INTERNATIONAL STANDARD

ISO 14817-1

First edition
2015-10-15

# Intelligent transport systems — ITS central data dictionaries —

## Part 1: Requirements for ITS data definitions

*Systèmes intelligents de transport — Dictionnaires de données centrales des ITS —*

*Partie 1: Exigences pour les définitions des données des ITS*

Reference number
ISO 14817-1:2015(E)

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

## 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 on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/TC 204, *Intelligent transport systems*.

This first edition of ISO 14817-1, together with ISO 14817-2, cancels and replaces ISO 14817:2002, which has been technically revised.

ISO 14817 consists of the following parts, under the general title *Intelligent transport systems — ITS data dictionaries*:

— *Part 1: Requirements for ITS data definitions*

— *Part 2: Governance of the Central ITS Data Concept Registry*

— *Part 3: Object identifier assignments for ITS data concepts*

# Introduction

## Background

This International Standard has been developed by ISO TC 204, in order to provide a framework for the documentation and registration of data that passes through system interfaces within the intelligent transport systems (ITS) domain. It is designed to maximize interoperability and facilitate information re-use across system interfaces.

## Vision statement

This International Standard envisions a harmonized approach to ITS data concepts to promote maximum interoperability of data within the ITS sector by the creation and maintenance of the “Central ITS Data Concept Registry” (CIDCR), supported by interface and application specific ITS data dictionaries, created and maintained in a common and interoperable form, and to ensure the minimization of duplication by clear rules for data concept definition and data concept registry management.

## Mission statement

The mission is to develop tools that will promote a holistic, integrated approach involving vehicle technology, infrastructure, and the road user to increase transport safety and efficiency. Specifically, this International Standard defines the principles and concepts; scope; field of application; rules and procedures; definition and concept of operation for the CIDCR and ITS functional data dictionaries; and makes provision for the migration of data concepts from ITS functional data dictionaries to the CIDCR so as to maximize interoperability and minimize proliferation of similar (but inconsistently defined) data concept entries.

This International Standard defines the framework, formats, and procedures used to define information and information exchanges within the ITS sector. This International Standard is designed to be used by the ITS community at large, but should be of special interest to application developers, equipment providers, and data concept registry managers.

This International Standard specifies a set of meta-attributes for ITS data concepts, as well as associated conventions and schemes that enable the description, standardization and management of all exchanged ITS data. Through consistent use of these common structures and associated conventions and schemes, interchange of data and information among the various ITS functional subsystems via their specific application systems can be maximized. This International Standard also supports re-use of data elements and other data concepts across various ITS functional subsystems and their specific application systems.

The formats and processes defined within this International Standard are consistent with implementation(s) of the ISO ITS System Architecture defined in the ISO 14813 Standardization deliverables, particularly ISO 14813-2 and ISO 14813-3. This does not preclude the application of data concept registries using alternative international, regional or national system architecture methodologies or techniques, indeed, common formats and processes will ease migration and interoperability between such approaches.

The ITS data concepts that populate the CIDCR or data dictionary may originate from a Computer-Aided Software Engineering (CASE) tool implementation of the ISO 14813 ITS Reference Architecture, from International Standards for ITS, from national implementations for ITS, or from the submission by relevant users. Data dictionary entries are not limited to those generated by object oriented methodologies.

## Document overview

This clause provides an overview of this International Standard. Clause 1 identifies the scope of this part of ISO 14817. Clause 2 identifies requirements for conformance to this part of ISO 14817. Clause 3 identifies references required for proper implementation of this part of ISO 14817. Clause 4 defines terms used in this part of ISO 14817 and Clause 5 lists the abbreviations.

Clause 6 declares the fundamental ITS data concepts and Clause 7 identifies meta-attributes used to document the data concepts declared in Clause 6.

Annex A prescribes which meta-attributes are required for each type of data concept. Annex B specifies the naming conventions and the process for converting among various naming conventions (e.g. between the ITS descriptive name and the ASN.1 name). Annex C contains a listing of preferred data concepts within the ITS domain. Annex D contains the rules for representing data in a data model, along with examples.

The Bibliography includes a list of documents related to this International Standard.

**Other parts**

ISO 14817-2 defines the operation of the Central ITS Data Concept Registry (CIDCR). ISO 14817-3 specifies how to assign object identifiers.

# Intelligent transport systems — ITS central data dictionaries —

# Part 1: Requirements for ITS data definitions

## 1 Scope

This part of ISO 14817 specifies the logical structure (framework) and the data content (substance) of intelligent transport systems (ITS) data dictionaries (DDs).

Specifically, this part of ISO 14817 specifies the following:

- framework used to identify and define all data concepts;
- meta-attributes used to describe, standardize and manage each of the data concepts defined within this framework;
- requirements used to record these definitions;
- naming conventions for the data concepts;
- a set of preferred data concepts within the ITS domain;
- data modelling method for defining ITS data concepts, when used.

DDs support data concepts derived from any number of international, regional or national system architecture methodologies and/or techniques. Common data formats and operating procedures will ease migration and interoperability between such approaches.

A data concept registry is an electronic data dictionary that supports some additional features. The CIDCR refers to the specific implementation of an ITS data concept registry that is operated under the auspices of ISO/TC 204. The term “data concept registries” may refer to the CIDCR and/or any other national or regional data concept registry that chooses to conform to this part of ISO 14817.

## 2 Conformance

This part of ISO 14817 prescribes a conceptual model, not a physical implementation. An implementation of this part of ISO 14817 may use different data concepts, different meta-attributes, or different data concepts and different meta-attributes; however, a conforming implementation of this part of ISO 14817 shall provide an unambiguous mapping to and from the physical implementation model and the conceptual meta-model defined by this part of ISO 14817.

Regional and National DDs have the option of adopting data concept definitions from the CIDCR, but are not required to do so.

Table 1 indicates the conformance requirements of data concept registries and data dictionaries.

**Table 1 — Data dictionary and data concept registry conformance**[a]

| Feature | Data Dictionary | Data Concept Registry[b] |
|---|---|---|
| Support all data concepts[c] | ✓ | ✓ |
| Support all mandatory identification meta-attributes[d] | ✓ | ✓ |
| Support all mandatory definitional meta-attributes[e] | ✓ | ✓ |
| Support all mandatory relational meta-attributes[f] | ✓ | ✓ |
| Support all mandatory representational meta-attributes[g] | ✓ | ✓ |
| Support all mandatory administrative meta-attributes[h] |  | ✓ |
| Electronic storage with automated administrative rules[i] |  | ✓ |

[a] Annex A identifies which meta-attributes are mandatory for specific data concepts.

[b] For data concept registries, “mandatory” meta-attributes shall also include all “assigned” meta-attributes.

[c] As defined in Clause 6.

[d] As defined in 7.1.

[e] As defined in 7.2.

[f] As defined in 7.3.

[g] As defined in 7.4.

[h] As defined in ISO 14817-2.

[i] As defined in ISO 14817-2.

## 3 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 14817-3:—[1]), *Intelligent transport systems — ITS data dictionaries — Part 3: Object identifier assignments for ITS data concepts*

ISO/IEC 8824-1:—[2]), *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation*

ISO/IEC 9834-1, *Information technology — Procedures for the operation of object identifier registration authorities: General procedures and top arcs of the international object identifier tree*

NIMA TR8350.2, Third Edition – Amendment 1, January 2000, Department of Defence – World Geodetic System 1984, Its Definition and Relationships With Local Geodetic Systems, issued by National Imagery and Mapping Agency (NIMA), US Department of Defence

## 4 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 14817-3 and the following apply.

**4.1**
**abstract**
indication of whether the object class is purely abstract or can be instantiated with member objects; abstract object classes typically have non-abstract specializations

1) To be published.

2) To be published. (Revision of ISO 8824-1:2008)

**4.2**
**aggregate domain**
data concept that defines a grouping of data elements and/or data frames

**4.3**
**ASN.1 name**
name of a data concept expressed as a valid "typereference" as defined by ISO/IEC 8824-1

**4.4**
**association**
semantic relationship between two object classes

**4.5**
**constraint**
notation which can be used in association with a type, to define a subtype of that type

[SOURCE: ISO/IEC 8824-1:2008]

**4.6**
**context**
universe of discourse in which a name or definition is used

[SOURCE: ISO/IEC 11179-3:2003, 3.3.31]

**4.7**
**contextual name**
descriptive word or group of words that labels a data concept within its defined context

**4.8**
**data**
re-interpretable representation of information in a formalized manner suitable for communication, interpretation or processing

Note 1 to entry: Data can be processed by human or automatic means.

[SOURCE: ISO 11179-1:2004]

**4.9**
**data concept**
item that may be stored in a data dictionary that refers to an abstraction or thing in the natural world that can be identified with explicit boundaries and meaning and whose properties and behaviour all follow the same rules

Note 1 to entry: Data concepts can be classified into the following types: object class, value domain, data element, aggregate domain, data frame, message, interface dialogue, dictionary document, or module

**4.10**
**data concept identifier**
numeric identifier that shall uniquely and unambiguously identify a data concept within the scope of the DD being managed

**4.11**
**data concept instance**
occurrence of a data concept

**4.12**
**data concept registry**
electronic data dictionary that follows precise documented rules for the registration and management of stored data concepts; it typically also includes advanced features for adding, retrieving, and working with its contents

Note 1 to entry: The data concept registry contains meta-attributes about data concepts in terms of their names and representational forms as well as the semantics associated with the data concepts. A data concept registry may contain data that assists information interchange and re-use, both from the perspective of human users and for machine-interpretation of data concepts.

**4.13**
**data concept revision**
integer that represents the number of edits that have been made to the data concept since the last update to the major version number

**4.14**
**data concept type**
categorization of the kind of data concept

**4.15**
**data concept version**
integer that represents the number of normative changes that have been made to approved versions of the associated data concept

**4.16**
**data dictionary**
listing of data concepts and their meta-attributes in a consistent format

**4.17**
**data element**
data concept represented by a specific value domain and that describes a single atomic property about an object class

Note 1 to entry: A data element is composed of an object class, a property of the represented object class and a value domain.

**4.18**
**data frame**
data concept represented by a specific aggregate domain and that describes information of interest through a useful grouping of more atomic properties about one or more object classes

Note 1 to entry: The grouping may be a set, sequence, or a choice.

**4.19**
**data model**
graphical and/or lexical representation of data, specifying their properties, structure, and inter-relationships

[SOURCE: ISO 11179-1:2004, 3.2.7]

EXAMPLE A data model might specify that a “Vehicle” may be described by a variety of properties, such as: “make”, “model”, “year”, and “vin” (vehicle identification number). Likewise, a “Collision” might be described by properties such as “occurenceTime”, “severity”, and “vehiclesInvolvedCount”. Finally, the model might depict that a Collision has a many-to-many relationship to a Vehicle. The following provides a sample graphic of this data model.

Note 1 to entry: Within the scope of this part of ISO 14817, data models are depicted using UML Class Diagrams.

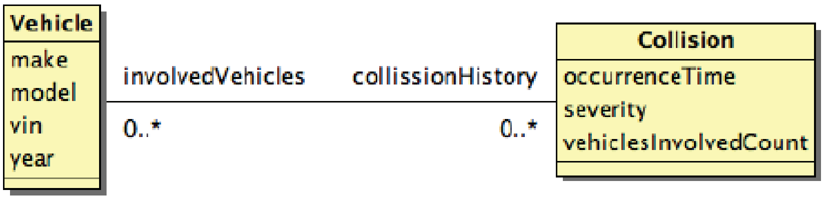

**Figure 1 — Sample data model**

**4.20**
**data type**
set of distinct values, characterized by properties of those values and by operations on those values

[SOURCE: ISO/IEC 11404:2007, 3.12]

**4.21**
**definition**
representation of a concept by a descriptive statement that serves to differentiate it from related concepts

[SOURCE: ISO 11179-1:2004, 3.2.8]

**4.22**
**descriptive name**
descriptive word or group of words that uniquely labels a data concept within a module

**4.23**
**dialogue**
see interface dialogue

**4.24**
**dialogue order rules**
rules governing the sequencing of messages to be sent among systems in order to achieve a specified service

**4.25**
**dictionary document**
data concept that represents a data dictionary along with supplemental information that may be standardized

**4.26**
**document identifier**
identifier that uniquely identifies the document

**4.27**
**format**
natural language description of the logical layout of the data concept in relation to interchange of data

**4.28**
**generalization**
taxonomic relationship between a more general element and a more specific element where the more specific element is fully consistent with the more general element and contains additional information

[SOURCE: ISO 14813-5:2010]

Note 1 to entry: The more general class is referred to as the superclass.

Note 2 to entry: The more specific class is referred to as the subclass.

Note 3 to entry: “Fully consistent” means that the subclass has all of the *properties* (4.47) and relationships of the superclass.

**4.29**
**historic ASN.1 name**
ASN.1 name assigned to a data concept that does not follow the current naming convention

**4.30**
**historic descriptive name**
descriptive name assigned to a data concept that does not follow the current naming convention

**4.31**
**identifier**
sequence of characters, capable of uniquely identifying that with which it is associated, within a specified context

[SOURCE: ISO/IEC 11179-1]

**4.32**
**interface dialogue**
data concept that defines bi-directional communication sequence between two parties in accordance with predetermined protocols

**4.33**
**international object identifier tree**
tree whose root corresponds to ISO/IEC 9834-1 and whose nodes correspond to Registration Authorities responsible for allocating arcs from a parent node

[SOURCE: ISO/IEC 9834-1:2012, 3.5.5]

**4.34**
**lower camel case term**
string consisting of one or more words, where each word within the string, except for the first, starts with an upper-case letter and all other letters are lowercase; the words follow each other without any space; hyphens and numbers may be used, but the first character of the string must be a lower case alphabetic character; a hyphen may not be the last character or occur multiple times in sequence

**4.35**
**message**
data concept that is a grouping of data elements, data frames, or data elements and data frames that is used to convey a complete set of information

Note 1 to entry: For the purposes of this part of ISO 14817, a message is an abstract description; it is not a specific instance.

