INTERNATIONAL STANDARD

ISO 21219-2

First edition
2019-07

Intelligent transport systems — Traffic and travel information (TTI) via transport protocol experts group, generation 2 (TPEG2) —

Part 2: UML modelling rules (TPEG2-UMR)

Systèmes intelligents de transport — Informations sur le trafic et le tourisme via le groupe expert du protocole de transport, génération 2 (TPEG2) —

Partie 2: Règles de modelage UML

Reference number
ISO 21219-2:2019(E)

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

Published in Switzerland

Contents

Foreword

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

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

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

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

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

This document was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*.

This first edition cancels and replaces ISO/TS 21219-2:2014 which has been technically revised. The main changes compared to the previous edition are as follows:

A list of all parts in the ISO 21219 series can be found on the ISO website.

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

Introduction

History

TPEG technology was originally proposed by the European Broadcasting Union (EBU) Broadcast Management Committee, who established the B/TPEG project group in the autumn of 1997 with a brief to develop, as soon as possible, a new protocol for broadcasting traffic and travel-related information in the multimedia environment. TPEG technology, its applications and service features were designed to enable travel-related messages to be coded, decoded, filtered and understood by humans (visually and/or audibly in the user's language) and by agent systems. Originally, a byte-oriented data stream format, which may be carried on almost any digital bearer with an appropriate adaptation layer, was developed. Hierarchically structured TPEG messages from service providers to end-users were designed to transfer information from the service provider database to an end-user's equipment.

One year later, in December 1998, the B/TPEG group produced its first EBU specifications. Two documents were released. Part 2 (TPEG-SSF, which became ISO/TS 18234-2) described the syntax, semantics and framing structure, which was used for all TPEG applications. Meanwhile, Part 4 (TPEG-RTM, which became ISO/TS 18234-4) described the first application for road traffic messages.

Subsequently, in March 1999, CEN/TC 278, in conjunction with ISO/TC 204, established a group comprising members of the former EBU B/TPEG and this working group continued development work. Further parts were developed to make the initial set of four parts, enabling the implementation of a consistent service. Part 3 (TPEG-SNI, ISO/TS 18234-3) described the service and network information application used by all service implementations to ensure appropriate referencing from one service source to another.

Part 1 (TPEG-INV, ISO/TS 18234-1) completed the series by describing the other parts and their relationship; it also contained the application IDs used within the other parts. Additionally, Part 5, the public transport information application (TPEG-PTI, ISO/TS 18234-5), was developed. The so-called TPEG-LOC location referencing method, which enabled both map-based TPEG-decoders and non-map-based ones to deliver either map-based location referencing or human readable text information, was issued as ISO/TS 18234-6 to be used in association with the other applications of parts of the ISO/TS 18234 series to provide location referencing.

The ISO/TS 18234 series has become known as TPEG Generation 1.

TPEG Generation 2

When the Traveller Information Services Association (TISA), derived from former forums, was inaugurated in December 2007, TPEG development was taken over by TISA and continued in the TPEG applications working group.

It was about this time that the (then) new Unified Modelling Language (UML) was seen as having major advantages for the development of new TPEG applications in communities who would not necessarily have binary physical format skills required to extend the original TPEG TS work. It was also realized that the XML format for TPEG described within the ISO/TS 24530 series (now superseded) had a greater significance than previously foreseen, especially in the content-generation segment and that keeping two physical formats in synchronism, in different standards series, would be rather difficult.

As a result, TISA set about the development of a new TPEG structure that would be UML-based. This has subsequently become known as TPEG Generation 2.

TPEG2 is embodied in the ISO/TS 21219 series and it comprises many parts that cover introduction, rules, toolkit and application components. TPEG2 is built around UML modelling and has a core of rules that contain the modelling strategy covered in ISO 21219-2, ISO 21219-3 and ISO 21219-4 and the conversion to two current physical formats: binary and XML; others could be added in the future. TISA uses an automated tool to convert from the agreed UML model XMI file directly into an MS Word document file, to minimize drafting errors, that forms the annex for each physical format.

TPEG2 has a three-container conceptual structure: message management (ISO 21219-6), application (several parts) and location referencing (ISO/TS 21219-7). This structure has flexible capability and can accommodate many differing use cases that have been proposed within the TTI sector and wider for hierarchical message content.

