INTERNATIONAL STANDARD

ISO 17572-2

Third edition
2018-09

Intelligent transport systems (ITS) — Location referencing for geographic databases —

Part 2: Pre-coded location references (pre-coded profile)

Systèmes intelligents de transport (SIT) — Localisation pour bases de données géographiques —

Partie 2: Localisations précodées (profil précodé)

Reference number
ISO 17572-2:2018(E)

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

Published in Switzerland

Contents

Page

Foreword

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

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

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

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

For an explanation on 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 the following URL: www.iso.org/iso/foreword.html.

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

This third edition cancels and replaces the second edition (ISO 17572-2:2015), which has been technically revised.

The main changes compared to the previous edition are as follows:

— added the description of Extended TMC Location reference (Annex E, F & G).

A list of all parts in the ISO 17572 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

A Location Reference (LR) is a unique identification of a geographic object. In a digital world, a real-world geographic object can be represented by a feature in a geographic database. An example of a commonly known LR is a postal address of a house. Examples of object instances include a particular exit ramp on a particular motorway, a road junction or a hotel. For efficiency reasons, LRs are often coded. This is especially significant if the LR is used to define the location for information about various objects between different systems. For Intelligent Transport Systems (ITS), many different types of real-world objects are addressed. Amongst these, Location Referencing of the road network, or components thereof, is a particular focus.

Communication of an LR for specific geographic phenomena, corresponding to objects in geographic databases, in a standard and unambiguous manner is a vital part of an integrated ITS system in which different applications and sources of geographic data are used. Location Referencing Methods (LRM), methods of referencing object instances, differ by applications, by the data model used to create the database, or by the enforced object referencing imposed by the specific mapping system used to create and store the database. A standard LRM allows for a common and unambiguous identification of object instances representing the same geographic phenomena in different geographic databases produced by different vendors, for varied applications, and operating on multiple hardware/software platforms. If ITS applications using digital map databases are to become widespread, data reference across various applications and systems must be possible. Information prepared on one system, such as traffic messages, must be interpretable by all receiving systems. A standard method to refer to specific object instances is essential to achieving such objectives.

Japan, Korea, Australia, Canada, the US and European ITS bodies are all supporting activities of Location Referencing. Japan has developed a Link Specification for VICS. Japan has developed the Road Section Identification Data set (RSIDs) which uses road sections and reference points. In Europe, the RDS-TMC traffic messaging system has been developed. In addition, methods have been developed and refined in the EVIDENCE and AGORA projects based on intersections identified by geographic coordinates and other intersection descriptors. In the US, standards for Location Referencing have been developed to accommodate several different LRMs.

This document provides specifications for location referencing for ITS systems (although other committees or standardization bodies may subsequently consider extending it to a more generic context). This document is consistent with other International Standards such as ISO 14825.

In addition, this edition of this document does not deal with public transport location referencing; this issue will be dealt with in a later edition.

Intelligent transport systems (ITS) — Location referencing for geographic databases —

Part 2:
Pre-coded location references (pre-coded profile)

1 Scope

The ISO 17572 series specifies LRMs that describe locations in the context of geographic databases and are used to locate transport-related phenomena in an encoder system as well as in the decoder side. The ISO 17572 series defines what is meant by such objects and describes the reference in detail, including whether or not components of the reference are mandatory or optional, and their characteristics.

The ISO 17572 series specifies two different LRMs:

— pre-coded location references (pre-coded profile);

— dynamic location references (dynamic profile).

The ISO 17572 series does not define a physical format for implementing the LRM. However, the requirements for physical formats are defined.

This document specifies the pre-coded LRM, comprising:

— specification of pre-coded location references (pre-coded profile);

— logical format for VICS link location (Annex A);

— TPEG physical format for ALERT-C (TMC) location references (Annex B, C & D);

— TPEG physical format for ETTLs (Annex E, F & G);

— TPEG physical format for Korean node-link ID references (Annex H, I & J).

— logical format for Road Section Identification Data set (Annex K).

2 Normative references

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

ISO 17572-1, *Intelligent transport systems (ITS) — Location referencing for geographic databases — Part 1: General requirements and conceptual model*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 17572-1 and the following apply.

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

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

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

3.1
major link
directed link in a road network

4 Abbreviated terms

ALERT	Advice and Problem Location for European Road Traffic
DATEX	DATa EXchange (protocol for exchange of traffic and travel information between traffic centres)
ETL	Extended TMC Location reference
GCId	Generic Component Identifier
GDF	Geographic Data File
ID	Identifier
ITRF	International Terrestrial Reference Frame
LDB	Location DataBase
LI	Location Information
LR	Location Referencing (or Reference)
LRM	Location Referencing Method
LRS	Location Referencing System
LRP	Location Referencing Procedure
MOCT	Ministry of Construction and Transportation (Republic of Korea)
RDS	Radio Data System
RSIDs	Road Section Identification Data set
SOEI	System Operating and Exchanging Information
TMC	Traffic Message Channel
TPEG	Transport Protocol Expert Group
TLR	TMC Location Reference
TTI	Traffic and Traveller Information
UTM	Universal Transverse Mercator
VICS	Vehicle Information and Communication System

5 Requirements for a location referencing standard

For details, see ISO 17572-1:2008, Clause 4.

For an inventory of LRMs, see ISO 17572-1:2008, Annex A.

6 Conceptual data model for LRMs

For details, see ISO 17572-1:2008, Clause 5.

For examples of conceptual data model use, see ISO 17572-1:2008, Annex B.

7 Specification of pre-coded location references

7.1 General concept

Pre-coded location referencing is a method which makes use of end-user client devices carrying an LDB that is exactly the same as the corresponding LDB used by a service provider of a particular message being exchanged. All pre-coded LRMs shall share the concept of defining a commonly used database of IDs. This concept has been developed in the past for technologies such as RDS-TMC and VICS to allow an (over-the-air) interface to be designed that uses compact code values (IDs) in the corresponding databases to express particular pre-coded locations of various types.

The LRM here is divided into three steps performed to implement the LRS. The first step is a process of defining the database of location IDs for a given area and the corresponding road network. In this step different service providers and systems provider agree on a defined database containing all locations to be codable (LDB creation). In the second step, this database is provisioned via various means into the service providers database as well as into all receiving systems (LDB provisioning). The third step is in real-time where a service provider can now make use of that database and reference to locations by using the newly-introduced IDs (LDB usage). Figure 1 illustrates this concept.

