON and CSV formats.

2.4.1.3 The standard data types used to define data structures and independent of the implementation language are described in this section.

2.4.1.4 Standard data types may be replaced by those defined in the implementation language.

2.4.1.5 The purpose of the generic data list structure and time series data channel list structure specified in this section is to share or submit the relevant generic data list or data channel list to ISC according to the requirements of the list format before the data transmission. Then, the received data is identified and parsed by ISC according to the list.

2.4.2 Implementation language

2.4.2.1 Refer to Appendix 4 for an example of the generic data structure format specified in this section.

2.4.2.2 The time series data structure specified in this section shall be implemented in accordance with the requirements of Appendix 5.

2.4.3 Standard data type

2.4.3.1 To define the data structure, the following standard data types derived from the basic types of standard modeling language (UML) listed in Table 2.4.3.1 are used.

Standard data type **Table 2.4.3.1**

Standard data type	Basic type	Constraints	Remarks
Integer	Integer		
NonNegativeInteger	Integer	＞-1	
PositiveInteger	Integer	＞0	
Real	Real		
Boolean	Boolean		
String	String		
DateTime	String	Formats comply with ISO 8601	For ABNF expression, see RFC3339
Null			Null represents missing value (applicable to any data type)

2.4.4 List structure of generic data

2.4.4.1 Data model

The list structure of generic data should contain the following five elements:

(1) Package

A package is a data packet consisting of a header and a data subject.

(2) Header

The header indicates when and by whom the generic data channel list is created.

(3) GeneralDataList

A general data list consists of a series of generic data type identifiers and generic data type attributes.

(4) Type identification of generic data(TypeID)

The type identification of generic data is the type identification specified in Section 2, Chapter 2 of the Guidelines.

(5) Type attribute of generic data(Property)

Type attribute of generic data the type attribute of generic data specifies the property.

These elements are arranged in a hierarchical structure as shown in Figure 2.4.4.1.

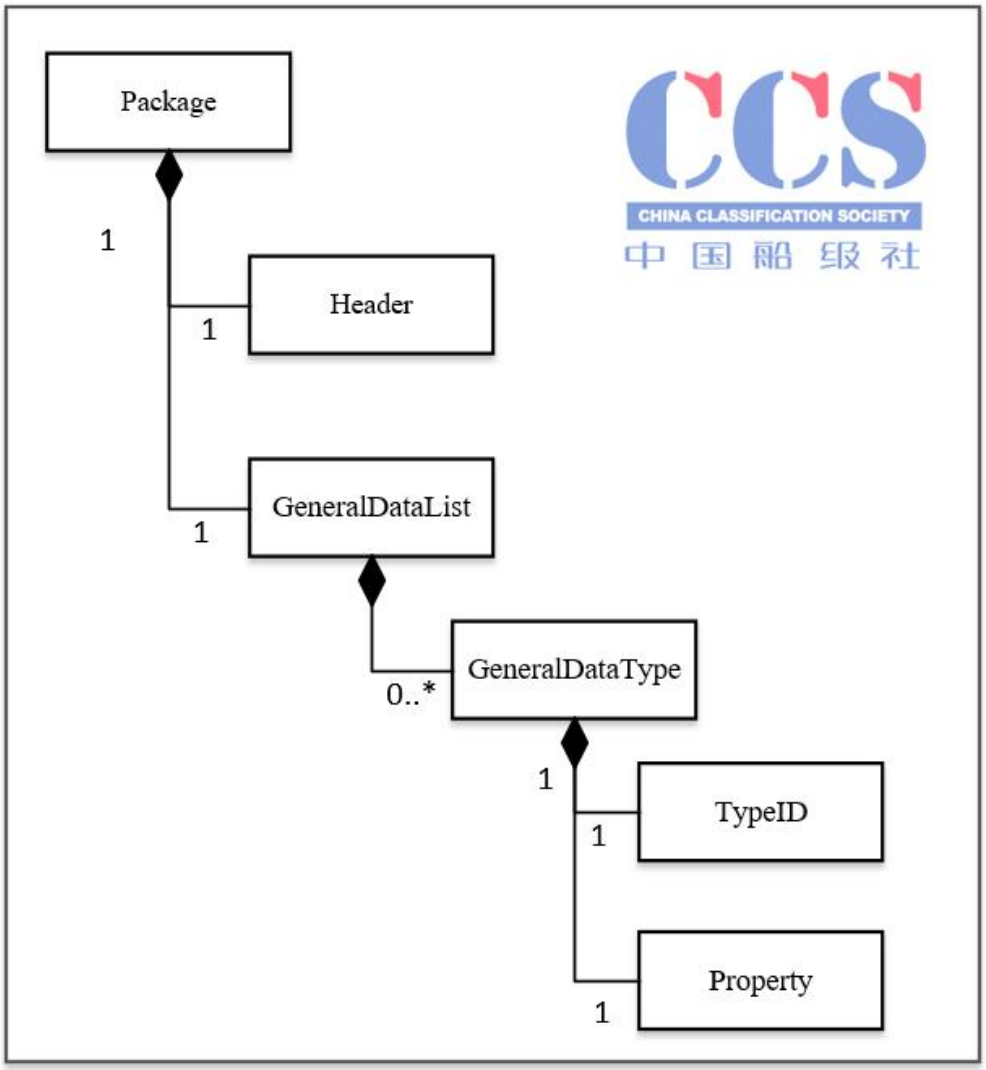

Figure 2.4.4.1 List Structure Model of Generic Data

2.4.4.2 Logical structure

(1) The logical structure of the list structure of generic data type is as shown in Figure 2.4.4.2.

(2) The list structure of generic data type is embodied as a data packet, and the packet elements are composed of the header and the list elements of generic data type. The type elements of generic data include the type identification of generic data and attribute elements.

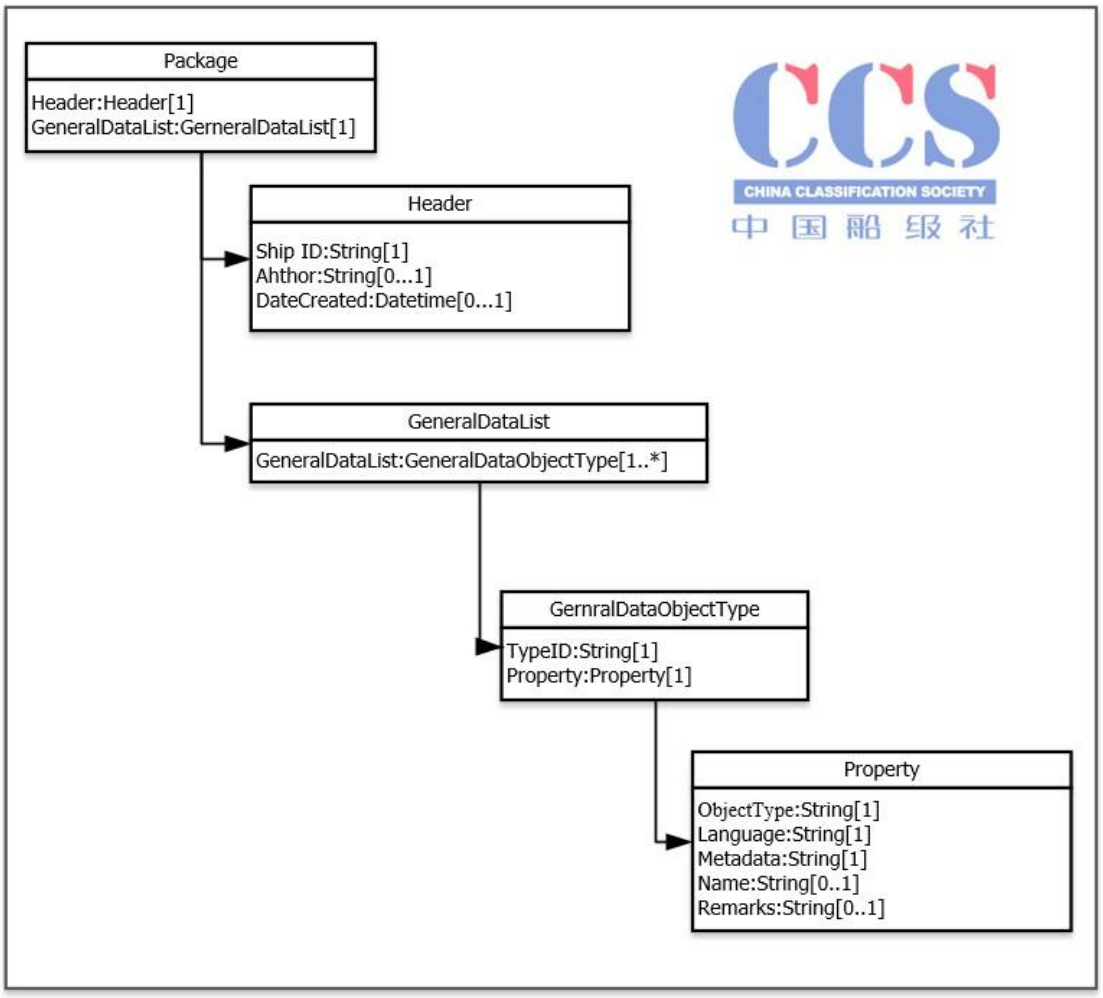

Figure 2.4.4.2 Logic Diagram of List Structure of Generic Data

2.4.4.3 The details of each element are described below.

(1) Package structure

Package structure **Table2.4.4.3 (1)**

Name	Data type	Remarks	Mandatory/ optional	Maximum
Header	(2) Header	See (2)	Mandatory	1
GerneralDataList	(3) GerneralDataList	See (3)	Mandatory	1

(2) Header structure

Header structure **Table 2.4.4.3 (2)**

Name	Data type	Remarks	Mandatory/optional	Maximum
ShipId	String	Ship identification	Mandatory	1
Author	String	List owner	Optional	1
DataCreated	Datetime	Package creation date	Optional	1

(3) GerneralDataList structure

Data Channel List structure **Table 2.4.4.3 (3)**

Name	Data type	Remarks	Mandatory/optional	Maximum
GeneralDataList	(4) GeneralDataObjectType	See (4)	Mandatory	*

(4) GeneralDataObjectType structure

GeneralDataObjectType structure **Table 2.4.4.3 (4)**

Name	Data type	Remarks	Mandatory/optional	Maximum
Type ID	String	See 2.2.2 of Chapter 2	Mandatory	1
Property	5) Property	See (5)	Mandatory	1

(5) Property structure

Property structure **Table2.4.4.3 (5)**

Name	Data type	Remarks	Mandatory/optional	Maximum
ObjectType	String	See 2.2.3.2 (1) of Chapter 2	Mandatory	1
Language	String	See 2.2.3.2 (2) of Chapter 2	Mandatory	1
Metadata	String	See 2.2.3.2 (3) of Chapter 2	Mandatory	1
Name	String	See 2.2.3.2 (4) of Chapter 2	Optional	1
Remarks	String	See 2.2.3.2 (5) of Chapter 2	Optional	1

2.4.5 Structure of generic data

2.4.5.1 Data model

The general type data consists of the following elements:

(1) Package

The package element is composed of a header (metadata) and the generic data (data subject).

(2) Header

① The header element contains the necessary metadata to represent when the data is created and by whom.

② The header element can be omitted if it is provided in advance by any means.

(3) GeneralTypeData

① The element of general type data is the subject of the package, containing the general type data in Section 2.

② The element of general type data also represents a set of data sets. The element should be newly generated as soon as the generic data list is updated, which affects the data value.

(4) DataSet

DataSet refers to a data set of object metadata structures composed of a set of type identifications of specific generic data

These elements are arranged in a hierarchical structure as shown in Figure 2.4.5.1.

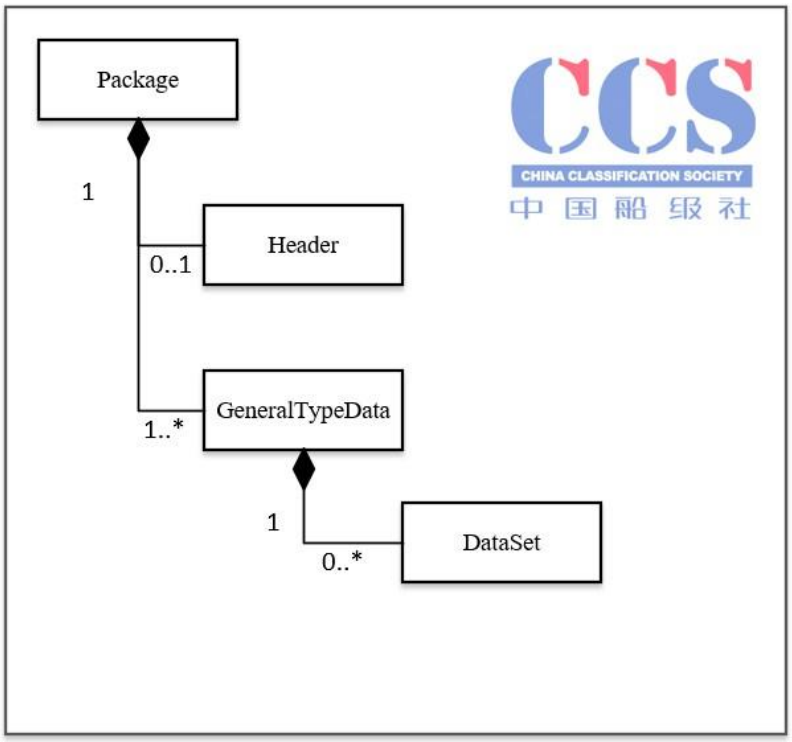

Figure 2.4.5.1 Model of General Type Data

2.4.5.2 Logical structure

(1) General type data should have the logical structure shown in Figure 2.4.5.2.

