INTERNATIONAL STANDARD

ISO 19848

First edition
2018-10

Ships and marine technology — Standard data for shipboard machinery and equipment

Navires et technologie maritime — Données normalisées pour les machines et équipements à bord des navires

Reference number
ISO 19848:2018(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 8, *Ships and marine technology*, Subcommittee SC 6, *Navigation and ship operations*.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

On-board computer applications for safety and energy-efficient operations have become popular. These applications require access to data of shipboard machinery and equipment.

To access data of navigational equipment, a data exchange standard, the IEC 61162 series can be used. However, access of data from other on-board components and systems (e.g. machinery, safety equipment and hull) have not yet been standardised.

Exchanging non-standardised data between and/or among applications requires name-based aggregation and format mapping. However, this requires a large amount of labour, which hinders the use of such data.

To improve these situations, this document defines unified rules for developing machine and human-readable identifiers and data structures for shipboard machinery and equipment, with the objective to facilitate exchange and processing of sensor data from ships.

This document defines two concepts and their models for data exchange: one is Data Channel, and the other is Time Series Data. This document thus defines two distinct data structures and file formats: A Data Channel List, which contains the necessary meta-data, and a Time Series Data format for measurements. The time-series format is designed to be lightweight and it therefore contains minimal meta-data information only in the form of a reference to the channel list.

Data Channel is a concept that represents virtual data transmission channels, and defines time-invariant properties. Data Channel can be viewed as a static description for the different sensor data streams.

Data Channel is composed of Data Channel ID and Data Channel Property.

Data Channel ID uniquely identifies the logical data channels. Data Channel Property defines attributes of Data Channel.

There are three types of Data Channel ID. One is Local ID, which is a unique identifier used on-board a ship, and another is Universal ID, which is a universal identifier, composed of Name Entity, Ship ID (e.g., IMO numbers) and Local ID. The other ID is Short ID, a short alternative ID of Local ID.

The purpose of this document is for exchanging data on-board a ship; however, in the future, shipboard machinery and equipment may be connected directly to the Internet.

Therefore, considering the compatibility between Data Channel ID and URLs, which are used to identify data on the Internet, Data Channel ID has a hierarchical structure with slashes as delimiters. To represent a hierarchy, Data Channel is categorised in accordance with the standardised categorising rule and named by concatenating these category names with slashes.

In Annexes B and C, two types of categorising rules and example of codebook, lists of standardised category names given in accordance with the rules, are defined for reference.

They are not designed to unify Data Channel ID, but it is assumed that some entities will develop, maintain and manage codebook and that they will be disclosed widely.

Data Channel Property is assumed to be used to automate data processing and help understanding of data. Data Channel Property shall be used because it is considered to be essential to both computer applications and humans for the reasons mentioned above.

Time Series Data is a concept that represents collection of time-stamped data. Time Series Data is assumed to be used for sharing latest data and for analysing trends made over time-stamped data.

For reliable data exchange, this document mandates the use of XML (Extensible Markup Language) and XML Schema for data encoding and data structure definition. Using XML and XML schemas makes it possible to define data structures precisely and validate data in accordance with such definitions.

As a result, it is believed that data can be exchanged more reliably between and/or among computer applications.

Further, for convenience and efficiency, this document also defines data-structures in JSON and CSV format.

It is assumed that data from shipboard machinery and equipment will be collected by shipboard data servers, which are defined in ISO 19847. Then, the data encoded in accordance with this document, in some cases, could be encrypted for security reasons, will be shared between and/or among computer applications in a wide variety of means, such as in Hyper Text Transfer Protocol (HTTP), in Message Queue Telemetry Transport (MQTT) or by e-mail through the servers. As described above, external computer applications can retrieve on-board data uniformly by accessing the servers.

Ships and marine technology — Standard data for shipboard machinery and equipment

1 Scope

This document applies to the structure of the ship and to shipboard machinery and equipment, and is intended for implementers of software used for the capture and processing of sensor data from the objects mentioned above.

For those purposes, this document describes the way to name the sensor, required data item, and the way to describe the data above.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 8601, *Data elements and interchange formats — Information interchange — Representation of dates and times*

ISO/IEC 80000 series, *Quantities and units*

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

W3C XML Schema Part 1: XML Schema Part 1: Structures, W3C Recommendation

W3C XML Schema Part 2: XML Schema Part 2: Datatypes, W3C Recommendation

RFC 3339, *Date and Time on the Internet: Timestamps*

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

RFC 5234, *Augmented BNF for Syntax Specifications: ABNF*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— IEC Electropedia: available at http://www.electropedia.org/

— ISO Online browsing platform: available at https://www.iso.org/obp

3.1
alert data
information that represents abnormal conditions of shipboard machinery and equipment

3.2
analogue data
numerical information obtained from sensors such as temperature sensors and pressure sensors

Note 1 to entry: Analogue data is a physical value converted from raw electric signals, such as 4-20 mA or 0-5 V.

3.3
codebook
list of standardised names

3.4
data
measurement value from shipboard machinery and equipment to which a timestamp is added

3.5
Data Channel
virtual channel for data transmission from shipboard machinery and equipment to shipboard data server, defining static properties of data

3.6
Data Channel ID
identifier for Data Channel that identifies Data Channel universally and on-board a ship

Note 1 to entry: There are three types of Data Channel ID: Universal ID, Local ID and Short ID.

3.7
Data Channel List
list of definitions for Data Channel that define Data Channel ID and Data Channel Property, and is shared through the shipboard data server

3.8
Data Channel Property
attributes of Data Channel, such as units and ranges

3.9
Data Set
set of Data having the same timestamp

3.10
Extensible Markup Language
XML
text-based data description language used for exchanging data on the Internet

3.11
Hyper Text Transfer Protocol
HTTP
communication protocol used to exchange HTML (Hyper Text Markup Language) or other content on the Internet

3.12
IMO Number
unique reference number for ships that is given by the International Maritime Organisation (IMO)

3.13
logical structure
structure of data that is independent of physical implementation

3.14
measurement value
numeric value or a status symbol, produced as a result of measuring, calculating or estimating the state of various objects

3.15
metadata
data that describes information about other data

3.16
Name Object
building block of Data Channel ID used to define the hierarchical structure of Data Channel ID

3.17
Namespace
set of names that is used in order to avoid conflicts of names

3.18
Shipboard Data Server
ship's "information hub" that stores data from shipboard machinery and equipment, shares data at sea including machine data, and sends stored data outboard

Note 1 to entry: See ISO 19847 for details.

3.19
shipboard machinery and equipment
various systems located in ships machinery space, such as main engine, generator, pumps, fans, valves, pipelines and electric control systems

3.20
status data
information that represents the condition of shipboard machinery and equipment

3.21
Time Series Data
collection of a Data Set

3.22
XML Schema
data definition language used for XML

4 Abbreviated terms

ABNF	Augmented Backus Naur Form
AMS	Alarm Monitoring System
BNF	Backus Naur Form
IAS	Integrated Automation System
IMO	International Maritime Organisation
HIN	Hull Identification Number
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
RFC	Request for Comments
SI	The International Systems of Units
URI	Uniform Resource Identifier

UTC	Universal Time Coordinated
UTF-8	UCS Transformation Format 8
XML	Extensible Markup Language

5 Data Channel

5.1 General

Data Channel is composed of Data Channel ID and Data Channel Property.

Data Channel ID is an identifier for Data Channel, and Data Channel Property represents attributes of Data Channel.

5.2 Data Channel ID

There are three types of Data Channel ID:

— Universal ID;

— Local ID;

— Short ID.

Universal ID is for identifying on-board Data Channel universally.

Local ID, meanwhile, is for identifying on-board Data Channel locally. For instance, on-board computer systems, such as the Integrated Automation System (IAS) and the Alarm Monitoring System (AMS), have their own Data Channel List, which is composed of unique Channel ID. This Channel ID can correspond to Local ID.

Short ID is an optional short alternative identifier of Local ID for usability and data compression. This short identifier, for instance, can be used as Data Channel identifier in the Time Series Data format.

Local ID and Short ID shall be unique for a ship.

These IDs shall be case-insensitive to avoid unexpected mistyping.

NOTE Data Channel of the same kind of sensors on different ships is expected to have a same Local ID. Conversely, even if Data Channel has a same meaning, Short ID may be different for each ship.

5.2.1 Universal ID

Universal ID is an URI conforming to the requirements below, in addition to those of the URI definition. The URI definition allows for many different compositions, but the Universal ID will be a subset of these and shall be in the following format.

Universal ID composition is defined by using Augmented BNF(ABNF), which is defined in RFC 5234, as follows.

UniversalID	= [protocol] "//" NamingEntity ShipID LocalID
NamingEntity	= authority
ShipID	= path-element
path-element	= "/" unreserved
path-elements	= path-element | path-element path-elements

Definition of the “Local ID” element is mentioned in 5.2.2.

The “authority” and “unreserved” element is defined in the URI definition. The “protocol” element is optional.

NOTE 1 Though path element of URI that is defined in RFC 3986 accepts much more characters, such as RFC 3986 “sub-delimiters”, “:”, “@”, etc., this document only accepts RFC 3986 “unreserved” characters since these characters may be used as control character in ISO 19847. (Definition of sub-delimiters and unreserved are defined in RFC 3986).

The slash (“/”) is a reserved character for describing hierarchies.

a) Naming Entity

Naming Entity element shall be a domain owned or controlled by the entity producing Local ID.

EXAMPLE

— data.shipdatacenter.jp

— data.dnvgl.com

b) Ship ID

Ship ID is for identifying ships universally.

Usually, an IMO number or HIN should be used for Ship ID.

If ships have no IMO number or HIN, an identifier provided by countries or regions, or other means may be used instead.

EXAMPLE

— /IMO1234567

— /JP-HXAB7A33G293

5.2.2 Local ID

Local ID consists of Naming Rule and Local Data Name.

Local ID composition is defined by using ABNF as follows

LocalID	= NamingRule LocalDataName
NamingRule	= path-element
LocalDataName	= path-elements
path-element	= "/" unreserved
path-elements	= path-element | path-element path-elements

a) Naming Rule

Naming Rule shall be the designated name for the rule used to name Data Channel.

This name can be set freely under the supervision of Naming Entity, and shall have a symbol that represents Naming Entity in front to eliminate duplications.

A Naming Rule is a set of requirements that define a naming scheme (or an identification scheme) for components and systems on-board the ship. A Naming Rule shall define how identification strings are composed, and the method of developing an identification string.

EXAMPLE

— /jsmea_mac

— /dnvgl-vis

b) Local Data Name

Local Data Name is an identifier for Data Channel that is named in accordance with Naming Rule.

The syntax of the identification string shall be disclosed and precisely defined using ABNF.

EXAMPLE

— /MainEngine/Cylinder1/ExhaustGas/Temp

— /411.1/C101.31+1/ExhGas+t(C)

5.2.3 Short ID

Short ID is an optional short alternative to Local ID. There must be a one-to-one correspondence between Data Channel and Short ID; Short ID shall therefore be unique for a ship.

Definition of Short ID is as follows.

ShortID	= unreserved

Short ID shall be as short as practical and represented as machine-friendly symbols, human-friendly short word, or a combination of those symbols and short word.

EXAMPLE

— 0001

— TAH001

— ME_RPM

5.2.4 Example of Data Channel ID

In the following example, Ship ID, Naming Rule and Local Data Name will be understood as the above definition, but without the leading slash.