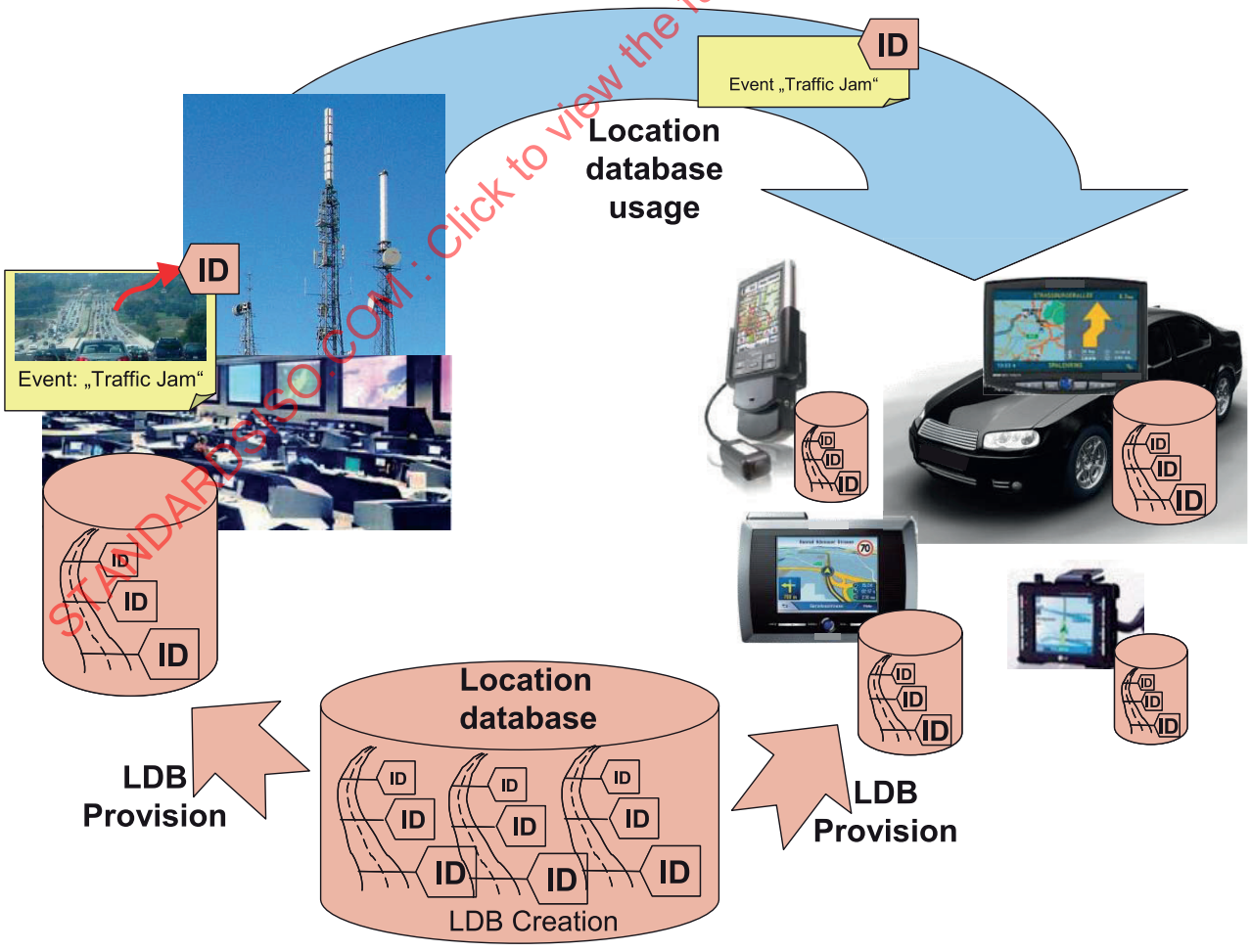

Figure 1 — General concept of pre-coded LRS

7.2 LDB creation and updating

The different LRSs more or less support standardized ways to create a new release of a LDB. All of them share a conceptual model specifying how the different location categories specified in ISO 17572-1 are related to each other. This specification together with some guiding literature helps the community to create new releases of the LDB.

7.3 LDB provision

After the finalization of the creation process, the newly-created LDB is provisioned into the devices with maintenance service agreements. This is mostly done on a regular map release update. The LRS has to ensure that the encoding and the decoding entities are able to distinguish which release (version) of the database is in use, because no conclusion regarding the correctness of the location can be made based on the contents of the IDs alone.

7.4 LDB usage

A service provider, using the current release dataset, now creates messages with location references according to specified rules of a location reference method out of the list of location IDs available and may put additional attributes to it, to define more precisely which part of the road network is referred to. The location reference sent to the receiving system then consists of a list of one or more location IDs and some additional attributes. Presuming that the receiving system has the actual database available it seeks for the given location IDs and applies the additional attributes according to the location referencing specification. Doing so, the decoder provides the same location definition as requested by the service provider.

8 Implementations at present

8.1 General

Different implementations of pre-coded location referencing have been already specified for a while. Some of them are captured in another ISO standard and some of them need some more specification here. This clause provides a list of presently known pre-coded LRMs and introduces them shortly. It also refers to the different documents needed to fully apply the different implementations.

8.2 VICS

8.2.1 LDB creation

VICS specifies in Reference [2] a digital map database as the basis for other map providers to adopt the different map IDs into their own digital map. The digital base map consists of nodes and road elements which build up a complete street map on level zero. Figure 2 defines the conceptual data model for this map.