**4.36**
**message instance**
occurrence of a message containing the actual values for the data elements and/or data frames

**4.37**
**meta-**
Greek prefix denoting a description that is one level of abstraction above the concept being described

**4.38**
**meta-attribute**
documenting characteristic of a data concept that is stored in a DD

**4.39**
**metadata**
documenting characteristic of a data concept that is provided in a message

Note 1 to entry: Documenting characteristics are termed "meta-attributes" when stored in a DD, but are termed metadata when provided within the same message instance as the actual value. For example, a data element may be defined within the DD with a specific unit of measure, such as meters; the Unit of Measure field is a defined meta-attribute. Alternatively, the unit of measure may be defined at runtime within a message, especially for items such as unit of currency. The field within a message that defines the unit of currency for an included value would be termed "metadata".

**4.40**
**module**
data concept that contains the formal syntactic definition, and optionally the semantic definition, of a defined set of other data concepts that are all version-controlled as a single unit; a module can be represented in multiple languages (e.g., ASN.1 or XML Schema) and compiled by computer systems

**4.41**
**multiplicity**
number of instances of the subject data concept that may be associated with the object class that it describes

**4.42**
**name**
indexical term used by humans as a means of identifying data elements and other data concepts

**4.43**
**nested object class**
object class that represents a logical group of data elements and data frames that describe some aspect of the larger object class by which the nested object class is contained

Note 1 to entry: Nested object classes are used to describe object classes that are contained within another object class and are used for conceptual objects rather than tangible objects.

EXAMPLE A message sign object class might have a nested object class for the messages stored in its library, where each message is described by a number of properties, such as message number, message content, message owner, etc.

**4.44**
**nominal version**
identifier that represents a version number that the data concept is more generally known by

**4.45**
**object class**
description of a set of objects that share the same properties, relationships, and semantics

Note 1 to entry: Adapted from ISO/IEC 11179-1; an object class is conceptually similar to an ISO/IEC 11179 object, but it does not include operations or methods and ISO/IEC 11179 "attributes" are called "properties" in this part of ISO 14817.

**4.46**
**object identifier**
ordered list of primary integer values from the root of the international object identifier tree to a node, which unambiguously identifies that node

[SOURCE: ISO/IEC 9834-1:2012, 3.5.11]

**4.47**
**parent object class**
object class that the data concept describes

**4.48**
**precursor**
historical, semantically similar data concept within the same DD, which this data concept has replaced or is replacing

**4.49**
**property**
characteristic common to all members of an object class

[SOURCE: ISO/IEC 11179-3]

Note 1 to entry: This is defined as a distinct data concept within ISO 11179, but is rolled into the definition of a data element within this part of ISO 14817 to simplify DD design.

**4.50**
**referenced data element**
data element that is referenced by the current data concept

**4.51**
**referenced data frame**
data frame that is referenced by the current data concept

**4.52**
**referenced message**
message that is employed in the current interface dialogue

**4.53**
**remarks**
comments or other information pertinent to the data concept

**4.54**
**semantics**
branch of linguistic science that deals with the meaning of words

[SOURCE: ISO/IEC 11179-5]

**4.55**
**source**
document or other reference that was used to develop the pertinent data concept

**4.56**
**successor**
newer, semantically similar data concept within the same DD, which has replaced or is replacing this data concept

**4.57**
**superclass**
object class that is a generalization of the current object class

**4.58**
**synonym**
semantically similar data concept

**4.59**
**syntax**
set of rules defining the way in which data is put together with appropriate identifiers, delimiters, separator character(s), and other non-data characters to form messages

[SOURCE: ISO 21849:2006]

**4.60**
**uniform resource locator**
string for identifying resources on the Internet (such as Web pages) by specifying the address of the resource and the access protocol used

**4.61**
**unit of measure**
actual units in which the associated values are measured

[SOURCE: ISO 11179-3:2003, 3.3.1334 modified.]

**4.62**
**upper camel case term**
string of one or more words where each word within the string starts with an upper-case letter and the remainder of each word is in lowercase; the words follow each other without any space; hyphens and numbers may be used, but the first character of the string must be an upper case alphabetic character; a hyphen may not be the last character or occur multiple times in sequence

**4.63**
**value domain**
data concept that defines a set of permissible values

**4.64**
**valid value rule**
natural language text definition of the rule(s) by which permissible legal instances of a data element or a value domain are identified

## 5 Symbols and abbreviated terms

| | |
|---|---|
| ASN.1 | Abstract Syntax Notation One |
| ANSI | American National Standards Institute |
| CASE | Computer-Aided Software Engineering |
| CIDCR | Central ITS Data Concept Registry |
| DCI | Data concept identifier |
| DD | Data dictionary |

NOTE By definition, a data concept registry is a specialized type of a data dictionary; thus the DD symbol also applies to data concept registries.

| | |
|---|---|
| IEC | International Electrotechnical Commission |
| ISO | International Organization for Standardization |
| ITS | Intelligent transport system(s) |
| N/A | not applicable |
| OID | object identifier |
| OSI | Open System Interconnection |
| TC | Technical Committee |
| UBL | Universal Business Language |

| | |
|---|---|
| UML | Unified Modelling Language |
| URL | Uniform Resource Locator |

# 6 Data concepts

## 6.1 Summary of data concepts

This Clause explains the nine *data concepts* (4.9) applicable to this part of ISO 14817, as listed in 7.2.3. Data concepts refer to abstractions and things in the natural world that can be identified with explicit boundaries and meaning. The properties and behaviour of these data concepts all follow the same set of rules. Within ITS, there may be data concepts to represent, for example, a bus route and relevant information about it.

At the lowest level, a *value domain* (4.64) is a data concept that defines allowed syntax that can be used to express a piece of information. Value domains provide minimal semantic information other than what the set of values are to be used for. C.1 defines several value domains, for example, “text” is a value domain that represents a human pronounceable textual string that is represented by an ASN.1 UTF8String.

The primary purpose of value domains is to provide standard representational forms for data elements. A *data element* (4.17) describes a single atomic *property* (4.50) of an *object class* (4.46). For example, a vehicle (object class) may have a colour (property) that can be represented by a colour code list (value domain). The object class “vehicle” describes the core concept being described, the property “colour” identifies what is being described about the object class, and the value domain “colour code” provides a representational form. The three concepts combined are called a data element.

Some concepts are complex and are represented with multiple related data elements. For example, a two-dimensional location is often described by a latitude and a longitude. These two items are grouped together into a structure known as an *aggregate domain* (4.2) called GeoLocation.2D, as defined in C.4.9. Similar to a value domain, an aggregate domain is a generic representation that can be used in multiple contexts; it provides a representational form, but provides little in the way of semantics (i.e. location of what). Each item within an aggregate domain should be defined as its own data element or data frame. For example, the latitude field within the GeoLocation.2D structure is defined as the “latitude” property of the “GeoLocation” object class with a value domain of MeasureType.

A *data frame* (4.18) is a complex data element; in other words, whereas a data element is by definition elemental and represented by a value domain, a data frame is complex and represented by an aggregate domain. For example, a vehicle (object class) may have a location (property) represented by the GeoLocation.2D aggregate domain. The three concepts combine form the data frame.

A *message* (4.36) is a set of data elements and/or data frames that convey a complete thought. An *interface dialogue* (4.33) defines the permissible sequence(s) of messages that may be exchanged between entities.

A *module* (4.41) represents a set of other data concepts that are all version-controlled as a single unit. This is useful so that one portion of a standard can be updated without breaking backwards compatibility with other portions of a standard.

A *dictionary document* (4.25) is any document approved by some authority that defines data interface details. A dictionary document is typically a formal standard approved by a standards development organization such as ISO, but a governmental entity or private company may also develop dictionary documents to record their custom designs.

Figure 2 provides an overview of how the nine data concepts inter-relate and can be documented.

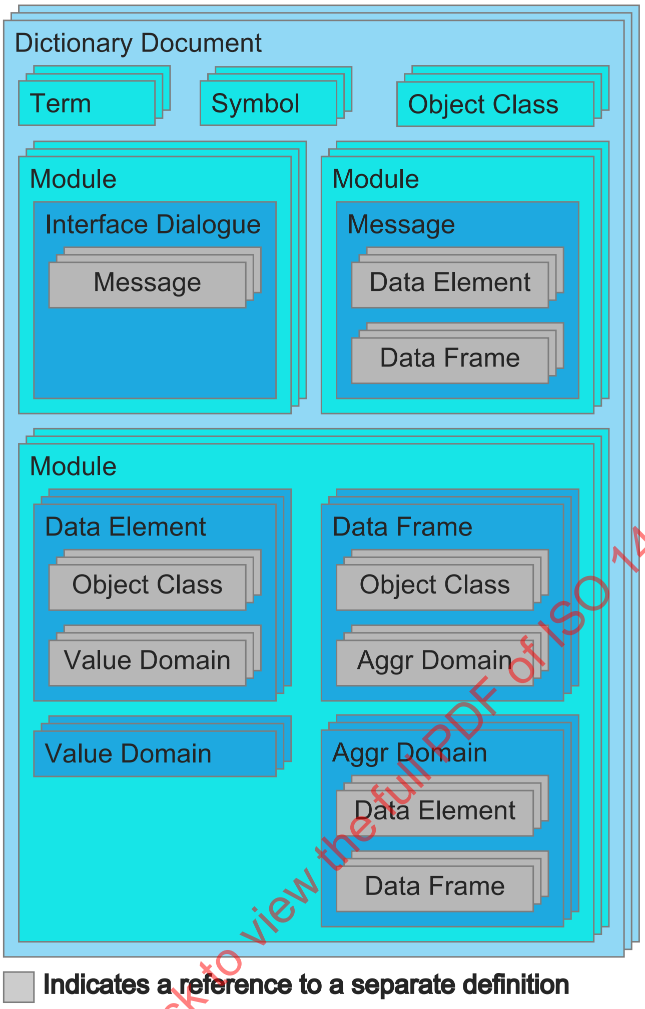

Figure 2 — Documenting Data Concepts

A dictionary document should formally define the *object classes* described within the document. The development of object classes forces the developers of the dictionary document to consider the exact structure of the data that they are defining. This is important so that when data concepts are recorded in the CIDCR, they fully define their own context in a clear and unambiguous way.

A dictionary document will typically define a number of modules. Modules can be categorized into one of three types.

Each interface dialogue should be defined within its own module to improve version control. If a future version of the dictionary document needs to only modify a single interface dialogue, the changes can be made without impacting the backwards compatibility of any of the other interface dialogues. Each interface dialogue will reference potentially multiple messages.

Likewise, each message should be defined within its own module to promote proper version control for similar reasons. Each message will reference some combination of data elements and data frames.

The third type of module defines the fundamental data concepts that are contained within messages, i.e. data elements, value domains, data frames, and aggregate domains. All of this information can be defined within a single module, or may be broken into sub-modules, often with references between them. In order to promote reuse of modules across dictionary document, it is often useful to have the following modules:

— the ISO-14817-1-Domains-1 module as defined in C.6;

- a custom value domain module, if needed;
- a data element module for each object class included in the dictionary document;
- a minimal set of modules for each object class that define aggregate domains and data frames.

Care should be taken when defining modules for aggregate domains and data frames so that circular references are not created between modules. In addition, the developers should also consider dividing modules if there is a perceived benefit for re-usability by other dictionary documents.

The definition of a data element includes a reference to both a parent object class and a value domain. The definition of a data frame includes a reference to both a parent object class and an aggregate domain.

Each aggregate domain will reference some combination of data elements and data frames.

The nine data concepts can be categorized into three main groups, which are as follows.

- Documentation data concepts include dictionary document and module.
- Data model data concepts include object class, value domain, and data element.
- Interface data concepts include interface dialogue, message, data frame, and aggregate domain.

Each of these groups of data concepts is described in the following clauses.

## 6.2 Documentation Data Concepts

This part of ISO 14817 encourages the use of the Unified Modelling Language (UML) to document relationships among data. However, UML can also be used to document a metamodel, i.e. a data model that specifies one or more other data models. In other words, a metamodel is at a higher level of abstraction. Rather than defining data relationships among ITS data concepts (e.g. “Vehicle” “Collision”, etc.) it is used to define relationships between the types of data concepts defined in this part of ISO 14817 (i.e. “object class”, “data element”, “dictionary document”, etc.). Within this part of ISO 14817, metamodels are always designated by labelling each UML class with the stereotype “metaclass”.

Figure 3 depicts the metamodel that defines relationships among the documentation data concepts. Documentation data concepts include the top-level data concepts that we described in Figure 2.

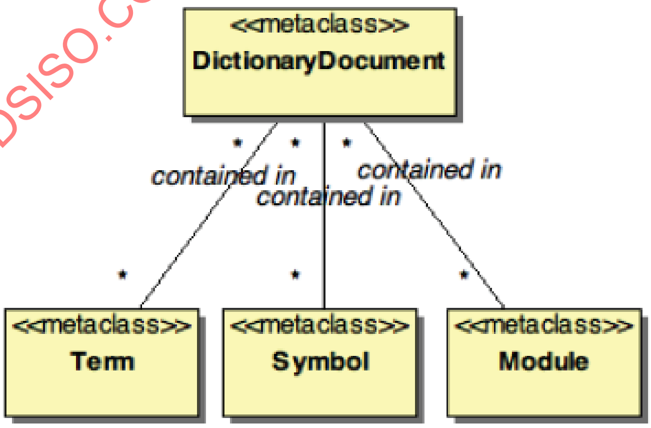

**Figure 3 — Framework of documentation data concepts**

Each box in the diagram corresponds to a data concept defined in this part of ISO 14817. The lines between data concepts indicate associations. The asterisk (*) next to the end of each line indicates that there may be multiples of the data concept. Thus, the diagram indicates that a dictionary document may contain many modules. It also indicates that a given module may be contained in multiple dictionary documents.

### 6.2.1 Dictionary Document

A dictionary document data concept may represent a standard or any other formal specification that defines or references other data concepts. It shall be comprised of zero or more modules and object classes that are approved as a single unit in order to fulfil a specific set of requirements. The actual document may also contain additional text and/or information that are not recorded within the data concept representation.