TPEG2 also has many location referencing options as required by the service provider community, any of which may be delivered by vectoring data included in the location referencing container.

The following classification provides a helpful grouping of the different TPEG2 parts according to their intended purpose. Note that the list below may be incomplete, e.g. new TPEG2 parts may be introduced after publication of this document.

— Toolkit parts: TPEG2-INV (ISO/TS 21219-1), TPEG2-UML (ISO 21219-2), TPEG2-UBCR (ISO 21219-3), TPEG2-UXCR (ISO 21219-4), TPEG2-SFW (ISO 21219-5), TPEG2-MMC (ISO 21219-6), TPEG2-LRC (ISO/TS 21219-7).

— Special applications: TPEG2-SNI (ISO/TS 21219-9), TPEG2-CAI (ISO/TS 21219-10), TPEG2-LTE (ISO/TS 21219-24).

— Location referencing: TPEG2-OLR (ISO/TS 21219-22), TPEG2-GLR (ISO/TS 21219-21), TPEG2-TLR (ISO 17572-2), TPEG2-DLR (ISO 17572-3).

— Applications: TPEG2-PKI (ISO/TS 21219-14), TPEG2-TEC (ISO/TS 21219-15), TPEG2-FPI (ISO/TS 21219-16), TPEG2-TFP (ISO 21219-18), TPEG2-WEA (ISO/TS 21219-19), TPEG2-RMR (ISO/TS 21219-23), TPEG2-EMI (ISO/TS 21219-25), TPEG2-VLI (ISO/TS 21219-26).

TPEG2 has been developed to be broadly (but not totally) backward compatible with TPEG1 to assist in transitions from earlier implementations, while not hindering the TPEG2 innovative approach and being able to support many new features, such as dealing with applications having both long-term, unchanging content and highly dynamic content, such as parking information.

This document is based on the TISA specification technical/editorial version reference: SP10031.

Intelligent transport systems — Traffic and travel information (TTI) via transport protocol experts group, generation 2 (TPEG2) —

Part 2: UML modelling rules (TPEG2-UMR)

1 Scope

This document specifies rules for the creation and extending of TPEG application UML models. The rules are intended to ensure that TPEG application UML models can be interpreted unambiguously for conversion to physical format representations. TPEG application UML models that are defined according to these rules can be used for automatic generation of TPEG standards and for automatic generation of TPEG application physical format descriptions.

This document also specifies the preferred structure of TPEG application specifications.

The TPEG abstract data types and the set of TPEG tables of common use are specified in the annexes.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

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

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

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

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

3.1
abstract data type
data type of atomic nature

3.2
attribute compartment
graphical section of a UML class box positioned directly under the *class name compartment* (3.3)

3.3
class name compartment
top most graphical section of a class box defining the name of the class and optionally a stereotype, inherited class and package scope

3.4
data structure
data type being composed of other data types being either of *abstract* (3.1) or complex data type, not having a *component header* (3.6), stereotyped as <<DataStructure>>

3.5
component
revisable, named, complex data type, not stereotyped as <<DataStructure>>

3.6
component header
data structure (3.4) consisting of a component identifier, component length indicator and attribute length indicator

3.7
element
component (3.5) or *data structure* (3.4)

3.8
link
relation between two or more *elements* (3.7)

3.9
TPEG Application
set of classes and rules defining TPEG information services at the highest layer of the ISO OSI model

4 Abbreviated terms

IPR	Intellectual Property Right(s)
ISO	International Organization for Standardization
LRC	Location Referencing Container
MMC	Message Management Container
OSI	Open Systems Interconnection
PTI	PTI Public Transport Information
TISA	Traveller Information Services Association
TPEG	Transport Protocol Experts Group
UML	Unified Modelling Language

5 TPEG UML model definition

5.1 Allowed UML elements

TPEG UML models are based on the UML standard[1], but only use a subset of the elements defined in the standard. This clause provides a description of the elements of UML that are used for modelling TPEG. This clause also defines restrictions on these elements. TPEG UML models shall only use the UML elements described in this clause. The defined restrictions shall be obeyed.