Universal ID	http://data.shipdatacenter.jp/imo1234567/jsmea_mac/MainEngine/Cylinder1/ExhaustGas/Outlet/Temp
Local ID	/jsmea_mac/MainEngine/Cylinder1/ExhaustGas/Outlet/Temp
Short ID	0001
Ship ID	imo1234567
Naming Entity	data.shipdatacenter.jp
Naming Rule	jsmea_mac
Local Data Name	MainEngine/Cylinder1/ExhaustGas/Outlet/Temp

Universal ID	http://data.dnvgl.com/imo1234567/dnvgl-vis/411.1/C101.31+1/ExhGas+t(C)
Local ID	/dnvgl-vis/411.1/C101.31+1/ExhGas+t(C)
Short ID	0001
Ship ID	imo1234567
Naming Entity	data.dnvgl.com
Naming Rule	dnvgl-vis
Local Data Name	411.1/C101.31+1/ExhGas+t(C)

NOTE 1 It is not a requirement that the Universal ID be a resolvable URI, i.e., the URI is not necessarily a valid URL.

5.3 Data Channel Property

Data Channel Property shall be defined to provide the attributes of Data Channel.

The reserved property types are as follows.

— Data Channel Type

— Format

— Range

— Unit

— Quality Coding

— Name

— Remarks

The properties above shall be described in accordance with the rules in this Clause.

Properties that are not listed above may be used if these are clearly distinguished from the properties defined in the standard.

Details of each property are as follows.

a) Data Channel Type

Data Channel Type is used to identify the types of Data Channel, such as row numeric value, average value, alarms and status. Data Channel Type is composed of the following sub-properties.

— Type

— Update Cycle

— Calculation Period

Type sub-property defines type of Data Channel and the value of the property follows the definitions mentioned in Table 1.

Table 1 — Type name of Data Channel Type

Type	Description
Inst	Measuring value at a certain point in time.
Average	Average of the value within a certain time period. "Average" does not mean average of values from multiple sensors at the same time but average of time-series values from single sensor.
Max	Maximum value within a certain time period. "Maximum" does not mean maximum of values from multiple sensors at the same time but maximum of time-series values from single sensor.
Min	Minimum value within a certain time period. "Minimum" does not mean minimum of values from multiple sensors at the same time but minimum of time-series values from single sensor.
StandardDeviation	Standard deviation of the value within a certain time period. "StandardDeviation" does not mean standard deviation of values from multiple sensors at the same time but standard deviation of time-series values from single sensor.
Calculated	Value obtained from calculation instead of measurement.
SetPoint	Target value for automatic control.
ControlOutput	Manipulated value of automatic control.
Alert	Alarm values that can be obtained are also described.
Status	Status values that can be obtained are also described.
ManuallyInput	Value input by crew. Value assumed here is reading of indicator.

Update Cycle represents the cycle of updating measurement value. This sub-property shall be used when measurement value is updated periodically.

When a value of Data Channel is a result of calculation that uses measurement value of specific time periods, Calculation Period shall be used to describe the said period.

Update Cycle and Calculation Period shall be described with a decimal number that is larger than zero. The unit of Update Cycle and Calculation Period shall be the “second”.

Type sub-property is mandatory and the others are optional.

EXAMPLES

— Type Average

— Calculation Period 60

— Update Cycle 1

b) Format

Format is used for describing data formats and defined by the following sub-properties.

— Type

— Restriction

Type sub-property is mandatory and Restriction sub-property is optional. More than one Restriction sub-property may exist under the Format property.

Available Types are as follows. Definitions of these data types comply with W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatype.

Table 2 — Available Datatype for Format property

Type	Description
Decimal	Decimal represents a subset of the real numbers, which can be represented by decimal numerals. The value space of decimal is the set of numbers that can be obtained by dividing an integer by a non-negative power of ten, i.e., expressible as $i / 10^n$ where i and n are integers and $n \geq 0$. Precision is not reflected in this value space; the number 2.0 is not distinct from the number 2.00. The order relation on decimal is the order relation on real numbers, restricted to this subset.
Integer	Integer is·derived·from decimal by fixing the value of fraction digits to be 0 and disallowing the trailing decimal point. This results in the standard mathematical concept of the integer numbers. The value space of integer is the infinite set {...,-2,-1,0,1,2,...}. The base type of integer is decimal.
Boolean	Boolean represents the values of two-valued logic.
String	The string datatype represents character strings in XML.
DateTime	Date and time data types are used for values that contain date and time. Format shall follow ISO 8601 "YYYY-MM-DDThh:mm:ssZ" where: YYYY indicates the year MM indicates the month DD indicates the day T indicates the start of the required time section hh indicates the hour mm indicates the minute ss indicates the second Z indicates UTC

Following restrictions defined in W3C XML Schema are available to define acceptable values. Validation rules and available constraint for each data types shall follow W3C XML Schema.

Table 3 — Restrictions for Format property

Restriction	Description	Data Type
Enumeration	Defines a list of acceptable values.	string
FractionDigits	Specifies the maximum number of decimal places allowed. Must be equal to or greater than zero.	nonNegativeInteger
Length	Specifies the exact number of characters or list items allowed. Must be equal to or greater than zero.	nonNegativeInteger
MaxExclusive	Specifies the upper bounds for numeric values (the value must be less than this value).	A value from the ·value space· of the {base type definition}.
MaxInclusive	Specifies the upper bounds for numeric values (the value must be less than or equal to this value).	A value from the ·value space· of the {base type definition}.
MaxLength	Specifies the maximum number of characters or list items allowed. Must be equal to or greater than zero.	nonNegativeInteger
MinExclusive	Specifies the lower bounds for numeric values (the value must be greater than this value).	A value from the ·value space· of the {base type definition}.
MinInclusive	Specifies the lower bounds for numeric values (the value must be greater than or equal to this value).	A value from the ·value space· of the {base type definition}.

Table 3 *(continued)*

Restriction	Description	Data Type
MinLength	Specifies the minimum number of characters or list items allowed. Must be equal to or greater than zero.	nonNegativeInteger
Pattern	Defines the exact sequence of characters that are acceptable.	string
TotalDigits	Specifies the exact number of digits allowed. Must be greater than zero.	positiveInteger
WhiteSpace	Specifies how white space (line feeds, tabs, spaces, and carriage returns) is handled.	

EXAMPLES

— Type Decimal

— Restriction TotalDigits 8, FractionDigits 3

c) Range

Range is for describing Data Range. Range property is composed of the following sub-properties.

— Low

— High

Low sub-property represents lower limit of the analogue data, and High sub-property represents its upper limit.

The value type of these sub-properties is decimal value or empty. The value of these sub-properties can be empty only if lower and/or upper limit cannot be specified.

Further, High value must be greater than and not equal to Low value.

Range property is mandatory only of the analogue data.

NOTE Range does not mean upper and lower boundary of the value. The value can exceed the range in case of sensor failure or other abnormal condition.

EXAMPLES

— Low 0

— High 120

d) Unit

Unit defines the unit and quantity applied to the measurement value. Unit property is composed of the following sub-properties.

— Unit Symbol

— Quantity Name

Unit symbols and quantity names defined in ISO 80000 or Table 4 shall be used for Unit Symbol and Quantity Name sub-properties.

Unit Symbol sub-property represents unit symbol of the measurement value (e.g., “m” for length and “kg” for mass). The value of Unit Symbol sub-property can be empty only for non-dimensional quantities.

Quantity Name sub-property may be used to identify the variable that is measured or computed for the physical item defined by the Data Channel.

The measurement value shall be described according to the Unit defined in this property.

NOTE On-board computer applications (Data consumer) can convert Unit according to ISO 80000.

EXAMPLES

— Unit Symbol kW

— Quantity Name active power

— Unit Symbol Pa

— Quantity Name pressure

Table 4 — Additional units and quantities of measurement value

Quantity Name	Unit Symbol	Remarks
Cargo capacity	TEU	Twenty-foot Equivalent Unit
	FEU	Forty-foot Equivalent Unit

e) Quality Coding

Quality Coding represents a name of data quality evaluation scheme and the scheme shall be able to distinguish at least between valid and invalid measurement values of the Data Channel.

EXAMPLES

— "OPC-QUALITY".

— "IEC 61162-STATUS".

In the case of "IEC 61162-STATUS", "A" (Data valid) and "V" (Data invalid) are used for the data quality.

f) Name

Name can be used to describe names assigned for on-board control systems and other instruments.

EXAMPLE

— “Main Engine Revolution”.

g) Remarks

In Remarks, arbitrary complementary information on Data Channel is described.

Remarks should include locations, manufacturers and types of equipment.

EXAMPLE

— “Location: ECR, Manufacturer: AAA Company, Type: TYPE-AAA”.

6 Time Series Data

6.1 General

Time Series Data is a collection of measurement values in which all the values have a corresponding time of measurement. Usually such data is arranged and recorded in chronological order.

6.2 Type of Time Series Data

For practical purposes (efficient data transport), two distinct representations of Time Series Data are defined:

— Tabular Data;

— Event Data.

These representations are used depending on update interval of measurement values.

6.2.1 Tabular Data

Tabular Data is a vector of a fixed number of values expected to be reported at regular interval. Data Channel List defines the interval.

Some examples of Tabular Data are:

— multiple raw numeric values from sensors/transmitters sampled at the same time;

— result of the calculation (e.g. time average, standard deviation, etc.) to be performed regularly.

Measurement values are grouped per timestamp, thus Tabular Data forms like Table 5.

Table 5 — Example of Tabular Data

Time Stamp	Data Channel 1	Data Channel 2	Data Channel 3	Data Channel 4
2017-01-01T00:00:00Z	101.2	0.30	10.2	CLOSE
2017-01-01T00:00:01Z	0.0	0.30	10.2	CLOSE
2017-01-01T00:00:02Z	110.9	0.32	10.2	OPEN

6.2.2 Event Data

Event Data is a collection of data for which the number of values at a specific time is not fixed.

Some examples of Event Data are:

— alarm information;

— status information;

— manually input data.

Event Data typically forms in tabular form like Table 6:

Table 6 — Example of Event Data by tabular form

Time Stamp	Data Channel 1	Data Channel 2	Data Channel 3	Data Channel 4
2017-01-01T00:00:00Z	101.2	—	—	—
2017-01-01T00:00:01Z	—	—	—	—
2017-01-01T00:00:02Z	110.9	0.32	—	OPEN

Tabular form mentioned above is not efficient for Event Data, thus measurement values are grouped per both timestamp and Data Channel ID and forms like Table 7:

Table 7 — Example of Event Data

Time Stamp	Data Channel ID	Value
2017-01-01T00:00:00Z	Data Channel 1	101.2
2017-01-01T00:00:02Z	Data Channel 1	110.9
2017-01-01T00:00:02Z	Data Channel 2	0.32
2017-01-01T00:00:02Z	Data Channel 4	OPEN

6.3 Time Series Data Composition

Each row of Tabular/Event Data is called Data Set.

Tabular Data is composed of a collection of Data Set without Data Channel IDs, since value can be identified by its order.

Tabular Data of Table 5 is described as follows.

```
DataSet("2017-01-01T00:00:00Z","101.2","0.30","10.2","CLOSE")
DataSet("2017-01-01T00:00:01Z","0.0","0.30","10.2","CLOSE")
DataSet("2017-01-01T00:00:02Z","110.9","0.32", "10.2","OPEN")
```

Event Data is composed of a collection of Data Set with Data Channel ID, since a number of values at a specific time cannot be fixed.