NOTE The "dictionary document" data concept is designed to allow for the proper management and version control of data concepts that are used by multiple documents. It does not replace the actual documents; it merely represents them within a DD.

A dictionary document contains definitions of other data concepts. Ideally, the definitions of these other data concepts should be generated from the CIDCR to ensure consistency among the various dictionary documents that reference the same data concepts.

### 6.2.2 Module

A module shall be a group of interface and data model data concepts that are version controlled as a unit and can be represented in a computer readable file in order to exchange these data concepts. A given module may appear in multiple dictionary documents. Each such use should be recorded in the module's "context" meta-attribute. In order to best manage version control, a module should either contain a single interface dialogue, a single message, or some combination of other data concepts.

## 6.3 Data model Data Concepts

Data model data concepts are data concepts that appear in a data model. Figure 4 depicts the various data model data concepts used in this part of ISO 14817 and their relationships.

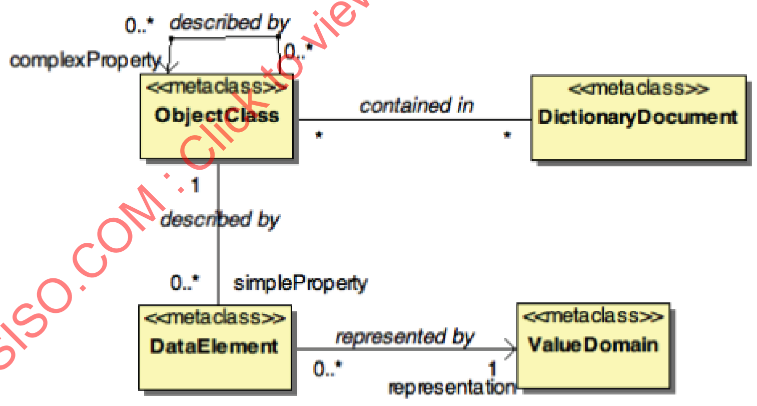

**Figure 4 — Framework of data model data concepts**

NOTE ISO 11179-3 defines several other data concepts, such as "property", "data element concept", "conceptual domain", etc. This part of ISO 14817 acknowledges the theoretical existence of such concepts, but it does not require the specification of these data concepts within ITS dictionary documents. Instead, this part of ISO 14817 simplifies the number of data concepts being defined.

An object class may be contained in multiple dictionary documents and each dictionary document may contain multiple object classes. Data elements and value domains are contained in zero or more modules (not shown in Figure 4 for brevity) and each module may contain multiple data elements and value domains (not shown in Figure 4 for brevity.)

An object class may be described by zero or more simple properties, each of which is termed a "data element". Data elements describe some aspect of exactly one object class and have exactly one representation as defined by a value domain. A value domain defines the exact representational form for a piece of data and may be used by multiple data elements.

An object class may also be described by zero or more complex properties, represented as other object classes. In other words, an object class (e.g. "Collision") can have an association with another object class (e.g. "Vehicle") to describe some property of the first object class (e.g. "involvedVehicles"); "involvedVehicles" must be represented as an object class because it has its own set of data elements (e.g. "make", "model", "vin", etc.) that describe each involved vehicle. While this part of ISO 14817 recognizes these associations exist, it does not require DDs to define these.

A sample data model is provided in Figure 5. This sample indicates that the object class Collisions has a complex property of reports that is represented by zero or more Collision object classes. Each Collision object class is described by three simple properties (i.e. data elements): occurrenceTime (represented as a date-time), severity (represented as a code), and vehiclesInvolved (represented as a quantity). Each Collision object class is also described by a complex property called involvedVehicles, which is represented by zero or more Vehicle object classes. Each Vehicle object class is described by five simple properties: make (represented as a name), model (represented as a name), vin (represented as an identifier), year (represented as an identifier), and licensePlate (represented as an identifier).

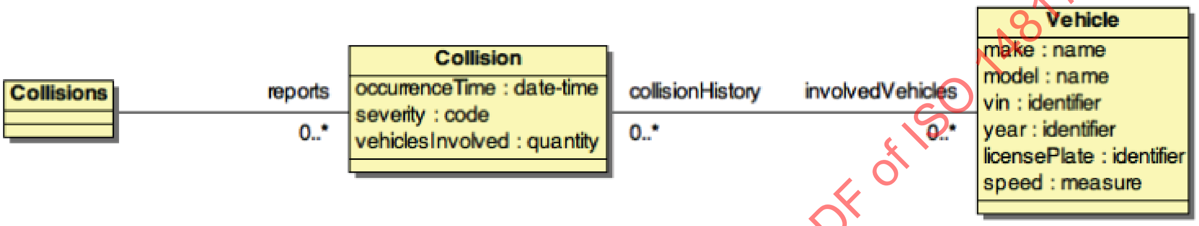

**Figure 5 — Sample data model**

### 6.3.1 Object class

An object class shall describe a set of object instances that share the same properties, relationships and semantics within a given domain of discourse about which there is a need to represent some information. Modifiers that qualify or further specialize the object class may be used. In Figure 5, the object classes illustrated are "Collisions", "Collision", and "Vehicle".

### 6.3.2 Data element

A data element shall be a formalized representation of some information (i.e. a property) about an object class, with an explicit value domain. Figure 5 depicts eight data elements:

- "Collision.occurrenceTime:dateTime";
- "Collision.severity:code";
- "Collision.vehiclesInvolved:quantity";
- "Vehicle.make:name";
- "Vehicle.model:name";
- "Vehicle.vin:identifier";
- "Vehicle.year:identifier";
- "Vehicle.licencePlate:identifier".

A data element may further refine a referenced value domain.

### 6.3.3 Value domain

A value domain shall be a term that indicates, precisely and unambiguously, the representational form, including the syntactic form, format, and units for data concept instance values. Figure 5 includes the value domains of “code”, “date-time”, “identifier”, “measure”, “name”, and “quantity”.

## 6.4 Interface Data Concepts

Figure 6 presents interface data concepts and their relationships.

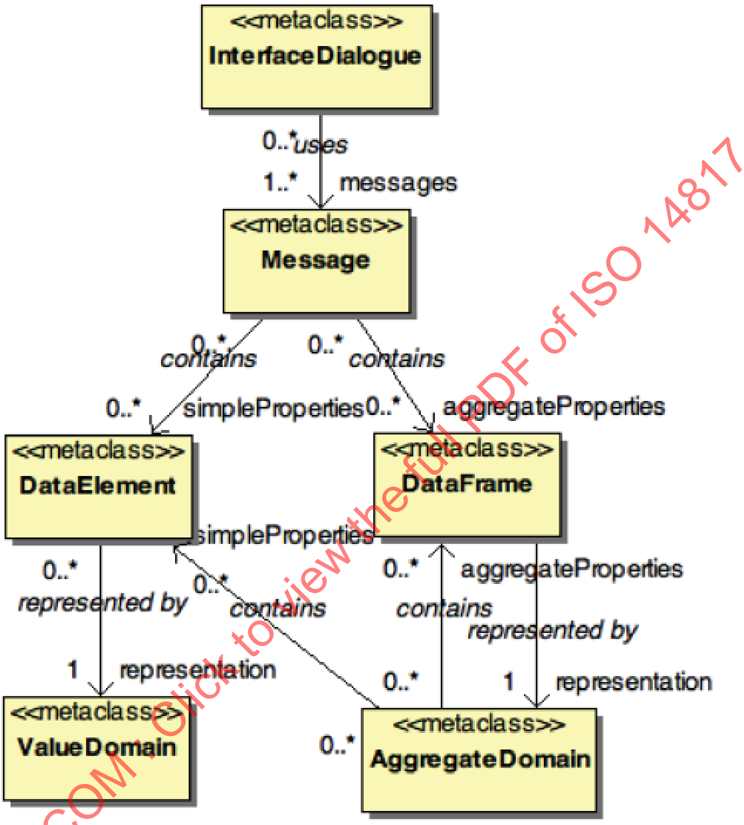

**Figure 6 — Framework of interface data concepts**

An information interchange between two ITS components for a specific service shall be defined by an interface dialogue. An interface dialogue shall use a set of one or more messages whose order and timing for transmission is predicated upon a defined operational concept or scenario.

A message shall contain zero or more data elements and/or data frames. A message may be used by zero or more interface dialogues.

A data element is semantically similar to a data frame; the only difference is that a data element is an atopic piece of data whereas a data frame is an aggregation of data. A data element is represented by exactly one value domain whereas a data frame is represented by exactly one aggregate domain.

An aggregate domain shall contain zero or more data elements and/or data frames.

Both data elements and data frames may be contained within zero or more messages and/or within zero or more aggregate domains.

While not shown for brevity, all of the data concepts shown in Figure 6 may be contained in zero or more modules and each module may contain multiples of these data concepts, although it is recommended that interface dialogues and messages be placed in their own modules in order to improve version control.

An interface can be presented in an interface model that looks similar to a data model, but an interface model only represents the data contained in a single data exchange whereas a data model typically provides a larger context such as indicating all information that may exist in a database. Therefore, an interface model is typically a subset of the information contained in a corresponding data model. An example data model is provided in Figure 7. Interface models are indicated by the stereotypes applied to the UML classes.

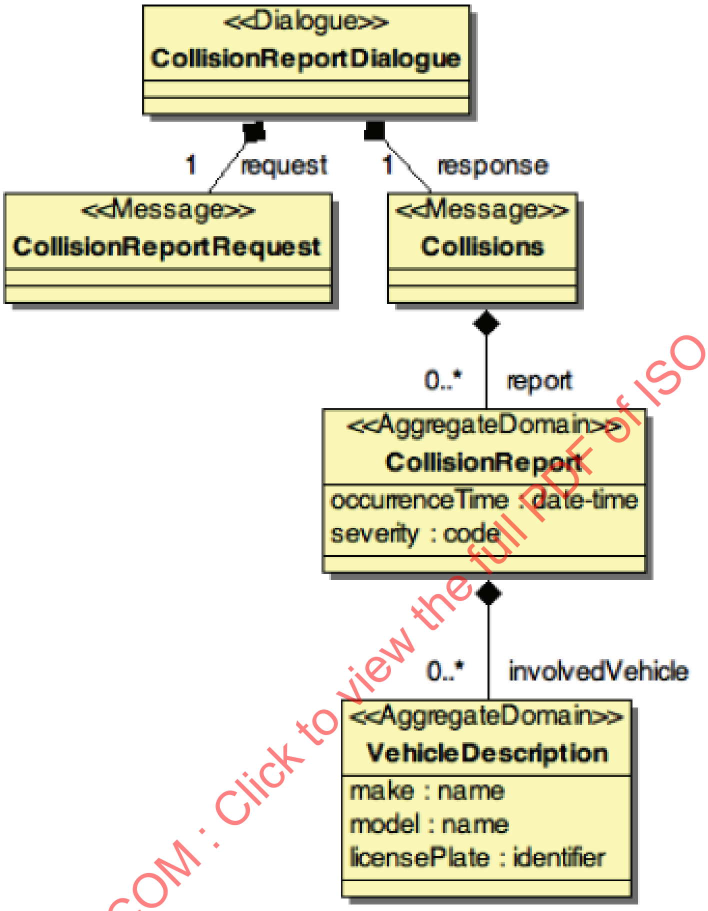

**Figure 7 — Sample Interface Model**

Figure 7 indicates that the CollisionReportDialogue consists of exactly one CollisionReportRequest request and exactly one Collisions response. The Collisions response contains zero or more report data frames, represented by the CollisionReport aggregate domain (which is a simplified view of the Collision object class presented in Figure 5). Each CollisionReport aggregate domain includes an occuranceTime data element (represented as a date-time value domain) and a severity data element (represented by a code value domain).

Each CollisionReport aggregate domain also includes a listing of zero or more involvedVehicle data frames, represented by the VehicleDescription aggregate domain (which is a simplified view of the Vehicle object class presented in Figure 5). Each VehicleDescription aggregate domain includes a make data element (represented by a name value domain), a model data element (represented by a name value domain) and a licensePlate data element (represented by an identifier value domain).

### 6.4.1 Interface dialogue

An interface dialogue shall be a temporal sequence of messages, including variants, among two or more system components that are used to accomplish a service/observable result. Within an interface model, an interface dialogue appears as a UML class with a stereotype of “Dialogue”. In Figure 7, the interface dialogue example is “CollisionReportDialogue”.

### 6.4.2 Message

A message shall be a structured grouping of data elements and/or data frames. Within an interface model, a message shall appear as a UML class with a stereotype of “Message”. In Figure 7, there are two messages: “CollisionReportRequest” and “Collisions”.

### 6.4.3 Data frame

A data frame may be used to mimic an association between object classes in the data model (e.g. “involvedVehicles” appears as an association in Figure 5 and as a data frame in Figure 7) or may be used to group data elements in a useful way (e.g. combining a number of data elements together into a reusable package). For example, a “Message.header” might be defined to include several data elements that are commonly used together in various messages. Within an interface model, a data frame shall appear as a UML association role name that connects to an aggregate domain. In Figure 7, there are two data frames: “Collisions.report” and “CollisionReport.involvedVehicle”.

### 6.4.4 Aggregate domain

An aggregate domain shall be a representation of an object class, but usually representing only a subset of the properties defined for the object class. It defines precisely and unambiguously the representational form of a data frame. While some aggregate domains may only be used by one data frame, others are likely to be used by several data frames. For example, the Location aggregate domain may be used by Vehicle.position, Vehicle.origin, and Vehicle.destination. Within an interface model, an aggregate domain shall appear as an UML class with a stereotype of “AggregateDomain”. In Figure 7, there are two aggregate domains: “CollisionReport” and “VehicleDescription”. An aggregate domain may have the same name as the object class it is describing, although it is often useful to distinguish the subset of information included in a particular aggregate domain.

## 7 Meta-attributes

Every data concept shall be described by a number of meta-attributes. The specific meta-attributes that apply to a given data concept are presented in Annex A. This Clause provides a complete definition of each meta-attribute defined by this part of ISO 14817.