5.1.1 Class

A class provides a description of the structure of the data stored in an instance of a class. The data are stored in the class attributes.

Classes shall be named according to the following regular expression: “^[a-zA-Z]\w*$” which is equivalent to “class names shall start with an English alphabet character, followed by zero or more English alphabet characters, digits or underscores”. As an exception, classes stereotyped as enumerations shall be named according to the following regular expression: “^[a-z]{3}\d{3}:[azA-Z]\w*$” which is equivalent to “enumeration class names shall start with three lower-case English alphabet characters, followed by three digits, a colon, a English alphabet character, then zero or more English alphabet characters, digits or underscores”. Upper camel case should be used for class names, with the first letter of every word capitalized. Additionally, class names shall be unique within their TPEG application or TPEG toolkit scope.

5.1.2 Abstract class

Abstract classes may be used to define shared properties of specialized child classes.

5.1.3 Attribute

An attribute provides a data type description of data that is stored in a class. Attributes can be either of primitive data type or compound data type.

An attribute shall be named according to the following regular expression: “^[a-zA-Z]\w*$” which is equivalent to “attribute names shall start with an English alphabet character, followed by zero or more English alphabet characters, digits or underscores“. As an exception, attributes within classes stereotyped as enumerations shall be named as defined in 5.2.3.5. Lower camel case should be used for attribute names, with the first letter in lowercase and the first letters of subsequent words in uppercase. Additionally, attributes shall be named uniquely within their class.

Within a class, an attribute has a multiplicity. If not explicitly indicated, the multiplicity is one. Other multiplicities may be indicated between square brackets: [minOccurs .. maxOccurs].

minOccurs and maxOccurs shall be positive integers, where maxOccurs shall be greater or equal to minOccurs. Alternatively, maxOccurs may be theoretically unbounded, in which case an asterisk “*” is modelled. However, in physical formats there may be upper bounds due to length restrictions and size indicator data type limitations.

Attribute multiplicity shall be interpreted as listed in Table 1. If no multiplicity is indicated, a multiplicity of *one* (mandatory attribute) is implied.

Table 1 — TPEG multiplicity

Multiplicity	TPEG meaning
minOccurs=1 and maxOccurs=1	mandatory attribute
minOccurs=1 and maxOccurs>1	mandatory list of attributes
minOccurs=0 and maxOccurs=1	optional attribute
minOccurs=0 and maxOccurs>1	optional list of attributes
minOccurs>1	mandatory list of attributes with a minimum occurrence

Attributes in classes are always modelled as public. Each attribute must have a data type. Attributes occur in the order as listed in the class definition in TPEG physical formats, unless this is overruled by the stereotype <<UnorderedComponentGroup>> .

NOTE Optional list of attributes with a minimum occurrence can be modelled as optional attribute including a data structure including a mandatory list of attributes with minOccurs>1.

5.1.4 Dependency

Figure 1 is the graphical representation used for ordered components (attributes stereotyped as <<OrderedComponentGroup>>) and DataStructures to show the hierarchical structure of the UML model.

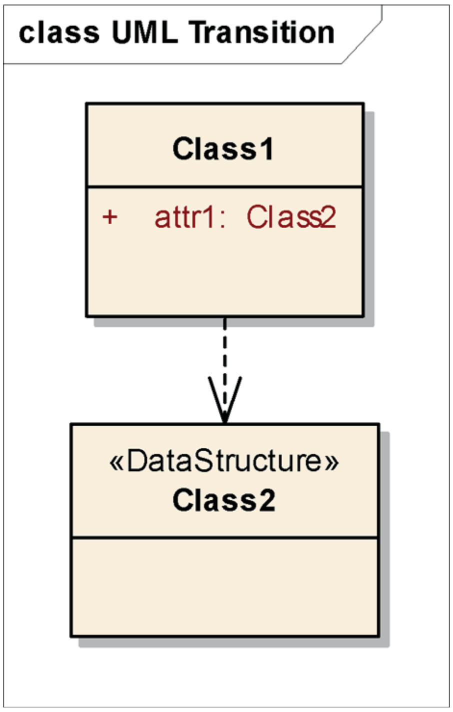

Figure 1 — UML dependency relation

NOTE Aggregation and composition associations are deprecated in TPEG models; the preferred method is inclusion as an attribute.