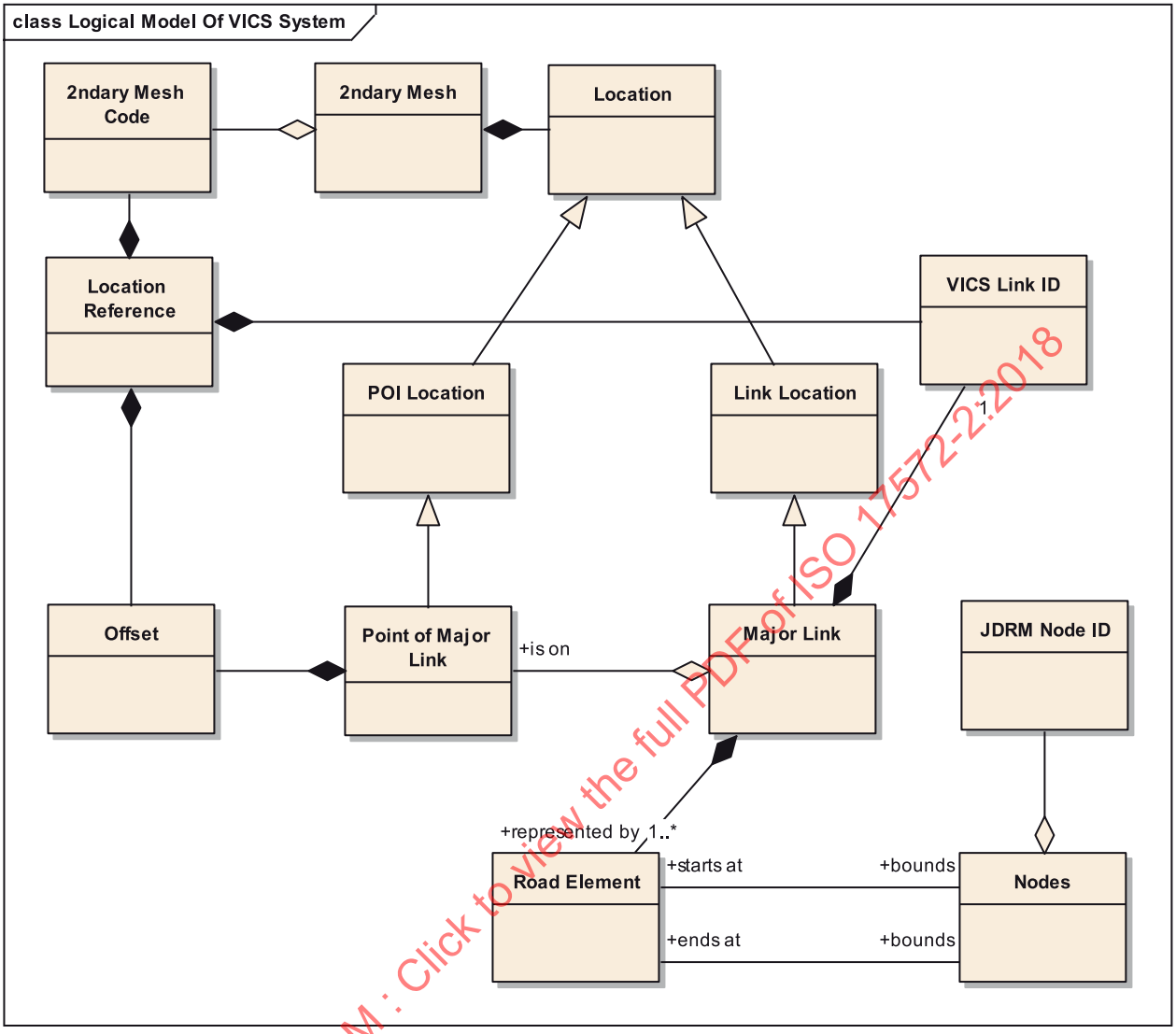

Figure 2 — Logical data model of VICS digital base map

8.2.2 LDB usage

All or any part of the specified digital map database can be referred to by a location reference consisting of VICS-Link-IDs, 2ndary-Mesh-Codes and offsets. The specification in Reference [1] defines how the digitized location IDs shall be coded to build up a more sophisticated location.

8.3 TMC/ALERT-C Specification

8.3.1 General

The location referencing rules defined in ISO 14819-3:2013[12] address the specific requirements of TMC systems, which use abbreviated coding formats to provide TTI messages over mobile bearers (e.g. GSM, DAB) or via exchange protocols like DATEX II. In particular, the rules address the RDS-TMC, a means of providing digitally-coded traffic and travel information to travellers using a silent data channel (RDS) on FM radio stations, based on the ALERT-C protocol[9].

8.3.2 LDB creation

Location types and subtypes are required for language independence of the information given, and to tell the receiving system what data fields to expect.

At the highest level, locations fall into three categories:

1. area locations
2. linear locations
3. point locations

RDS-TMC location tables use a hierarchical structure of pre-defined locations. Locations are identified using a location ID. A system of pointers provides upward references to higher-level locations of which the specified location forms a part. As such, all point locations belong to linear locations and they refer to area locations. Point locations additionally refer to a succeeding and a preceding point location which builds up a connected network of point locations. Further information can be found in a coding hand book that has been written by the TMC forum[4].

8.3.3 LDB usage

A location ID in such a message refers and serves as a tabular "address" of the pre-stored location details in the LDB used by the service. A real world location may have more than one point location within the same location table, which can be expressed by one point location code and an additional attribute extent which counts the steps of succeeding point location to be added to the location. Another additional attribute direction allows to extend from a point location into positive or into negative direction according to the point location direction defined in the LDB.

8.4 Korean node link ID system

8.4.1 General

The MOCT of Korea has developed a standard Node-Link System for ITS in 2004 for effective exchange of real-time traffic information. The Node and Link ID is made up of 10 digits. Korean standard Node-Link ID is the standard location ID for TPEG-Loc services in Korea[3].

8.4.2 LDB creation

In principle, road authorities create and manage standard Node-Link IDs and digital base map for those standard Node/Link according to Reference [6] which was published by MOCT. MOCT verifies the IDs and digital base map, and then officially distributes them.

8.4.3 LDB usage

Any Node or Link ID can be served as location ID in LRS, but only Link ID is used in currently implemented systems.

8.5 RSIDs

8.5.1 General

The RSIDs was developed to enable exchanging various static/dynamic information on road network.

The location of the information is represented by an appropriate road section with a reference point and a distance from the reference point.

This document provides a profile of road section identification and reference point identification required in the RSIDs. Definition of each link and node corresponding to respective road maps is out of scope.

As permanent ID set is specified in the system, RSIDs is independent from avoidable change caused by road map revisions.

8.5.2 LDB creation

RSIDs creates an authority table for section IDs and reference point IDs.

8.5.3 LDB usage

RSIDs is expected to use for exchanging various LI-related road between different players. For example, road authorities use this method to provide road information to private sectors.

Annex A
(informative)

Logical format for VICS link location

A.1 Description of the logical structure

A.1.1 General

The subsequent clauses define data elements used for building up the VICS Link location reference (database usage). Different descriptions of the datastructure help to understand the concept. It consists of an LI Header and Location Content as shown in Figure A.1, with the latter further subdivided functionally into Coordinates, Descriptors and Offset information. Figure A.2 describes the structure of the LI main in the form of a UML Diagram. A.3 and A.4 define different views on a logical format.

All or any part of the LI may be omitted optionally if it is possible to refer to a location between databases without all or any part of LI content by defining unambiguous rules for a physical format and by establishing a management system.