### 7.1 Identification and naming meta-attributes

Identification meta-attributes shall differentiate one data concept from another. Data concepts have multiple identification meta-attributes to assist in differentiating data concepts from one another within different contexts. For example, the “data concept identifier” meta-attribute can be used to identify a data concept within a database and the “data concept version” meta-attribute can be used to distinguish among multiple historic versions of the same data concept as stored within the database. However, these numeric identifiers are not very user-friendly, and thus, each data concept also has a “descriptive name” to assist in human usage. Other identification meta-attributes are defined for other contexts.

### 7.1.1 Data concept identifier

The local DD shall assign this identifier, preferably in an automated fashion and shall never change it. The value of this meta-attribute allows for proper database management and allows tracking of a given data concept even there have been changes to its Descriptive Name, Object Identifier, etc.

NOTE A data concept copied from one DD to another will be assigned a new DCI for the new DD.

### 7.1.2 Data concept version

Versions are established to record normative changes, which include any normative change in semantic and/or representational information about a data concept. The version shall start with the value one (1) and increment upon any normative change to the data concept if, and only if, the current version has already been approved. Table 2 defines the normative meta-attributes of data concept and their relationship to version numbering.

**Table 2 — Meta-attributes that may result in a version change**

| Clause | Meta-attribute | Version change requirements |
|---|---|---|
| 7.1.11 | Object identifier | Must be updated with each version change |
| 7.2.1 | Definition | Forces a version change, except for strictly editorial corrections[a] |
| 7.2.3 | Data concept type | Can never be changed once the data concept has been recorded |
| 7.2.6 | Dialogue order rules | Forces a version change, except for strictly editorial corrections[a] |
| 7.3.1 | Parent object class | Forces a version change[a] |
| 7.3.5 | Abstract | Forces a version change[a] |
| 7.3.6 | Multiplicity | Forces a version change[a] |
| 7.3.7 | Superclass | Forces a version change[a] |
| 7.4.1 | Data type | Forces a version change[a] |
| 7.4.2 | Format | Forces a version change[a] |
| 7.4.3 | Unit of measure | Forces a version change[a] |
| 7.4.4 | Valid value rule | Forces a version change[a] |
| 7.4.5 | Constraint | Forces a version change[a] |

[a] Only if the current version has already been approved.

NOTE See ISO 14817-2 for detailed processes such as handling draft versions that are never approved (i.e. dead branches of version control).

### 7.1.3 Data concept revision

Revisions are established to track minor changes, which do not include changes in semantic or representational information about a data concept. The revision shall start with the value zero (0) and shall increment with each change to the data concept that does not result in a new data concept version. The revision shall reset to zero (0) upon incrementing the data concept version.

NOTE Revisions are updated even when minor modifications are made such as adding a Remark to the data concept.

### 7.1.4 Nominal version

Humans often assign their own version numbers to data concepts. For example, the nominal version of an International Standard is defined to be the year in which it was approved.

### 7.1.5 Document identifier

This is typically composed of the acronym of the standardizing body and the number assigned by that body. For example, the document identifier for this part of ISO 14817 is ISO 14817-1.

### 7.1.6 Contextual name

The contextual name is that portion of the descriptive name that is unique to the data concept. For example, the complete definition of a data element includes a formal reference to a defined object class and a defined value domain. The contextual name of the data element would only include the property term, as it would inherit the object class name and value domain name from the referenced data concepts. See B.1 for additional information.

### 7.1.7 Descriptive name

Descriptive names shall be constructed in accordance with requirements of B.2. Descriptive names represent the meaning of the data concept and facilitate semantic understanding. They can be used to quickly locate a data concept within a DD. The Descriptive Name may be generated automatically if the DD supports Contextual Names.

### 7.1.8 Historic descriptive name

The Historic Name meta-attribute allows the DD to maintain a link from a data concept to any previously known name for the same data concept that has been updated with a name conforming to the Annex B naming conventions. This meta-attribute was formerly called “Synonymous Descriptive Name”, but since a DD should harmonize synonyms into a single data element, the name of this meta-attribute has been changed to “Historic Name” to more accurately reflect its purpose. There should not be any normative differences between the historic data concept and the recorded data concept, other than the name. If there is a normative difference, both data concepts should be entered and linked through the use of the predecessor/successor meta-attributes or the synonym meta-attribute, if appropriate. Historic descriptive names do not need to conform to the naming conventions in Annex B.

### 7.1.9 ASN.1 name

The name of a data concept expressed in ASN.1 syntax. ASN.1 names shall be constructed in accordance with requirements of B.3.

### 7.1.10 Historic ASN.1 name

Data concepts are often developed and put into use before they are fully conformant with the naming conventions of this part of ISO 14817. This meta-attribute allows the DD to record a previously defined ASN.1 Name for a data concept that might have been based on a Historic Name or on different naming conventions.

NOTE This meta-attribute replaces the “Symbolic Name” meta-attribute of the previous version of this part of ISO 14817 because the ASN.1 name is the only application program name that a DD needs to be concerned with.

### 7.1.11 Object identifier

An OID allows application programs to quickly and uniquely identify and reference data concepts in data exchanges. In order to maximize interoperability

a) the OID does not change for an approved version of a data concept, and

b) the OID changes whenever the data concept version changes.

The OID shall be constructed in accordance with requirements in ISO/IEC 9834-1.[3)]

### 7.1.12 Uniform resource locator

A URL provides an easy way of accessing additional information about a data concept via the Internet. The URL of a module shall be the namespace of the module.

3) It is recommended that the OID be constructed in accordance with the requirements in ISO 14817-3.

## 7.2 Definitional meta-attributes

Definitional meta-attributes shall describe the semantic aspects of a data concept. These meta-attributes may directly address semantic meanings (e.g. Definition, Remarks, Abstract) or indirectly provide insights into the semantic aspects of a data concept (e.g. Source, Data Concept Type).

### 7.2.1 Definition

The definition provides the formal, normative meaning of the data concept in natural-language text.

### 7.2.2 Source

The source is a reference to the original document (e.g. white paper, architecture or standard) that defines the requirement for the data concept.

### 7.2.3 Data concept type

The data concept type of a data concept shall not be changed after its initial creation. Valid values for data concept type are:

- Dictionary document;
- Module;
- Object class;
- Data element;
- Value domain;
- Interface dialogue;
- Message;
- Data frame;
- Aggregate Domain.

### 7.2.4 Remark

This meta-attribute is unconstrained as to its textual content.

### 7.2.5 Context

The context of most data concepts shall be the descriptive name of the module in which it is contained. For the data concepts module and object class, the context shall be the document identifier of the dictionary document. A dictionary document shall not have a context.

NOTE By allowing multiples entries of this meta-attribute for a single data concept, a DD will be able to track reuse of data concepts.

### 7.2.6 Dialogue Order Rules

Ordering rules should describe the allowed message sequences, including error conditions, and may discuss issues such as frequency of transmission, priority logic, message verification, etc.

## 7.3 Relational meta-attributes

Relational meta-attributes shall document relationships among or between data concepts.

### 7.3.1 Parent object class

This indicates the object class that a data element or data frame describes.

### 7.3.2 Precursor

The corresponding data concept should reference this data concept as its successor. The referenced data concept shall be represented by its fully qualified descriptive name when presented to the user.

### 7.3.3 Successor

The corresponding data concept should reference this data concept as its precursor. The referenced data concept shall be represented by its fully qualified descriptive name when presented to the user.

### 7.3.4 Synonym

This may include any data concept that the developer believes is similar enough to be of interest. It may have a slightly different semantics or different representational form. The referenced data concept shall be represented by its fully qualified descriptive name when presented to the user.

### 7.3.5 Abstract

This meta-attribute only applies to object classes. An abstract object class shall have this value set to “true” meaning that it cannot be directly instantiated (only specializations can). Non-abstract classes shall have this meta-attribute set to “false.”

### 7.3.6 Multiplicity

A specification of the minimum and maximum number of instances that may exist of this data concept for the corresponding parent object class. The value may be a single number, which indicates the minimum equals the maximum; or may be a range indicated by a number two periods and another number (e.g. “0..5”). The maximum may be represented by an asterisk (“*”) rather than a number to show an unbounded range. For example, when defining a data element describing vehicle colour, this meta-attribute would indicate whether a vehicle is allowed to be associated with 0, 1, 2, etc. colours. In this case, the designer may decide that all vehicles must have at least one colour or at the most 2; in which case, the value of this meta-attribute would be “1..2”.

### 7.3.7 Superclass

The superclass meta-attribute of a data concept shall reference the superclass, if any, of the specific data concept being described. The subclass (i.e. the data concept being described) inherits all properties of the superclass.

### 7.3.8 Referenced message

A data concept registry should implement this meta-attribute in a manner that allows cross-referencing from the message to the interface dialogue as well.

### 7.3.9 Referenced data frame

A data concept registry should implement this meta-attribute in a manner that allows cross-referencing from the data frame to the data frame or message as well.

### 7.3.10 Referenced data element

A data concept registry should implement this meta-attribute in a manner that allows cross-referencing from the data element to the data frame or message as well.

## 7.4 Representational meta-attributes

Representational meta-attributes describe requirements for physical representation of data concepts. These meta-attributes define how data elements are exchanged across system interfaces and do not necessarily restrict how the data is stored in databases or appears in user interfaces.

### 7.4.1 Data type

The logical representation of the data concept as expressed as a valid ASN.1 Type, as defined in ISO/IEC 8824-1. The exact form of this meta-attribute for messages, data frames, aggregate domains, data elements and value domains is specified in the following sub-clauses.

NOTE Additional meta-attributes to support other syntaxes (e.g. CORBA IDL, EDIFACT Graphical Syntax, XML Schema) may be added in future revisions and some existing mandatory attributes may become optional.

#### 7.4.1.1 Data type for messages

The Data Type meta-attribute of a Message data concept will typically be a “SEQUENCE”, “SEQUENCE OF” or “CHOICE”, but any ASN.1 Type shall be allowed. If the Data Type is a “SEQUENCE”, it shall follow the rules defined in 7.4.1.1.1. If the Data Type is a “SEQUENCE OF”, it shall follow the rules defined in 7.4.1.1.2. If the Data Type is a “CHOICE”, it shall follow the rules defined in 7.4.1.1.3.

##### 7.4.1.1.1 Presentation of a “sequence” type

The Type field of each NamedType within an ASN.1 “SEQUENCE” Type shall reference the ASN.1 Name meta-attribute of a defined Data Element or Data Frame. If the referenced data concept is not defined in the same module, it shall be written as an externalTypeReference, as defined in ISO/IEC 8824-1. There are no restrictions placed on the identifier field, other than those specified by the normal rules of ASN.1.

Designers should carefully consider version control and backwards compatibility. In particular, SEQUENCE Types should typically include an extension marker to allow for expansion in a later version, unless the designer is highly confident that no extension will be needed.

EXAMPLE In the following example, Vehicle-make and Vehicle-model must be the ASN.1 Names of defined data elements or data frames within the IS14817-1-bc-1 module:

```
Data Type   SEQUENCE {
                a    IS14817-1-bc-1.Vehicle-make,
                b    IS14817-1-bc-1.Vehicle-model,
                ... }
```

##### 7.4.1.1.2 Presentation of a “sequence of” type

The Type field within an ASN.1 “SEQUENCE OF” Type shall reference the ASN.1 Name meta-attribute of a defined Data Element or Data Frame. If the referenced data concept is not defined in the same module, it shall be written as an externalTypeReference, as defined in ISO/IEC 8824-1.

EXAMPLE In the following example, Vehicle-information must be the ASN.1 Name of a defined data element or date frame within the IS14817-1-ac-1 module:

```
Data Type   SEQUENCE OF IS14817-1-ac-1.Vehicle-information
```

##### 7.4.1.1.3 Presentation of a “choice” type

The Type field of each NamedType within an ASN.1 “CHOICE” Type shall reference the ASN.1 Name meta-attribute of a defined Data Element or Data Frame. If the referenced data concept is not defined in the same module, it shall be written as an externalTypeReference, as defined in ISO/IEC 8824-1. There are no restrictions placed on the identifier field, other than those specified by the normal rules of ASN.1.

Designers should carefully consider version control and backwards compatibility. In particular, CHOICE Types should typically include an extension marker to allow for expansion in a later version, unless the designer is highly confident that no extension will be needed.

EXAMPLE In the following example, Vehicle-make and Vehicle-model must be the ASN.1 Names of defined data elements or data frames within the IS14817-1-bc-1 module:

```
Data Type   CHOICE {
                     a    IS14817-1-bc-1.Vehicle-make,
                     b    IS14817-1-bc-1.Vehicle-model,
                     ... }
```

### 7.4.1.2 Data type for data frames

The Data Type meta-attribute of a Date Frame data concept shall be the ASN.1 Name of an Aggregate Domain data concept.

### 7.4.1.3 Data type for aggregate domains

The Data Type meta-attribute of an Aggregate Domain data concept shall be a “SEQUENCE”, “SEQUENCE OF” or “CHOICE”; it shall not be any primitive Type. If the Data Type is a “SEQUENCE”, it shall follow the rules defined in 7.4.1.1.1. If the Data Type is a “SEQUENCE OF”, it shall follow the rules defined in 7.4.1.1.2. If the Data Type is a “CHOICE”, it shall follow the rules defined in 7.4.1.1.3.

### 7.4.1.4 Data type for data elements

The Data Type meta-attribute of a Data Element data concept shall be the ASN.1 Name of a Value Domain.

### 7.4.1.5 Data type for value domains

The Data Type meta-attribute of a Value Domain shall be a valid ASN.1 primitive Type.

An integer type may be used to represent a fixed-point decimal type if the Unit of Measure meta-attribute or metadata within the Data Type indicates the offset of the decimal.

UTF8String shall be used for the character string type in the case of international information exchange. BMPString and IA5String may be used in regional or country-specific DDs.

Enumerated value domains should use the INTEGER type with namedValues if there is a desire to rigidly associate a numeric value with a meaning. The ENUMERATED type does not provide this type of rigid value association.

Designers should carefully consider version control and backwards compatibility. In particular, ENUMERATED Types should typically include an extension marker to allow for expansion in a later version, unless the designer is highly confident that no extension will be needed.