5.1.5 Specialization

A specialization (see Figure 2) relates a parent class to a child class. The child class inherits properties from the parent class. Classes shall not inherit from multiple parent classes. Classes shall only inherit from classes with the same stereotype.

Derived classes copy all attributes from the parent class. Parent classes shall contain classes not stereotyped as <<DataStructure>> or <<BinaryDataContainer>>. Parent classes shall be modelled as abstract class. In future versions of a standard, parent classes shall not be extended. Classes shall not be both parent and child class.

NOTE Extending parent classes in future versions of a standard breaks backwards compatibility.

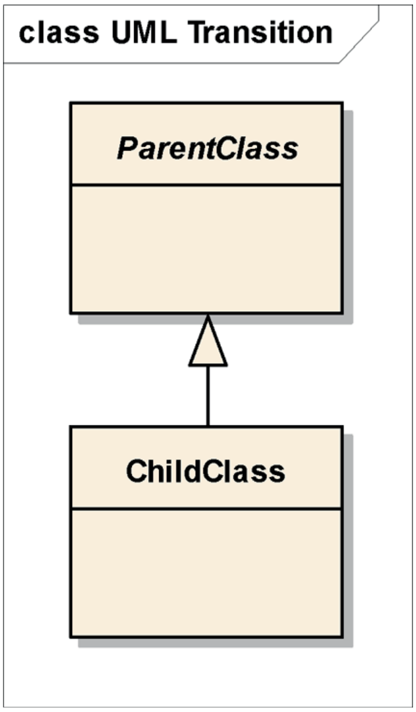

Figure 2 — UML specialization relation

5.1.6 Stereotype

A stereotype is used to provide an additional classification of UML properties. A physical format specification may use stereotype information to select a rule set for converting UML to the physical format.

The stereotypes as listed in Table 2 may be used for UML modelling of TPEG applications. Other stereotypes shall not be used.

Table 2 — TPEG stereotypes

UML element	Stereotype	TPEG meaning
Package	TPEG Application	Self standing protocol specification for a given application.
Package	TPEG Toolkit	Specification of general interest being referenced by different other specifications.
Package	TPEG DataTypes	Specification defining data structures and tables belonging to one single package.
Class	DataStructure	TPEG data structure.
Class	Enumeration	List of defined, constant expressions not containing attributes or sub data elements.
Class	External	TPEG Component defined in an external document.
Class	BinaryDataContainer	Component carrying binary data in its body that is not further specified within TPEG.
Attribute	OrderedComponentGroup	Attribute is of component type, and belongs to the group of components occurring in the order as defined by the attribute order.
Attribute	UnorderedComponentGroup	Attribute is of component type, and belongs to the group of components that may occur in random order, after all other attributes.

5.1.7 Tagged values

Tagged values may be used to provide additional information on a UML element, used for the creation of the specification document.

Only the tagged values listed in Table 3 shall be used.

Table 3 — Allowed tagged values

Tag	UML Element	TPEG meaning	Example
ApplicationAbbreviation	Package	Abbreviation of the application name as string.	TEC
ApplicationName	Package	Name of the application as string.	Traffic Event Compact
ApplicationNo	Package	Part number to be included in the target specification title.	A value of „2“ will result in: Intelligent transport systems — Traffic and Travel Information (TTI) via Transport Protocol Expert Group, Generation 2 (TPEG2) — Part 2: UML Modeling Rules (TPEG2-UMR_1.1/001)
ApplicationRoot	Package	Name of the root class of an application as string.	TECMessage
ApplicationVersion	Package	Version of the specification that is generated from this model. Major.Minor VersionNumber.	3.1
ApplicationRelease	Package	Integer that is increased each time the model is changed	Integer/345
TableEntryExample	Enumeration attribute	Comment for a table entry	
Documentation		Description of generic properties of a class	
Description		Description of single attributes within a class	
GCId	Class	Positive integer value in the range of [0..255] that uniquely identifies a class in the binary physical format.	1
ModelingRulesVersion	Package	Technical version number (MajorMinorVersion) of the UML Modelling Rules specification to which this model complies.	1.2
tpegMLprefix	Class	For classes stereotypes as <<External>> an namespace prefix may be given here.	Tec
tpegMLuri	Class	For classes stereotypes as <<External>> an namespace URI may be given here.	http://www.tisa.org/tpeg/...