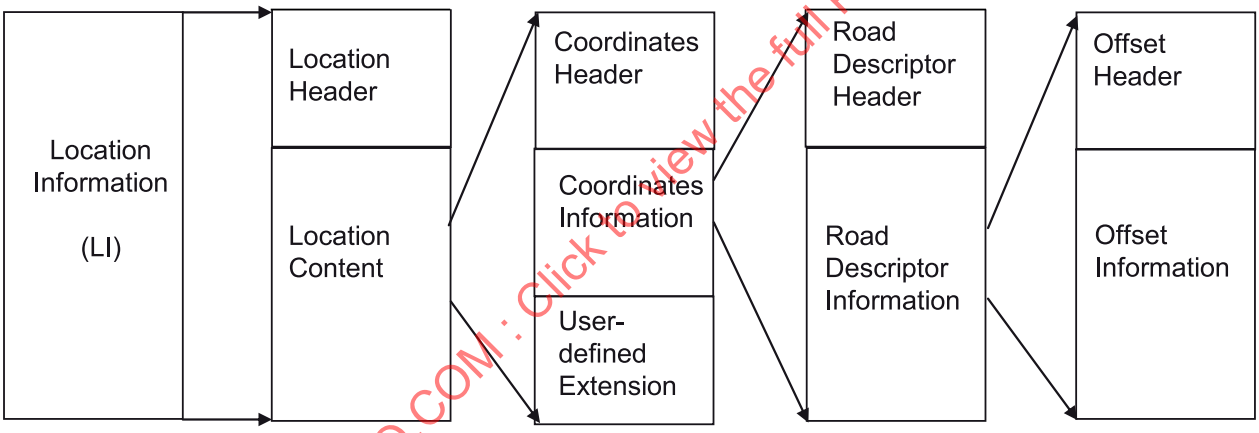

Figure A.1 — Outline diagram of the logical structure

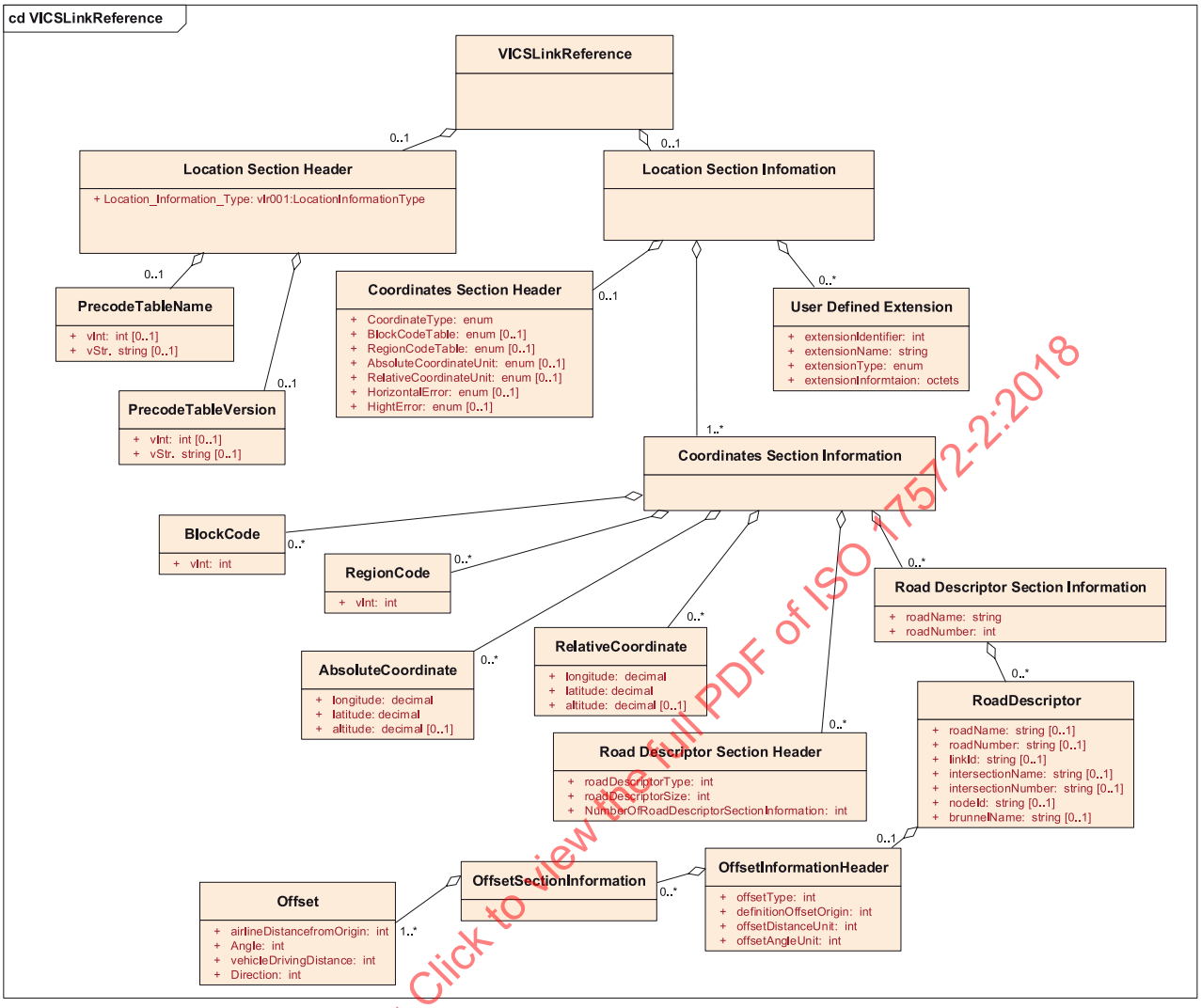

Figure A.2 — Outline diagram of the logical structure in UML

A.1.2 Data values

Table A.1 lists all the specific values of enumerations used in the Annex A location reference format.

Table A.1 — Enumerations used in the Annex A location reference format

Data value name	Definition
basemap1	A parameter specifying that the location is digitized on a map of 1/2 500~1/10 000 scale
basemap2	A parameter specifying that the location is digitized on a map of 1/25 000~1/50 000 scale
basemap3	A parameter specifying that the location is digitized on a map of more than 1/100 000 scale
ddmmss	A parameter specifying a coordinate is expressed using decimal integer value of degree, minute, and second
degree	A parameter specifying that a unit of coordinates is degree
error1	A parameter specifying that a height error is less than 1 m
error2	A parameter specifying that a height error is less than 10 m

Table A.1 *(continued)*

Data value name	Definition
extensiontype1|2|...|n	The type of user-defined extension of which n different enumerated values are specified in the User-Defined Extension Header data frame
absolute	A parameter specifying that a coordinate system is absolute
relative	A parameter specifying that a coordinate system is relative
grid	A parameter specifying that the coordinate has a grid code
Relative_X	The horizontal X value of relative coordinates of a point
Relative_Y	The horizontal Y value of relative coordinates of a point
Relative_Z	The value of height in relative coordinates of a point
pgrid1	A parameter specifying that a coordinate has a private grid
rct1	A parameter specifying that region code table 1 is used
rct2	A parameter specifying that region code table 2 is used
rectc	A parameter specifying that a coordinate system is rectangular
second	A parameter specifying that a unit of absolute coordinates is second
secondu1	A parameter specifying that a unit of relative coordinates is normalized
secondu2	A parameter specifying that a unit of relative coordinates is some value
utmp	A parameter specifying that the blocking is the UTM primary mesh dividing
utms	A parameter specifying that the blocking is the UTM secondary mesh dividing
x	The value of horizontal axis on relative coordinates
y	The value of vertical axis on relative coordinates
z	The value of height in relative coordinates

A.1.3 Data elements

In Table A.2, the maximum value (labelled with MAX) in the column "Valid value rule" is specified first when defining unambiguous rules for a physical format by establishing the system implementation.