EXAMPLE In the following example, the value domain is represented by an integral value from zero to 255, a useful range as this can be encoded within a single byte within some encoding schemes.

```
Data Type      INTEGER (0..255)
```

NOTE “Submitters” of version 1 ASN.1 definitions are strongly encouraged to provide extension markers where appropriate. For example, in an element “vehicle-type ENUMERATED {unknown, car, heavy-goods-vehicle, public-service-vehicle, ...} the ellipsis should certainly be included to indicate possible additions in version 2.

### 7.4.2 Format

The format meta-attribute shall not be interpreted to override the restrictions in either the data type or valid value rule meta-attributes. The specific layout depends upon the data type of the value domain.

### 7.4.3 Unit of measure

Units shall be defined in accordance with ISO 80000-1. For units of enumeration, such as equipment or units of issue the measure shall be defined using this meta-attribute.

### 7.4.4 Valid value rule

In no case shall the Valid Value Rule allow values that are not in accordance with the Data Type meta-attribute. While the precise abstract data exchange format is defined by the Data Type meta-attribute, a valid value rule may be used to further constrain valid values (e.g. due to relationships to other data concepts) or to provide a natural language text definition of the data format.

### 7.4.5 Constraint

The constraint shall be an ASN.1 Subtype Element as defined in ISO/IEC 8824-1:—, Clause 47. The subtype shall be applied to the Data Type, if it is a primitive ASN.1 type or shall be applied to the Type of the “value” component if the Data Type resolves to a “SEQUENCE” type. While the ASN.1 Subtype Element can be included in the Data Type meta-attribute, it is recommended to place it in the Constraint meta-attribute to allow for the greatest possible reuse of data concepts and to accommodate the two-pass validation process allowed by XML implementations. For example, this allows a single generic “code” value domain to be defined and used for all enumerated values. A specific data element can then reference the “code” Value Domain while defining its own appropriate Constraint to modify the more generic Value Domain. With the XML two-pass validation, the first pass will ensure that the data type is valid while the second pass will ensure that the actual encoded value is valid.

The constraint should be defined as tightly as appropriate in order to minimize the encoding size of the information while also accommodating the need of the data exchange.

# Annex A
(normative)

# Meta-attribute requirements

## A.1 General

This Annex specifies the applicability of meta-attributes to each data concept.

## A.2 Overview

The definitions of the meta-attributes referenced in this Annex are found in Clause 7.

In the tables contained in this Annex, each row denotes applicability of a specific meta-attribute for each data concept. The first column of each row provides the name of a meta-attribute. The clause number, wherein the referenced meta-attribute is defined, is given in the second column. The next 11 columns list applicability of the meta-attribute to each of the data concepts. The last column is for any notes pertinent to the meta-attribute and its relationship to the data concepts.

The applicability of a meta-attribute is defined by a code defined as follows:

- "M" = mandatory. Mandatory meta-attributes are required for the referenced data concept, without exception.
- "O" = optional. Optional meta-attributes may be implemented, if desired.
- "I" = indicative. Indicative meta-attributes depend upon an "if" condition that is independent of any other meta-attribute. If the "if" condition evaluates to 'true', then the "I" coded meta-attribute is mandatory; otherwise, it is not applicable.
- "A" = Assigned. The value of the meta-attribute is automatically assigned for data concepts by the data concept registry. The meta-attribute is optional for the data concept when used in a data dictionary.
- "N/A" = not applicable.

The note column of each table explains the nature of each indicative meta-attribute and provides other explanatory information.

NOTE Only single values are permitted for each meta-attribute unless specifically identified as "Multiples Allowed".

## A.3 Meta-attribute requirements

Table A.1 defines the applicability of each meta-attribute for each defined data concept.

## Table A.1 — Meta-attribute requirements

| | | Data Concepts | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Meta-attribute** | **Clause** | **Dic-tionary docu-ment** | **Module** | **Object class** | **Data ele-ment** | **Value domain** | **Inter-face dialog** | **Mes-sage** | **Data Frame** | **Aggre-gate domain** | **Notes** |
| Data concept identifier | 7.1.1 | A | A | A | A | A | A | A | A | A | The same data concept identifier may be used for different versions of a data concept. |
| Data concept version | 7.1.2 | A | A | A | A | A | A | A | A | A | The same version may be used for different revisions of a data concept. |
| Data concept revision | 7.1.3 | A | A | A | A | A | A | A | A | A | |
| Nominal version | 7.1.4 | O | O | O | O | O | O | O | O | O | Typically only used for the standard data concept |
| Document identifier | 7.1.5 | M | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | |
| Contextual name | 7.1.6 | M | M | M | M | M | M | M | M | M | Shall conform to B.1. Data dictionaries may indirectly provide this meta-attribute by defining a descriptive name and/or ASN.1 name that conforms to Annex B. |
| Descriptive name | 7.1.7 | A | A | A | A | A | A | A | A | A | |
| Historic descriptive name | 7.1.8 | O | O | O | O | O | O | O | O | O | Multiples allowed |
| ASN.1 name | 7.1.9 | A | A | A | A | A | A | A | A | A | |
| Historic ASN.1 name | 7.1.10 | N/A | O | N/A | O | O | N/A | O | O | O | Multiples allowed |
| Object identifier[a] | 7.1.11 | M | M | M | M | M | M | M | M | M | Shall conform to ISO 14817-3 |
| Uniform resource locator | 7.1.12 | O | I | O | O | O | O | O | O | O | Required for modules if they may be implemented in XML |

[a] Normative meta-attribute; any change to the value of this meta-attribute to an approved data concept shall result in a new version.

**Table A.1** *(continued)*

| | | Data Concepts | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Meta-attribute** | **Clause** | **Dictionary document** | **Module** | **Object class** | **Data element** | **Value domain** | **Interface dialog** | **Message** | **Data Frame** | **Aggregate domain** | **Notes** |
| Definition[a] | 7.2.1 | M | M | M | M | M | M | M | M | M | |
| Source | 7.2.2 | O | O | O | O | O | O | O | O | O | Multiples allowed |
| Data concept type[a] | 7.2.3 | M | M | M | M | M | M | M | M | M | |
| Remark | 7.2.4 | O | O | O | O | O | O | O | O | O | Multiples allowed |
| Context | 7.2.5 | N/A | M | M | M | M | O | M | M | M | Multiples allowed |
| Dialogue order rules[a] | 7.2.6 | N/A | N/A | N/A | N/A | N/A | M | N/A | N/A | N/A | |
| Parent object class[a] | 7.3.1 | N/A | N/A | N/A | M | N/A | N/A | N/A | M | N/A | |
| Precursor | 7.3.2 | O | O | O | O | O | O | O | O | O | Multiples allowed (when merging) |
| Successor | 7.3.3 | O | O | O | O | O | O | O | O | O | Multiples allowed (when diverging) |
| Synonym | 7.3.4 | O | O | O | O | O | O | O | O | O | Multiples allowed |
| Abstract[a] | 7.3.5 | N/A | N/A | M | N/A | N/A | N/A | N/A | N/A | N/A | |
| Multiplicity[a] | 7.3.6 | N/A | N/A | N/A | M | N/A | N/A | N/A | M | N/A | |
| Superclass[a] | 7.3.7 | N/A | N/A | I | N/A | N/A | N/A | N/A | N/A | N/A | Required, if the object class is a subclass |
| Referenced message | 7.3.8 | N/A | N/A | N/A | N/A | N/A | M | N/A | N/A | N/A | Multiples allowed |
| Referenced data frame | 7.3.9 | N/A | N/A | N/A | N/A | N/A | N/A | O | N/A | O | Multiples allowed |
| Referenced data element | 7.3.10 | N/A | N/A | N/A | N/A | N/A | N/A | O | N/A | O | Multiples allowed |
| Data type[a] | 7.4.1 | N/A | N/A | N/A | M | M | O | M | M | M | |
| Format [a] | 7.4.2 | N/A | N/A | N/A | I | I | N/A | N/A | N/A | N/A | Required, if applicable and not already defined |
| Unit of measure[a] | 7.4.3 | N/A | N/A | N/A | I | I | N/A | N/A | N/A | N/A | Required, if applicable and not already defined |

[a] Normative meta-attribute; any change to the value of this meta-attribute to an approved data concept shall result in a new version.

**Table A.1** *(continued)*

| | | Data Concepts | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Meta-attribute** | **Clause** | **Dictionary document** | **Module** | **Object class** | **Data element** | **Value domain** | **Interface dialog** | **Message** | **Data Frame** | **Aggregate domain** | **Notes** |
| Valid value rule[a] | 7.4.4 | N/A | N/A | N/A | I | I | N/A | N/A | I | I | Required, if applicable and not already defined |
| Constraint[a] | 7.4.5 | N/A | N/A | N/A | I | N/A | N/A | N/A | I | N/A | Required, if applicable and not already defined |

[a] Normative meta-attribute; any change to the value of this meta-attribute to an approved data concept shall result in a new version.

# Annex B
(normative)

# Naming conventions

## B.1 Contextual names

Contextual names shall be formulated when developing data concepts according to the requirements described in this Clause. These requirements shall apply to the entire ITS data environment.

The contextual name shall represent the meaning of a data concept within a known context and serve as a synopsis of the data concept's definition.

The contextual name shall be unique within its context.

### B.1.1 Overview

Table B.1 provides a summary of descriptive name formats; precise rules are provided in the indicated clause.

**Table B.1 — Contextual name formats for ITS data concepts**

| Data concept | Clause | Example format |
|---|---|---|
| Dictionary document | B.1.2 | Data concept contextual name |
| Module | B.1.3 | ORG-DictionaryDocument-Module |
| Object class | B.1.5 | DataConceptContextualName |
| Data element | B.1.6 | dataConceptContextualName |
| Value domain | B.1.7 | data-concept-contextual-name |
| Interface dialogue | B.1.4 | DataConceptContextualName:dialogue |
| Message | B.1.4 | DataConceptContextualName:message |
| Data frame | B.1.6 | dataConceptContextualName |
| Aggregate domain | B.1.5 | DataConceptContextualName |

### B.1.2 Dictionary document contextual name format

The contextual name of a dictionary document shall be the full textual name of the dictionary document, including spaces and punctuation, using its normal capitalization rules for the document.

EXAMPLE Intelligent transport systems — ITS data dictionaries — Part 1: Requirements for ITS data definitions

### B.1.3 Module contextual name format

The contextual name of a module shall consist of an organization identifier, a document identifier, and a module identifier.

The organization identifier shall be 3 to 6 uppercase, alphabetic (A-Z) characters that uniquely identify the organization responsible for the module. As of the publication of this part of ISO 14817, recognized organization identifiers include:

— CEN: European Committee for Standardization

— ISO: International Organization for Standardization

The organization shall then assign a document identifier to each dictionary document and shall ensure that each document identifier is unique within its organization. The document identifier may include part numbers, which may be separated by hyphens.

The editor of the document shall define the module identifier for each module within the dictionary document. The module identifier shall be an upper camel case term of 1 to 48 characters.

The three components of the name are then combined, separated by hyphens.

EXAMPLE ISO-14817-1-ValueDomains

### B.1.4 Contextual name format for interface dialogues and messages

The contextual names for interface dialogues and messages shall consist of an upper camel case term of 1 to 48 characters. Each word within the term should be singular, unless a single instance of the data concept represents a plural concept.

EXAMPLE ContextualName

### B.1.5 Contextual name format for object classes and aggregate domains

The contextual name of object classes and aggregate domains shall be an upper camel case term of 1 to 64 characters. Each word within the term should be singular, unless a single instance of the data concept represents a plural concept. Proper names, prepositions and conjunctions are discouraged. The use of hyphen is discouraged because of potential conflicts with name conversion rules.

EXAMPLE ContextualName

### B.1.6 Contextual name format for data elements and data frames

The contextual name of data elements and data frames shall be a lower camel case term of 1 to 64 characters. The term shall indicate the fact, proposition, or observation about the object class being described. An object may have multiple instances of the property, in which case the term shall be in the plural form of a noun; otherwise it shall be in singular noun form. Proper names, prepositions and conjunctions are discouraged. If further disambiguation is needed, a hyphen ("-") may be added along with the contextual name of the value domain or aggregate domain referenced by the data concept's data type meta-attribute.

EXAMPLE contextualName

### B.1.7 Value domain contextual name format

#### B.1.7.1 General

The contextual name of a value domain shall be constructed as a series of one or more words or abbreviations, designed to quickly convey the format in which the information is represented. Each word or abbreviation shall be in all lowercase letters with hyphens separating words. The contextual name shall start with one of the representation terms identified in B.1.7.2 in order to identify the general nature of value domain. The remaining words shall uniquely characterize the specific value domain. The maximum length of the contextual name shall be 64 characters.

EXAMPLE code-country (e.g. a two letter country code)

#### B.1.7.2 Representation class terms

The contextual name of a value domain shall start with one of the representation class terms or abbreviations as listed below. A representation class abbreviation should not be further abbreviated.
**amount** (amt): A numeric quantification of a monetary value expressed in monetary units, such as dollars

and cents. Some example explicit value domains for this representation term are $$$$.cc, €€€€,cc, and £££££.pp, where "$$$$", "€€€€" and "££££" represents dollars, euros or pounds sterling (or other main currency denomination) to whatever number of significant digits is required, and "cc" or "pp" represents cents or pence. For non-monetary numeric values, use the "quantity" value domain term.

**binary** (bin): A finite-length sequence of binary octets that represents some sort of information.

**code** (cd): A set of values where each value represents a specific meaning.

NOTE Example value domains: ANSI X3.38–1988, ISO 3166-1:2013, ISO 3166-2:2013, ISO 3166-3:2013, and ISO/IEC 5218:2004 are all of the representation class "code".

**date** (dt): A specific calendar day

**date-time** (dtm): A specific instant in all of history

**duration** (dur): A length of time.

**identifier** (id): A value used to uniquely identify an object within a known context.

**indicator** (ind): A list of two mutually exclusive values that express the only possible states of an entity.

**image** (img): A graphical or pictorial item, such as a map, diagram, picture, motion picture, or icon.

**measure** (ms): A numeric value that can be measured with standard SI units and can be subjected to computational manipulations.