Event Data of Table 7 is described as follows.

```
DataSet("2017-01-01T00:00:00Z ", "DataChannel1",    "101.2")
DataSet("2017-01-01T00:00:02Z ", "DataChannel1",    "110.9")
DataSet("2017-01-01T00:00:02Z ", "DataChannel2",    "0.32" )
DataSet("2017-01-01T00:00:02Z ", "DataChannel4",    "OPEN" )
```

7 Data structure

7.1 General

Data structure is the definition of logical structures that are independent from data implementation languages, such as XML, JSON, CSV and others.

This Clause defines two types of data structure. One is for Data Channel List, the list of Data Channel definitions in Clause 5, and the other is for serialisation and transport of Time Series Data in Clause 6.

These structures can be described by using XML, JSON and CSV.

In addition, standard data type, which is used to define the data structure and also independent of implementation languages, is defined.

Standard data type can be replaced with the data types defined in the implementation language.

Data implemented in accordance with the data structure can be shared between and/or among computer applications.

7.2 Implementation language

Data structure defined in this Clause shall be implemented in accordance with Annex A.

7.3 Standard data types

To define the data Structure, the following standard data types derived from UML primitive types listed in Table 8 are used.

Table 8 — Standard data types

Standard data type	Primitive type	Restriction	Remarks
Integer	Integer		
NonNegativeInteger	Integer	larger than –1	
PositiveInteger	Integer	larger than 0	
Real	Real		
Boolean	Boolean		
String	String		
DateTime	String	formatted by ISO 8601	Refer to RFC3339 for ABNF expression.
Null			Null specifies the lack of a value (can be used for any data types).

7.4 Structure of Data Channel List

7.4.1 Data model

Data Channel List shall consist of the following five elements.

a) Package

Package is a data package that is made up of Header, which is a set of metadata, and DataChannel, which is a main data body.

b) Header

Header indicates when Data Channel List is created and who creates it.

c) DataChannel

DataChannel consists of Property and DataChannelId, which indicates Data Channel definitions.

d) DataChannelID

DataChannelID is an identifier of Data Channel defined in 5.2.

e) Property

Property defines the attributes of Data Channel defined in 5.3.

These elements are arranged in the hierarchic structure shown in Figure 1.

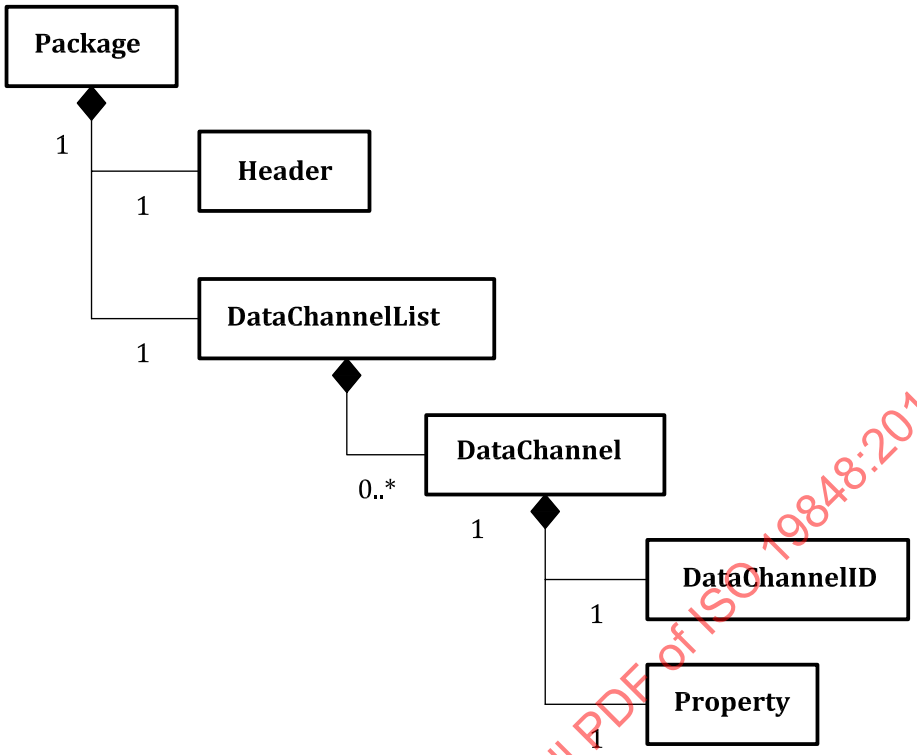

Figure 1 — Model of Data Channel List

7.4.2 Logical structure

Data Channel List shall have the logical structure shown in Figure 2.

Data Channel List has a Package element. The Package element is composed of a Header and a DataChannelList element. The DataChannelList element includes one or more DataChannel elements with its IDs and properties.

Further, NameObject element may be added to define structure of Local ID.

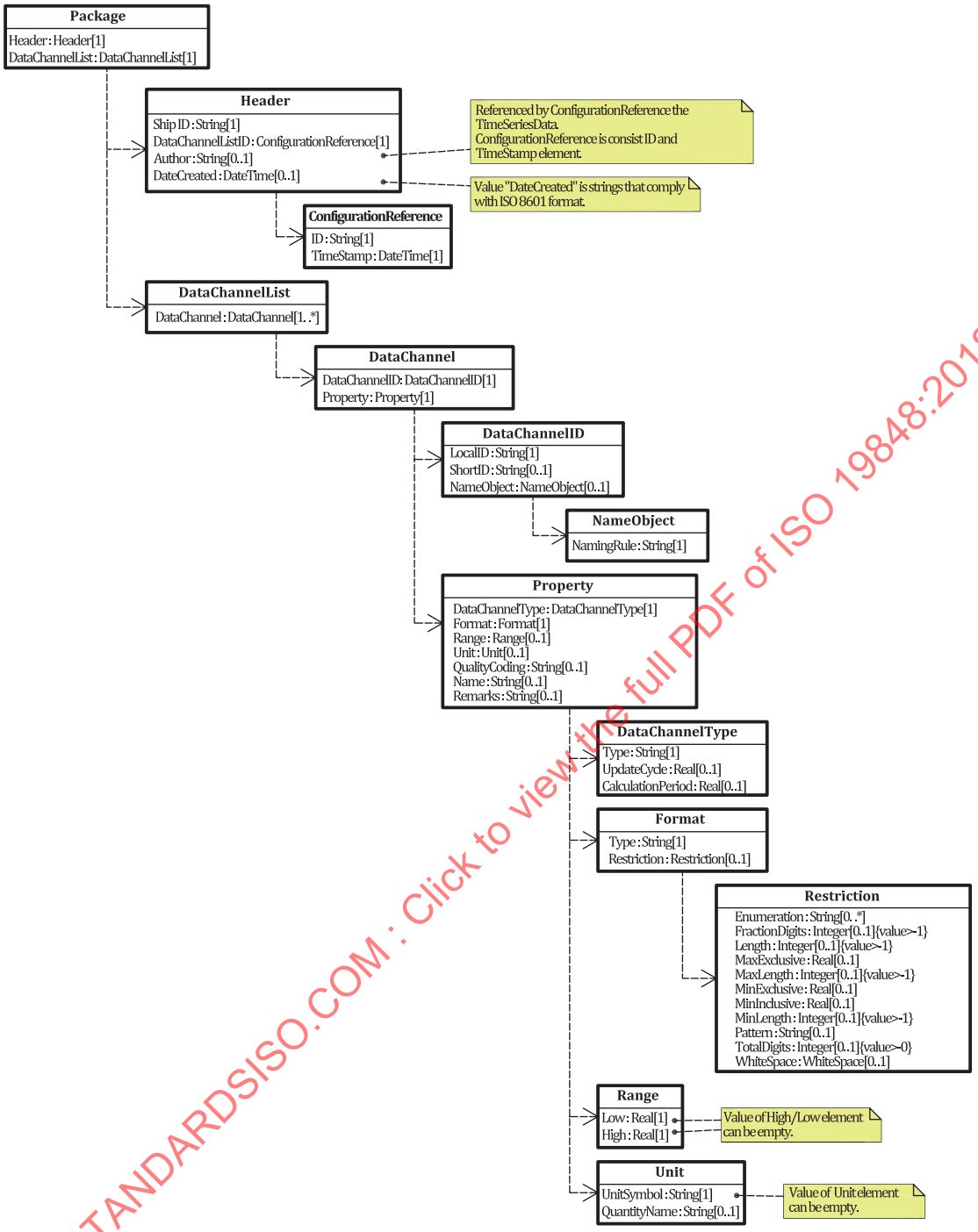

Figure 2 — Logical structure of Data Channel List

Details of each element are described below.

a) Package structure

Name	Data type	Note	Mandatory/ Option	Max count
Header	b) Header	See b).	Mandatory	1
DataChannelList	d) DataChannelList	See d).	Mandatory	1

b) Header structure

Name	Data type	Note	Mandatory/ Option	Max count
ShipID	String	See 5.2.1.	Mandatory	1
DataChannelListID	c) ConfigurationReference	See c).	Mandatory	1
Author	String	Author of Data Channel List.	Optional	1
DateCreated	DateTime	Date when Package is created.	Optional	1

c) ConfigurationReference structure

Name	Data type	Note	Mandatory/ Option	Max count
ID	String	Identifier of Data Channel List itself.	Mandatory	1
TimeStamp	DateTime	Modified Date & Time.	Mandatory	1

d) DataChannelList structure

Name	Data type	Note	Mandatory/ Option	Max count
DataChannel	e) DataChannel	See e).	Mandatory	*

e) DataChannel structure

Name	Data type	Note	Mandatory/ Option	Max count
DataChannelID	f) DataChannelID	See f).	Mandatory	1
Property	g) Property	See g).	Mandatory	1

f) DataChannelID structure

Name	Data type	Note	Mandatory/ Option	Max count
LocalID	String	See 5.2.2.	Mandatory	1
ShortID	String	See 5.2.3.	Optional	1
NameObject	m) NameObject	Definition of Local Data Name structure.	Optional	1

g) Property structure

Name	Data type	Note	Mandatory/ Option	Max count
DataChannelType	h) DataChannelType	See h) and 5.3 a).	Mandatory	1
Format	i) Format	See i) and 5.3 b).	Mandatory	1
Range	k) Range	See k) and 5.3 c).	Mandatory*	1
Unit	l) Unit	See l) and 5.3 d).	Mandatory*	1
* Range and Unit are required only when Data Channel Type is “Decimal”.				