(2) The general type data is embodied as the package, which is composed of the header and the general type data.

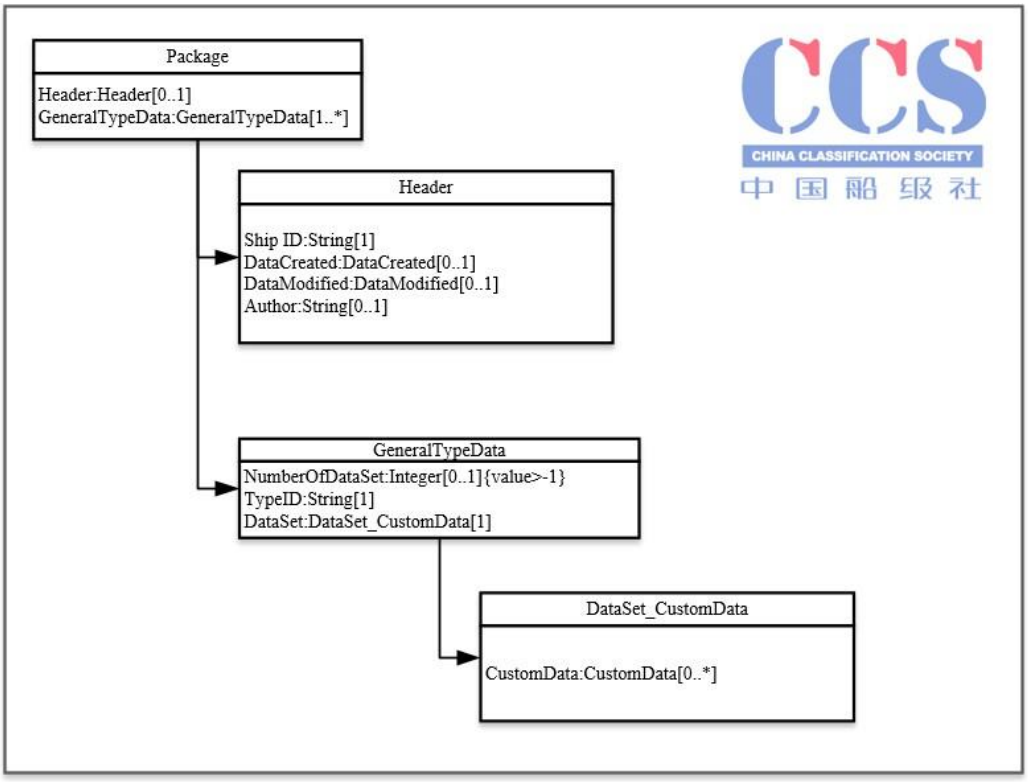

Figure 2.4.5.2 Logic Diagram of General Type Data

2.4.5.3 The details of each element are described below.

(1) Package structure

Package Structure **Table 2.4.5.3 (1)**

Name	Data type	Remarks	Mandatory /optional	Maximum
Header	2) Header	See (2)	Optional	1
GeneralTypeData	3) GeneralTypeData	See (3)	Mandatory	*

(2) Header structure

Header Structure **Table 2.4.5.3 (2)**

Name	Data type	Remarks	Mandatory /optional	Maximum
ShipId	String	IMO number, CMSA ID, etc.	Mandatory	1
DataCreated	DateTime	Package creation date	Optional	1
DataModified	DateTime	Package modification date	Optional	1
Author	String	Data owner	Optional	1

(3) GeneralTypeData structure

GeneralTypeData Structure **Table 2.4.5.3 (3)**

Name	Data type	Remarks	Mandatory/optional	Maximum
NumberOfDataSet	NonNegativeInteger	Number of datasets in a tabular data element	Optional	1
TypeID	String	An array of type identifications of generic data in 2.2.2	Mandatory	1
DataSet	(4) DataSet_CustomData	See 2.4.5.1	Mandatory	*

(4) DataSet_CustomData structure

Dataset_CustomData Structure **Table 2.4.5.3 (4)**

Name	Data type	Remarks	Mandatory/optional	Maximum
CustomData	CustomData	The data in user-defined format is organized according to a custom metadata structure in Section 2.2.3.2 (3)	Mandatory	*

Note: * means that the maximum number is unlimited.

2.4.6 Channel list structure of time series data

2.4.6.1 Data model

The data channel list shall contain the following five elements:

(1) Package

A package is a data packet consisting of a header and a data channel (data subject).

(2) Header

The header indicates when and by whom the generic data channel list is created.

(3) DataChannel

A data channel consists of the attribute and the data channel ID.

(4) DataChannelID

The data channel ID is the data channel identifier specified in 2.3.2 of Chapter 2 of the Guidelines.

(5) Property

The property is used to define the attributes of the data channel specified in 2.3.3 of Chapter 2 of the Guidelines.

These elements are arranged in a hierarchical structure as shown in Figure 2.4.6.1.

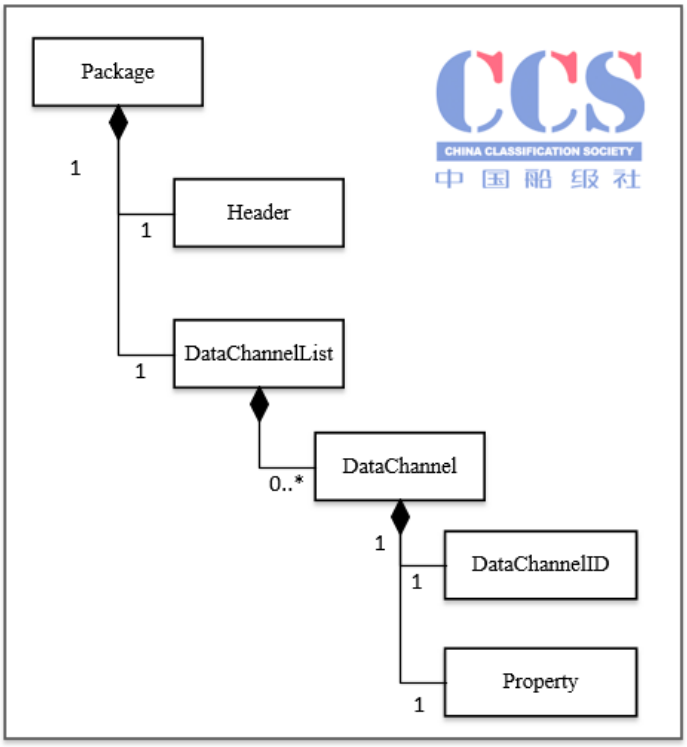

Figure 2.4.6.1 List Model of Data Channel

2.4.6.2 Logical structure

(1) The logical structure of the data channel list is as shown in Figure 2.4.6.2.

(2) The data channel list is embodied as a data packet, and the packet elements are composed of the header and the elements of data channel list, which include one or more data channel elements with ID(s) and attribute(s).

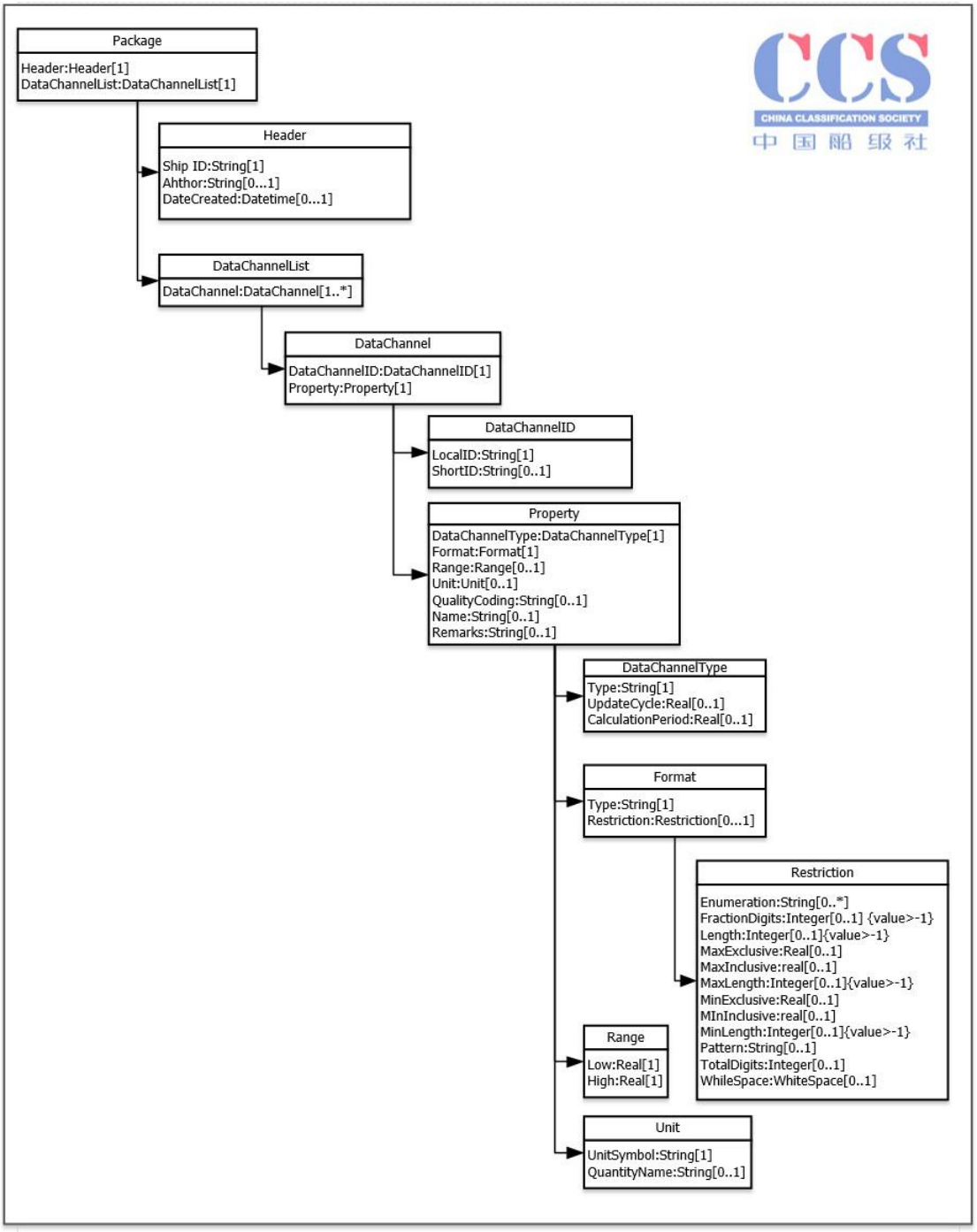

Figure 2.4.6.2 Logic Diagram of Data Channel List

2.4.6.3 The details of each element are described below.

(1) Package structure

Package structure **Table 2.4.6.3 (1)**

Name	Data type	Remarks	Mandatory/ optional	Maximum
Header	(2) Header	See (2)	Mandatory	1

DataChannelList	(3) DataChannelList	See (3)	Mandatory	1

(2) Header structure

Header structure **Table 2.4.6.3 (2)**

Name	Data type	Remarks	Mandatory/ optional	Maximum
ShipId	String	Ship identification	Mandatory	1
Author	String	Owner of data channel list	Optional	1
DataCreated	Datetime	Package creation date	Optional	1

(3) DataChannelList structure

DataChannelList structure **Table 2.4.6.3 (3)**

Name	Data type	Remarks	Mandatory/ optional	Maximum
DataChannel	(4) DataChannel	See (4)	Mandatory	*

(4) DataChannel structure

DataChannel structure **Table 2.4.6.3 (4)**

Name	Data type	Remarks	Mandatory/ optional	Maximum
DataChannelID	(5) DataChannelID	See (5)	Mandatory	1
Property	(6) Property	See (6)	Mandatory	1

(5) DataChannelID structure

DataChannelID structure **Table 2.4.6.3 (5)**

Name	Data type	Remarks	Mandatory/ optional	Maximum
LocalID	String	See 2.3.2.2 of Chapter 2	Mandatory	1
ShortID	String	See 2.3.2.3 of Chapter 2	Optional	1

(6) Property structure

Property structure **Table2.4.6.3 (6)**

Name	Data type	Remarks	Mandatory/ optional	Maximum
DataChannelType	(7) DataChannelType	See (7) and 2.3.3.2 (1) of Chapter 2	Mandatory	1
Format	(8) Format	See (8) and 2.3.3.2 (2) of Chapter 2	Mandatory	1
Range	(10) Range	See (10) and 2.3.3.2 (3) of Chapter 2	Mandatory*	1

Unit	(11) Unit	See (11) and 2.3.3.2 (4) of Chapter 2	Mandatory*	1
QualityCoding	String	See 2.3.3.2 (5) of Chapter 2	Optional	1
Name	String	See 2.3.3.2 (6) of Chapter 2	Optional	1
Remarks	String	See 2.3.3.2 (7) of Chapter 2	Optional	1
* Range and unit are only required if the data channel type is "decimal".				