Table A.2 — Data elements used in the Annex A location reference format

Data element name	Definition	Data type	Valid value rule
Absolute_Coordinate_Unit	A unit for an Absolute Coordinate	BIT STRING	ENUMERATED { degree, ddmmss, second}
Airline_Distance_from_Origin	The shortest distance from an origin to a point	Distance_type	CHOICE {INTEGER, REAL}
Altitude	The geographic altitude of a node	Altitude_type	CHOICE {INTEGER, REAL}
Angle	An Integer angle from a starting point to a feature, in units defined by Offset_Angle_Unit	Angle_type	CHOICE {INTEGER, REAL}
Block_Code	A code given to an area such as one of UTM's meshes or some rectangles	INTEGER	SIZE (0..MAX)
Block_Code_Table	A table of Block Codes	BIT STRING	ENUMERATED { utms, utmp, rectc, pgrid1}
Brunnel_Name	Text name of a Brunnel	UTF8String	SIZE (0..255)
Coordinate_Type	A type of the coordinate such as the absolute, the relative and the composite	BIT STRING	ENUMERATED { absolute, relative, grid}
Definition_Offset_Origin	A parameter specifying whether an origin of offset is starting node or end node	BIT STRING	ENUMERATED { Start point, End point}
Direction	A parameter specifying the direction from a starting point to a feature, in units defined by Offset_Distance_Unit either as integer value or as real value	Direction_type	CHOICE {INTEGER, REAL}

Table A.2 *(continued)*

Data element name	Definition	Data type	Valid value rule
Extension_Identifier	Identifier of a user defined data element	Extension_Id_type	CHOICE {INTEGER, UTF8String}
Extension_Name	Name of a user defined data element	UTF8String	SIZE (0..255)
Extension_Type	Type of a user defined data element	BIT STRING	ENUMERATED { extensiontype1, extensiontype2, ... extensiontypen}
Height_Error	The height (vertical) error specifying an altitude accuracy of coordinates	BIT STRING	ENUMERATED { error1, error2}
Horizontal_Error	The horizontal error specifying a horizontal accuracy of coordinates	BIT STRING	ENUMERATED { basemap1, basemap2, basemap3}
Intersection_Name	Text name of an intersection	UTF8String	SIZE (0..255)
Intersection_Number	Integer number of an intersection	INTEGER	SIZE (0..MAX)
Latitude	ITRF geographic latitude of a node	Latitude_type	CHOICE {INTEGER, REAL}
Link_ID	Link identifier	INTEGER	SIZE (0..MAX)
Location_Information_Type	A parameter specifying whether the Location Information used for the location referencing is information of a point, lines, or area	BIT STRING	ENUMERATED{ point, line, face}
Longitude	ITRF geographic longitude of a node	Longitude_type	CHOICE {INTEGER, REAL}
Node_ID	Node Identifier	INTEGER	SIZE (0..MAX)

Table A.2 *(continued)*

Data element name	Definition	Data type	Valid value rule
Number_of_Absolute_Coordinates	The number of absolute coordinates in a Coordinate_Information Data Frame	INTEGER	SIZE (0..MAX)
Number_of_Coordinates_Information	The number of Coordinate_Information Data Frames	INTEGER	SIZE (0..MAX)
Number_of_Blocks	The number of Blocks in a Coordinate_Information Data Frame	INTEGER	SIZE (0..MAX)
Number_of_Offsets	The number of Offsets in an Offset_Information_Data Frame	INTEGER	SIZE (0..MAX)
Number_of_Offset_Information	The number of Offset_Information Data Frames	INTEGER	SIZE (0..MAX)
Number_of_Regions	The number of Regions in a Coordinate_Information Data Frame	INTEGER	SIZE (0..MAX)
Number_of_Relative_Coordinates	The number of relative coordinates in a Coordinate_Information Data Frame	INTEGER	SIZE (0..MAX)
Number_of_Road_Descriptors	The number of Road_Descriptors in a Road_Descriptor_Information Data Frame	INTEGER	SIZE (0..MAX)
Number_of_Road_Descriptor_Information	The number of Road_Descriptor_Information Data Frames	INTEGER	SIZE (0..MAX)
Offset_Angle_Unit	The unit of an angle defining the direction from a starting point to the feature such as a degree or a radian	BIT STRING	ENUMERATED{ degree, radian}
Offset_Distance_Unit	The unit defining a distance from a starting point to the feature such as 1 m or 10 m	BIT STRING	ENUMERATED{ 1m, 10m, 100m}
Offset_Type	A type of the offset, on-road or airline	BIT STRING	ENUMERATED{ on-route, airline}
Precoded_Table_Name	The text identifier of a pre-coded table containing location information	UTF8String	SIZE (0..255)
Precoded_Table_Version	Version of the pre-coded table	Precoded_Table_Version_type	CHOICE {INTEGER, UTF8String}
Region_Code	A code of a region such as a country, city or state	INTEGER	SIZE (0..MAX)

Table A.2 *(continued)*

Data element name	Definition	Data type	Valid value rule
Region_Code_Table	A table of Region_Codes	BIT STRING	ENUMERATED { rct1, rct2}
Relative_Coordinate_Unit	A unit for a Relative Coordinate	BIT STRING	ENUMERATED { secondu1, secondu2}
Relative_X	The value of horizontal axis on relative coordinates	INTEGER	SIZE (0..MAX)
Relative_Y	The value of vertical axis on relative coordinates	INTEGER	SIZE (0..MAX)
Relative_Z	The value of height in relative coordinates	INTEGER	SIZE (0..MAX)
Road_Descriptor_Size	A number of characters of a Road Descriptor	INTEGER	SIZE (0..MAX)
Road_Descriptor_Type	Classification of a Road Descriptor such as express way, national road and others	INTEGER	SIZE (0..MAX)
Road_Name	Text name of a road	UTF8String	SIZE (0..255)
Road_Number	Integer number of a road	INTEGER	SIZE (0..MAX)
Vehicle_Driving_Distance	The distance a vehicle drives to an offset point	Distance_type	CHOICE {INTEGER, REAL}

A.1.4 Data frames

See Table A.3.