Name	Data type	Note	Mandatory/ Option	Max count
QualityCoding	String	5.3 e).	Optional	1
Name	String	See 5.3 f).	Optional	1
Remarks	String	See 5.3 g).	Optional	1
* Range and Unit are required only when Data Channel Type is “Decimal”.				

h) DataChannelType structure

Name	Data type	Note	Mandatory/ Option	Max count
Type	String	Described in accordance with Data Channel Type format shown in 5.3 a).	Mandatory	1
UpdateCycle	Real		Optional	1
CalculationPeriod	Real		Optional	1

i) Format structure

Name	Data type	Note	Mandatory/ Option	Max count
Type	String	Described in accordance with Data Channel Type format shown in 5.3 b).	Mandatory	1
Restriction	j) Restriction		Optional	1

j) Restriction structure

Name	Data type	Note	Mandatory/ Option	Max count
Enumeration	String	Described in accordance with Data Channel Type format shown in 5.3 b) Table 3.	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
MinInclusive	Real		Optional	1
MinLength	Integer		Optional	1
Pattern	String		Optional	1
TotalDigits	Integer		Optional	1
WhiteSpace	One of the following: “Preserve” “Replace” “Collapse”		Optional	1

k) Range structure

Name	Data type	Note	Mandatory/ Option	Max count
Low	Real	Lower limit of measuring range.	Mandatory	1
High	Real	Upper limit of measuring range.	Mandatory	1

l) Unit structure

Name	Data type	Note	Mandatory /Option	Max count
UnitSymbol	String	Unit symbol defined in ISO 80000 or 5.3 d) Table 4.	Mandatory	1
QuantityName	String	Quantity name defined in ISO 80000 or 5.3 d) Table 4.	Optional	1

m) NameObject structure

Name	Data type	Note	Mandatory/ Option	Max count
NamingRule	String	See 5.2.2	Mandatory	1

7.5 Structure of Time Series Data

7.5.1 Data model

Time Series Data shall consist of the following elements.

a) Package

Package element is data made up of a Header, which is a metadata, and TimeSeriesData, which is a main data body.

b) Header

Header element contains the necessary metadata to link the Data in the Package to the right channels in the Data Channel List and the time in the real world.

This can be achieved by indicating when Time Series Data is created, who creates Time Series Data, an explicit reference to a specific Data Channel List, what Data Channels are included, the number of data and the measurement period of Time Series Data.

If such kind of metadata is shared in advance by any means, Header element itself may be omitted or just formed by list of Short IDs.

c) TimeSeriesData

TimeSeriesData element is a main body of the Package and contains Time Series Data defined in Clause 6.

TimeSeriesData element also represents a group of DataSet and is grouped by Data Channel List editions. Once Data Channel List is updated and the updates will affect the value of Time Series Data, this element shall be newly generated.

d) TabularData

TabularData element contains a list of values defined in 6.2.

Tabular means that an ordered list of measurement value that is reported with the same timestamp and the same update interval.

TabularData shall be grouped by the update interval.

e) EventData

EventData element contains a list of (typically) irregularly updated values where a reference to the Data Channel List (LocalID or ShortID) is given for each data point (see 6.2 for details).

(6) DataSet

DataSet is a set of measurement values having the same timestamp. There are two types of DataSet, one is for TabularData, and the other is for EventData.

These elements are arranged in the hierarchic structure shown in Figure 3.

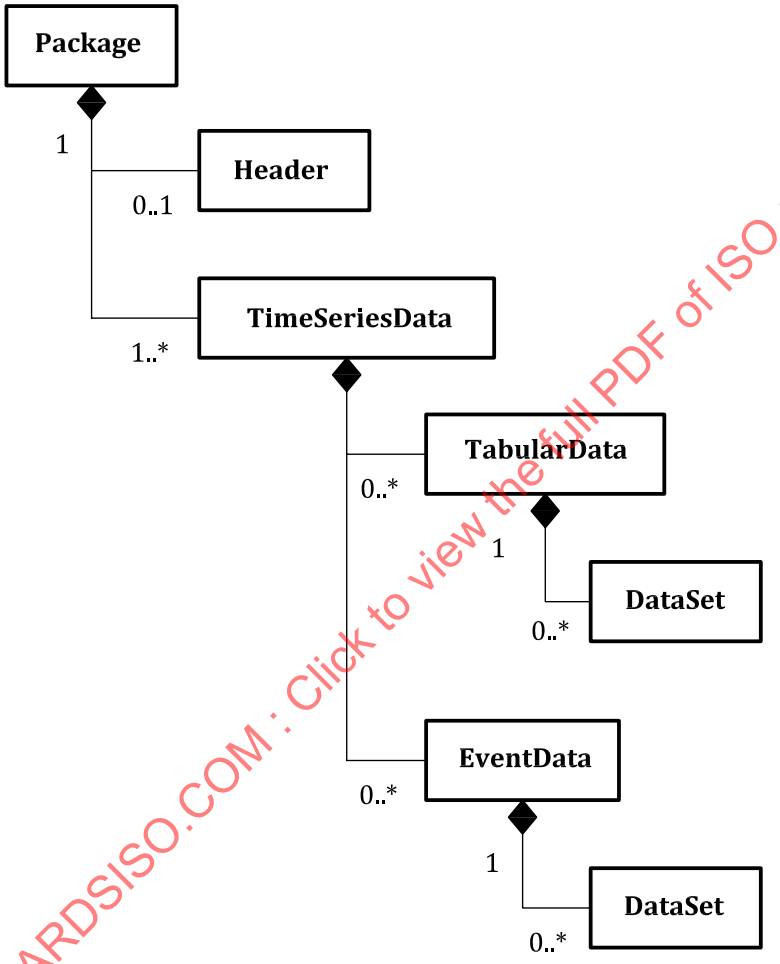

Figure 3 — Model of Time Series Data

7.5.2 Logical structure

Time Series Data shall have the logical structure shown in Figure 4.

Time Series Data has Package. Package has Header and TimeSeriesData. TimeSeriesData includes one or more DataSet. DataSet has one TimeStamp and one or more Data.

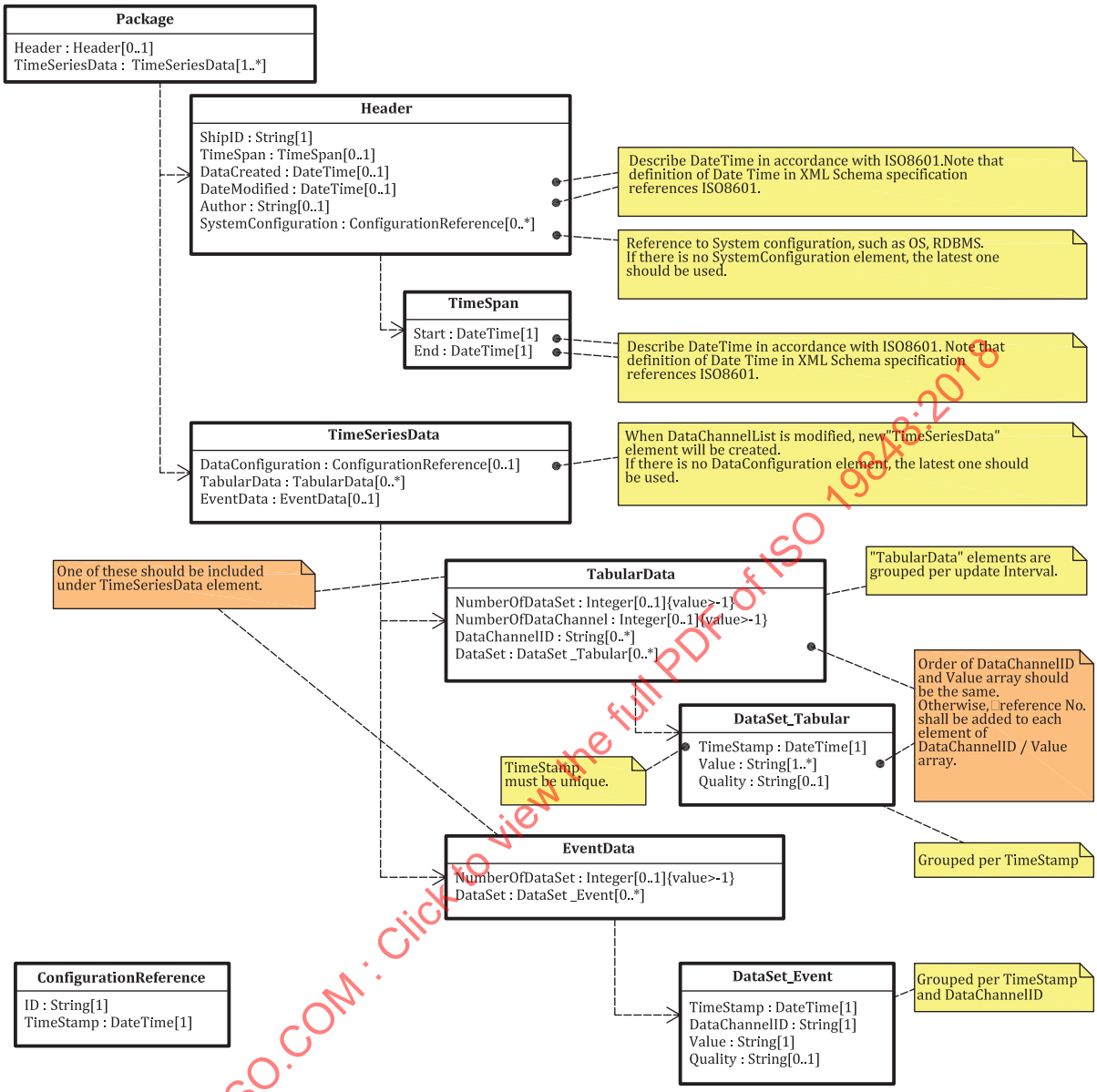

Figure 4 — Logical structure of Time Series Data

Details of each element are as follows.

a) Package structure

Name	Data type	Note	Mandatory/ Option	Max count
Header	b) Header	See b). For periodic data exchange, this element can be omitted to reduce data size.	Optional	1
TimeSeriesData	e) TimeSeriesData	See e). TimeSeriesData is grouped by DataChannelList editions.	Mandatory	*

b) Header structure

Name	Data type	Note	Mandatory/ Option	Max count
ShipID	String	IMO number, HIN, etc.	Mandatory	1
TimeSpan	c) TimeSpan	See c).	Optional	1
DateCreated	DateTime	Date when Package is created.	Optional	1
DateModified	DateTime	Date when Package is modified.	Optional	1
Author	String	Author of data.	Optional	1
SystemConfiguration	d) ConfigurationReference	See d). Reference to the configuration of systems, such as OS, RDBMS, etc.	Optional	*

c) TimeSpan structure

Name	Data type	Note	Mandatory/ Option	Max count
Start	DateTime	Timestamp of the oldest Data Set.	Mandatory	1
End	DateTime	Timestamp of the newest Data Set.	Mandatory	1

d) ConfigurationReference structure

Name	Data type	Note	Mandatory/ Option	Max count
ID	String	Identifier of the configuration.	Mandatory	1
TimeStamp	DateTime	Modified date & time of the configuration.	Mandatory	1

e) TimeSeriesData structure

Name	Data type	Note	Mandatory/ Option	Max count
DataConfiguration	d) ConfigurationReference	See d). Reference to the DataChannel-List. If there is a no reference, the latest DataChannelList shall be used.	Optional	1
TabularData	f) TabularData	See f) and Clause 6.	Optional	*
EventData	g) EventData	See g) and Clause 6.	Optional	1

f) TabularData structure

Name	Data type	Note	Mandatory/ Option	Max count
NumberOfDataSet	NonNegativeInteger	Number of DataSet in the TabularData element.	Optional	1
NumberOfDataChannel	NonNegativeInteger	Number of DataChannel in the TabularData element.	Optional	1
DataChannelID	String	Array of one of the Data Channel ID in 5.2. Order of DataChannelID elements shall be same as f) TabularData/Value. If the order of DataChannelID elements cannot be specified, reference number shall be added. Further, If the order of DataChannelID elements can be fixed and never changed, these elements may be omitted.	Optional	*
DataSet	h) DataSet_Tabular	See Clause 6.	Optional	*

g) EventData structure

Name	Data type	Note	Mandatory/ Option	Max count
NumberOfDataSet	NonNegativeInteger	Number of DataSet in the EventData element.	Optional	1
DataSet	i) DataSet_Event	See 6.3 b).	Optional	*

h) DataSet_Tabular structure

Name	Data type	Note	Mandatory/ Option	Max count
Timestamp	DateTime	Time of measurement.	Mandatory	1
Value	String	Array of measurement value of each data channel at TimeStamp. Order of Value elements shall be same as f) TabularData/DataChannelID. If the order of Value elements cannot be specified, reference number shall be added.	Mandatory	*
Quality	String	Data quality of the Value above.	Optional	1

i) DataSet_Event structure

Name	Data type	Note	Mandatory/ Option	Max count
Timestamp	DateTime	Time of measurement.	Mandatory	1
DataChannelID	String	One of the Data Channel ID in 5.2.	Mandatory	1
Value	String	Measurement value.	Mandatory	1
Quality	String	Data quality of the Value above.	Optional	1

Annex A
(Normative)

Implementation

A.1 General

A.1.1 Exchanging Data

Data which have structure as written in Clause 8, will be implemented by XML, JSON or CSV.