(7) DataChannelType structure

DataChannelType structure **Table 2.4.6.3 (7)**

Name	Data type	Remarks	Mandatory/ optional	Maximum
Type	String	Described according to the type format of data channel in 2.3.3.2 (1) of Chapter 2	Mandatory	1
UpdateCycle	Real		Optional	1
CalculationPeriod	Real		Optional	1

(8) Format structure

Format structure **Table 2.4.6.3 (8)**

Name	Data type	Remarks	Mandatory/ optional	Maximum
Type	String	Described according to the type format of data channel in 2.3.3.2 (2) of Chapter 2	Mandatory	1
Restriction	(9) Restriction		Optional	1

(9) Restriction structure

Restriction structure **Table2.4.6.3 (9)**

Name	Data type	Remarks	Mandatory/ optional	Maximum
Enumeration	String	Described according to the type format of data channel in 2.3.3.2 (2) of Chapter 2	Optional	*
FractionDigits	Integer		Optional	1
Length	Integer		Optional	1
MaxExclusive	Real		Optional	1
MaxInclusive	Real		Optional	1
MaxLength	Integer		Optional	1
MinExclusive	Real		Optional	1
Inclusive	Real		Optional	1
MinLength	Integer		Optional	1
Pattern	String		Optional	1
TotalDigits	Integer		Optional	1
WhiteSpace	One of the following: "Preserve"		Optional	1

	"Replace" "Collapse"			

(10) Range structure

Range structure **Table 2.4.6.3 (10)**

Name	Data type	Remarks	Mandatory/ optional	Maximum
Low	Real	Lower limit of measuring range	Optional	1
High	Real	Upper limit of measuring range	Optional	1

(11) Unit structure

Unit structure **Table 2.4.6.3 (11)**

Name	Data type	Remarks	Mandatory/ optional	Maximum
UnitSymbol	String	See ISO 80000 or 2.3.3.2 (4) of Chapter 2	Mandatory	1
QuantityName	String	The quantity name is defined in ISO 80000	Optional	1

2.4.7 Structure of time series data

2.4.7.1 Data classification

(1) Classification of time series data

① Time series data is a collection of measured values, where all values have a measurement time. Usually, the data is arranged and recorded in chronological order.

② In practical applications, for effective data transmission, two different representations methods of time series data, tabular data and event data, are specified, and they are used reasonably, depending on the update interval of the measured value.

(2) Tabular data

① Tabular data is a vector of a fixed number of values reported at regular intervals.

② The interval is specified in the data channel list.

Example:

——Multiple raw values sampled from sensor/transmitter at the same time

——Results calculated on a regular basis (e.g.: time average, standard deviation, etc.)

③ The measured values are grouped by timestamp, and the tabular data form is as shown in Table 2.4.7.1 (1).

Data Example **Table 2.4.7.1 (1)**

Timestamp	Data channel l	Data channel 2	Data channel 3	Data channel 4

2022-01-01T00:00:00Z	101.2	0.30	10.2	Close
2022-01-01T00:00:01Z	0.0	0.30	10.2	Close
2022-01-01T00:00:02Z	110.9	0.32	10.2	Open

(3) Event data

① Event data is a set of data that is not fixed at a specific time.

Example:

——Alarm information

——Status information

——Manual input information

② The measured value of event data is usually grouped by the timestamp, data channel ID and form in Table 2.4.7.1 (2).

Event Data Example **Table 2.4.7.1 (2)**

Timestamp	Data channel type	Value
2022-01-01T00:00:00Z	Data channel 1	101.2
2022-01-01T00:00:02Z	Data channel 1	110.9
2022-01-01T00:00:02Z	Data channel 2	0.32
2022-01-01T00:00:02Z	Data channel 4	OPEN

(4) Composition of time series data

① Each row of table/event data is called a DataSet.

② Tabular data consists of a set of datasets without a data channel ID, and values can be identified by their chronological order.

③ The tabular data of Table 2.4.7.1 (1) is as follows:

DataSet ("2022-01-01T00:00:00Z", "101.2", "0.30", "10.2", "CLOSE")
DataSet ("2022-01-01T00:00:01Z", "0.0", "0.30", "10.2", "CLOSE")
DataSet ("2022-01-01T00:00:02Z", "110.9", "0.32", "10.2", "OPEN")

④ Event data consists of collections of datasets with data channel IDs since multiple values cannot be fixed at a particular time.

⑤ The event data of Table 2.4.7.1 (2) is as follows:

DataSet ("2022-01-01T00:00:00Z", "Data channel 1", "101.2")
DataSet ("2022-01-01T00:00:02Z", "Data channel 1", "110.9")
DataSet("2022-01-01T00:00:02Z", "Data channel 2", "0.32")
DataSet ("2022-01-01T00:00:02Z", "Data channel 4", "OPEN")

2.4.7.2 Data model

The time series data consists of the following elements:

(1) Package

The package element is composed of a header (metadata) and the time series data (data subject).

(2) Header

① The header element contains the necessary metadata to link the data in the package to the correct channel in the data channel list and the time in the real world.

② It indicates when and by whom the time series data is created, and the measurement period of the time series data.

③ The header metadata may be omitted or is composed of only short ID list if it is provided in advance by any means.

(3) TimeSeriesData

① The element of the time series data is the subject of the package, containing the time series data in Section 3.

② The element also represents a set of datasets, which are grouped by the version of the data channel list. The element should be re-generated as soon as the data channel list is updated, which affects the value of the time series data.

(4) TabularData

① The tabular data elements contain the list of values specified in 2.4.7.1 (2).

② The table represents an ordered list of measured values reported with the same timestamp and update interval.

③ Tabular data should be grouped by the update interval.

(5) EventData

The event data element contains a (usually) irregularly-updated list of values. For each data point, the reference to the data channel list (local ID or short ID) is provided. See 2.4.7.1 (3) for details.

(6) DataSet

A dataset is a set of measured values with the same timestamp, including two types, one for tabular data and one for event data.

These elements are arranged in a hierarchical structure as shown in Figure 2.4.7.2.

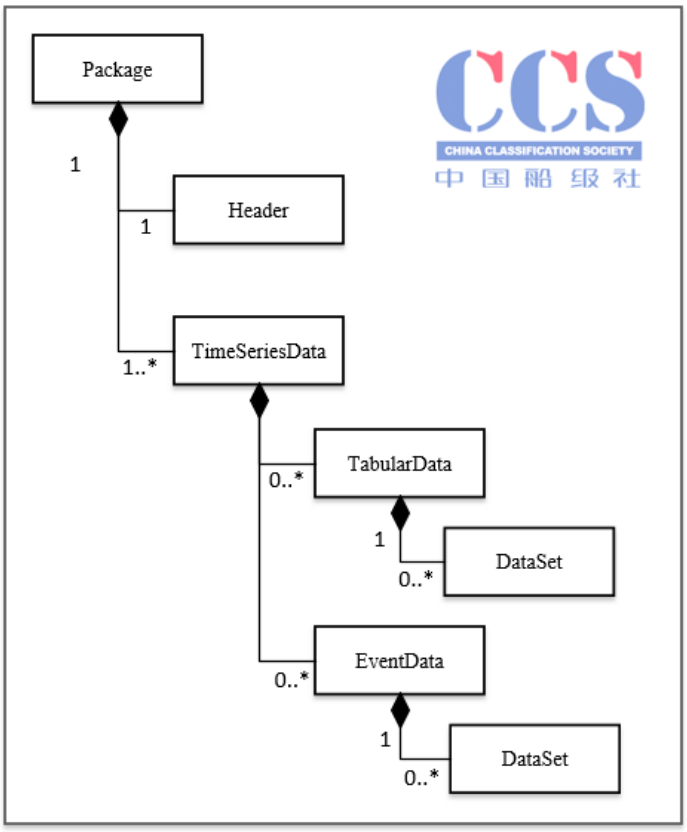

Figure 2.4.7.2 Model of Time series Data

2.4.7.3 Logical structure

① The time series data shall have the logical structure shown in Figure 2.4.7.3.

② Time series data is embodied as a package, which is composed of a header and time series data including one or more data sets, and the data set contains a timestamp and one or more data.

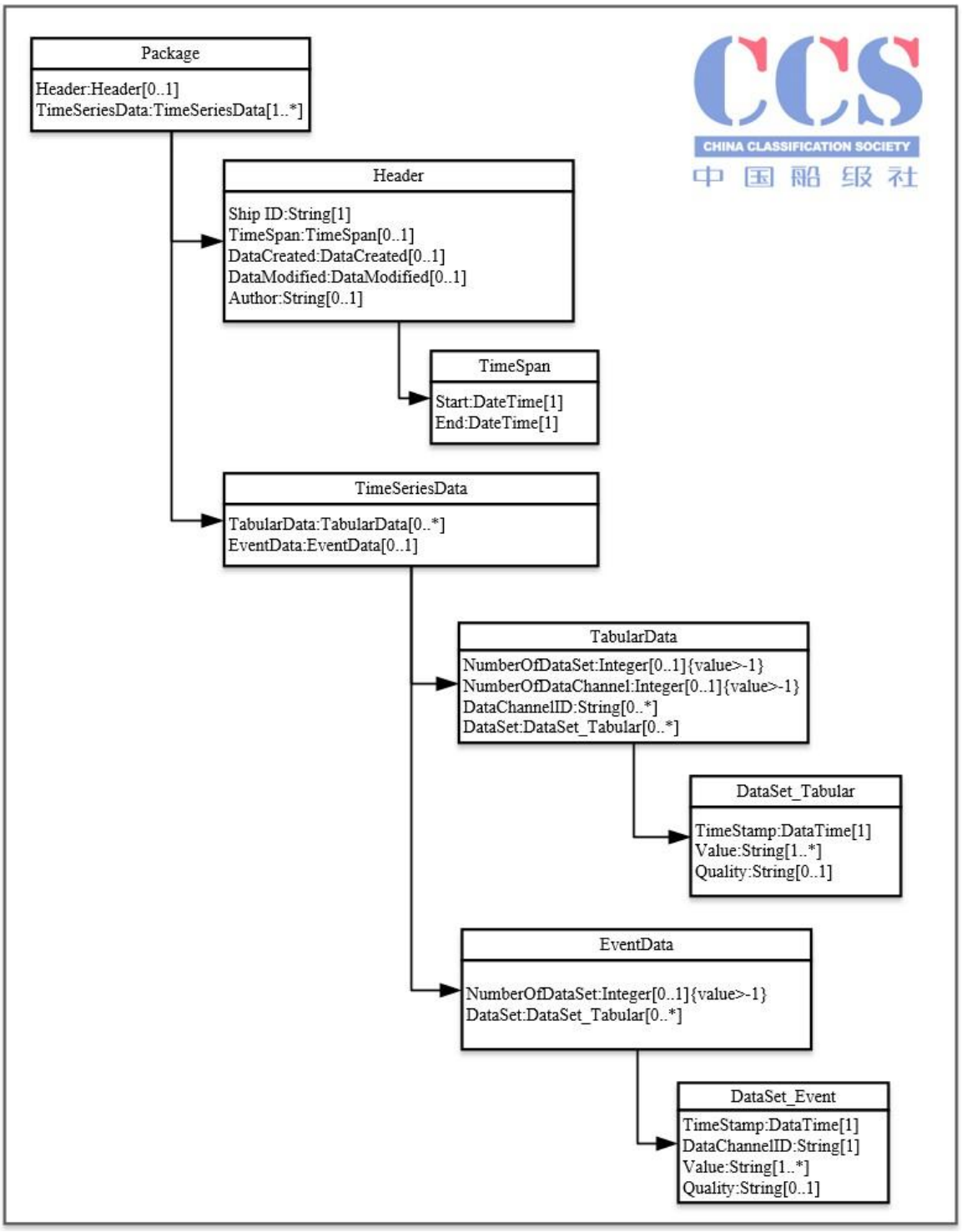

Figure 2.4.7.3 Logic Diagram of Time series Data

2.4.7.4 The details of each element are described below.

(1) Package structure

Package structure **Table 2.4.7.4 (1)**

Name	Data type	Remarks	Mandatory	Maximum

			/optional	
Header	(2) Header	See (2) For periodic data exchange, this element can be omitted to reduce data size	Optional	1
TimeSeriesData	(4) TimeSeriesData	See (4) The time series data is grouped by the version of the data channel list	Mandatory	*