Table A.3 — Used in the Annex A location reference format

Data Frame name	Definition	Data elements/nested frames
Location_Information	A Data Frame containing a Location_Section_Header and Location_Section_Information	Location_Section_Header
		Location_Section_Information
Location_Section_Header	A Data Frame containing header information for location information	Precoded_Table_Name
		Precoded_Table_Version
		Location_Information_Type
Location_Section_Information	A Data Frame containing coordinate, road descriptor, and offset information and headers	Coordinates_Section_Header
		Coordinates_Section_Information[a]
		User_Defined_Extension
Coordinates_Section_Header	A Data Frame containing header information for coordinate information	Coordinate_Type
		Block_Code_Table
		Region_Code_Table
		Absolute_Coordinate_Unit
		Relative_Coordinate_Unit
		Horizontal_Error
		Height_Error
		Number_of_Coordinates_Section_Information
Coordinates_Section_Information[a]	A Data Frame containing block, region, absolute, and relative coordinate information	Number_of_Blocks
		Number_of_Regions
		Number_of_Absolute_Coordinates
		Number_of_Relative_Coordinates
		Road_Descriptor_Section_Header
		Road_Descriptor_Section_Information[a]
Blocks[a]	A Data Frame containing Block_Codes	Block_Code
Regions[a]	A Data Frame containing Region_Codes	Region_Code
Absolute_Coordinates[a]	A Data Frame of ITRF geographic coordinates	Latitude
		Longitude
		Altitude

[a] A Data Frame is repeatable; Data Frame 'X' occurs 'Number_of_X' times.

Table A.3 *(continued)*

Data Frame name	Definition	Data elements/nested frames
Relative_Coordinates[a]	A Data Frame containing relative x, y, and z coordinate information	Relative_X
		Relative_Y
		Relative_Z
Road_Descriptor_Section_Header	A Data Frame containing road descriptors header information	Road_Descriptor_Type
		Road_Descriptor_Size
		Number_of_Road_Descriptor Section_Information
Road_Descriptor_Section_Information[a]	A Data Frame containing road descriptor section information	Number_of_Road_Descriptors
		Road_Descriptor[a]
Road_Descriptor[a]	A Data Frame containing road descriptors	Road_Name
		Road_Number
		Link_ID
		Intersection_Name
		Intersection_Number
		Node_ID
		Brunnel_Name
		Offset_Section_Header
		Offset_Section_Information[a]
Offset_Section_Header	A Data Frame containing header information for offsets	Offset_Type
		Definition_Offset_Origin
		Offset_Distance_Unit
		Offset_Angle_Unit
		Number_of_Offset_Section_Information
Offset_Section_Information[a]	A Data Frame containing offsets	Number_of_Offsets
		Offsets[a]
Offsets[a]	A Data Frame containing offset information on a distance from a base point to the feature	Airline_Distance_from_Origin
		Angle
		Vehicle_Driving_Distance
		Direction

[a] A Data Frame is repeatable; Data Frame 'X' occurs 'Number_of_X' times.

Table A.3 *(continued)*

Data Frame name	Definition	Data elements/nested frames
User_Defined_Extension	A Data Frame containing user defined extension information	User_Defined_Extension_Header
		User_Defined_Extension_Information
User_Defined_Extension_Header	A Data Frame containing header information for user defined information	Extension_Identifier
		Extension_Name
		Extension_Type
User_Defined_Extension_Information	A Data Frame containing data elements and/or nested data frames are user-defined and not specified	<user defined>

[a] A Data Frame is repeatable; Data Frame 'X' occurs 'Number_of_X' times.

A.2 Detailed diagram of logical structure

Lrp1-linf : Location information of Location referencing procedure 1

- **lsheader**: Location Section Header
  - **ptname** : Precoded Table Name
  - **ptversion** : Precoded Table Version
  - **linft** : Location Information Type
- **lsinf** : Location Section Information
  - **cshi** : Coordinates Section Header
    - **ct** : Coordinate Type
    - **bct** : Block Code Table
    - **rct**: Region Code Table
    - **acu** : Absolute Coordinate Unit
    - **rcu** : Relative Coordinate Unit
    - **horerr** : Horizontal Error
    - **hterr** : Height Error
    - **numcsi** : Number of Coordinates Section Information
  - **csi** : Coordinates Section Information (numcsi occurrences)
    - **numblocks** : Number of Blocks
    - **bcodes** : Block Code (numblocks occurrences)
    - **numregion** : Number of Regions
    - **rcodes** : Region Code (numregions occurrences)
    - **numac** : Number of Absolute Coordinates
    - **ac** : Absolute Coordinate (numac occurrences)
      - **lat** : Latitude
      - **lon** : Longitude
      - **alt** : Altitude
    - **numrc** : Number of Relative Coordinates
    - **rc** : Relative Coordinate (numrc occurrences)
      - **x** : Relative X
      - **y** : Relative Y
      - **z** : Relative Z

rdsh : Road Descriptor Section Header

rdt : Road Descriptor Type

rdsize : Road Descriptor Size

numrdsinf : Number of Road Descriptor Section Information

rdsinf : Road Descriptor Section Information (numrdsinf occurrences)

numrd : Number of Road Descriptor

rds : Road Descriptor (numrd occurrences)

rname : Road Name

rnum : Road Number

linkid : Link ID

intersectname : Intersection Name

intersected : Intersection Number

nodeid : Node ID

brunnelname : brunnel Name

offsecthdr : offset Information Header

offsett : Offset Type

defoffsetorg : Definition Offset Origin

offsetdu : Offset Distance Unit

offsetau : Offset Angle Unit

numoffinf : Number of Offset Section Information

offsectinf : Offset Section Information (numoffinf occurrences)

noffset : Number of Offsets

offsets : Offsets (noffset occurrences)