Regardless of the implementation language, data shall comply as follows.

— Data shall be written by text and encoded by UTF-8 WITHOUT Byte Order Mark (BOM).

XML/JSON is recommended to exchange Data Channel List and snap shot(*) of Time Series Data (*: contains single Data Set or a few number of Data Sets). When using JSON, since there is no schema validation mechanism, it is recommended to develop applications with an environment that can communicate with actual shipboard data server. CSV is recommended for especially large Time Series Data. In this case, Data Channel List shall be shared in advance. XML/JSON is recommended for exchanging Data Channel List.

A.1.2 Exchanging data as a file

The data file shall be designed to be able to process in various file systems.

To do this, the file shall comply with the following requirements.

— The name of the file shall be case-insensitive.

— The name of the file shall not start with "." (period).

— The name of the file shall not contain [\ ] [/] [:] [*] [?] ["] [<] [>] [|] and any white space characters.

— The length of the name of the file shall be less than or equal to 255 characters.

A.2 XML implementation

A.2.1 General

In this Clause, requirements are shown of cases in which the data formats discussed in Clause 7 are implemented with the use of XML and XML Schemas.

XML Schema is formulated in accordance with the rules described below.

— The minimum and maximum numbers of occurrences shall be defined.

— XML Schema shall be created to command XML to add Namespace to all elements and attributes.

EXAMPLE <**nr**:NamingRule **nr**:ID="Naming_Rule"/>

— Naming Rule specific elements and attributes should belong to its own Namespace.

— Any newline or indent characters in the file shall not be treated as significant information.

A.2.2 Standard & XML Schema data type

Standard data type in 7.3 is replaced with XML Schema data type in Table A.1.

Table A.1 — Correspondences between standard data type and XML Schema data type

Standard Data Type	XML Schema Data Type	Note
Integer	Integer	Integer number
PositiveInteger	PositiveInteger	Integer number that is 1 or larger
NonNegativeInteger	NonNegativeInteger	Integer number that is 0 or larger
Real	Decimal	Decimal number
DateTime	DateTime	ISO 8601 time and date
CharacterString	String	Random string of characters
Boolean	Boolean	Truth value
Null	XML Schema <element name="example" type="float" nillable="true"/> XML Document <example nil="true"/>	When designating Null, designate Attribute nillable=”true” in the Element definition of XML Schema, and Attribute nillable=”true” in the corresponding Element in XML Document.

A.2.3 Data Channel List

a) Namespace

The following Namespace shall be added to all the elements and attributes in Data Channel List.

XMLNamespace:sdd = urn:ISO19848:Ship_Data_Definition

In case of using Naming Rule specific elements/attributes, namespace shall be to avoid duplication.

EXAMPLE XMLNamespace:nr = urn:ISO19848:Ship_Data_Definition:Naming_Rule

b) XML Schema

Data structures in 7.4 shall be defined and validated by using following XML Schema.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:sdd="urn:ISO19848:SHIP_DATA_DEFINITION"
           attributeFormDefault="unqualified" elementFormDefault="qualified"
           targetNamespace="urn:ISO19848:SHIP_DATA_DEFINITION"
           xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <!--7.4.2 a) Package structure-->
  <xs:element name="Package" type="sdd:Package"/>

  <!--7.4.2 a) Package structure-->
  <xs:complexType name="Package">
    <xs:sequence>
      <!--7.4.2 b) Header structure-->
      <xs:element name="Header" type="sdd:Header" minOccurs="1" maxOccurs="1"/>
      <!--7.4.2 d) DataChannelList structure-->
      <xs:element name="DataChannelList" type="sdd:DataChannelList" minOccurs="1"
maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>

  <!--7.4.2 b) Header structure-->
  <xs:complexType name="Header">
    <xs:sequence>
      <!--IMO Number, HIN or other ship's identification number-->
      <xs:element name="ShipID" type="xs:string"  minOccurs="1" maxOccurs="1"/>
```

```
      <!--Identifier of this list-->
      <xs:element name="DataChannelListID" type="sdd:ConfigurationReference"
            minOccurs="1" maxOccurs="1"/>
      <!--Author of data-->
      <xs:element name="Author" type="xs:string"  minOccurs="0" maxOccurs="1"/>
      <!--Date when data are created-->
      <xs:element name="DateCreated" type="xs:dateTime"  minOccurs="0" maxOccurs="1"/>
      <!--Extension point for custom headers-->
      <xs:any processContents="lax" namespace="##other" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <!--7.4.2 c) ConfigurationReference structure-->
  <xs:complexType name="ConfigurationReference">
    <xs:sequence>
      <xs:element name="ID" type="xs:string"  minOccurs="1" maxOccurs="1"/>
      <xs:element name="TimeStamp" type="xs:dateTime"  minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>

  <!--7.4.2 d) DataChannelList structure-->
  <xs:complexType name="DataChannelList">
    <xs:sequence>
      <!--7.4.2 e) DataChannel structure-->
      <xs:element name="DataChannel" type="sdd:DataChannel"  minOccurs="1"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <!--7.4.2 e) DataChannel structure-->
  <xs:complexType name="DataChannel">
    <xs:sequence>
      <!--7.4.2 f) DataChannelID structure-->
      <xs:element name="DataChannelID" type="sdd:DataChannelID" minOccurs="1"
maxOccurs="1"/>
      <!--7.4.2 g) Property structure-->
      <xs:element name="Property" type="sdd:Property" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>

  <!--7.4.2 f) DataChannelID structure-->
  <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"/>
      <!--7.4.2 m) NameObject structure-->
      <xs:element name="NameObject" type="sdd:NameObject" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>

  <!--7.4.2 g) Property structure-->
  <xs:complexType name="Property">
    <xs:sequence>
      <!--5.3 a) Identifier of Data Channel Type, such as raw numeric value,
                              average value, alarm and status, etc.-->
      <xs:element name="DataChannelType" type="sdd:DataChannelType" minOccurs="1"
maxOccurs="1"/>
      <!--5.3 b) Formats are for describing data formats-->
      <!--Assumed data types are "Float","Integer","Boolean","Text" and "Symbol"-->
      <xs:element name="Format" type="sdd:Format"  minOccurs="1" maxOccurs="1"/>
      <!--5.3 c) Range of measurement value-->
      <xs:element name="Range" type="sdd:Range" minOccurs="0" maxOccurs="1"/>
      <!--5.3 d) Unit and quantity of measurement value-->
      <xs:element name="Unit" type="sdd:Unit"  minOccurs="0" maxOccurs="1" />
      <!--5.3 e) Name of data quality evaluation scheme-->
      <xs:element name="QualityCoding" type="xs:string"  minOccurs="0" maxOccurs="1"/>
      <!--5.3 f) Names assigned in on-board control systems and other instruments-->
      <xs:element name="Name" type="xs:string"  minOccurs="0" maxOccurs="1"/>
      <!--5.3 g) Remarks shall contains location, manufacturer and type of equipment-->
      <xs:element name="Remarks" type="xs:string"  minOccurs="0" maxOccurs="1"/>
```

```
      <!--Extension point for custom properties-->
      <xs:any processContents="lax" namespace="##other" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <!--7.4.2 h) DataChannelType structure-->
  <!--5.3 a) Identifier of Data Channel Type, such as raw numeric value,
                                average value, alarm and status, etc.-->
  <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>

  <!--7.4.2 i) Format structure-->
  <!--5.3 b) Format of measurement value-->
  <xs:complexType name="Format">
    <xs:sequence>
      <!--Type of the Format-->
      <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>
      <!--7.4.2 j) Restriction of the value-->
      <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>

  <!--7.4.2 k) Range structure-->
  <!--5.3 c) Range of measurement value-->
  <xs:complexType name="Range">
    <xs:sequence>
      <!--Higher limit of measurement value-->
      <xs:element name="High" type="sdd:emptyOrDecimal" minOccurs="1" maxOccurs="1"/>
      <!--Lower limit of measurement value-->
      <xs:element name="Low" type="sdd:emptyOrDecimal" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>

  <!--7.4.2 l) Unit structure-->
  <!--5.3 d) Unit and quantity of measurement value-->
  <xs:complexType name="Unit">
    <xs:sequence>
      <!--Symbol of the unit-->
      <xs:element name="UnitSymbol" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <!--Quantity name of the measurement value, that is defined in ISO80000 or Table2 -->
      <xs:element name="QuantityName" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <!--Extension point for custom Unit element, such as Quantity Symbol, Scale Factor,
etc.-->
      <xs:any processContents="lax" namespace="##other" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <!--7.4.2 m) NameObject structure-->
  <xs:complexType name="NameObject">
    <xs:sequence>
      <!--Name of NamingRule, see 5.2.3-->
      <xs:element name ="NamingRule" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <!--Extension point for custom NameObjects, Definition of LocalDataName structure-->
      <xs:any processContents="lax" namespace="##other" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <!--empty value 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>
```

c) XML representation

This example uses namespace:nr for custom elements. Custom elements are written in *Italic* for reference.

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