(2) Header structure

Header structure **Table 2.4.7.4 (2)**

Name	Data type	Remarks	Mandatory /optional	Maximum
ShipId	String	IMO number, CMSA ID, etc.	Mandatory	1
TimeSpan	(3) TimeSpan	See (3)	Optional	1
DataCreated	DateTime	Package creation date	Optional	1
DataModified	DateTime	Package modification date	Optional	1
Author	String	Data owner	Optional	1

(3) TimeSpan structure

TimeSpan structure **Table 2.4.7.4 (3)**

Name	Data type	Remarks	Mandatory /optional	Maximum
Start	DateTime	The start timestamp of the dataset	Mandatory	1
End	DateTime	The end timestamp of the dataset	Mandatory	1

(4) TimeSeriesData structure

TimeSeriesData structure **Table 2.4.7.4 (4)**

Name	Data type	Remarks	Mandatory /optional	Maximum
TabularData	(5) TabularData	See (5)	Optional	*
EventData	(6) EventData	See (6)	Optional	1

(5) TabularData structure

TabularData structure **Table 2.4.7.4 (5)**

Name	Data type	Remarks	Mandatory/optional	Maximum
NumberOfDataSet	NonNegativeInteger	Number of datasets in a tabular data element	Optional	1
NumberOfDataChannel	NonNegativeInteger	Number of channels in a tabular data element	Optional	1

DataChannelID	String	An array of data channel IDs in 2.3.2 of Chapter 2 The order of the data channel IDs should be the same as that of (7) tabular data/value. If the order of data channel IDs cannot be specified, the reference number should be added. Furthermore, these elements may be omitted if the order of data channel IDs can be kept changed.	Optional	*
DataSet	(7) DataSet_Tabular	See 2.4.7.1	Optional	*

(6) EventData structure

EventData structure **Table 2.4.7.4 (6)**

Name	Data type	Remarks	Mandatory /optional	Maximum
NumberOfDataSet	NonNegativeInteger	Number of datasets in an event data element	Optional	1
DataSet	(8) DataSet_Event	See 2.4.7.1	Optional	*

(7) DataSet_Tabular structure

DataSet_Tabular structure **Table 2.4.7.4 (7)**

Name	Data type	Remarks	Mandatory /optional	Maximum
Timestamp	DateTime	Measurement time	Mandatory	1
Value	String	Array of measured values at timestamps for each data channel. The order of the value elements should be the same as that of (5) tabular data/ data channel ID. If the order of value elements cannot be specified, the reference number should be added.	Mandatory	*
Quality	String	Data quality of above value	Optional	1

(8) DataSet_Event structure

DataSet_Event structure **Table 2.4.7.4 (8)**

Name	Data type	Remarks	Mandatory /optional	Maximum
Timestamp	DateTime	Measurement time	Mandatory	1

DataChannelID	String	A set of data channel IDs in 2.3.2 of Chapter 2	Mandatory	1
Value	String	Measured value	Mandatory	1
Quality	String	Data quality of above value	Optional	1

CHAPTER 3 DATA EXCHANGE INTERFACE

Section 1 General Provisions

3.1.1 General requirements

3.1.1.1 The purpose of the interface rules stipulated in this Chapter is to guide ship owners or shipping companies to specify interface services in accordance with the rules and provide ISC with data required for digital surveys.

3.1.1.2 The data transmission format shall comply with the data identification and structure standards specified in Chapter 2 of the Guidelines.

3.1.1.3 The interface service specified by the ship owner or ship management company shall support the log monitoring, and the log content shall at least cover all complete historical records within the last 30 days.

3.1.1.4 If data transmission fails, the interface service shall support the retransmission after the troubleshooting.

3.1.1.5 The clock of the server where the interface service is arranged shall be synchronized with UTC.

Section 2 Interface Service

3.2.1 General requirements

3.2.1.1 The Guidelines specify interface format specifications suitable for data exchange of ship survey.

3.2.1.2 The interface service should meet the test verification requirements in Section 2 of Chapter 4.

3.2.2 Interface standards

3.2.2.1 Interface programming should be subject to the principles of REST architecture.

3.2.2.2 The frame structure of interface data exchange is as shown in Figure 3.2.2.2, and the specific requirements are as follows.

(1) The data exchange rules between the ship and the enterprise's ship digital platform are defined by the enterprise.

(2) The enterprise's ship data platform shall support the ship survey data sharing or submit data to ISC in accordance with the rules of the Guidelines.

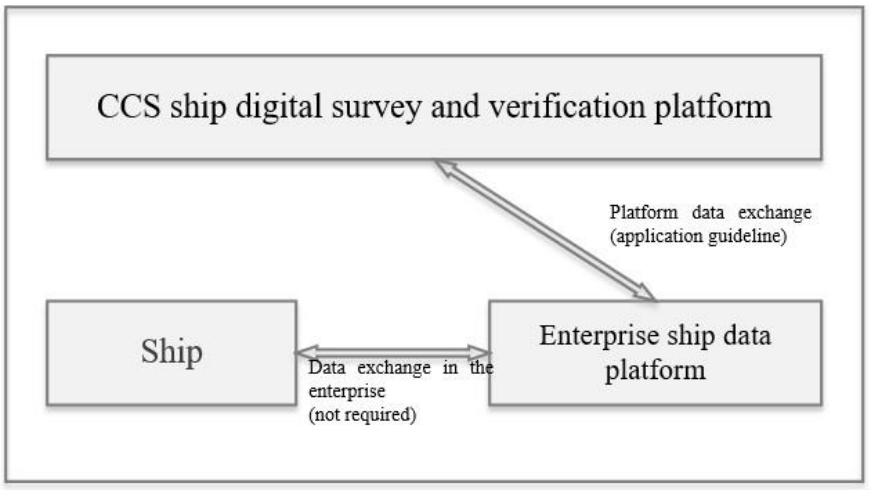

Figure 3.2.2.2 Framework of Interface Data Exchange

3.2.2.3 Call process

(1) The interface supports two modes of push and call, as shown in Figure 3.2.2.3. The request processing is achieved with the Http/Https protocol. The Http status code will be returned if the request fails, otherwise, the call result will be obtained.

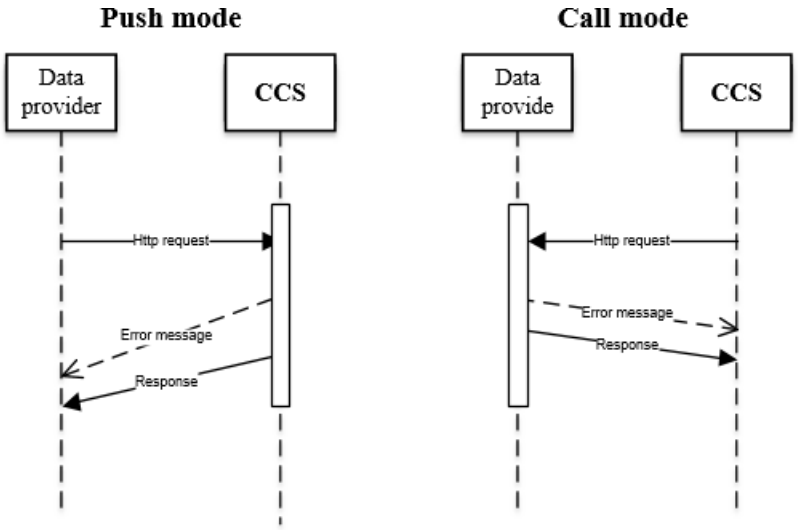

Figure 3.2.2.3 Example of Request-response Call Process

3.2.3 Interface format

3.2.3.1 Request-response protocol structure

(1) The structure of the request-response protocol is as shown in Table 3.2.3.1.

(2) Data channel type options, query options, and label options should be used as required by RFC 3986, and the options and parameters should be case-insensitive (letter).

Structure of Request-response Protocol **Table 3.2.3.1**

Method	Service root	Resources path	Type option	Data channel type option	Query option	Label option
See 3.2.3.2	See 3.2.3.3	See 3.2.3.4	See 3.2.3.5	See 3.2.3.6	See 3.2.3.7	See 3.2.3.8

3.2.3.2 Methods

(1) The method used for the request-response protocol is as shown in Table 3.2.3.2.

(2) When receiving the request, the interface issuer shall record the Http status code, Http method type, time and date, and the result in the log.

Method of Request-response Protocol **Table 3.2.3.2**

Method	Description
GET	The GET method is to retrieve the specified URL resource from the target server. It is only applicable when the data provider server obtains the list of the time series data channel and the time series data.
POST	The POST method is to add the data of the message subject to the target server, without specifying resources path.
PUT	The PUT method is to update the data of the message subject to the target server, without specifying resources path.
TRACE	The TRACE method is to acquire the counts of URL resources specified by the target server. It is only applicable to resource counts of time series data. The elements of the counting information are as follows: type=ts_data:[Time Series].[Local ID] type=ts_sdd:[Local ID].[Data Channel Type].[History]

3.2.3.3 Service root

(1) Service root format and details are as shown in Table 3.2.3.3.

Details of Service Root **Table 3.2.3.3**

Service root	Description	Mandatory/optional	Example
<Host>	IP address or domain name of interface server	Mandatory	192.168.1.253
<Port>	HTTP port number	Optional	8080

3.2.3.4 Resources path

(1) The resources path is only applicable to GET and TRACE methods.

(2) Special requirements for time series data transmission are as follows:

① When the data channel type is "Local ID (LocalID)", the resources path format and details are as shown in Table 3.2.3.4.

② When the data channel type is "Short ID (ShortID)", a short ID from the data channel list is specified. Multiple short IDs shall not be specified by separating the short IDs with "/" or ",".

③ When the data channel type is "Local ID (LocalID)", and the GET method is used, the wildcard character can be used in the resources path. For more information on the wildcard, see 3.2.3.9.

Details of Resources Path **Table 3.2.3.4**

Resources path	Description	Mandatory/optional	Example

<Ship ID>	Ship identification	Optional	imo1234567
<Naming Rule>	See 2.3.2 of Chapter 2. The version number needs to be specified based on emd	Optional	/emd /v1.0
<Local Data Name>	Only available if type option is 'ts_data' and 'ts_sdd'.	Optional	Time series data: P8.003.006+1/Cylinder.Exhaust_Gas+2/

3.2.3.5 Type option

(1) In Table 3.2.3.5, the data types that can be processed and the setting method are listed.

(2) The type option is specified in the Http header and can also be used for methods other than the GET method.

(3) The type option defaults to the "ts_data" type.

Details of Type Options **Table 3.2.3.5**

Data type	Description	Number of appearances	Example
ts_data	To process time series data	1	type:ts_data
ts_sdd	To process data channel list	1	type:ts_sdd
ge_data	Generic type data	1	type:ge_data
ge_sdd	List of generic data type	1	type:ge_sdd

3.2.3.6 Data channel type

(1) The ID type of resources in the shipboard data server to be accessed is specified, which is only applicable to the acquisition of channel data related to time series data.

(2) In Table 3.2.3.6, the data channel types that can be processed and the specifying method are listed. Data channel type option defaults to “local ID”.

(3) Data channel type option is specified in the message subject.

Details of Data Channel Type Options **Table 3.2.3.6**

Data type	Description	Number of appearances	Example
Local ID	The local ID is used to access resources on the shipboard data server.	1	idtype=LocalID
Short ID	The short ID is used to access resources on the shipboard data server.	1	idtype=ShortID

3.2.3.7 Query option

(1) The query option is used to specify the data format, change the sorting order and method, and for filtering.

(2) Table 3.2.3.7 lists the available query option and the specifying method.

(3) Multiple query options can be specified at one time.

(4) When date and time are used for option parameters, ISO 8601 format shall be used.

(5) Query option is specified in the message subject.