**name** (nm): A human pronounceable character string that is used for identification.

**numeric** (nbr): A non-computational numeric or alphanumeric string used to designate an item, e.g. a serial number, telephone number, street number, apartment number, or social security number.

**percent** (pct): A ratio of two quantities expressed in numeric format as a decimal number multiplied by 100.

**quantity** (qty): A countable, non-monetary numeric value (e.g. number of vehicles) that can be subjected to computational manipulations.

**rate** (rt): A numeric ratio without any units indicating the quantity or measure of one item to another with the same units, for example, 1:10.

**sound** (snd): An audio sequence with explicit beginning and end. An example explicit value domain applicable to a data element using this value domain term should be specified in its value domain meta-attribute, e.g. "wav."

**text** (txt): An alphanumeric string (formatted or unformatted), potentially including spaces; e.g. a street name, or the contents of a document, message, or other file. An example explicit value domain for "text" is ISO/IEC 10646.

**time** (tm): An instant of time that occurs every day.

**video** (vid): A video graphic image

## B.2 Descriptive names

### B.2.1 Descriptive name — General rules

Every data concept has a descriptive name and a fully-qualified descriptive name.

Except for modules, dialogues, messages, data elements and data frames, the descriptive name of a data concept shall be identical to its contextual name.

The fully-qualified descriptive name shall be as defined in B.3.8.

### B.2.2 Descriptive name for a module

The descriptive name of a module shall consist of the contextual name followed by a hyphen followed by the data concept version.

EXAMPLE ISO-14817-1-ValueDomains-1

### B.2.3 Descriptive name for dialogues and messages

The descriptive name of interface dialogues and messages shall consist of the contextual name followed by a colon and followed by the extension term identified in Table B.2.

**Table B.2 — Extension terms**

| Data concept | Extension term |
|---|---|
| Interface dialogue | dialogue |
| Message | message |

### B.2.4 Descriptive name for data elements and data frames

The descriptive name of data elements and data frames shall consist of three parts. The first part shall be the contextual name of the object class referenced by the data element's parent object class meta-attribute. The second part shall be the contextualName of the data element or data frame. The third part shall be the descriptive name of the value domain or aggregate domain referenced by the data element's (or data frame's) data type meta-attribute. The first two parts shall be separated by a period (".") while the last two parts shall be separated with a colon ("":).

EXAMPLE 1 For a data element: ObjectClassTerm.propertyTerm:value-domain-term

EXAMPLE 2 For a data frame: ObjectClassTerm.propertyTerm:AggregateDomainTerm

### B.2.5 Fully-qualified descriptive name formats

Because there is a large number of independent groups that may develop data concepts, it would be problematic to require globally unique descriptive names for every data concept. Instead, this part of ISO 14817 only requires a descriptive name to be unique within the scope of the defining context, as defined by the context meta-attribute.

Whenever a reference is made to a data concept that is formally defined in another module, the reference shall use the fully qualified descriptive name. The fully qualified descriptive name of any data concept is obtained by concatenating the descriptive name of the data concept's context meta-attribute with a double colon and the descriptive name of the data concept.

EXAMPLE ISO-14817-1-Location-1::GeoLocation.latitude:measure

## B.3 ASN.1 names

### B.3.1 Overview

The convention for developing descriptive names for data concepts represented in DDs is given above. However, conventions used in forming the descriptive names are in conflict with some of the rules of ASN.1, therefore the names must be altered slightly when used in an ASN.1 context. The descriptive names can be converted to ASN.1 names using the rules described in this clause. The ASN.1-conforming name shall be represented in an ITS data dictionary using the meta attribute called ASN.1 Name.

B.3.2 provides an overview of the constraints placed on names by the ASN.1 standard. B.3.3 to B.3.7 provide the detailed rules for converting descriptive names to ASN.1 names. B.3.8 provides the rules for converting an ASN.1 Name to a fully-qualified ASN.1 Name.

### B.3.2 Use of ASN.1 syntax

The ASN.1 naming rules are as follows;

- the set of characters from which names can be formed in ASN.1 are: A-Z, a-z, hyphen, and 0-9;
- typereferences are names that must start with a capital letter;
- identifiers are names that must start with a lower case letter;
- a name (i.e. a typereference or an identifier) cannot contain two or more contiguous hyphens, nor start or end with a hyphen. Note that ASN.1 names are case-sensitive;
- while no maximum length is placed on names by ASN.1; this part of ISO 14817 limits names by way of this Annex.

In the following example

```
PersonInfo ::= SEQUENCE {
    name               Person-name,
    age                Person-age,
    address            Person-homeAddress
}
Person-name            ::= Name
Person-age             ::= Quantity (0..160)
Person-homeAddress ::= Address

Name                   ::= VisibleString (SIZE(1..64))
Quantity               ::= INTEGER
Address                ::= VisibleString (SIZE(1..100))
```

PersonInfo, Person-name, Person-age, Person-homeAddress, Name, Quantity, and Address are typereferences; name, age, and address are identifiers.

### B.3.3 ASN.1 name of a dictionary document

The ASN.1 name of a dictionary document shall be identical to its document identifier with spaces removed.

### B.3.4 ASN.1 name of modules, object classes, and aggregate domains

The ASN.1 name of modules, object classes, and aggregate domains shall be identical to its contextual name.

### B.3.5 ASN.1 name of data elements and data frames

The ASN.1 name of data elements and data frames shall start with the contextual name of the of the object class referenced by the data concept's parent object class meta-attribute. This shall be followed by a hyphen ("-") and then immediately followed by the contextual name of the data concept.

EXAMPLE 1 GeoLocation-latitude

EXAMPLE 2 GeoLocation-latitude-measure

EXAMPLE 3 Vehicle-position

EXAMPLE 4 Vehicle-position-GeoLocation

### B.3.6 ASN.1 name of a value domain

The ASN.1 name of a value domain shall be identical to its contextual name with the initial letter capitalized and immediately followed by the word “Type”.

EXAMPLE CodeType

### B.3.7 ASN.1 name of messages and dialogues

The ASN.1 name of messages and interface dialogues shall be identical to its contextual name with the “:message” or “:dialogue” extension removed.

EXAMPLE CollisionRequestMessage

### B.3.8 Fully-qualified name formats

Because there is a large number of independent groups that may develop data concepts, it would be problematic to require globally unique ASN.1 names for every data concept. Instead, this part of ISO 14817 only requires a name to be unique within the scope of the defining module.

Whenever a reference is made to a data concept that is formally defined in another module, the reference shall use a fully qualified name. The fully qualified descriptive name of any data concept is obtained by concatenating the module ASN.1 name with a period and the ASN.1 name of the data concept.

EXAMPLE ISO-14817-1-Location-1.GeoLocation-latitude

# Annex C
(normative)

# Preferred data concepts

## C.1 General

NOTE Preferred data concepts define the recommended syntax and semantics to be used when discussing the concept anywhere within the ITS domain. Data concepts using alternate syntax and/or sematics are allowed; the preferred data concepts are designated as such to encourage their adoption where feasible. The developers of a standard are considered the ultimate experts on the needs of the standard and are still allowed the full flexibility of ASN.1 in their design.

## C.2 Object classes

Table C.1 defines the object classes used in the context of this part of ISO 14817

**Table C.1 — Object Classes**

| Contextual Name | OID[a] | Definition | Source | Abstract | Superclass |
|---|---|---|---|---|---|
| Agency | {its 2} | entity that provides some service | | False | |
| Amount | {its 128} | numeric quantification of monetary value specified in an explicit currency; the unit of currency may be defined within the data concept definition and/or within the presentation of the amount in an encoding | UBL 2.1 | False | |
| BinaryObject | {its 129} | set of finite-length sequences of binary octets | UBL 2.1 | False | |
| CodeList | {its 130} | series of shorthand representations of definitive concepts | | False | |
| GeoLocation | {its 134} | point on or near the surface of Earth | | False | Location |
| Location | {its 131} | locatable place or position | | False | |
| MeasureObject | {its 132} | numeric value determined by measuring an object along with the specified unit of measure | | False | |
| TextObject | {its 133} | pronounceable character string, generally in the form of words in a language | UBL 2.1 | False | |
| Vehicle | {its 1} | machine for the transportation of goods and people on land | ISO 11783-1:2007, 3.70 | False | |

[a] For definition of the parent arcs of the OID, see ISO 14817-3.

## C.3 Modules

Table C.2 defines the modules defined within the context of this part of ISO 14817.

**Table C.2 — Modules**

| Contextual Name | OID[a] | Definition |
|---|---|---|
| ISO-14817-1-Domains-1 | {value-domains-modules 1} | A module that defines the dialogue for exchanging collision reports. |
| [a] For definition of the parent arcs of the OID, see ISO 14817-3. | | |

## C.4 Data Frames

Table C.3 defines the data frames defined within the ISO-14817-1-Vehicle-1 module.

**Table C.3 — Data Frames**

| ASN.1 Name | OID[a] | Definition | Multiplicity | Data Type |
|---|---|---|---|---|
| Vehicle-position3D | {vehicle 1} | The current physical location of the vehicle | 1 | ISO-14817-1-Location-1.GeoLocation3D |
| Vehicle.position2D | {vehicle 2} | The current physical location of the vehicle | 1 | ISO-14817-1-Location-1.GeoLocation2D |
| [a] For definition of the parent arcs of the OID, see ISO 14817-3. | | | | |

## C.5 Aggregate Domains

Table C.4 defines the aggregate domains defined within the ISO-14817-1-Domains-1 module.

Table C.5 defines the aggregate domains defined within the ISO-14817-1-Location-1 module.

**Table C.4 — Aggregate Domains**

| ASN.1 Name | OID[a] | Definition | Parent Object Class | Data Type |
|---|---|---|---|---|
| CodeList | {codelist 0 1} | An set of printable character strings where each string represents a a specific meaning. The Agency referenced by agencyID and agencyName shall be the Agency that maintains the CodeList. | CodeList | `SEQUENCE {`<br>`id      CodeList-id OPTIONAL,`<br>`agencyID  Agency-id OPTIONAL,`<br>`agencyName    Agency-name OPTIONAL,`<br>`name        CodeList-name OPTIONAL,`<br>`version     CodeList-version OPTIONAL,`<br>`language    CodeList-language OPTIONAL,`<br>`uri     CodeList-uri OPTIONAL,`<br>`scheme CodeList-scheme OPTIONAL,`<br>`value       CodeList-value`<br>`}` |
| Measure | {measure 0 1} | A numeric value that can be measured with standard units. The unit shall indicate the units used in the measurement. When available the units shall be one of the codes listed at http://www.schemacentral.com/sc/ubl20/a-unitCode-3.html | Measure | `SEQUENCE {`<br>`unit        Measure-unit,`<br>`value       Measure-value`<br>`}` |
| Text | {text 0 1} | A natural language expression. | Text | `SEQUENCE {`<br>`language    Text-laguage,`<br>`value       Text-value`<br>`}` |

[a] For definition of the parent arcs of the OID, see ISO 14817-3.

Table C.5 — Location Aggregate Domains

| ASN.1 Name | OID[a] | Definition | Parent Object Class | Data Type |
|---|---|---|---|---|
| GeoLocation3D | {geoLocation 0 1} | A three-dimensional geographic location. | GeoLocation | `SEQUENCE {`<br>`lat GeoLocation-latitude,`<br>`lon GeoLocation-longitude,`<br>`alt GeoLocation-altitude`<br>`}` |
| GeoLocation2D | {geoLocation 0 2} | A two-dimensional geographic location. | GeoLocation | `SEQUENCE {`<br>`lat GeoLocation-latitude,`<br>`lon GeoLocation-longitude`<br>`}` |

[a] For definition of the parent arcs of the OID, see ISO 14817-3.

## C.6 Data Elements

Table C.6 defines the data elements defined within the ISO 14817-1-Domains-1 module.

**Table C.6 — Data Elements**

| Descriptive Name | OID[a] | Definition | Source | Multiplicity | Data Type |
|---|---|---|---|---|---|
| Agency.id:oid | {agency 1} | The globally unique identifier for the Agency. | | 0..1 | OidType |
| Agency.name:name | {agency 2} | The name by which the Agency is known. | | 0..* | NameType |
| Amount.currencyID:identifier | {amount 1} | The three letter currency code according to the current version of ISO 4217. | UBL 2.1 | 1 | IdentifierType |
| Amount.value:amount[b] | {amount 2} | The amount of currency. | UBL 2.1 | 1 | AmountType |
| BinaryObject.mimeCode:identifier | {binaryObject 1} | The registered mime type that describes the encoding of the binary value. | UBL 2.1 | 1 | IdentifierType |
| BinaryObject.format:identifier | {binaryObject 2} | The valid layout information for the value. | UBL 2.1 | 0..1 | IdentifierType |
| BinaryObject.encodingCode:identifier | {binaryObject 3} | The decoding algorithm for the binary object. | UBL 2.1 | 0..1 | IdentifierType |
| BinaryObject.characterSet:identifier | {binaryObject 4} | The characterSet used by the value. | UBL 2.1 | 0..1 | IdentifierType |
| BinaryObject.uri:identifier | {binaryObject 5} | The valid uniform resource identifier that specifies where the binary object is located. | UBL 2.1 | 0..1 | IdentifierType |

[a] For definition of the parent arcs of the OID, see ISO 14817-3.

[b] Unit of Measure: 1/100ths of the currencyID, unless otherwise indicated

**Table C.6** *(continued)*

| Descriptive Name | OID[a] | Definition | Source | Multiplicity | Data Type |
|---|---|---|---|---|---|
| BinaryObject.filename:identifier | {binaryObject 6} | The name of the file used to store the binary object. | UBL 2.1 | 0..1 | `IdentifierType` |
| BinaryObject.value.binary | {binaryObject 7} | The octets that represent the binary value. | UBL 2.1 | 1 | `BinaryType` |
| CodeList.id:oid | {codelist 1} | The globally unique identifier assigned to the CodeList. | UBL 2.1 | 0..1 | `OidType` |
| CodeList,name:name | {codelist 2} | The name by which the CodeList is known. | UBL 2.1 | 0..* | `NameType` |
| CodeList.version:identifier | {codelist 3} | The version of the CodeList | UBL 2.1 | 1 | `IdentifierType` |
| CodeList.language:identifier | {codelist 4} | The language used by the CodeList as identified by RFC 5646 and RFC 4647. | UBL 2.1 | 1 | `IdentifierType` |
| CodeList.uri:identifier | {codelist 5} | The valid uniform resource identifier that specifies where the CodeList can be found. | UBL 2.1 | 0..1 | `IdentifierType` |
| CodeList.scheme:identifier | {codelist 6} | The valid uniform resource identifier that specifies where the CodeList scheme can be found. | UBL 2.1 | 0..1 | `IdentifierType` |

[a] For definition of the parent arcs of the OID, see ISO 14817-3.