airlinedfo : Airline Distance fromOrigin

angle : Angle

vehicledrivingd : Vehicle Driving Distance

direction : Direction

UDExt : User-defined Extension

UDExtHeader : User-defined ExtensionHeader

extID : Extension Identifier

extName : Extension Name

extType : Extension Type

UDExtinf : User-defined ExtensionInformation

< user-defined data elements and/or nested data frames >

A.3 Structure in ASN.1

```
Lrp1-linf DEFINITIONS ::=              --Lrp1-linf: Location information of Location
referencing procedure 1
BEGIN
EXPORTS
   Lrp1-linf;
IMPORTS
Precoded-Table, Blockcode-Table;

   Lrp1-linf ::= SEQUENCE {
      lsheader       Lsheader          OPTIONAL,       -Location_Section_Header
      lsinf          Lsinf                             -Location_Section_Information
      }

   Lisheader ::= SEQUENCE {
      ptname        UTF8String         OPTIONAL,       -Precoded_Table_Name
      ptversion     Ptversion          OPTIONAL,       -Precoded_Table_Version
      linft         Linft              OPTIONAL        -Location_Information_Type
      }

   Ptversion ::= CHOICE {
      vint           INTEGER,
      vstr           UTF8String
      }

   Linft ::= ENUMERATED {
      point,
      line,
      area
      }

   Lsinf ::= SEQUENCE {
      cshi           Csih              OPTIONAL,       -Coordinates_Section_Header
      csi            Csi               OPTIONAL,       -Coordinates_Section_Information
      udext          UDExt             OPTIONAL        -User-Defined Extension
      }

   Csih ::= SEQUENCE {
      ct             Ct                OPTIONAL,       -Coordinate_Type
      bct            Bct               OPTIONAL,       -Block_Code_Table
      rct            Rct               OPTIONAL,       -Region_Code_Table
      acu            Acu               OPTIONAL,       -Absolute_Coordinate_Unit
      rcu            Rcu               OPTIONAL,       -Relative_Coordinate_Unit
      horerr         Horerr            OPTIONAL,       -Horizontal_Error
      hterr          Hterr             OPTIONAL,       -Height_Error
      numcsi         INTEGER           OPTIONAL        -Number_of_Coordinates_Information
      }

   Ct ::= ENUMERATED {
      absolute,
      relative,
      grid
      }

   Bct ::= ENUMERATED {
      utms,
      utmp,
      rectc,
      pgrid1
      }
   Rct ::= ENUMERATED {
```

```
    rct1,
    rct2,
    }

Acu ::= ENUMERATED {
    degree,
    ddmmss,
    second
    }

Rcu ::= ENUMERATED {
    secondu1,
    secondu2
    }

Horerr ::= ENUMERATED {
    basemap1,
    basemap2,
    basemap3
    }

Hterr ::= ENUMERATED {
    error1,
    error2
    }

Csi ::= SEQUENCE {
    numblock  INTEGER              OPTIONAL,          -Number_of_Blocks
    bcodes    SEQUENCE (SIZE (0..MAX)) OF INTEGER, -Block_Codes
    numregion INTEGER              OPTIONAL,          -Number_of_Regions
    rcodes    SEQUENCE (SIZE (0..MAX)) OF INTEGER, -Region_Codes
    numac     INTEGER              OPTIONAL,          -Number_of_Absolute_Coordinates
    ac        SEQUENCE (SIZE (0..MAX)) OF Ac,      -Absolute_Coordinates
    numrc     INTEGER              OPTIONAL,          -Number_of_Relative_Coordinates
    rc        SEQUENCE (SIZE (0..MAX)) OF Rc,      -Relative_Coordinates
    rdsh      Rdsh                 OPTIONAL,          -Road_Descriptor_Section_Header
    rdsinf    Rdsinf               OPTIONAL           -Road_Descriptor_Section_Information
    }

Ac ::= SEQUENCE {
    lat       Latitude,
    lon       Longitude,
    alt       Altitude             OPTIONAL
    }

Latitude ::= CHOICE {
    lat_int       INTEGER       (-90000000..90000000),     -microdegrees
    lat_real      REAL          (-90..90)                  -degrees
    }

Longitude ::= CHOICE {
    lon_int       INTEGER       (-180000000..180000000),   -microdegrees
    lon_real      REAL          (-180..180)                -degrees
    }

Altitude ::= CHOICE {
    int           INTEGER       (-1000..10000),            -meters
    real          REAL          (-1..10)                   -kilometers
    }

Rc ::= SEQUENCE {
    x           INTEGER,                          -Relative_X
    y           INTEGER,                          -Relative_Y
    z           INTEGER          OPTIONAL         -Relative_Z
    }

Rdsh ::= SEQUENCE {
    rdt         INTEGER          OPTIONAL,        -Road_Descriptor_Type
    rdsize      INTEGER          OPTIONAL,        -Road_Descriptor_Size
    numrdsinf   INTEGER          OPTIONAL         -Number_of_Road_Descriptor_Section_
                                                  -Information
```

```
      }

   Rdsinf ::= SEQUENCE {
      numrd       INTEGER          OPTIONAL,        -Number_of_Road_Descriptors
      rds         SEQUENCE (SIZE (0..255)) OF Rd    -Road_Descriptors
      }

   Rd ::= SEQUENCE {
      rname         UTF8String        OPTIONAL,            -Road_Name
      rnum          INTEGER           OPTIONAL,            -Road_Number
      linkid        INTEGER           OPTIONAL,            -Link_ID
      intersectname UTF8String        OPTIONAL,            -Intersection_Name
      intersectid   INTEGER           OPTIONAL,            -Intersection_Number
      nodeid        INTEGER           OPTIONAL,            -Node_ID
      brunnelname   UTF8String        OPTIONAL,            -Brunnel_Name,
      offsecthdr    Offsecthdr        OPTIONAL,            -Offset_Section_Header
      offsectinf    SEQUENCE (SIZE (0..255)) OF Offsectinf -Offset_Section_Information
      }

   Offsecthdr ::= SEQUENCE {
      offsett        Offsett          OPTIONAL,            -Offset_Type
      defoffsetorg   Defoffsetorg     OPTIONAL,            -Definition_Offset_Origin
      offsetdu       Offsetdu         OPTIONAL,            -Offset_Distance_Unit
      offsetau   Offsetau         OPTIONAL,           -Offset_Angle_Unit
      numoffinf   INTEGER          OPTIONAL            -Number_of_Offset_Section_
Information
      }

   Offsett ::= ENUMERATED {
      on-route,
      airline
      }

   Defoffsetorg ::= ENUMERATED {
      Start point,
      End point
      }

   Offsetdu ::= ENUMERATED {
      1m,
      10m,
      100m
      }

   Offsetau ::= ENUMERATED {
      degree,
      radian
      }


   Offsectinf ::= SEQUENCE {
      noffset         INTEGER           OPTIONAL,           -Number_of_Offsets
      offsets         SEQUENCE (SIZE (0..255)) OF Offsets
      }

   Offsets ::= SEQUENCE {
      airlinedfo      INTEGER           OPTIONAL,           -Airline Distance_from_
Origin
      angle           Angle_type        OPTIONAL,           -Angle
      vehicledrivingd INTEGER           OPTIONAL,           -Vehicle_Driving_Distance
      direction       Direction_type    OPTIONAL            -Direction
      }

   Angle_type ::= CHOICE {
      ang_int       INTEGER,
      ang_real      REAL
      }

   Direction_type ::= CHOICE {
      direc_int       INTEGER,
      direc_real      REAL
```

```
      }

   UDExt ::= SEQUENCE {
      uDEExtHeader    UDExtHeader        OPTIONAL,            -User-Defined Extension
Header
      uDExtinf        UDExtinf                                -User-Defined Extension
Information
      }

   UDExtHeader ::= SEQUENCE {
      extID           ExtID              OPTIONAL,            -Extension_Identifier
      extName         UTF8String         OPTIONAL,            -Extension_Name
      extType         UTF8String         OPTIONAL             -Extension_Type
      }

   ExtID ::= CHOICE {
      extIDint        INTEGER,
      extIDstr        UTF8String
      }

   UDExtinf ::= SEQUENCE {               -user-defined set of data elements and/or nested
data frames
      byte            BYTE
      }
END
```