In UML packages that are stereotyped as <<TPEG Application>> , the ApplicationAbbreviation, ApplicationName, ApplicationRoot ApplicationNo and ModelingRulesVersion tagged values are mandatory.

5.1.8 Notes

Notes may be used to provide additional information that is used for generating the specification document.

5.2 Modelling rules and recommendations

5.2.1 General

TPEG UML models are used to generate TPEG specifications. A fundamental assumption is that applications will develop and new features will be added. Correct designs permit applications to be upgraded and extended over time, providing new features to new decoders, and yet permit existing decoders to continue to operate. This clause describes design principles that shall be obeyed when building and upgrading TPEG applications.

5.2.2 Order of elements

In a physical format, attributes shall occur in the same order as listed in the UML class definition (see Figure 3).

When components may occur in any order (independent of the order in which they are listed in the UML class definition), they should be modelled as attributes with the stereotype <<UnorderedComponentGroup>> and of the type of the corresponding class. The unordered components shall be linked by the embedding class using a dependency relation.

When components shall occur in a specific order, they shall be modelled as attributes with the stereotype <<OrderedComponentGroup>> and of the type of the corresponding class. The ordered components shall be linked by the embedding class using a dependency relation.

Mandatory attributes should occur *before* optional attributes. Mandatory Booleans should occur *after* the other mandatory attributes. Optional attributes should occur *after* mandatory attributes. Components shall occur *after* all other attributes. Ordered components shall occur *before* unordered components.

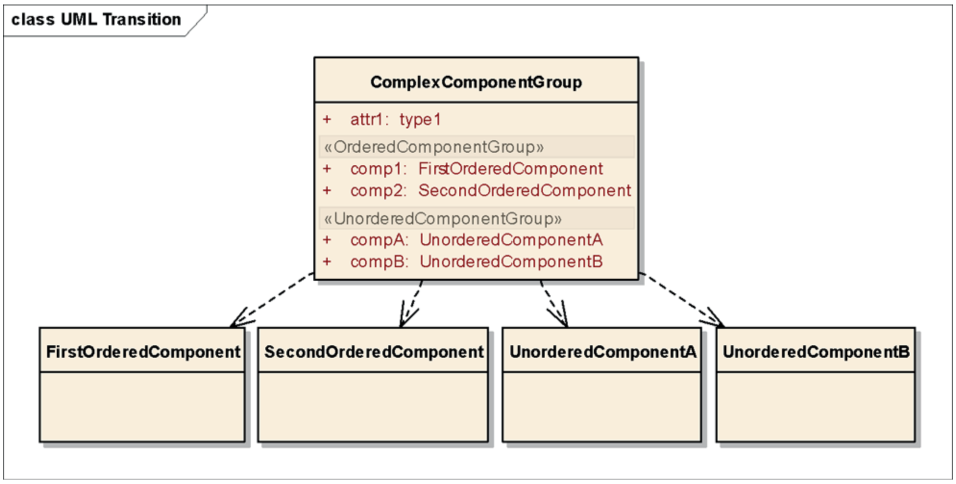

Figure 3 — Ordering of class elements

NOTE Special rules for extending TPEG UML models for newer revisions of a standard are provided in 5.3. Extending TPEG UML models in a backwards compatible way can break the recommendations for ordering mandatory and optional elements described in 5.2.2.

5.2.3 Stereotypes

5.2.3.1 TPEG Application

The <<TPEG Application>> stereotype is used to identify a UML package as TPEG Application.

NOTE UML packages that are stereotyped as <<TPEG Application>> can as well be stereotyped as <<TPEG Toolkit>> if the use of classes is encouraged in other applications.

5.2.3.2 TPEG Toolkit

TPEG Toolkits are used to share common functionality between different TPEG Applications. For example the Location Referencing Container and Message Management Container are toolkits that are used by all TPEG applications. A TPEG Application therefore can refer to a data type definition not specified in the same model.