Details of Query Options **Table 3.2.3.7**

Option name	Description	Applicable object	Default	Example
?offset	To specify the UTC time and date for fetching data.	Time series data; Generic data	UTC when the server receives the "method"	?offset=2
?before	To specify whether to search before or after the reference time and date (before: true, after: false). Only available if type option is 'data'.	Time series data	TRUE	?before=FALSE
?implemented	To specify the format adopted by the data entity: XML, JSON, CSV, and JSON (default)	Time series data; Generic data	JSON	?implemented=JSON
?limit	To specify the time frame obtained on the second timescale when the type is 'ts_data'. To specify the number of items in the modification history of data channel list to be fetched when the type is 'ts_sdd'. When 1 is specified, the latest information is returned. To return the latest data channel list when the type is "ts_sdd". To return the latest data source information when the type is "siod".	Time series data	1	?limit=5
?revisionfrom	To specify the date and time when the revision history of data channel list is fetched. The revisionfrom option is available when the type option is "ts_sdd".	Time series data		?revisionfrom= 2016-03-31T00:00:00Z
?revisionto	To specify the date and time when the revision history of data channel list is fetched. It can be used simultaneously with revisionfrom. The revisionto option is available when the type option is "ts_sdd".	Time series data		?revisionto=2015-04-01T 23:59:59Z
?orderby	To specify the ascending or descending chronological order for	Time series data	asc	? orderby= ASC

	the obtained data (asc: ascending; desc: descending)			

3.2.3.8 Label option

(1) The label option is only applicable to the interface calling method of the time series data.

(2) When the GET method is used to obtain time series data, the priority of the output format of column title shall be specified.

(3) Otherwise, short ID and local ID shall be used in order.

(4) Table 3.2.3.8 lists the available label option and the specifying method.

(5) The label option is specified in the message subject.

Details of Label Options **Table 3.2.3.8**

Option type	Description	Number of appearances	Example
Short ID	The ShortID with the highest priority is used. The column header is used together with the priority of ShortID and LocalID.	1	?label=ShortID
Local ID	The LocalID with the highest priority is used.	1	?label= LocalID

3.2.3.9 Wildcard in resources path

(1) It is only applicable to the interface GET method of the time series data.

(2) For the structure of the resources path, refer to the local ID in 2.3.2, Section 3, Chapter 2.

(3) When the time series data, data channel list and data source information stored in the shipboard data server are obtained, wildcards in the resources path shall be used to create a filter.

(4) Wildcard string:

#: Multi-level wildcard: It is used to match any quantity level in a URI.

+: Single-level wildcard: It is used to match only one URI level.

3.2.3.10 Return code of method

(1) Data transmission processing results and Http status codes (see Table 3.2.3.10).

Http status code **Table3.2.3.10**

Status code	Message	Description
200	Completed	
201	Created	Return the URL of the newly-generated resource.
400	Bad request	Syntax error
401	Uncertificated	User confirmation error
403	Prohibited	Access unauthorized directories and files
404	Not found	Data unavailable

405	Method not allowed (client error)	
408	Request time-out	Failure to generate request within a specific time
413	Request entity too large	Accept unexpected requests
500	Internal server error	Execute unauthorized methods

3.2.4 Interface example

3.2.4.1 Example of tabular data

Data Example **Table 3.2.4.1**

Timestamp	Ship identification	Local ID	Value
2022/01/25 06:43:00	IMO1234567	/emd/v1.0/P4.006.003+1//DrivingEnd/Speed	268
2022/01/25 06:43:00	IMO1234567	/emd/v1.0/P4.006.003+2//DrivingEnd/Speed	269
2022/01/25 06:43:00	IMO1234567	/emd/v1.0/P4.006.003+3//DrivingEnd/Speed	255
2022/01/25 06:43:00	IMO1234567	/emd/v1.0/P4.006.003+1//DrivingEnd/Temperature	80
2022/01/25 06:43:00	IMO1234567	/emd/v1.0/P4.006.003+2//DrivingEnd/Temperature	81
2022/01/25 06:43:00	IMO1234567	/emd/v1.0/P4.006.003+3//DrivingEnd/Temperature	83
2022/01/25 06:45:00	IMO1234567	/emd/v1.0/P4.006.003+1//DrivingEnd/Speed	270
2022/01/25 06:45:00	IMO1234567	/emd/v1.0/P4.006.003+2//DrivingEnd/Speed	271
2022/01/25 06:45:00	IMO1234567	/emd/v1.0/P4.006.003+3//DrivingEnd/Speed	280
2022/01/25 06:45:00	IMO1234567	/emd/v1.0/P4.006.003+1//DrivingEnd/Temperature	82
2022/01/25 06:45:00	IMO1234567	/emd/v1.0/P4.006.003+2//DrivingEnd/Temperature	83
2022/01/25 06:45:00	IMO1234567	/emd/v1.0/P4.006.003+3//DrivingEnd/Temperature	84
2022/01/25 06:47:00	IMO1234567	/emd/v1.0/P4.006.003+1//DrivingEnd/Speed	270
2022/01/25 06:47:00	IMO1234567	/emd/v1.0/P4.006.003+2//DrivingEnd/Speed	271
2022/01/25 06:47:00	IMO1234567	/emd/v1.0/P4.006.003+3//DrivingEnd/Speed	280
2022/01/25 06:47:00	IMO1234567	/emd/v1.0/P4.006.003+1//DrivingEnd/Temperature	82
2022/01/25 06:47:00	IMO1234567	/emd/v1.0/P4.006.003+2//DrivingEnd/Temperature	83
2022/01/25 06:47:00	IMO1234567	/emd/v1.0/P4.006.003+3//DrivingEnd/Temperature	84

3.2.4.2 POST method

POST http://localhost/?implemented=JSON

{Data format}

(1) See Appendix 5 for the data format.

(2) The processing result is returned in the Http request, and the return value is as shown in Table 3.2.3.10 Http Status Code.

3.2.4.3 GET method

(1) The GET method is used for the actual record data in Table 3.2.4.1.

GET http://localhost/IMO1234567/+/+/P4.006.003/+/+?offset=2022-01-25T06:45:03Z&before=true&limit=3

(2) Search criteria:

IMO1234567 ship local ID contains "P4.006.003" (6:45:03 on Nov. 25, 2022) or earlier date and time, and data within past 3 seconds is searched.

(3) The search results are as follows:

Timestamp	Ship identification	Local ID	Value
2022/01/25 06:43:00	IMO1234567	/emd/v1.0/P4.006.003+1//DrivingEnd/Speed	268
2022/01/25 06:43:00	IMO1234567	/emd/v1.0/P4.006.003+2//DrivingEnd/Speed	269
2022/01/25 06:43:00	IMO1234567	/emd/v1.0/P4.006.003+3//DrivingEnd/Speed	255
2022/01/25 06:43:00	IMO1234567	/emd/v1.0/P4.006.003+1//DrivingEnd/Temperature	80
2022/01/25 06:43:00	IMO1234567	/emd/v1.0/P4.006.003+2//DrivingEnd/Temperature	81
2022/01/25 06:43:00	IMO1234567	/emd/v1.0/P4.006.003+3//DrivingEnd/Temperature	83
2022/01/25 06:45:00	IMO1234567	/emd/v1.0/P4.006.003+1//DrivingEnd/Speed	270
2022/01/25 06:45:00	IMO1234567	/emd/v1.0/P4.006.003+2//DrivingEnd/Speed	271
2022/01/25 06:45:00	IMO1234567	/emd/v1.0/P4.006.003+3//DrivingEnd/Speed	280
2022/01/25 06:45:00	IMO1234567	/emd/v1.0/P4.006.003+1//DrivingEnd/Temperature	82
2022/01/25 06:45:00	IMO1234567	/emd/v1.0/P4.006.003+2//DrivingEnd/Temperature	83
2022/01/25 06:45:00	IMO1234567	/emd/v1.0/P4.006.003+3//DrivingEnd/Temperature	84

3.2.4.4 PUT method

PUT http://localhost/

{Data format}

(1) See Appendix 5 for the data format. It is to update the actual record data corresponding to the timestamp and the local ID of the data set structure.

(2) The processing result is returned in the Http request, and the return value is as shown in Table 3.2.3.10 Http Status Code.

3.2.4.5 TRACE method

(1) The TRACE method is used for the actual record data in Table 3.2.4.1.

TRACE http://localhost/IMO1234567/+/+/ExhaustGas/#?offset=2022-01-25T06:43:02Z&before=true

(2) Search criteria:

IMO1234567 ship local ID contains "P4.006.003" (6:43:02 on Jan. 25, 2022) or earlier date and time, and the direction earlier than the time node is selected.

(3) The return message of normal end:

{Result: 6}

Section 3 Interface and Data Security

3.3.1 General requirements

3.3.1.1 The appropriate authentication method and the authorization function of the interface data range shall be adopted for the interface service.

3.3.1.2 For the interface service, the data transmission shall be achieved with the SSL protocol, to provide security support.

3.3.1.3 In addition, the security of interface service shall meet the relevant requirements of 3.3.3 Communication Security in the Guidelines for Ship Network System Requirements and Security Assessment of ISC.

3.3.1.4 It is recommended that ship owner or ship management company apply blockchain technology in data exchange to realize the data right confirmation, the tamper-proofing, and non-repudiation.

3.3.1.5 The ownership of the data submitted to ISC through data exchange belongs to the data authorizer, and it is only used for the ship digital survey by ISC.

CHAPTER 4 DATA EXCHANGE IMPLEMENTATION AND TEST VERIFICATION

Section 1 Data Exchange Implementation

4.1.1 General requirements

4.1.1.1 The ship owner or ship management company implementing ship digital survey shall make an implementation plan in accordance with the requirements of Chapter 2 of the Guidelines for the Application of Ship Digital Survey of ISC.

4.1.1.2 The survey data exchange process shall be identified according to the relevant requirements of the Guidelines, and the data structure should be organized.

4.1.1.3 Before the data exchange, it is necessary to clarify with ISC that either the push or the call mode is adopted for the data exchange interface.

4.1.1.4 Before the data exchange, it is necessary to clarify the ownership and purpose of the data to be exchanged with ISC.

4.1.2 Implementation steps

4.1.2.1 Before the exchange of survey data, the ship owner or ship management company shall prepare the data required for digital survey according to the implementation plan.

4.1.2.2 The ship owner or ship management company (or entrusted supplier service organization) shall provide data exchange interface services and conduct the test verification.

4.1.2.3 After the verification and approval by ISC, data exchange shall be carried out according to the implementation plan.

Section 2 Test Verification

4.2.1 General requirements

4.2.1.1 The interface issuer shall test and verify the data exchange interface module, and record the methods and results in the test verification report.

4.2.1.2 The test verification report shall cover the test verification items, methods, judgment criteria and results.

4.2.1.3 Test verification is applicable to ships applying for the class notation of digital survey.

4.2.1.4 ISC will evaluate and match the interface with reference to the test verification report from the interface issuer to ensure that its functional status is monitored normally and the data required for digital survey is submitted to ISC.

4.2.2 Test verification items

4.2.2.1 System clock

(1) The internal clock of the data server system of the interface issuer supports the function of synchronizing with UTC.

(2) Regardless of whether the data server is synchronized with UTC or not, it is necessary to confirm that the error between the server system clock and UTC is 1 second/hour or less.

(3) If it is confirmed that the shipboard data server is not synchronized with UTC and abnormality occurs, the system clock should support the alarm function.

4.2.2.2 Data format

The interface document is checked to confirm that the data format processed by the data exchange interface complies with the requirements in Section 4, Chapter 2 of the Guidelines.

4.2.2.3 Data transmission service

(1) Confirm the data requested by the interface data transmission service visually.

(2) Confirm that the data sent in the data transmission service is processed correctly according to the method visually.

(3) Confirm that all methods that may be employed by the data transmission service are tested.

4.2.2.4 Log management

(1) Confirm that the interface service supports the system log function, all system log records are saved and available for reference within at least 30 days through analysis and evaluation, and the following matters can be recorded in the log.

① The access control log of the request source can be used to record the audit pass or failure log.

② The interface method and parameters requested by the request source, the call success and failure logs can be recorded.

③ The time consumed by interface calling can be recorded.

APPENDIX 1 ISC SHIP DIGITAL MODEL

1 General requirements

1.1 In the ISC ship digital model, the hierarchical division is carried out for the digital classification of ships, the code and name are assigned to each item, and the coding is based on the universal decimal classification (UDC) coding system. For the detailed description, tools and interfaces of the model, visit http://digitalship.ISC.org.cn/emd/.

2 Primary classification of ship digital model

Primary Classification of Standard Model **Table 1**

Code	Name
P1	Document
P2	Hull Structure
P3	Deck Equipment
P4	Main Propulsion System
P5	Steering System
P6	Boiler and Associated Equipment
P7	Pollution Prevention System
P8	Electric Power System
P9	Navigation aid Equipment
P10	Radio Communications Equipment
P11	Lifesaving and Fire Fighting
P12	Dynamic Positioning System
P13	Lifting Appliance
P14	Energy Efficiency Management System
OTH	Other

Note: The classes not listed in the above table can be used after the assessment and confirmation by ISC.