[b] Unit of Measure: 1/100ths of the currencyID, unless otherwise indicated

Table C.6 *(continued)*

| Descriptive Name | OID[a] | Definition | Source | Multiplicity | Data Type |
|---|---|---|---|---|---|
| CodeList.value:identifier | {codelist 7} | The selected code from the CodeList that represents the value to be encoded. | UBL 2.1 | 1 | `IdentifierType` |
| Measure.unit:identifier | {measure 1} | The units used to measure the value. | UBL 2.1 | 1 | `IdentifierType` |
| Measure.value:measure | {measure 2} | The number of units to be conveyed by the value. | UBL 2.1 | 1 | `MeasureType` |
| Text.language:identifier | {text 1} | The language used by the text field as identified by RFC 5646 and RFC 4647. | UBL 2.1 | 1 | `IdentifierType` |
| Text.value:text | {text 2} | The text to be conveyed. | UBL 2.1 | 1 | `TextType` |

[a] For definition of the parent arcs of the OID, see ISO 14817-3.

[b] Unit of Measure: 1/100ths of the currencyID, unless otherwise indicated

Table C.7 defines the data elements defined within the ISO-14817-1-Location-1 module.

Table C.7 — Location Data Elements

| Descriptive Name | OID[a] | Definition | Multiplicity | Data Type | Unit of Measure |
|---|---|---|---|---|---|
| GeoLocation.latitude:measure | {geoLocation 1} | The latitude of the location using the WGS-84 realization that is in effect at the time the data are recorded.[b] | 1 | `ISO-14817-1-Domains-1.MeasureType`<br>`(-900000000..900000001)` [c] | 1/10 micro degrees (10^-7 degrees) |

[a] For definition of the parent arcs of the OID, see ISO 14817-3.

[b] The earth's surface is constantly moving; this data element should typically be used with a timestamp, especially if positional accuracy is needed in relation to what layment would consider to be static features on the earth's surface. At the time that this part of ISO 14817 was written, the most recent realizations of WGS-84 was G1762.

[c] The value 900000001 represents an error condition or missing value.

[d] The value 1800000001 represents an error condition or missing value.

[e] The value 57344 represents and error condition or missing value.

**Table C.7** *(continued)*

| Descriptive Name | OID[a] | Definition | Multiplicity | Data Type | Unit of Measure |
|---|---|---|---|---|---|
| GeoLocation.longitude:measure | {geoLocation 2} | The longitude of the location using the WGS-84 realization that is in effect at the time the data are recorded.[b] | 1 | `ISO-14817-1-Domains-1.MeasureType`<br>`(-1800000000..1800000001)`[d] | 1/10 micro degrees (10^-7 degrees) |
| GeoLocation.altitude:measure | {geoLocation 3} | The altitude of the location using the WGS-84 realization that is in effect at the time the data are recorded.[b] | 1 | `ISO-14817-1-Domains-1.MeasureType`<br>`(-8191..57344)`[e] | 1/10 m |
| Vehicle-vin | {vehicle 3} | | | | |

[a] For definition of the parent arcs of the OID, see ISO 14817-3.

[b] The earth's surface is constantly moving; this data element should typically be used with a timestamp, especially if positional accuracy is needed in relation to what laymen would consider to be static features on the earth's surface. At the time that this part of ISO 14817 was written, the most recent realizations of WGS-84 was G1762.

[c] The value 900000001 represents an error condition or missing value.

[d] The value 1800000001 represents an error condition or missing value.

[e] The value 57344 represents and error condition or missing value.

## C.7 Value Domains

Table C.8 defines the value domains defined within the ISO-14817-1-Domains-1 module.

**Table C.8 — Value Domains**

| ASN.1 Name | OID[a] | Definition | Remarks | Source | Data Type | Unit of Measure |
|---|---|---|---|---|---|---|
| AmountType | {value-domains 1} | numeric quantification of monetary value specified in an explicit currency | | UBL 2.1 | `INTEGER` | Shall be defined by data element |
| BinaryType | {value-domains 2} | finite-length sequence of binary octets that represents some sort of information. | | UBL 2.1 | `OCTET STRING` | |

[a] For definition of the parent arcs of the OID, see ISO 14817-3.

**Table C.8** *(continued)*

| | | | | | | |
|---|---|---|---|---|---|---|
| CodeType | {value-domains 3} | list of integer values where each value is explicitly defined to have a precise meaning, typically unrelated to the mathematical meaning of the integer value | | | INTEGER | |
| DateType | {value-domains 4} | Gregorian calendar date | | | DATE | |
| DateTimeType | {value-domains 5} | specific point in time specified by the Gregorian calendar date and time of day | | | DATE-TIME | |
| DurationType | {value-domains 6} | period of time spanning from a start point to an end point | | | DURATION | |
| IdentifierType | {value-domains 7} | character string used to uniquely distinguish one instance of an object within a known context | Integral identifiers should use the value domain number. | UBL 2.1 | VisibleString | |
| IndicatorType | {value-domains 8} | list of two mutually exclusive values that express the only possible states of an entity | | | BOOLEAN | |
| MeasureType | {value-domains 9} | numeric value that can be measured with standard units | | UBL 2.1 | INTEGER | Shall be defined by data element |
| NameType | {value-domains 10} | human pronounceable character string that is used for identification | | UBL 2.1 | UTF8String | |
| NumericType | {value-domains 11} | Numeric information that is assigned and may be used for identification | Numeric types are not intended to be used in calculations. For example, this may be a telephone number. | UBL 2.1 | INTEGER | |

[a] For definition of the parent arcs of the OID, see ISO 14817-3.

**Table C.8** *(continued)*

| | | | | | | |
|---|---|---|---|---|---|---|
| OidType | {value-domains 12} | globally unique identifier registered on the international object identifier tree and used to uniquely distinguish one instance of an object within a known context. | | | `OBJECT IDENTIFIER` | |
| PercentType | {value-domains 13} | ratio of two quantities. The percentage shall be the ratio multiplied by 100 and rounded to the nearest integral value, unless otherwise indicated. | | UBL 2.1 | `INTEGER` | 1 percent, unless otherwise indicated by the data element |
| QuantityType | {value-domains 14} | countable, non-monetary numeric value (e.g. number of vehicles) that can be subjected to computational manipulations | Values that use SI units should use the measure value domain. | UBL 2.1 | `INTEGER` | countable unit; shall be further defined by data element |
| RateType | {value-domains 15} | ratio of two quantities. The rate shall be the ratio rounded to the nearest integral number, unless otherwise indicated (e.g. a 10:1 ratio would be represented as 10) | | UBL 2.1 | `INTEGER` | whole value, unless otherwise indicated by data element |
| TextType | {value-domains 16} | human pronounceable textual string | | UBL 2.1 | `UTF8String` | |
| TimeType | {value-domains 17} | time within a 24 h day | | | `TIME-OF-DAY` | |

a For definition of the parent arcs of the OID, see ISO 14817-3.

## C.8 ASN.1 Modules

### C.8.1 Domains module, version 1.0

ISO-14817-1-Domains-1 DEFINITIONS AUTOMATIC TAGS::= BEGIN

```
-- Value Domains
AmountType  ::= INTEGER
BinaryType  ::= OCTET STRING
CodeType ::= INTEGER
DateType ::= DATE
DateTimeType  ::= DATE-TIME
DurationType  ::= DURATION
IdentifierType   ::= VisibleString
Id-oidType  ::= OBJECT IDENTIFIER
IndicatorType   ::= BOOLEAN
```

```
MeasureType ::= INTEGER
NameType ::= UTF8String
NumericType  ::= INTEGER
PercentType  ::= INTEGER
QuantityType  ::= INTEGER
RateType ::= INTEGER
TextType ::= UTF8String
TimeType ::= TIME_OF-DAY

-- Data Elements
Agency-id ::= Id-oidType
Agency-name  ::= NameType
Amount-currencyID   ::= IdentifierType
Amount-value ::= AmountType
Binary-mimeCode  ::= IdentifierType
Binary-format ::= IdentifierType
Binary-encodingCode  ::= IdentifierType
Binary-characterSet  ::= IdentifierType
Binary-uri ::= IdentifierType
Binary-filename  ::= IdentifierType
Binary-value ::= BinaryType
CodeList-id ::= Id-oidType
CodeList-name  ::= NameType
CodeList-version   ::= IdentifierType
CodeList-language   ::= IdentifierType
CodeList-uri ::= IdentifierType
CodeList-scheme  ::= IdentifierType
CodeList-value  ::= CodeType
Measure-unit ::= IdentifierType
Measure-value ::= MeasureType
Text-language ::= IdentifierType
Text-value ::= TextType

-- Aggregate Domains
CodeList-details ::= SEQUENCE {
    ·id               CodeList-id           OPTIONAL,
    ·agencyID         Agency-id             OPTIONAL, ·
    agencyName    Agency-name       OPTIONAL, ·
    name          CodeList-name     OPTIONAL,
    ·version       CodeList-version      OPTIONAL,
    ·language         CodeList-language     OPTIONAL, ·
    uri           CodeList-uri ·    OPTIONAL, ·
    scheme        CodeList-scheme   OPTIONAL, ·
    value         CodeList-value·
    }

Measure-details  ::= SEQUENCE { ·
    unit         Measure-unit,
    ·value          Measure-value·
    }

Text-details ::= SEQUENCE {
    ·language    Text-laguage,
    ·value       Text-value
    }

END
```

### C.8.2 Location module, version 1.0

```
ISO-14817-1-Location-1 DEFINITIONS AUTOMATIC TAGS ::= BEGIN
GeoLocation-latitude ::= ISO-14817-1-Domains-1.MeasureType
GeoLocation-longitude ::= ISO-14817-1-Domains-1.MeasureType
GeoLocation-altitude ::= ISO-14817-1-Domains-1.MeasureType
GeoLocation-3-d ::= SEQUENCE { ·
   lat GeoLocation-latitude, ·
   lon GeoLocation-longitude,
   ·alt GeoLocation-altitude·
   }
GeoLocation-2-d ::= SEQUENCE {
   ·lat GeoLocation-latitude, ·
   ·lon GeoLocation-longitude·
   }
END
```

### C.8.3 Vehicle module, version 1.0

```
ISO-14817-1-Vehicle-1 DEFINITIONS AUTOMATIC TAGS::= BEGIN
-- Data Frames
Vehicle-position ::= ISO-14817-1-Location-1.GeoLocation-3D
Vehicle-position2D ::= ISO-14817-1-Location-1.GeoLocation-2D
END
```

# Annex D
(informative)

# Data models

## D.1 General

In order to exchange information and make effective use of it, communicating systems should have a common understanding of the data to be exchanged. This includes a common understanding of the semantic relationships between various pieces of data as well as its syntactic representation. Any disagreement over these relationships is likely an indication of differing semantic meanings of the data, which must be resolved. However, when developing the consensus about these relationships, parties should remember that a common understanding of data across an interface does not require identical implementations internal to a system.

The data relationships defined for an interface should be designed to support a superset of all information exchange needs across the interface. However, any given system may support only a small subset of the defined interface functions and may additionally provide other features not associated with the interface. The most appropriate design of a system is dependent on all of the functional requirements of that system with the interface definition being just one requirement. As such, each system will likely use its own implementation-specific representation of data internally. The internal relationships are defined by a database schema, object model, or some other technique and are then translated as needed into the generic representation defined by the interface definition.

Because many system designers must understand the interface definition, it is critical that it is well documented. Experience suggests that one of the best human-readable ways to express these relationships is through the use of data models. These data models may then be converted into computer readable formats as needed.

This Annex defines guidelines for producing data models within the ITS domain. They are based on the data concepts defined in Clause 6 and illustrated in the example below using the graphical notation of the Unified Modelling Language.

NOTE See the specification for UML 1.3 in ISO/IEC 19501.

## D.2 Data modelling example

Data modelling is herein described through the use of an example. A single example is used in this Annex to present the basic philosophy as well as to demonstrate the rules and guidelines defined by this part of ISO 14817. The example describes a generic relationship between a Stop Point and a Line, as defined in the Public Transport Feature of the Geographic Data File standard (GDF 4.0, ISO CD 14825).

In GDF, all elements regarding the public transport network are grouped in a feature theme called *Public Transport*. This theme contains all the public transport basic features that can be related with a geometrical position. A *Stop Point* is a point where passengers can board or alight a public transport vehicle. A *Line* is a group of Routes, which is known by a common name or number. In its simplest form, the relationship between Stop Point and Line could be viewed as a simple association between two object classes as shown in Figure D.1.