TPEG Toolkits are designed such that its root components are defined as templates which can be used as external reference within other packages. A TPEG Application using a toolkit template therefore needs to specify a unique interface class for this instantiation of the imported toolkit interface’s component.

All subsequent components in a toolkit are defined as out of the scope of the TPEG application, i.e. the toolkit on its own defines subcomponents beginning with local identifiers.

The <<TPEG Toolkit>> stereotype is applied at UML package level.

5.2.3.3 TPEG DataTypes

General TPEG datatypes and TPEG Application specific datatypes as well as classes stereotyped as <<enumeration>> are defined in separate UML packages. This only applies for elementary data types and classes that are stereotyped as <<DataStructure>>.

The <<TPEG DataTypes>> stereotype is applied at UML package level.

5.2.3.4 DataStructure

The TPEG binary format distinguishes between *components* and *datastructures*. In this physical representation, a component is a compound data type, containing a header, providing type and length information. The type information of a component shall be unique within an application. A datastructure is a compound datatype not containing this type and length information.

Datastructures shall explicitly be stereotyped as <<DataStructure>> . Classes that are not explicitly stereotyped as <<DataStructure>> shall be interpreted as component. The differentiation between components and datastructures is not relevant for tpegML as both variants are represented as xs: element.

Components shall be included in datastructures by using attributes typed as this component.

The <<DataStructure>> stereotype is applied at UML class level.

The specific usage of datastructures or components depends on the requirements of the particular application. Components should be used wherever future extensions are envisioned, and where ‘future proofing’ is a strong requirement. Datastructures are more bandwidth efficient as they contain no header information but are not extendible in a backwards-compatible manner.

5.2.3.5 Enumeration

TPEG Tables are modelled as enumerations and shall have no more than 256 entries. Each enumeration shall have a unique name which consists of two parts. The first part is the abbreviation of the application where the enumeration is specified in, appended with a three-digit decimal number, starting at value

001. The second part is a description of the content of the table in camel-case. The two parts are separated by a colon.

EXAMPLE typ001:LanguageCode.

When applicable, the enumeration entry with index zero should have the value “unknown”. If tables are likely to be extended in future versions of a standard, a default value shall be defined. This default value shall have index 255.

Additionally, enumeration entry names shall be named according to the following regular expression: “^(25[0- 5]|2[0-4][0-9]|[01]?[0-9][0-9]?):.+$”, which is equivalent to “enumeration entry names shall start with a numerical value between 0 and 255, followed by a colon and one or more arbitrary characters”, where the text following the colon should only contain printable characters and whitespaces.

EXAMPLE See Figure 4.

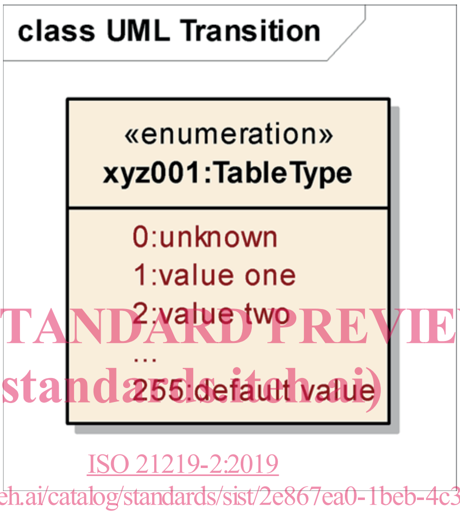

Figure 4 — Enumeration stereotype

The <<Enumeration>> stereotype is applied at UML class level.

5.2.3.6 External

TPEG applications can use classes that are not defined in the same package, but are defined elsewhere. These classes shall be stereotyped as <<External>> .

The <<External>> stereotype is applied at UML class level.

Classes stereotyped as <<External>> shall not contain or inherit any attributes, but define the container formats only, while the content is defined in the referenced class/specification.

There are three different levels of external references:

1) Classes defined in other TPEG applications or TPEG toolkits.

   In this case, the class shall be derived from the class in the other package. Additionally, the tpegMLprefix and tpegMLuri shall be used.

   In the case where a class stereotyped as <<External>> has already been derived from another class, this second external generalisation is valid. Classes imported via this mechanism shall not be abstract.