APPENDIX 2 CODE OF LOCATIONS

1 The location means the position on board. The following 3 locations are specified:

1.1 Absolute locations: The position relative to the fixed coordinate system on the ship.

1.2 Generic locations: An item describing a typical position, usually on common ships.

1.3 Relative locations: The relative position of individuals relative to each other within a set of components.

2 The emd is used for naming the sensor (usually the location code; Table 2).

Locations **Table 2**

Types	Code	Description
Absolute location	Forward	Front part of the ship
	Bow	Bow of the ship
	Port	Left side of the ship towards the ship's heading
	Starboard	Right side of the ship towards the ship's heading
	ForwardPort	Left front side of the ship towards the ship's heading
	ForwardStarboard	Right front side of the ship towards the ship's heading
	AftPort	Left rear side of the ship towards the ship's heading
	AftStarboard	Right rear side of the ship towards the ship's heading
	Center	Center part between ship's port side and starbord side
	Middle	Center of the ship in the longitudinal direction
	MiddlePort	The port side of center the longitudinal direction of the ship
	MiddleStarboard	The starboard side of the center of the ship in the longitudinal direction
	Aft	Rear part of the ship
	n_Platform	Number Platform
	Deck_n	Dock Number
	Fr_n	Frame Number
	Stern	Stern of the ship
Common location	PhaseR	First phase, R phase, U phase in a three-phase AC system
	PhaseS	Second phase S phase, V phase in a three-phase AC system
	PhaseT	Third phase, T phase, W phase in a three-phase AC system
	Pipe	Pipeline
Relative location	Upper	Top part of the equipment
	Lower	Bottom part of the equipment

	Inlet	Inlet
	Outlet	Outlet
	Open	Open position
	Close	Closed position
	DrivingEnd	Near side of prime mover/motor
	FreeEnd	Far side of prime mover/motor

Note: The classes not listed in the above table can be used after the assessment and confirmation1 by ISC.

APPENDIX 3 SIGNAL TYPES OF UNIVERSAL SENSORS

The commonly-used signal types of sensors are as shown in Table 3.

Sensor Signal Type **Table 3**

No.	Signal code	Name
1	Pressure	Pressure
2	Temperature	Temperature
3	Speed	Speed
4	Flow	Flow
5	Amplitude	Amplitude
6	Power	Power
7	Current	Current
8	Frequency	Frequency
9	Angle	Angle
10	Voltage	Voltage
11	Latitude	Latitude
12	Longitude	Longitude
13	Failure	Failure
14	Alarm	Alarm
15	Abnormal	Abnormal
16	Length	Length
17	Direction	Direction
18	Aperture	Aperture
19	Concentration	Concentration
20	Level	Level
21	Distance	Distance
22	Torque	Torque
23	Resistance	Resistance
24	On-Off	On-Off Status
25	Height	Height
26	Stress	Stress
27	Strain	Strain

APPENDIX 4 EXAMPLE FOR GENERIC DATA METADATA

1 Example for metadata of file transfer object

1.1 XML format:

```
<Documents>
    <Row>
        <DocumentKey> Unique document identification </DocumentKey>
        <DocumentName/> Document name </DocumentName>
        <DocumentSize/> Document size </DocumentSize>
        <ValidDate/> Valid date </ValidDate>
        <DocumentBody/> Document content Base64 coding </DocumentBody>
        <Remark/> Remark </Remark>
    </Row>
</Documents>
```

1.2 JSON format:

```
{
  "Documents": [
    {
      "DocumentKey": "Unique document identification",
      “DocumentName": "Document name",
      “DocumentSize": "Document size",
      “ValidDate": "Valid date",
      "DocumentBody": "Document content base64 coding",
      “Remark": "Remark"
    }
  ]
}
```

2 Example for ship operation experiment and maintenance

2.1 XML format

```
<Definition>
    <DataObjectCode> Data object code </DataObjectCode>
    <DataObject> Data object </DataObject>
    <OperationProjects>
        <Row>
            <OperationProjectCode> Operation project code </OperationProjectCode>
            <OperationProject> Operation project </OperationProject>
            <EquipmentNo> Equipment No. </EquipmentNo>
            <OperationProjectData>
                <Row>
                    <OperationType> Operation type </OperationType>
                    <OperationResult> Operation result </OperationResult>
                    <No> Number </No>
```

```
                    <Temperature> Temperature </Temperature>
                    <Speed> Speed </Speed>
                    <OperationTime> Operation time </OperationTime>
                    <Remarks> Remarks </Remarks>
                    <Enclosures>
                        <Row>
                            <AttachmentID></AttachmentID>
                            <AttachmentName ></AttachmentName>
                            <AttachmentAddress ></AttachmentAddress>
                        </Row>
                    </Enclosures>
                </Row>
            </OperationProjectData>
        </Row>
    </OperationProjects>
</Definition>
```

2.2 JSON format

```
{
    “dataObjectCode": "Data object code",
    "dataObject": "Data object",
    "operationProjects": [{
                “operationProjectCode": "Operation project code",
                “operationProject": "Operation project",
                "equipmentNo": "Equipment No.",
                "operationProjectData": [{
                        "operationType": "Operation type",
                        “operationResult": "Operation result",
                        "no": "Number",
                        “temperature": "Temperature",
                        "speed": "Speed",
                        “operationTime": "Operation time",
                “remarks": “Remarks",
                        "enclosures": [{
                                "attachmentID": "",
                                "attachmentName": "",
                                "attachmentAddress": ""
                        }]
                }]
    }
```

APPENDIX 5 REQUIREMENTS AND IMPLEMENTATION OF TIME SERIES DATA

1 General

1.1 Data exchange

1.1.1 The data of the structure described in Section 4 of Chapter 2 will be implemented with XML, JSON or CSV.

1.1.2 Regardless of the implementation language, the data shall meet the following requirements.

1.1.3 Data should be written in text form and encoded in UTF-8 without the byte order mark (BOM).

1.1.4 It is recommended that XML/JSON be used to exchange the data channel list and time series data (including a single dataset or multiple datasets). When JSON is used and there is no schema validation mechanism, it is recommended to develop the application using an environment capable of communicating with the actual shipboard data server. For particularly large time series data, CSV is recommended, in such case, the list of data channels needs to be shared in advance. It is recommended to use XML/JSON to exchange the list of data channels.

2 XML implementation

2.1 General requirements

2.1.1 This section describes that XML and XML schema are used to implement the data format specified in Section 4, Chapter 2 of the Guidelines.

2.1.2 XML schema is formulated according to the rules described below.

2.1.2.1 The minimum and maximum numbers of occurrences shall be specified.

2.1.2.2 An XML schema shall be created to instruct XML to add namespaces to all elements and attributes.

Example: <nr:NamingRule nr:ID="Naming_Rule"/>

2.1.2.3 Specific elements and attributes of the naming rules shall belong to their own namespace.

2.1.2.4 Any newline or indentation characters in the file shall not be considered as important information.

2.2 Standard & XML schema data type.

The standard data types in Section 2.4.3, Chapter 2 of the Guidelines are replaced by the XML schema data types in Table 2.2.

Correspondence Between Standard Data Types and XML Schema Data Types Table 2.2

Standard data type	XML schema data type	Remarks

Integer	Integer	Integer
PositiveInteger	PositiveInteger	1 or greater (integer)
NotNegativeInteger	NotNegativeInteger	0 or greater (integer)
Real	Decimal	Decimal
DateTime	DateTime	Time and date in ISO 8601
CharacterString	String	Random string
Boolean	Boolean	True value
Null	XML schema <element name="example"type="float" nillable="true"/> XML file <example nil="true"/>	When a null value is specified, the attribute nillable="true" should be defined in the element definition of the XML schema, and the attribute nillable="true" should be assigned in the corresponding element of the XML document

2.3 Data Channel List

2.3.1 Namespace

2.3.1.1 The following namespace shall be added to all elements and attributes in the data channel list.

XML naming: sdd=urn:ISO 19848: Ship_data_definition

2.3.1.2 When the specific elements/attributes of naming rules are used, the duplication should be avoided in the namespace.

Example:

XMLNamespace:nr = urn:ISO19848:Ship_Data_Definition:ISC_EMODEL

2.3.2 XML schema syntax

Data structure in Section 2.4.6T of Chapter 2 should be defined and verified with the following XML schema.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
    xmlns:sdd="urn:ISO19848:SHIP_DATA_DEFINITION:ISC_EMODEL"
attributeFormDefault="unqualified" elementFormDefault="qualified"
targetNamespace="urn:ISO19848:SHIP_DATA_DEFINITION:ISC_EMODEL"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <!--2.4.6.3(1) Package-->
    <xs:element name="Package" type="sdd:Package"/>
    <xs:complexType name="Package">
        <xs:sequence>
            <!--2.4.6.3(2) Header-->
            <xs:element name="Header" type="sdd:Header" minOccurs="1" maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
    <!--2.4.6.3(2) Header-->
    <xs:complexType name="Header">
        <xs:sequence>
            <!--IMO number, uniform ship identification number or other identification code -->
            <xs:element name="ShipID" type="xs:string" minOccurs="1" maxOccurs="1"/>
```

```
                <!-- Author ->
                <xs:element name="Author" type="xs:string" minOccurs="0" maxOccurs="1"/>
                <!--Data creation time-->
                <xs:element name="DateCreated" type="xs:dateTime" minOccurs="0" maxOccurs="1"/>
                <!--Extended definition of a custom header-->
                <xs:any processContents="lax" namespace="##other" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    <!--2.4.6.3 (3) Data channel list-->
    <xs:complexType name="DataChannelList">
        <xs:sequence>
            <!--2.4.6.3 (4) Data channel-->
            <xs:element name="DataChannel" type="sdd:DataChannel" minOccurs="1"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <!--2.4.6.3 (4) Data channel-->
    <xs:complexType name="DataChannel">
        <xs:sequence>
            <!--2.4.6.3 (5) Data channel ID-->
            <xs:element name="DataChannelID" type="sdd:DataChannelID" minOccurs="1"
maxOccurs="1"/>
            <!--2.4.6.3 (6) Property-->
            <xs:element name="Property" type="sdd:Property" minOccurs="1" maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
    <!--2.4.6.3 (5) Data channel ID -->
    <xs:complexType name="DataChannelID">
        <xs:sequence>
            <xs:element name="LocalID" type="xs:string" minOccurs="1" maxOccurs="1"/>
            <xs:element name="ShortID" type="xs:string" minOccurs="0" maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
    <!--2.4.6.3 (6) Property-->
    <xs:complexType name="Property">
        <xs:sequence>
            <!--2.3.3.2 (1) Identification of data channel type, such as the average value, alarm and status-->
            <xs:element name="DataChannelType" type="sdd:DataChannelType" minOccurs="1"
maxOccurs="1"/>
            <!--2.3.3.2 (2) format definition used to describe the data format-->
            <!--Assuming that the data types include the "floating-point ","integer","Boolean","text", and "symbol".-->
            <xs:element name="Format" type="sdd:Format" minOccurs="1" maxOccurs="1"/>
            <!--2.3.3.2 (3) Range of measured value-->
            <xs:element name="Range" type="sdd:Range" minOccurs="0" maxOccurs="1"/>
            <!--2.3.3.2 (4) Unit and quantity of measured values-->
            <xs:element name="Unit" type="sdd:Unit" minOccurs="0" maxOccurs="1" />
            <!--2.3.3.2 (5) Name of the data quality assessment scheme for the measured value-->
            <xs:element name="QualityCoding" type="xs:string" minOccurs="0" maxOccurs="1"/>
            <!--2.3.3.2 (6) Name specified in the control system and other instruments-->
            <xs:element name="Name" type="xs:string" minOccurs="0" maxOccurs="1"/>
            <!--2.3.3.2 (7) Remarks -->
            <xs:element name="Remarks" type="xs:string" minOccurs="0" maxOccurs="1"/>
            <!--Custom extended attributes-->
            <xs:any processContents="lax" namespace="##other" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
```