```
    <DataChannelListID>
      <ID>DataChannelList.xml</ID>
      <TimeStamp>2016-01-01T00:00:00Z</TimeStamp>
    </DataChannelListID>
    <Author>Author1</Author>
    <DateCreated>2015-12-01T00:00:00+00:00</DateCreated>
    <nr:CustomHeaderElement>Vender specific headers</nr:CustomHeaderElement>
  </Header>
  <DataChannelList>
    <DataChannel>
      <DataChannelID>
        <LocalID>/Naming_Rule/MainEngine/AirCooler1/CoolingFreshWater/Outlet/Temp
</LocalID>
        <ShortID>0010</ShortID>
        <NameObject>
          <NamingRule>Naming_rule</NamingRule>
          <nr:CustomNameObject>Vender specific NameObject</nr:CustomNameObject>
        </NameObject>
      </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>M/E #1 Air Cooler CFW OUT Temp</Name>
        <Remarks> Location: ECR, Manufacturer: AAA Company, Type: TYPE-AAA </Remarks>
        <nr:CustomPropertyElement>Vender specific Property</nr:CustomPropertyElement>
      </Property>
    </DataChannel>
    <DataChannel>
      <DataChannelID>
        <LocalID>/Naming_Rule/MainEngine/AirCooler1/CoolingFreshWater/Outlet/Press
</LocalID>
        <ShortID>0020</ShortID>
        <NameObject>
          <NamingRule>Naming_Rule</NamingRule>
          <nr:CustomNameObject>Vender specific NameObject</nr:CustomNameObject>
        </NameObject>
      </DataChannelID>
      <Property>
        <DataChannelType>
          <Type>Average</Type>
          <UpdateCycle>60</UpdateCycle>
          <CalculationPeriod>3600</CalculationPeriod>
        </DataChannelType>
        <Format>
          <Type>Integer</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> M/E #1 Air Cooler CFW OUT Average Temp </Name>
        <Remarks> Location: ECR, Manufacturer: AAA Company, Type: TYPE-AAA </Remarks>
        <nr:CustomPropertyElement>Vender specific Property</nr:CustomPropertyElement>
      </Property>
    </DataChannel>
    <DataChannel>
      <DataChannelID>
        <LocalID>/Naming_Rule/MainEngine/AirCooler1/CoolingFreshWater/Outlet/Temp/Alert
</LocalID>
        <ShortID>0011</ShortID>
        <NameObject>
          <NamingRule>Naming_Rule</NamingRule>
          <nr:CustomNameObject>Vender specific NameObject</nr:CustomNameObject>
        </NameObject>
      </DataChannelID>
      <Property>
        <DataChannelType>
          <Type>Alert</Type>
        </DataChannelType>
        <Format>
          <Type>String</Type>
          <Restriction>
            <Enumeration>High</Enumeration>
            <Enumeration>Low</Enumeration>
            <Enumeration>Normal</Enumeration>
          </Restriction>
        </Format>
        <QualityCoding>OPC_QUALITY</QualityCoding>
        <Name> M/E #1 Air Cooler CFW OUT Temp Status </Name>
        <Remarks> Location: ECR, Manufacturer: AAA Company, Type: TYPE-AAA </Remarks>
        <nr:CustomPropertyElement>Vender specific Property</nr:CustomPropertyElement>
      </Property>
    </DataChannel>
    <DataChannel>
      <DataChannelID>
        <LocalID>/Naming_Rule/MainEngine/AirCooler1/CoolingFreshWater/Outlet/Press/Alert
</LocalID>
        <ShortID>0021</ShortID>
        <NameObject>
          <NamingRule>Naming_Rule</NamingRule>
          <nr:CustomNameObject>Vender specific NameObject</nr:CustomNameObject>
        </NameObject>
      </DataChannelID>
      <Property>
        <DataChannelType>
          <Type>Alert</Type>
        </DataChannelType>
        <Format>
          <Type>String</Type>
          <Restriction>
            <Enumeration>High</Enumeration>
            <Enumeration>Low</Enumeration>
            <Enumeration>Normal</Enumeration>
          </Restriction>
        </Format>
        <QualityCoding>OPC_QUALITY</QualityCoding>
        <Name> M/E #1 Air Cooler CFW OUT Press Status </Name>
        <Remarks> Location: ECR, Manufacturer: AAA Company, Type: TYPE-AAA </Remarks>
        <nr:CustomPropertyElement>Vender specific Property</nr:CustomPropertyElement>
      </Property>
    </DataChannel>
  </DataChannelList>
```

```
</Package>
```

A.2.4 Time Series Data

a) Namespace

In Time Series Data, the following Namespace shall be added.

XMLNamespace:**sdt**=urn:ISO19848:Ship_Data_Transport

b) XML Schema

Data structures in 7.5 shall be defined and validated by using following XML Schema.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:sdt="urn:ISO19848:SHIP_DATA_TRANSPORT"
           attributeFormDefault="unqualified" elementFormDefault="qualified"
           targetNamespace="urn:ISO19848:SHIP_DATA_TRANSPORT"
           xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <!--8.4.2(1) Package structure-->
  <xs:element name="Package" type="sdt:Package"/>

  <!--8.4.2(1) Package structure-->
  <xs:complexType name="Package">
    <xs:sequence>
      <!--8.4.2(2) Header structure-->
      <xs:element name="Header" type="sdt:Header"
                  minOccurs="0" maxOccurs="1"/>
      <!--8.4.2(5) TimeSeriesData structure -->
      <xs:element name="TimeSeriesData" type="sdt:TimeSeriesData"
                  minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <!--8.4.2(2) Header structure-->
  <xs:complexType name="Header">
    <xs:sequence>
      <!--IMO Number, HIN or other ship's identification number-->
      <xs:element name="ShipIDN" type="xs:string"
                  minOccurs="1" maxOccurs="1"/>
      <!--8.4.2(3) TimeSpan structure-->
      <xs:element name="TimeSpan" type="sdt:TimeSpan"
                  minOccurs="0" maxOccurs="1"/>
      <!--Date when data are created-->
      <xs:element name="DateCreated" type="xs:dateTime"
                  minOccurs="0" maxOccurs="1"/>
      <!--Date when data are modified-->
      <xs:element name="DateModified" type="xs:dateTime"
                  minOccurs="0" maxOccurs="1"/>
      <!--Author of data-->
      <xs:element name="Author" type="xs:string"
                  minOccurs="0" maxOccurs="1"/>
      <!--8.4.2(4) Reference to system configuration -->
      <xs:element name="SystemConfiguration" type="sdt:ConfigurationReference"
                  minOccurs="0" maxOccurs="unbounded"/>
      <!--Extension point for custom headers-->
      <xs:any processContents="lax" namespace="##other" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <!--8.4.2(3) TimeSpan structure-->
  <xs:complexType name="TimeSpan">
    <xs:sequence>
      <!--TimeStamp of the oldest DataSet-->
      <xs:element name="Start" type="xs:dateTime" minOccurs="1" maxOccurs="1"/>
      <!--TimeStanp of the newest DataSet-->
      <xs:element name="End" type="xs:dateTime" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
```

```
  </xs:complexType>

  <!--8.4.2(4) ConfigurationReference structure-->
  <xs:complexType name="ConfigurationReference">
    <xs:sequence>
      <xs:element name="ID" type="xs:string"  minOccurs="1" maxOccurs="1"/>
      <xs:element name="TimeStamp" type="xs:dateTime"  minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>

  <!--DataChannelID structure-->
  <!--Value must be Local ID or Shor ID strings. see 6.1 for the details.-->
  <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>

  <!--8.4.2(5) TimeSeriesData structure -->
  <xs:complexType name="TimeSeriesData">
    <xs:sequence>
      <!--8.4.2(4) ConfigurationReference structure -->
      <xs:element name="DataConfiguration" type="sdt:ConfigurationReference"
                  minOccurs="0" maxOccurs="1" />
      <!--8.4.2(6) TabularData structure -->
      <xs:element name="TabularData" type="sdt:TabularData"
                  minOccurs="0" maxOccurs="unbounded" />
      <!--8.4.2(7) EventData structure -->
      <xs:element name="EventData" type="sdt:EventData"
                  minOccurs="0" maxOccurs="1" />
      <!--Extension point for custom data kinds-->
      <xs:any processContents="lax" namespace="##other" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <!--8.4.2(6) TabularData structure -->
  <xs:complexType name="TabularData">
    <xs:sequence>
      <!--Number of DataSet elements under the hierarchy -->
      <xs:element name="NumberOfDataSet" type="xs:nonNegativeInteger"
                  minOccurs="0" maxOccurs="1" />
      <!--Number of DataChannels for each DataSources-->
      <xs:element name="NumberOfDataChannel" type="xs:nonNegativeInteger"
                  minOccurs="0" maxOccurs="1"/>
      <!--8.4.2(6) Identifiers of target DataChannels-->
      <xs:element name="DataChannelID" type="sdt:DataChannelID"
                  minOccurs="0" maxOccurs="unbounded"/>
      <!--8.4.2(8) DataSet structure -->
      <xs:element name="DataSet" type="sdt:DataSet_Tabular"
                  minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>

  <!--8.4.2(7) EventData structure -->
  <xs:complexType name="EventData">
    <xs:sequence>
      <!--Number of DataSet elements under the hierarchy -->
      <xs:element name="NumberOfDataSet" type="xs:nonNegativeInteger"
                  minOccurs="0" maxOccurs="1" />
      <!--8.4.2(9) DataSet structure -->
      <xs:element name="DataSet" type="sdt:DataSet_Event"
                  minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>

  <!--8.4.2(8) DataSet structure for Tabular Data -->
  <xs:complexType name="DataSet_Tabular">
```

```
    <xs:sequence>
      <!--Measurement value -->
      <!--Measurement value can be empty-->
      <xs:element name="Value" type="sdt:Value"
                  minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
    <!--ISO 8601 date & time structure shall be used-->
    <xs:attribute name="timeStamp" type="xs:dateTime" use="required"/>
  </xs:complexType>

  <!--8.4.2(9) DataSet structure for Event Data-->
  <xs:complexType name="DataSet_Event">
    <xs:sequence>
      <xs:element name="DataChannelID" type="xs:string"
                  minOccurs="1" maxOccurs="1"/>
      <!--Measurement value -->
      <!--Measurement value can be empty-->
      <xs:element name="Value" type="xs:string"
                  minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
    <!--ISO 8601 date & time structure shall be used-->
    <xs:attribute name="timeStamp" type="xs:dateTime" use="required"/>

    <xs:attribute name="quality" type="xs:string" use="optional"/>
  </xs:complexType>

  <!--Value structure for Tabular Data -->
  <xs:complexType name="Value">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <!--Ordinal number of the Value array.
        Order of the array shall be the same as TabularData/DataChannelID array.-->
        <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>
```