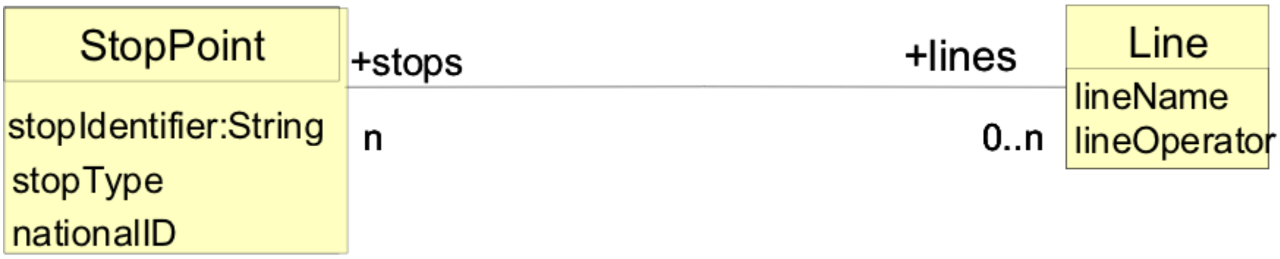

**Figure D.1 — Simple data model**

Figure D.1 depicts two object classes: StopPoint and Line, denoted as rectangles with the object class name in the top partition. A line connecting the two rectangles represents an association and the adornments at each end of the line characterize that association. Figure D.1 also depicts four properties that have been assigned to object class StopPoint which are stopIdentifier, stopType, nationalID, and lines. Three of the properties are included in the attribute partition of the object class rectangle while the fourth property (lines) is placed next to the object class Line to denote the role performed by the Line object class in association with the StopPoint object class. There are three properties assigned to Line: lineName, lineOperator, and stops. Two of the properties are included in the object class rectangle while the third (stops) is placed next to the object class StopPoint to denote the role performed by the StopPoint object class in association with the Line object class. The "stops" property of Line may have a number of instances (i.e. a Line is associated with n StopPoints) and a StopPoint is associated with zero to n instances of Line. Finally, the figure indicates that the value domain "String" defines the representational form of stopIdentifier.

However, some systems may wish to record more structural semantics pertaining to this public transport context. To this end, Figure D.2 provides a more comprehensive set of data and relationships. This representation provides all of the information provided in the previous example, plus additional information about the StopPoint's participation in different contexts. See Table D.1 for a summary and examples of StopPoint relationships. The example StopPoint instance, "Regent Street 1200-1," is part of an express route which may have only a few other StopPoints between its origin and destination. At the same time, "Regent Street 1200-1" is on two other Routes that act as feeder routes. Thus, "Regent Street 1200-1" is simultaneously registered with three Routes, while many or all of the StopPoints on the feeder routes may be part of just one Route.

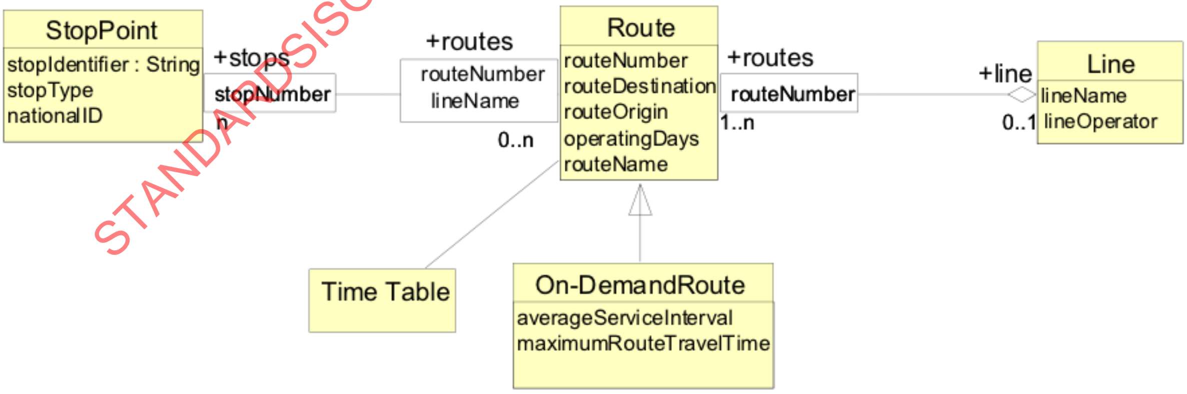

**Figure D.2 — Comprehensive data model**

Table D.1 — StopPoint relationships

| ObjectClass: | StopPoint | Route | | Line |
|---|---|---|---|---|
| **Property:** | **stopIdentifier** | **routeOrigin** | **routeName** | **lineName** |
| **Value Domain:** | **String** | **String** | **String** | **String** |
| **Instance:** | Regent Street 1200–1 | Green Square | Green Square inbound express | Penfield Station |
| **Instance:** | Regent Street 1200-1 | Templestowe | Templestowe-Lipton | Templestowe District |
| **Instance:** | Regent Street 1200-1 | Longfield | Longfield-Templestowe | Templestowe District |
| **Instance:** | Lavender Lane 200-1 | Templestowe | Templestowe-Lipton | Templestowe District |
| **Instance:** | Fountain Circuit 400-2 | Longfield | Longfield-Templestowe | Templestowe District |

The comprehensive model also indicates that Line identifies StopPoint through the assigned identifier (stopNumber), while StopPoint connects to its various Routes by routeNumber and lineName. The comprehensive model also indicates that a given Route (instance) may be associated with none or only one Line. Finally, the model indicates that the maximumRouteTravelTime is only relevant in the model if Route is an On-Demand Route, in which case, averageServiceInterval is also recorded for traveller information purposes.

An alternative third view of this model is depicted in Figure D.3. This diagram allows a system to store all of the information defined in Figure D.2, but it does not provide the reader with some of the more specific semantic relationships. For example, Figure D.3 omits any relationship between the first routeDestination and the first routeOrigin. This does not mean that none exists, but the model does not call out this detail. Nonetheless, an implementer could easily develop a system based on the third model while applying the rules defined in the second model and be able to successfully provide the same data as provided in Figures D.1 and D.2.

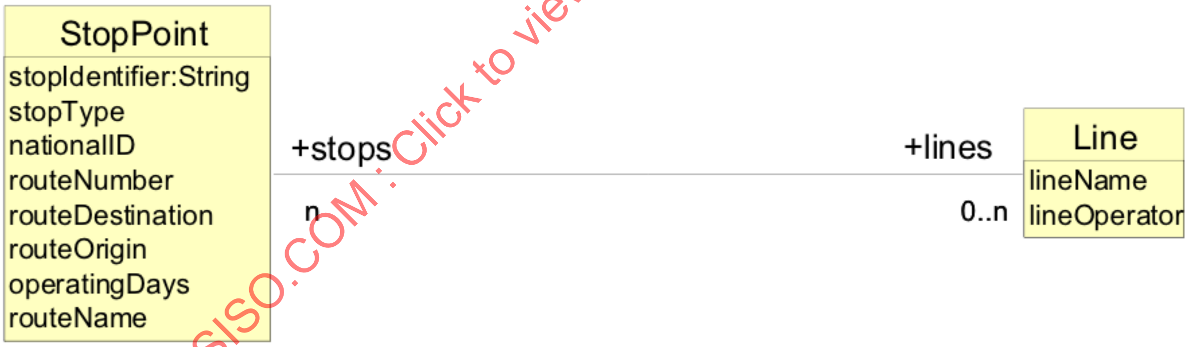

Figure D.3 — Implementation data model

Thus, Figure D.1 and D.3 are valid implementation models for the data model defined by Figure D.2 (although the system implementing Figure D.1 would only support a subset of the defined information). The intent behind data modelling is to precisely define the data and relationships so that all parties involved understand the intent of the model but an implementation of the model is allowed to simplify the design in order to meet other system requirements.

## D.3 Information interchange example

The data concept registry process defined within this part of ISO 14817 is consistent to enable implementation(s) of the ISO ITS architecture as defined in ISO/TR 14813-2 and ISO/TR 14813-3. However, this does not preclude the application of a DD using an alternative International, Regional or National System Architecture methodologies or techniques, indeed, the CIDCR will ease migration and interoperability between such approaches. The example below sketches how data concepts can be developed and illustrates some of the important meta-attributes to be recognized for a DD.

The meta-attributes required to document data concepts in a DD are listed in the tables in Annex A. The most important meta-attributes applicable to modelling an Interface Dialogue are illustrated by the following example. The example is based on the semantics of the data model of Figure D.2.

The example interface dialogue is derived from the public transport use case diagram in ISO 14813-2, Figure 19. A small part of that diagram is reproduced in Figure D.4. Figure D.4 depicts the provision of public transport information to a traveller.

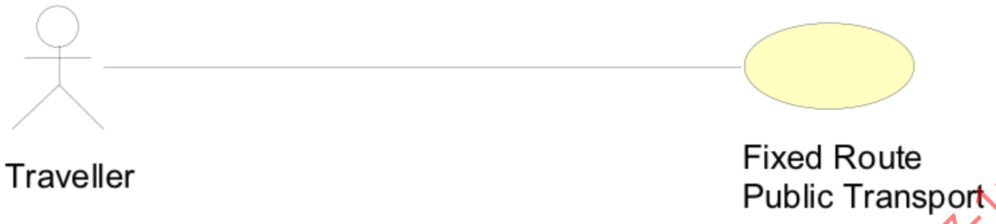

**Figure D.4 — Public transport use case diagram (subset)**

NOTE The graphic symbols in Figure D.4 are from the Unified Modelling Language. The stick figure denotes an actor, an external system or user, that obtains or participates in the functionality of the ITS system. The ellipse denotes a unit of functionality called a use case. A use case delivers meaningful results to one or more actors.

The interface dialogue pertaining to an information interchange between travellers and a bus fixed route public transport service could support many types of queries pertaining to routes and timetables. This is depicted by the high level interaction in ISO 14813-3, Figure 51 labelled “access public transport running data”. In Figure D.5 a UML sequence diagram is shown that would support the specific query “where does route 2 of the Penfield line stop?” The sequence diagram is representing the interface dialogue between one object class instance (a traveller at a particular travel terminal) and another object class instance (Penfield Line, a particular public transport line). By definition the UML sequence diagram represents time running vertically down the diagram. Additional examples pertinent to the proposed interface would require further elaboration of Table D.5 and corresponding elaboration of the TimeTable object class in Figure D.2.

The referenced data concepts of the example are described in Table D.2. The cells in Table D.2 correspond to instances of key mandatory meta-attributes specified in Table A.1 relating to the interface dialogue. The values in each cell (e.g. “id1”) denote unique identifiers for data concept instances. These values would be implemented using one of the meta-attributes defined in Table D.1 that are unique data concept identifiers (e.g. data concept identifier or object identifier). Using a top down approach, the interface dialogue, Traveller << BusQuery >> ISP, references messages with Data Concept Indentifiers (DCIs) of m1 and m2. Message m1, RouteRequest():message, references a single data element whose DCI is de2 and whose “Descriptive Name” is Route.routeNumber:identifier. Route.routeNumber:identifier references the object class whose DCI is c2 and whose “escriptive Name” is Route. This then may be related back to Figure D.2 where routeNumber is identified as a property of the object class Route. In like fashion, the relationships surrounding message m2 may be traced. Additional model information is contained in the association data concepts a1 and a2 which represent the relationships among the three object classes presented in Figure D.2.

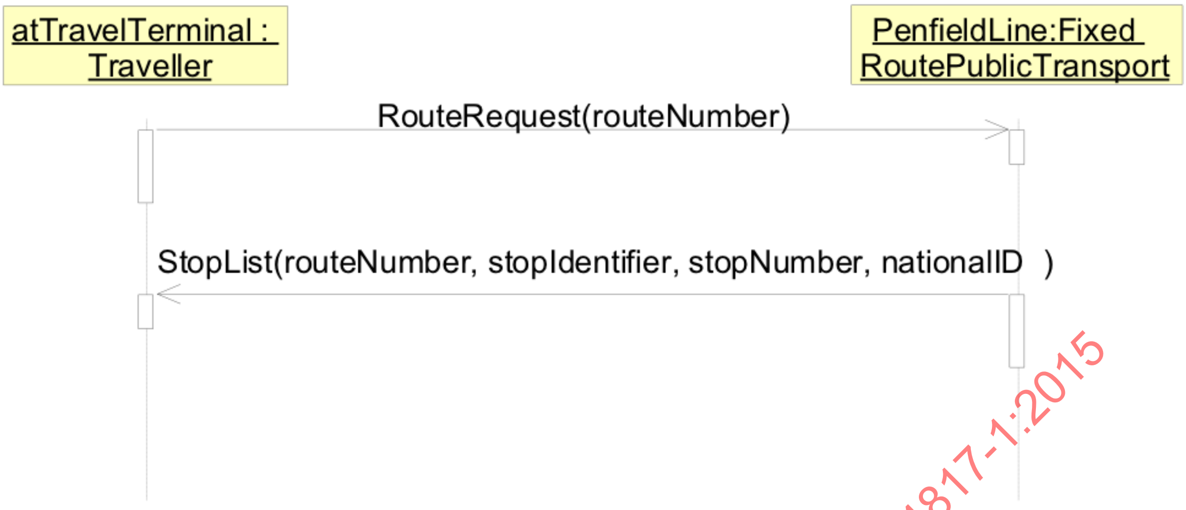
atTravelTerminal : Traveller
PenfieldLine:Fixed RoutePublicTransport
RouteRequest(routeNumber)
StopList(routeNumber, stopIdentifier, stopNumber, nationalID )

**Figure D.5 — Interaction diagram for bus query**

NOTE The interface dialogue may be a sequence of a set of messages and a set of messages. Time runs vertically down the diagram.

**Table D.2 — Interface dialogue meta attributes**

| Data-ConceptID | Data concept type | Descriptive name | Definition | Ref data elements | Ref classes | Ref messages |
|---|---|---|---|---|---|---|
| id1 | Interface Dialogue | Traveller <-Bus-Query-> ISP | Enables the information service provider to satisfy the query: what are the stop points on a route? | | c1, c2, c3 | m1, m2 |
| m1 | Message | RouteRequest:message | Enables remote traveller support to pose the query: what are the stop points on a route? | de2 | | |
| m2 | Message | Stop List:message | Enables the information service provider to issue the results of the query: what are the stop points on a route? | de2, de3, de4, de5 | | |
| de1 | Data Element | Line.lineName:-text | The name of a public transport Line | | c1 | |
| de2 | Data Element | Route. routeNumber: identifier | An identifier of a Route on a public transport Line | | c2 | |
| de3 | Data Element | StopPoint. stopIdentifier: identifier | An identifier in a limited context for a public transport Stop Point | | c3 | |
| de4 | Data Element | StopPoint.stopNumber: number | An ordering of a public transport StopPoint in a Route | | c3 | |
| de5 | Data Element | StopPoint. nationalID: identifier | A national identifier for a public transport StopPoint | | c3 | |