```
<!--2.4.6.3 (7) Data channel type-->
<!--2.3.3.2 (1) Identification of data channel type, such as the average value, alarm and status-->
<xs:complexType name="DataChannelType">
    <xs:sequence>
        <xs:element name="Type" minOccurs="1" maxOccurs="1">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="Inst"/>
                    <xs:enumeration value="Average"/>
                    <xs:enumeration value="Max"/>
                    <xs:enumeration value="Min"/>
                    <xs:enumeration value="StandardDeviation"/>
                    <xs:enumeration value="Calculated"/>
                    <xs:enumeration value="SetPoint"/>
                    <xs:enumeration value="Output"/>
                    <xs:enumeration value="Alert"/>
                    <xs:enumeration value="Status"/>
                    <xs:enumeration value="ManuallyInput"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="UpdateCycle" type="xs:decimal" minOccurs="0" maxOccurs="1"/>
        <xs:element name="CalculationPeriod" type="xs:decimal" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
</xs:complexType>
<!--2.4.6.3 (8) Format-->
<!--2.3.3.2 (2) Format of measured value-->
<xs:complexType name="Format">
    <xs:sequence>
        <!-—Format type-->
        <xs:element name="Type" minOccurs="1" maxOccurs="1">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="Decimal"/>
                    <xs:enumeration value="Integer"/>
                    <xs:enumeration value="Boolean"/>
                    <xs:enumeration value="String"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <!--2.4.6.3 (9) Value restriction-->
        <xs:element name="Restriction" minOccurs="0" maxOccurs="1">
            <xs:complexType>
                <xs:choice maxOccurs="unbounded">
                    <xs:element name="Enumeration" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
                    <xs:element name="FractionDigits" type="xs:integer" minOccurs="0" maxOccurs="1"/>
                    <xs:element name="Length" type="xs:integer" minOccurs="0" maxOccurs="1"/>
                    <xs:element name="MaxExclusive" type="xs:decimal" minOccurs="0" maxOccurs="1"/>
                    <xs:element name="MaxInclusive" type="xs:decimal" minOccurs="0" maxOccurs="1"/>
                    <xs:element name="MaxLength" type="xs:integer" minOccurs="0" maxOccurs="1"/>
                    <xs:element name="MinExclusive" type="xs:decimal" minOccurs="0" maxOccurs="1"/>
                    <xs:element name="MinInclusive" type="xs:decimal" minOccurs="0" maxOccurs="1"/>
                    <xs:element name="MinLength" type="xs:integer" minOccurs="0" maxOccurs="1"/>
                    <xs:element name="Pattern" type="xs:string" minOccurs="0" maxOccurs="1"/>
                    <xs:element name="TotalDigits" type="xs:integer" minOccurs="0" maxOccurs="1"/>
                    <xs:element name="WhiteSpace" minOccurs="0" maxOccurs="1">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
```

```
                                        <xs:enumeration value="preserve"/>
                                        <xs:enumeration value="replace"/>
                                        <xs:enumeration value="collapse"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                        </xs:choice>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
        <!--2.4.6.3 (10) Range-->
        <!--2.3.3.2 (3) Range of measured value-->
        <xs:complexType name="Range">
            <xs:sequence>
                <!--Upper limit of measured value-->
                <xs:element name="High" type="sdd:emptyOrDecimal" minOccurs="1" maxOccurs="1"/>
                <!--Lower limit of measured value-->
                <xs:element name="Low" type="sdd:emptyOrDecimal" minOccurs="1" maxOccurs="1"/>
            </xs:sequence>
        </xs:complexType>
        <!--2.4.6.3 (11) Unit ->
        <!--2.3.3.2 (4) Unit and quantity of measured values-->
        <xs:complexType name="Unit">
            <xs:sequence>
                <!-Unit symbol-->
                <xs:element name="UnitSymbol" type="xs:string" minOccurs="1" maxOccurs="1"/>
                <!--Quantity name defined in ISO80000 -->
                <xs:element name="QuantityName" type="xs:string" minOccurs="0" maxOccurs="1"/>
                <!--Custom extension points for unit elements, such as "quantity symbol", "scale factor", etc.-->
                <xs:any processContents="lax" namespace="##other" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
        <!--Null value and acceptable decimal-->
        <xs:simpleType name="emptyOrDecimal">
            <xs:union memberTypes="sdd:empty xs:decimal"/>
        </xs:simpleType>
        <xs:simpleType name="empty">
            <xs:restriction base="xs:string">
                <xs:enumeration value=""/>
            </xs:restriction>
        </xs:simpleType>
</xs:schema>
```

2.3.3 XML expression

In this example, the namespace:nr is used as a custom element. Custom elements are written in italics for reference.

```
<?xml version="1.0" encoding="utf-8"?>
<Package
    xmlns="urn:ISO19848:SHIP_DATA_DEFINITION"
    xmlns:nr="urn:ISO19848:SHIP_DATA_DEFINITION:ISC_EMODEL">
    <Header>
        <ShipID>IMO1234567</ShipID>
        <Author>Author1</Author>
```

```
        <DateCreated>2022-12-01T00:00:00+00:00</DateCreated>
        <nr:CustomHeaderElement>Vender specific headers</nr:CustomHeaderElement>
    </Header>
    <DataChannelList>
        <DataChannel>
            <DataChannelID>
                <LocalID>/emd/v1.0/P4.006.003 +1//DrivingEnd/Temperature</LocalID>
                <ShortID>0010</ShortID>
            </DataChannelID>
            <Property>
                <DataChannelType>
                    <Type>Inst</Type>
                    <UpdateCycle>1</UpdateCycle>
                </DataChannelType>
                <Format>
                    <Type>Decimal</Type>
                    <Restriction>
                        <FractionDigits>1</FractionDigits>
                        <MaxInclusive>200.0</MaxInclusive>
                        <MinInclusive>-150.0</MinInclusive>
                    </Restriction>
                </Format>
                <Range>
                    <High>150.0</High>
                    <Low>0.0</Low>
                </Range>
                <Unit>
                    <UnitSymbol>°C</UnitSymbol>
                    <QuantityName>Temperature</QuantityName>
                </Unit>
                <QualityCoding>OPC_QUALITY</QualityCoding>
                <Name>Hot water temperature of No.1 main engine </Name>
                <Remarks> Location: ECR, manufacturer: AAA Company, model: TYPE-AAA </Remarks>
                <nr:CustomPropertyElement>Vender specific Property</nr:CustomPropertyElement>
            </Property>
        </DataChannel>
        <DataChannel>
            <DataChannelID>
                <LocalID>/emd/v1.0/P4.006.003+1//DrivingEnd/SPEED</LocalID>
                <ShortID>0020</ShortID>
            </DataChannelID>
            <Property>
                <DataChannelType>
                    <Type>Average</Type>
                    <UpdateCycle>60</UpdateCycle>
                    <CalculationPeriod>3600</CalculationPeriod>
                </DataChannelType>
                <Format>
                    <Type>Integer</Type>
                    <Restriction>
                        <FractionDigits>1</FractionDigits>
                        <MaxInclusive></MaxInclusive>
                        <MinInclusive></MinInclusive>
                    </Restriction>
                </Format>
                <Range>
                    <High>7000</High>
                    <Low>0</Low>
```

```
                    </Range>
                    <Unit>
                        <UnitSymbol>RPM</UnitSymbol>
                        <QuantityName>Speed</QuantityName>
                    </Unit>
                    <QualityCoding>OPC_QUALITY</QualityCoding>
                    <Name>Supercharger speed of No. 1 main engine </Name>
                    <Remarks> Location: ECR, manufacturer: AAA Company, model: TYPE-AAA </Remarks>
                    <nr:CustomPropertyElement>Vender specific Property</nr:CustomPropertyElement>
                </Property>
            </DataChannel>
        </DataChannelList>
```

2.4 Time Series Data

2.4.1 Namespace

In time series data, the following namespaces shall be added.

XMLNamespace:sdt=urn:ISO19848:Ship_Data_Transport:ISC_EMODEL

2.4.2 XML schema syntax

Data structure in Section 2.4.5T of Chapter 2 should be defined and verified with the following XML schema.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
    xmlns:sdt="urn:ISO19848:SHIP_DATA_TRANSPORT:ISC_EMODEL "
 attributeFormDefault="unqualified" elementFormDefault="qualified"
 targetNamespace="urn:ISO19848:SHIP_DATA_TRANSPORT:ISC_EMODEL"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <!--2.4.7.4 (1) Package-->
    <xs:element name="Package" type="sdt:Package"/>
    <!--2.4.7.4 (1) Package-->
    <xs:complexType name="Package">
        <xs:sequence>
            <!--2.4.7.4 (2) Header-->
            <xs:element name="Header" type="sdt:Header"
 minOccurs="0" maxOccurs="1"/>
            <!--2.4.7.4 (4) Time series data-->
            <xs:element name="TimeSeriesData" type="sdt:TimeSeriesData"
minOccurs="1" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <!--2.4.7.4 (2) Package structure-->
    <xs:complexType name="Header">
        <xs:sequence>
            <!--IMO number, uniform identification number of the ship or other unique number of the ship-->
            <xs:element name="ShipID" type="xs:string"
 minOccurs="1" maxOccurs="1"/>
            <!--2.4.7.4 (3) Time span-->
            <xs:element name="TimeSpan" type="sdt:TimeSpan"
 minOccurs="0" maxOccurs="1"/>
```

```
                <!--Data creation time-->
                <xs:element name="DateCreated" type="xs:dateTime"
 minOccurs="0" maxOccurs="1"/>
                <!--Data modification time-->
                <xs:element name="DateModified" type="xs:dateTime"
 minOccurs="0" maxOccurs="1"/>
                <!-- Author ->
                <xs:element name="Author" type="xs:string"
 minOccurs="0" maxOccurs="1"/>
                <!--Extended definition of a custom header-->
                <xs:any processContents="lax" namespace="##other" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
        <!--2.4.7.4 (3) Time span-->
        <xs:complexType name="TimeSpan">
            <xs:sequence>
                <!--The start timestamp of the dataset-->
                <xs:element name="Start" type="xs:dateTime" minOccurs="1" maxOccurs="1"/>
                <!--The end timestamp of the dataset-->
                <xs:element name="End" type="xs:dateTime" minOccurs="1" maxOccurs="1"/>
            </xs:sequence>
        </xs:complexType>
        <!--Data channel ID-->
        <!--The value shall be a local ID or a short ID string-->
        <xs:complexType name="DataChannelID">
            <xs:simpleContent>
                <xs:extension base="xs:string">
                    <!--Ordinal number of the DataChannelID array-->
                    <xs:attribute name="id" type="xs:positiveInteger" use="required" />
                </xs:extension>
            </xs:simpleContent>
        </xs:complexType>
        <!--2.4.7.4 (4) Time series data-->
        <xs:complexType name="TimeSeriesData">
            <xs:sequence>
                <!--2.4.7.4 (5) Tabular data -->
                <xs:element name="TabularData" type="sdt:TabularData"
 minOccurs="0" maxOccurs="unbounded" />
                <!--2.4.7.4 (6) Event data -->
                <xs:element name="EventData" type="sdt:EventData"
 minOccurs="0" maxOccurs="1" />
                <!--Extended definition of custom data type-->
                <xs:any processContents="lax" namespace="##other" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
        <!--2.4.7.4 (5) Tabular data -->
        <xs:complexType name="TabularData">
```

```
        <xs:sequence>
            <!--Number of data binding elements in a tabular data structure -->
            <xs:element name="NumberOfDataSet" type="xs:nonNegativeInteger"
minOccurs="0" maxOccurs="1" />
            <!--Number of data hannel-->
            <xs:element name="NumberOfDataChannel" type="xs:nonNegativeInteger"
minOccurs="0" maxOccurs="1"/>
            <!--2.4.7.4 (5) Target data channel identifier-->
            <xs:element name="DataChannelID" type="sdt:DataChannelID”
minOccurs="0" maxOccurs="unbounded"/>
            <!--2.4.7.4 (7) Dataset -->
            <xs:element name="DataSet" type="sdt:DataSet_Tabular"
minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
    </xs:complexType>
    <!--2.4.7.4 (6) Event data -->
    <xs:complexType name="EventData">
        <xs:sequence>
            <!--Number of dataset elements in the event data structure-->
            <xs:element name="NumberOfDataSet" type="xs:nonNegativeInteger"
minOccurs="0" maxOccurs="1" />
            <!--2.4.7.4 (6) Dataset -->
            <xs:element name="DataSet" type="sdt:DataSet_Event"
minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
    </xs:complexType>
    <!--2.4.7.4 (7) Tabular data dataset -->
    <xs:complexType name="DataSet_Tabular">
        <xs:sequence>
            <!--Measured value-->
            <!--Measured value can be null-->
            <xs:element name="Value" type="sdt:Value"
minOccurs="1" maxOccurs="unbounded"/>
        </xs:sequence>
        <!--ISO 8601 date and time structure should be used-->
        <xs:attribute name="timeStamp" type="xs:dateTime" use="required"/>
    </xs:complexType>
    <!--2.4.7.4 (8) Event data dataset-->
    <xs:complexType name="DataSet_Event">
        <xs:sequence>
            <xs:element name="DataChannelID" type="xs:string"
minOccurs="1" maxOccurs="1"/>
            <!--Measured value-->
            <!--Measured value can be null-->
            <xs:element name="Value" type="xs:string"
minOccurs="1" maxOccurs="1"/>
        </xs:sequence>
        <!--ISO 8601 date and time structure should be used-->
        <xs:attribute name="timeStamp" type="xs:dateTime" use="required"/>
```

```
            <xs:attribute name="quality" type="xs:string" use="optional"/>
        </xs:complexType>
        <!--Tabular data value-->
        <xs:complexType name="Value">
            <xs:simpleContent>
                <xs:extension base="xs:string">
                    <!-- Array requirements for the value. The order of the array should be the same as that of the array of tabular data/data channel ID.
                    <xs:attribute name="ref" type="xs:positiveInteger" use="required" />
                    <xs:attribute name="quality" type="xs:string" use="optional"/>
                </xs:extension>
            </xs:simpleContent>
        </xs:complexType>
</xs:schema>
```