c) XML representation

Example of XML representation is as follows.

```
<?xml version="1.0" encoding="utf-8" ?>
<Package xmlns="urn:ISO19848:SHIP_DATA_TRANSPORT">
  <Header>
    <ShipID>IMO1234567</ShipID>
    <TimeSpan>
      <Start>2016-01-01T12:00:00Z</Start>
      <End>2016-01-03T12:00:00Z</End>
    </TimeSpan>
    <DateCreated>2016-01-03T12:00:00Z</DateCreated>
    <DateModified>2016-01-03T12:00:00Z</DateModified>
    <Author>Shipboard data server</Author>
    <SystemConfiguration>
      <ID>SystemConfiguration.xml</ID>
      <TimeStamp>2016-01-01T00:00:00Z</TimeStamp>
    </SystemConfiguration>
    <SystemConfiguration>
      <ID>SystemConfiguration.xml</ID>
      <TimeStamp>2016-01-03T00:00:00Z</TimeStamp>
    </SystemConfiguration>
  </Header>
  <TimeSeriesData>
    <DataConfiguration>
      <ID>DataChannelList.xml</ID>
      <TimeStamp>2016-01-01T00:00:00Z</TimeStamp>
    </DataConfiguration>
    <TabularData>
      <NumberOfDataSet>2</NumberOfDataSet>
      <NumberOfDataChannel>2</NumberOfDataChannel>
```

```
      <!--ShortID of the DataChannel -->
      <DataChannelID id="1">0010</DataChannelID>
      <DataChannelID id="2">0020</DataChannelID>
      <DataSet timeStamp="2016-01-01T12:00:00Z">
        <!--TabularData/DataSet/Value@ref references TabularData/DataChannelID@id -->
        <!--measurement value of Data Channel:0010 at 2016-01-01T12:00:00Z -->
        <Value ref="1" quality="0">100.0</Value>
        <!--measurement value of Data Channel:0020 at 2016-01-01T12:00:00Z -->
        <Value ref="2" quality="0">200.0</Value>
      </DataSet>
      <DataSet timeStamp="2016-01-02T12:00:00Z">
        <!--measurement value of Data Channel:0010 at 2016-01-02T12:00:00Z -->
        <Value ref="1" quality="0">100.5</Value>
        <!--measurement value of Data Channel:0020 at 2016-01-02T12:00:00Z -->
        <Value ref="2" quality="0">205.0</Value>
      </DataSet>
    </TabularData>
    <TabularData>
      <NumberOfDataSet>3</NumberOfDataSet>
      <NumberOfDataChannel>1</NumberOfDataChannel>
      <DataChannelID id="1">0010</DataChannelID>
      <DataSet timeStamp="2016-01-01T12:00:00Z">
        <Value ref="1" quality="0">100.0</Value>
      </DataSet>
      <DataSet timeStamp="2016-01-02T00:00:00Z">
        <Value ref="1" quality="0">100.2</Value>
      </DataSet>
      <DataSet timeStamp="2016-01-02T12:00:00Z">
        <Value ref="1" quality="0">100.5</Value>
      </DataSet>
    </TabularData>
    <EventData>
      <NumberOfDataSet>3</NumberOfDataSet>
      <DataSet timeStamp="2016-01-01T12:00:01Z" quality="0">
        <DataChannelID>0011</DataChannelID>
        <Value>HIGH</Value>
      </DataSet>
      <DataSet timeStamp="2016-01-01T12:00:01Z" quality="0">
        <DataChannelID>0021</DataChannelID>
        <Value>HIGH</Value>
      </DataSet>
      <DataSet timeStamp="2016-01-01T12:00:23Z" quality="0">
        <DataChannelID>0011</DataChannelID>
        <Value>NORMAL</Value>
      </DataSet>
    </EventData>
  </TimeSeriesData>
  <TimeSeriesData>
    <DataConfiguration>
      <ID>DataChannelList.xml</ID>
      <TimeStamp>2016-01-03T00:00:00Z</TimeStamp>
    </DataConfiguration>
    <TabularData>
      <NumberOfDataSet>1</NumberOfDataSet>
      <NumberOfDataChannel>2</NumberOfDataChannel>
      <DataChannelID id="1">0010</DataChannelID>
      <DataChannelID id="2">0020</DataChannelID>
      <DataSet timeStamp="2016-01-03T12:00:00Z">
        <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="2016-01-03T00:00:00Z">
        <Value ref="1" quality="0">100.8</Value>
      </DataSet>
      <DataSet timeStamp="2016-01-03T12:00:00Z">
        <Value ref="1" quality="0">101.0</Value>
```

```
      </DataSet>
    </TabularData>
  </TimeSeriesData>
</Package>
```

NOTE This example shows that both system configurations and Data Channel List are modified at 2016-01-03T00:00:00Z, therefore this example has two Time Series Data elements. Further, Data Channel 0010, 0020 have 24 hour update interval and 0110 have 12 hour update interval. Therefore this example has two TabularData elements for each TimeSeriesData element.

A.3 JSON implementation

A.3.1 General

This clause defines the way to describe the data by using JSON.

JSON implementation is not mandatory but shall be implemented in accordance with the following.

— Extension of the file shall be “json”.

— Any newline or indent characters in the file shall not be treated as significant information.

— XML element with namespace shall convert to JSON element as it is. For example, <xml: Element>Value</xml:Element> is converted to { "xml:Element" : "Value" }.

A.3.2 Data Channel List

JSON representation example is given below.

```
{
   "Package": {
      "Header": {
         "ShipID": "IMO1234567",
         "DataChannelListID": {
            "ID": "DataChannelList.xml",
            "TimeStamp": "2016-01-01T00:00:00Z"
         },
         "Author": "Author1",
         "DateCreated": "2015-12-01T00:00:00Z",
         "nr:CustomHeaderElement": "Vender specific headers"
      },
      "DataChannelList": {
         "DataChannel": [{
            "DataChannelID": {
               "LocalID":
 "/Naming_Rule/MainEngine/AirCooler1/CoolingFreshWater/Outlet/Temp",
               "ShortID": "0010",
               "NameObject": {
                  "NamingRule": "Naming_Rule",
                  "nr:CustomNameObject": "Vender specific NameObject"
               }
            },
            "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": "M/E #1 Air Cooler CFW OUT Temp",
               "Remarks": " Location: ECR, Manufacturer: AAA Company, Type: TYPE-AAA ",
               "nr:CustomPropertyElement": "Vender specific Property"
            }
         }, {
            "DataChannelID": {
               "LocalID":
"/Naming_Rule/MainEngine/AirCooler1/CoolingFreshWater/Outlet/Press",
               "ShortID": "0020",
               "NameObject": {
                  "NamingRule": "Naming_Rule",
                  "nr:CustomNameObject": "Vender specific NameObject"
               }
            },
            "Property": {
               "DataChannelType": {
                  "Type": "Average",
                  "UpdateCycle": "60",
                  "CalculationPeriod": "3600"
               },
               "Format": {
                  "Type": "Integer",
                  "Restriction": {
                     "FractionDigits": "1",
                     "MaxInclusive": "200.0",
                     "MinInclusive": "-150.0"
                  }
               },
               "Range": {
                  "High": "150.0",
                  "Low": "0.0"
               },
            "Unit": {
"UnitSymbol": "°C",
               "QuantityName": "Temp"
            },
            "QualityCoding": "OPC_QUALITY",
               "Name": " M/E #1 Air Cooler CFW OUT Average Temp ",
               "Remarks": " Location: ECR, Manufacturer: AAA Company, Type: TYPE-AAA ",
               "nr:CustomPropertyElement": "Vender specific Property"
            }
         }, {
            "DataChannelID": {
               "LocalID":
"/Naming_Rule/MainEngine/AirCooler1/CoolingFreshWater/Outlet/Temp/Alert",
               "ShortID": "0011",
               "NameObject": {
                  "NamingRule": "Naming_Rule",
                  "nr:CustomNameObject": "Vender specific NameObject"
               }
            },
            "Property": {
               "DataChannelType": {
                  "Type": "Alert"
               },
            "Format": {
              "Type": "String",
              "Restriction": {
                "Enumeration": [
                  "High",
                  "Low",
                  "Normal"
                ]
              }
```

```
            },
            "QualityCoding": "OPC_QUALITY",
                  "Name": " M/E #1 Air Cooler CFW OUT Temp Status ",
                  "Remarks": " Location: ECR, Manufacturer: AAA Company, Type: TYPE-AAA ",
                  "nr:CustomPropertyElement": "Vender specific Property"
               }
            }, {
               "DataChannelID": {
                  "LocalID":
"/Naming_Rule/MainEngine/AirCooler1/CoolingFreshWater/Outlet/Press/Alert",
                  "ShortID": "0021",
                  "NameObject": {
                     "NamingRule": "Naming_Rule",
                     "nr:CustomNameObject": "Vender specific NameObject"
                  }
               },
               "Property": {
                 "DataChannelType": {
                    "Type": "Alert"
                  },
            "Format": {
              "Type": "String",
              "Restriction": {
                "Enumeration": [
                  "High",
                  "Low",
                  "Normal"
                ]
              }
            },
            "QualityCoding": "OPC_QUALITY",
                  "Name": " M/E #1 Air Cooler CFW OUT Press Status ",
                  "Remarks": " Location: ECR, Manufacturer: AAA Company, Type: TYPE-AAA ",
                  "nr:CustomPropertyElement": "Vender specific Property"
               }
            }]
         }
      }
}
```

A.3.3 Time Series Data

Example of JSON representation is as follows.


```
{
   "Package": {
      "Header": {
         "ShipID": "IMO1234567",
         "TimeSpan": {
            "Start": "2016-01-01T12:00:00Z",
            "End": "2016-01-03T12:00:00Z"
         },
         "DateCreated": "2016-01-03T12:00:00Z",
         "DateModified": "2016-01-03T12:00:00Z",
         "Author": "Shipboard data server",
         "SystemConfiguration": [{
            "ID": "SystemConfiguration.xml",
            "TimeStamp": "2016-01-01T00:00:00Z"
         }, {
            "ID": " SystemConfiguration.xml ",
            "TimeStamp": "2016-01-03T00:00:00Z"
         }]
      },
      "TimeSeriesData": [{
         "DataConfiguration": {
            "ID": "DataChannelList.xml",
            "TimeStamp": "2016-01-01T00:00:00Z "
         },
         "TabularData": [{
            "NumberOfDataSet": "2",
            "NumberOfDataChannel": "2",
            "DataChannelID": ["0010", "0020"],
            "DataSet": [{
               "TimeStamp": "2016-01-01T12:00:00Z",
                  "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": "2016-01-01T12:00:00Z",
                  "Value": [ "100.0" ],
                  "Quality": ["0"]
            }, {
               "TimeStamp": "2016-01-02T00:00:00Z",
                  "Value": [ "100.2" ],
                  "Quality": ["0"]
            }, {
               "TimeStamp": "2016-01-02T12:00:00Z",
               "Value": [ "100.3" ],
                  "Quality": ["0"]
            }]
         }],
         "EventData": {
            "NumberOfDataSet": "3",
            "DataSet": [{
               "TimeStamp": "2016-01-01T12:00:01Z",
               "DataChannelID": "0011",
                  "Value": "HIGH",
                  "Quality": "0"
            }, {
               "TimeStamp": "2016-01-01T12:00:01Z",
               "DataChannelID": "0021",
                  "Value": "HIGH",
                  "Quality": "0"
```

```
            }, {
               "TimeStamp": "2016-01-01T12:00:23Z",
               "DataChannelID": "0011",
                  "Value": "NORMAL",
                  "Quality": "0"
            }]
         }
      }, {
         "DataConfiguration": {
            "ID": "DataChannelList.xml",
            "TimeStamp": "2016-01-03T00:00:00Z"
         },
         "TabularData": [{
            "NumberOfDataSet": "1",
            "NumberOfDataChannel": "2",
            "DataChannelID": ["0010", "0020"],
            "DataSet": [{
               "TimeStamp": "2016-01-03T12:00:00Z ",
                  "Value": [ "101.0", "210.0" ],
                  "Quality": ["0", "0"]
            }]
         }, {
            "NumberOfDataSet": "2",
            "NumberOfDataChannel": "1",
            "DataChannelID": ["0110"],
            "DataSet": [{
               "TimeStamp": "2016-01-03T00:00:00Z",
                  "Value": ["100.8"],
                  "Quality": ["0"]
            }, {
               "TimeStamp": "2016-01-03T12:00:00Z",
                  "Value": [ "101.0" ],
                  "Quality": ["0"]
            }]
         }]
      }]
   }
}
```


NOTE Same as XML implementation, this example shows that both system configuration and Data Channel List are modified at 2016-01-03T00:00:00Z. Therefore, this example has two Time Series Data elements. Further, Data Channel 0010, 0020 have 24-hour update interval and 0110 have 12-hour update interval. Therefore, this example has two TabularData elements for each TimeSeriesData element.

A.4 CSV implementation

A.4.1 General

This clause defines the way to describe the data by using CSV. CSV is only available for Time Series Data and shall be used in combination with Data Channel List described in XML/JSON.

CSV shall be implemented in accordance with the following.

— CSV shall be described in accordance with RFC 4180.

— Extension of the file shall be “csv”.

— Tabular Data and Event Data shall be described separately.

— CSV shall have the same number of columns in all rows.

— Header row (1st row) may be omitted when Data Channel in the data can clearly identified. This definition will be described in Alias List on ISO 19847.