2.4.3 XML expression

The XML is expressed as follows:

```
<?xml version="1.0" encoding="utf-8" ?>
<Package xmlns="urn:ISO19848:SHIP_DATA_TRANSPORT:ISC_EMODEL">
    <Header>
        <ShipID>IMO1234567</ShipID>
        <TimeSpan>
            <Start>2022-01-01T12:00:00Z</Start>
            <End>2022-01-03T12:00:00Z</End>
        </TimeSpan>
        <DateCreated>2022-01-03T12:00:00Z</DateCreated>
        <DateModified>2022-01-03T12:00:00Z</DateModified>
        <Author>Author1</Author>
    </Header>
    <TimeSeriesData>
        <TabularData>
            <NumberOfDataSet>2</NumberOfDataSet>
            <NumberOfDataChannel>2</NumberOfDataChannel>
            <!--ShortID of the DataChannel -->
            <DataChannelID id="1">0010</DataChannelID>
            <DataChannelID id="2">0020</DataChannelID>
            <DataSet timeStamp="2022-01-01T12:00:00Z">
                <Value ref="1" quality="0">100.0</Value>
                <Value ref="2" quality="0">200.0</Value>
            </DataSet>
            <DataSet timeStamp="2022-01-02T12:00:00Z">
                <Value ref="1" quality="0">100.5</Value>
                <Value ref="2" quality="0">205.0</Value>
            </DataSet>
        </TabularData>
        <TabularData>
            <NumberOfDataSet>3</NumberOfDataSet>
            <NumberOfDataChannel>1</NumberOfDataChannel>
            <DataChannelID id="1">0010</DataChannelID>
            <DataSet timeStamp="2022-01-01T12:00:00Z">
                <Value ref="1" quality="0">100.0</Value>
            </DataSet>
            <DataSet timeStamp="2022-01-02T00:00:00Z">
                <Value ref="1" quality="0">100.2</Value>
```

```
            </DataSet>
            <DataSet timeStamp="2022-01-02T12:00:00Z">
                <Value ref="1" quality="0">100.5</Value>
            </DataSet>
        </TabularData>
        <EventData>
            <NumberOfDataSet>3</NumberOfDataSet>
            <DataSet timeStamp="2022-01-01T12:00:01Z" quality="0">
                <DataChannelID>0011</DataChannelID>
                <Value>HIGH</Value>
            </DataSet>
            <DataSet timeStamp="2022-01-01T12:00:01Z" quality="0">
                <DataChannelID>0021</DataChannelID>
                <Value>HIGH</Value>
            </DataSet>
            <DataSet timeStamp="2022-01-01T12:00:23Z" quality="0">
                <DataChannelID>0011</DataChannelID>
                <Value>NORMAL</Value>
            </DataSet>
        </EventData>
    </TimeSeriesData>
    <TimeSeriesData>
        <TabularData>
            <NumberOfDataSet>1</NumberOfDataSet>
            <NumberOfDataChannel>2</NumberOfDataChannel>
            <DataChannelID id="1">0010</DataChannelID>
            <DataChannelID id="2">0020</DataChannelID>
            <DataSet timeStamp="2022-01-03T12:00:00Z">
                <Value ref="1" quality="0">101.0</Value>
                <Value ref="2" quality="0">210.0</Value>
            </DataSet>
        </TabularData>
        <TabularData>
            <NumberOfDataSet>2</NumberOfDataSet>
            <NumberOfDataChannel>1</NumberOfDataChannel>
            <DataChannelID id="1">0010</DataChannelID>
            <DataSet timeStamp="2022-01-03T00:00:00Z">
                <Value ref="1" quality="0">100.8</Value>
            </DataSet>
            <DataSet timeStamp="2022-01-03T12:00:00Z">
                <Value ref="1" quality="0">101.0</Value>
            </DataSet>
        </TabularData>
    </TimeSeriesData>
</Package>
```

3 JSON implementation

3.1 General requirements

3.1.1 This section specifies the method of describing data with JSON.

3.1.2 JSON implementation is not mandatory, and shall be carried out as follows.

3.1.2.1 Any newline or indentation characters in the document shall not be considered as important information.

3.1.2.2 XML elements with namespaces shall be converted to JSON elements. For example: <xml:Element>value</xml:Element> Converted to { "xml:Element" : "Value" }.

3.2 Data Channel List

The example of JSON expression is as follows:

```
{
    "Package": {
        "Header": {
            "ShipID": "IMO1234567",
            "Author": "Author1",
            "DateCreated": "2015-12-01T00:00:00Z",
            "nr:CustomHeaderElement": "Vender specific headers"
        },
        "DataChannelList": {
            "DataChannel": [{
                "DataChannelID": {
                    "LocalID": "/emd/v1.0/P4.006.003 +1//DrivingEnd/Temperature",
                    "ShortID": "0010",

                },
                "Property": {
                    "DataChannelType": {
                        "Type": "Inst",
                        "UpdateCycle": "1"
                    },
                    "Format": {
                        "Type": "Decimal",
                        "Restriction": {
                            "FractionDigits": "1",
                            "MaxInclusive": "200.0",
                            "MinInclusive": "-150.0"
                        }
                    },
                    "Range": {
                        "High": "150.0",
                        "Low": "0.0"
                    },
                    "Unit": {
                        "UnitSymbol": "°C",
                        "QuantityName": "Temperature"
                    },
                    "QualityCoding": "OPC_QUALITY",
                    "Name": "Hot water temperature of No.1 main engine",
                    "Remarks": "Location: ECR, manufacturer: AAA Company, specification: TYPE-AAA ",
                    "nr:CustomPropertyElement": "Vender specific Property"
                }
            }, {
                "DataChannelID": {
                    "LocalID": " /emd/v1.0/P4.006.003+1//DrivingEnd/Speed",
                    "ShortID": "0020",                    },
                "Property": {
                    "DataChannelType": {
                        "Type": "Average",
                        "UpdateCycle": "60",
                        "CalculationPeriod": "3600"
                    },
```

```
                                "Format": {
                                        "Type": "Integer",
                                        "Restriction": {
                                                "FractionDigits": "0",
                                                "MaxInclusive": "",
                                                "MinInclusive": ""
                                        }
                                },
                                "Range": {
                                        "High": "7000",
                                        "Low": "0"
                                },
                                "Unit": {
                                        "UnitSymbol": "RPM",
                                        "QuantityName": "Speed"
                                },
                                "QualityCoding": "OPC_QUALITY",
                                "Name": "Supercharger speed of No. 1 main engine",
                                "Remarks": "Location: ECR, manufacturer: AAA Company, specification: TYPE-AAA ",
                                "nr:CustomPropertyElement": "Vender specific Property"
                        }
                }
                }]
        }
    }
}
```


3.3 Time Series Data

The example of JSON expression is as follows:


```
{
    "Package": {
        "Header": {
            "ShipID": "IMO1234567",
            "TimeSpan": {
                "Start": "2022-01-01T12:00:00Z",
                "End": "2022-01-03T12:00:00Z"
            },
            "DateCreated": "2022-01-03T12:00:00Z",
            "DateModified": "2022-01-03T12:00:00Z",
            "Author": "Shipboard data server",

        },
        "TimeSeriesData": [{
            "TabularData": [{
                "NumberOfDataSet": "2",
                "NumberOfDataChannel": "2",
                "DataChannelID": ["0010", "0020"],
                "DataSet": [{
                    "TimeStamp": "2022-01-01T12:00:00Z",
                    "Value": ["100.0", "200.0"],
                    "Quality": ["0", "0"]
                }, {
                    "TimeStamp": "2016-01-02T12:00:00Z",
                    "Value": ["100.5", "205.0"],
                    "Quality": ["0", "0"]
                }]
```

```
        }, {
            "NumberOfDataSet": "3",
            "NumberOfDataChannel": "1",
            "DataChannelID": ["0110"],
            "DataSet": [{
                "TimeStamp": "2022-01-01T12:00:00Z",
                "Value": ["100.0"],
                "Quality": ["0"]
            }, {
                "TimeStamp": "2022-01-02T00:00:00Z",
                "Value": ["100.2"],
                "Quality": ["0"]
            }, {
                "TimeStamp": "2022-01-02T12:00:00Z",
                "Value": ["100.3"],
                "Quality": ["0"]
            }]
        }],
        "EventData": {
            "NumberOfDataSet": "3",
            "DataSet": [{
                "TimeStamp": "2022-01-01T12:00:01Z",
                "DataChannelID": "0011",
                "Value": "HIGH",
                "Quality": "0"
            }, {
                "TimeStamp": "2022-01-01T12:00:01Z",
                "DataChannelID": "0021",
                "Value": "HIGH",
                "Quality": "0"
            }, {
                "TimeStamp": "2022-01-01T12:00:23Z",
                "DataChannelID": "0011",
                "Value": "NORMAL",
                "Quality": "0"
            }]
        }
    }, {
        "TabularData": [{
            "NumberOfDataSet": "1",
            "NumberOfDataChannel": "2",
            "DataChannelID": ["0010", "0020"],
            "DataSet": [{
                "TimeStamp": "2022-01-03T12:00:00Z ",
                "Value": ["101.0", "210.0"],
                "Quality": ["0", "0"]
            }]
        }, {
            "NumberOfDataSet": "2",
            "NumberOfDataChannel": "1",
            "DataChannelID": ["0110"],
            "DataSet": [{
                "TimeStamp": "2022-01-03T00:00:00Z",
                "Value": ["100.8"],
                "Quality": ["0"]
            }, {
                "TimeStamp": "2022-01-03T12:00:00Z",
                "Value": ["101.0"],
```

```
                                "Quality": ["0"]
                            }]
                        }]
                    }]
                }
}
```


4 CSV implementation

4.1 General requirements

4.1.1 This section specifies the method for describing data with CSV is specified in this section. CSV is only applicable to time series data and shall be used in conjunction with the data channel list described in XML/JSON.

4.1.2 CSV shall be implemented in accordance with the following requirements.

- CSV shall be described according to RFC 4180
- Tabular data and event data shall be stated separately
- The number of columns in all rows of the CSV shall be the same

4.1.3 The header row (first row) may be omitted if the data channel in the data can be clearly identified.

4.2 Tabular Data

4.2.1 The tabular data described with CSV is as follows:

Timestamp	Data channel 1	Data channel 2	…	Data channel N
Timestamp 1	Monitoring value 1-1	Monitoring value 1-2	…	…
…	…	…	…	…
Timestamp M	…	…	…	Monitoring value M-N

Where:

(1) Time stamp M: the monitoring date and time.

(2) Data channel N: one of the following IDs, universal ID, local ID, and short ID.

(3) Monitoring value M-N: the monitoring value of data channel N at timestamp M.

A CSV example of tabular data using short IDs is as follows:

```
Timestamp                   0010, 0020
2022-01-01T12:00:00Z, 100.0, 200.0
2016-01-02T12:00:00Z, 100.5, 205.0
2022-01-03T12:00:00Z, 101.0, 210.0
```

Note: 0010 and 0020 are short Ids of some data channels.

4.3 Event data

4.3.1 The example of describing event data with CSV is as follows:

Timestamp, data channel ID monitoring value
Timestamp 1, data channel ID1 monitoring value 1-1
…, …, …
Timestamp_M, data channel ID_N, monitoring value M-N

Where:

(1) Timestamp M: the monitoring date and time.

(2) Data channel N: one of the following IDs, universal ID, local ID, and short ID.

(3) Monitoring value MN: the monitoring value of data channel N at timestamp M.

A CSV example of event data using short IDs is as follows:

Timestamp, data channel ID, monitoring value
2022-01-01T12:00:01Z,0011, HIGH
2022-01-01T12:00:01Z,0021, HIGH
2022-01-01T12:01:23Z,0011, NORMAL

Note: 0011 and 0021 are short Ids of some data channels.