A.4.2 Tabular Data

CSV describes Tabular Data as follows:

```
TimeStamp,      DataChannelID_1,  DataChannelID_2,  …,   DataChannelID_N
TimeStamp_1,    Value_1-1,        Value-1-2,        …,   …
…,              …,                …,                …,   …
TimeStamp_M,    …,                …,                …,   Value_M-N
```

where

TimeStamp_M is the measurement date and time;

DataChannelID_N is the one of the following: Universal ID, Local ID, Short ID;

Value_M-N is the measurement value of DataChannelID_N at TimeStamp_M.

Example of CSV for Tabular Data by using Short ID is as follows:

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

where 0010, 0020 is the Short ID of some Data Channel.

A.4.3 Event Data

CSV describes Event Data as follows:

```
TimeStamp,          DataChannelID,    Value
TimeStamp_1,        DataChannelID_1,  Value_1-1
…,                  …,                …
TimeStamp_M,        DataChannelID_N,  Value_M-N
```

where

TimeStamp_M is the measurement date and time;

DataChannelID_N is the one of the following: Universal ID, Local ID, Short ID;

Value_M-N is the measurement value of DataChannelID_N at TimeStamp_M.

Example of CSV for Event Data by using Short ID is as follows:

```
TimeStamp, DataChannelID,    Value
2016-01-01T12:00:01Z,0011,   HIGH
2016-01-01T12:00:01Z,0021,   HIGH
2016-01-01T12:01:23Z,0011,   NORMAL
```

where 0011, 0021 is the Short ID of some Data Channel.

Annex B
(informative)

Examples of Local ID definitions — jsmea_mac

B.1 Naming Rule

Local ID shall be named in accordance with rules that are unified respectively in each Naming Rule.

As an example of jsmea_mac Naming Rule, a method of naming Local ID by dividing Data into five Name Objects and combining these IDs is introduced.

Details of Name Object are shown in Table B.1.

Table B.1 — List of Categories

Name Object	Description	Example
System	Discrete equipment, systems, and units.	MainEngine, DieselGeneratorSet CoolingSeaWaterSystem
Component	Part of the System above or detail description of the system.	Cylinder TurboCharger FuelOilLine
Content	Contents that flow through pipelines or parts in the above component.	HeavyFuelOil CoolingFreshWater ExahstGas
Position	Position that indicates relative positions in System and Components.	Inlet Outlet Port
Item	Indicates temperatures, pressures and other measurement objects.	Temp Press Speed
Suffix (Option)	Only if Data Channels cannot be specified with the Elements shown above (from “System” to “Item”), suffix may be added after Local ID to identify the Data Channels. Suffix may consists of and DataChannelType as shown below. Suffix: ../<modifier>/<DataChannelType>	Modifier is typically used to describe correction parameter such as temperature for fuel oil viscosity, like "at15degC" .

According to the categorising and naming rules defined by “jsmea_mac”, “Main Engine #3 Air Cooler Cooling Fresh Water Outlet Temperature” for the ship whose IMO number is “IMO1234567” is as follows.

ShipID	Naming Rule	Local Data Name				
		System	Component	Content	Position	Item
IMO1234567	jsmea_mac	MainEngine	AirCooler3	CoolingFreshWater	Outlet	Temp

In this case, the <Universal ID> and <Local ID> are as follows.

<Universal ID> = http://IMO1234567/jsmea_mac/MainEngine/AirCooler3/CoolingFreshWater/Outlet/Temp//Inst

<Local ID> = /jsmea_mac/MainEngine/AirCooler3/CoolingFreshWater/Outlet/Temp//Inst

B.2 Name Object

When writing Local ID structure in Data Channel List, Name Object is added under Data Channel element.

Figure B.1 shows an example of adding Name Object in "jsmea_mac". In this example, three attributes —ID, Code and Name— are added under Name Object to describe details.

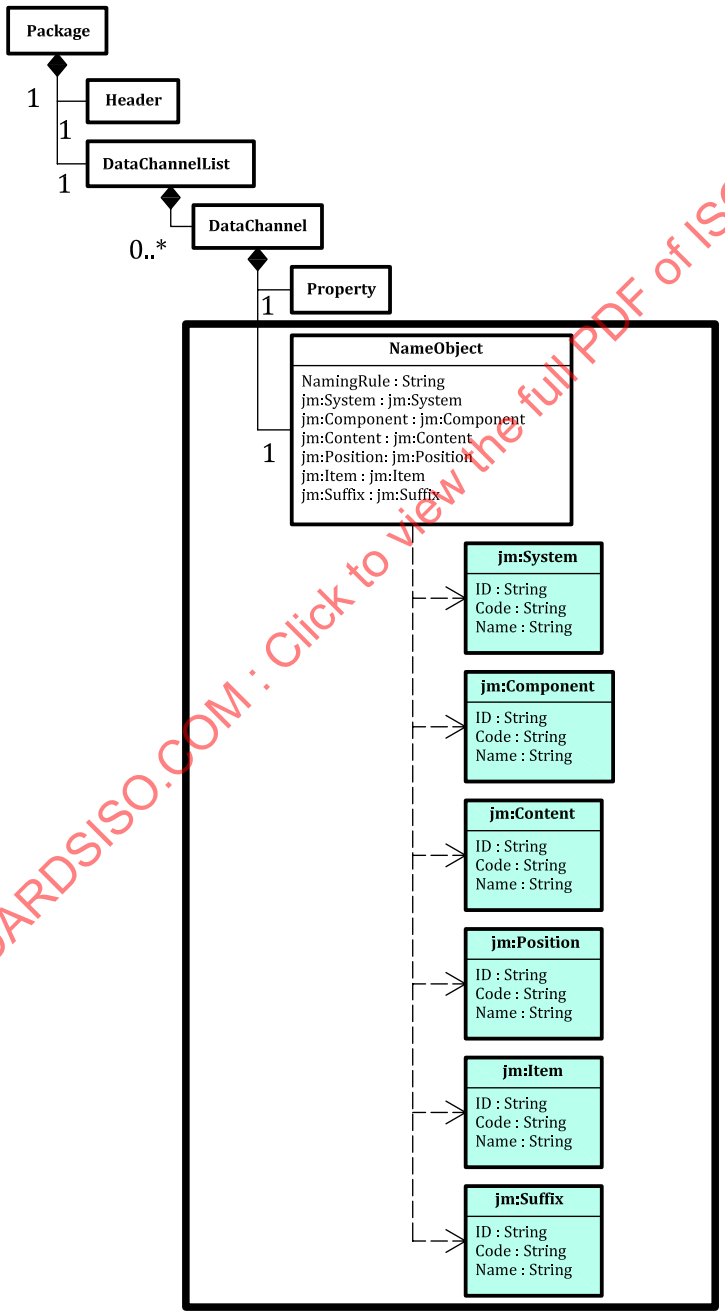

Figure B.1 — Logical structure of Data Channel List to which Name Object is added

B.3 Structure of engine and machinery codebook

Codebook is a list of standard names that corresponds to Name Object, and created for each Name Object.

In codebook: ID, which is part of Local ID; Code, machine-friendly name that corresponds to the ID; Name, that is for human readability; and its description are described.

EXAMPLE

ID	Code	Name	Description
MainEngine	1ME01	No.1 Main Engine	No. 1 Diesel engine for propulsion

Annex C
(informative)

Examples of LocalID definitions — DNVGL-VIS

C.1 General

The requirements regarding construction and contents of Local IDs under the DNVGL-VIS naming rule is provided at data.dnvgl.com/dnvgl-vis. The purpose of this informative Annex is to provide explanatory information and examples to assist implementers.

C.2 DNVGL-VIS introduction

VIS is an abbreviation for Vessel Information Systems, which has been used in DNV GL production tools for classification of vessels since 2005. VIS contains a hierarchy of functions and libraries of equipment, where a name and code is defined for each item. The coding is based on the Universal Decimal Classification (UDC) numbering system. The DNVGL-VIS is further described in documentation available at data.dnvgl.com/dnvgl-vis.

C.3 Local ID construction

C.3.1 Overall Local ID composition

When DNVGL-VIS is used as naming rule, the LocalID shall be constructed as follows:

Local ID = Function group/n*Sub-Function group/Suffix and quantity group, where n is an integer (0 to many) denoting how many times the Sub-Function group is repeated.

The naming elements within each group shall be separated by a plus character ("+"), whereas each group shall be separated by a slash character ("/").

Example of Local ID using DNVGL-VIS:

— 411.1/C101.31+2/ExhaustGasOut+t(C) *(Coded version)*

— PropulsionEngine/CylinderArrangement+No2/ExhaustGasOut+t(C) *(Verbose version)*

NOTE Further explanation of this and other examples are provided in C.4. below.

In order to avoid ambiguities for parser implementations, any space, plus or slash characters found in the naming elements sources (DNVGL-VIS or ISO 80000) shall be replaced by underscore character ("_") before constructing the Local ID.

C.3.2 Naming Elements groups

Three different Naming Elements groups are defined in the following:

a) Function group

Function group = <Function>+<Component>+<Location>

The Function group is mandatory and must start with one Function. The Component shall not be included if a Sub-Function group is appended. (This is because the same Component will be provided with the Sub-function at the next, and is thus redundant.) Although optional, it is highly

recommended that Location is used to separate between multiple instances of the given Function/Component pair, when applicable.

b) Sub-Function group

Sub-Function group = <Sub-function>+<Component>+<Location>

0 to n Sub-Function groups can be appended to the Function group. Each Sub-Function group must start with a Sub-Function. The Component is optional and may only be included in the last Sub-Function group. Although optional, it is highly recommended that Location is used to separate between multiple instances of the given Sub-Function/Component pair, when applicable.

Components assigned to a Function or Sub-Function can only be selected from those defined in VIS.

c) Suffix and quantity group

Suffix and quantity group = <Suffix>+<Quantity>(<UoM>)

The Suffix and quantity group is mandatory, but the individual elements (Suffix, Quantity, UoM) are optional. This implies that at least one of the elements must be used. The implementer must ensure that the Suffix and quantity group is constructed so that unique Local IDs are created.

The different Naming Elements that can be used to construct the above Naming Element groups have been further explained in Table C.1 below.

Table C.1 — Table C.1: Naming Elements used in DNVGL-VIS

Naming Element	Description/Example
Function	Functions represent the ability to perform or prevent certain actions by parts of a vessel. Functions are defined hierarchically. Function leaves may be assigned Components. The allowed Functions values are defined in “DNVGL-VIS Functions”. *Examples: 411.1 (Propulsion driver), 510 (Main electric power system)*
Component	Components are physical items that may exist on-board vessels. Components are organized according to “Disciplines” defined in “DNVGL-VIS Disciplines”. Components may provide a detailed description through sub-functions, which again may be assigned Components. The allowed Component values are defined in “DNVGL-VIS Equipment”. *Examples: C221 (Shaft), C101 (Reciprocating internal combustion engine), N52 (Gyro compass)*
Location	Locations refer to positions on-board a vessel. Three types of Locations are defined; absolute, generic and relative. The allowed Location values are defined in “DNVGL-VIS Locations”. Location may be omitted if not needed to identify the Function or Sub-Function group, e.g. if there is only one instance on-board. *Examples: S (Starboard), P (Port), 1,2,9*
Sub-Function	Sub-Functions express the ability to perform or prevent actions applicable by parts of a Component. Sub-functions are defined hierarchically. Sub-function leaves may be assigned Components. The allowed Sub-function values are defined in “DNVGL-VIS Equipment”. *Examples: C101.31 (Cylinder arrangement in a Reciprocating internal combustion engine), E15.3 (Transformer in a Shore Connection)*