A.4 Structure as XML schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:Lrp1linf="Lrp1linf" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                                 elementFormDefault="qualified">
       <xsd:simpleType name="UTF8String">
              <xsd:restriction base="xsd:string">
                     <xsd:maxLength value="255"/>
              </xsd:restriction>
       </xsd:simpleType>
       <xsd:simpleType name="Linft">
              <xsd:restriction base="UTF8String">
                     <xsd:enumeration value="point"/>
                     <xsd:enumeration value="line"/>
                     <xsd:enumeration value="area"/>
              </xsd:restriction>
       </xsd:simpleType>
       <xsd:simpleType name="Ct">
              <xsd:restriction base="UTF8String">
                     <xsd:enumeration value="absolute"/>
                     <xsd:enumeration value="relative"/>
                     <xsd:enumeration value="grid"/>
              </xsd:restriction>
       </xsd:simpleType>
       <xsd:simpleType name="Bct">
              <xsd:restriction base="UTF8String">
                     <xsd:enumeration value="utms"/>
                     <xsd:enumeration value="utmp"/>
                     <xsd:enumeration value="rectc"/>
                     <xsd:enumeration value="pgrid1"/>
              </xsd:restriction>
       </xsd:simpleType>
       <xsd:simpleType name="Rct">
              <xsd:restriction base="UTF8String">
                     <xsd:enumeration value="rct1"/>
                     <xsd:enumeration value="rct2"/>
              </xsd:restriction>
       </xsd:simpleType>
       <xsd:simpleType name="Acu">
              <xsd:restriction base="UTF8String">
                     <xsd:enumeration value="degree"/>
                     <xsd:enumeration value="ddmmss"/>
                     <xsd:enumeration value="second"/>
              </xsd:restriction>
       </xsd:simpleType>
```

```
<xsd:simpleType name="Rcu">
	<xsd:restriction base="UTF8String">
		<xsd:enumeration value="secondu1"/>
		<xsd:enumeration value="secondu2"/>
	</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="Horerr">
	<xsd:restriction base="UTF8String">
		<xsd:enumeration value="basemap1"/>
		<xsd:enumeration value="basemap2"/>
		<xsd:enumeration value="basemap3"/>
	</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="Hterr">
	<xsd:restriction base="UTF8String">
		<xsd:enumeration value="error1"/>
		<xsd:enumeration value="error2"/>
	</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ExtType">
	<xsd:restriction base="UTF8String"/>
</xsd:simpleType>
<xsd:simpleType name="Offsett">
	<xsd:restriction base="UTF8String">
		<xsd:enumeration value="on-route"/>
		<xsd:enumeration value="airline"/>
	</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="Defoffsetorg">
	<xsd:restriction base="UTF8String">
		<xsd:enumeration value="Start point"/>
		<xsd:enumeration value="End point"/>
	</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="Offsetdu">
	<xsd:restriction base="UTF8String">
		<xsd:enumeration value="1m"/>
		<xsd:enumeration value="10m"/>
		<xsd:enumeration value="100m"/>
	</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="Offsetau">
	<xsd:restriction base="UTF8String">
		<xsd:enumeration value="degree"/>
		<xsd:enumeration value="radian"/>
	</xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="Lsheader">
	<xsd:sequence>
		<xsd:element name="ptname" type="UTF8String" minOccurs="0"/>
		<xsd:element name="ptversion" type="Ptversion" minOccurs="0"/>
		<xsd:element name="linft" type="Linft" minOccurs="0"/>
	</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Lsinf">
	<xsd:sequence>
		<xsd:element name="cshi" type="Csih" minOccurs="0"/>
		<xsd:element name="csi" type="Csi" minOccurs="0"/>
		<xsd:element name="udext" type="UDExt" minOccurs="0"/>
	</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Ptversion">
	<xsd:choice>
		<xsd:element name="vint" type="xsd:integer"/>
		<xsd:element name="vstr" type="UTF8String"/>
	</xsd:choice>
</xsd:complexType>
<xsd:complexType name="Csih">
	<xsd:sequence>
		<xsd:element name="ct" type="Ct" minOccurs="0"/>
		<xsd:element name="bct" type="Bct" minOccurs="0"/>
```

```
            <xsd:element name="rct" type="Rct" minOccurs="0"/>
            <xsd:element name="acu" type="Acu" minOccurs="0"/>
            <xsd:element name="rcu" type="Rcu" minOccurs="0"/>
            <xsd:element name="horerr" type="Horerr" minOccurs="0"/>
            <xsd:element name="hterr" type="Hterr" minOccurs="0"/>
            <xsd:element name="numcsi" type="xsd:integer" minOccurs="0"/>
      </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Csi">
      <xsd:sequence>
            <xsd:element name="numblock" type="xsd:integer" minOccurs="0"/>
            <xsd:element name="bcodes" type="xsd:integer" minOccurs="0"
                                                maxOccurs="unbounded"/>
            <xsd:element name="numregion" type="xsd:integer" minOccurs="0"/>
            <xsd:element name="rcodes" type="xsd:integer" minOccurs="0"
                                                maxOccurs="unbounded"/>
            <xsd:element name="numac" type="xsd:integer" minOccurs="0"/>
            <xsd:element name="ac" type="Ac" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="numrc" type="xsd:integer" minOccurs="0"/>
            <xsd:element name="rc" type="Rc" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="rdsh" type="Rdsh" minOccurs="0"/>
            <xsd:element name="rdsinf" type="Rdsinf" minOccurs="0"/>
      </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Ac">
      <xsd:sequence>
            <xsd:element name="lat" type="Latitude"/>
            <xsd:element name="lon" type="Longitude"/>
            <xsd:element name="alt" type="Altitude" minOccurs="0"/>
      </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Rc">
      <xsd:sequence>
            <xsd:element name="x" type="xsd:integer">
                  <xsd:annotation>
                        <xsd:documentation>relative</xsd:documentation>
                  </xsd:annotation>
            </xsd:element>
            <xsd:element name="y" type="xsd:integer">
                  <xsd:annotation>
                        <xsd:documentation>relative</xsd:documentation>
                  </xsd:annotation>
            </xsd:element>
            <xsd:element name="z" type="xsd:integer" minOccurs="0">
                  <xsd:annotation>
                        <xsd:documentation>relative</xsd:documentation>
                  </xsd:annotation>
            </xsd:element>
      </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Rdsh">
      <xsd:sequence>
            <xsd:element name="rdt" type="xsd:integer" minOccurs="0"/>
            <xsd:element name="rdsize" type="xsd:integer" minOccurs="0"/>
            <xsd:element name="numrdsinf" type="xsd:integer" minOccurs="0"/>
      </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Rdsinf">
      <xsd:sequence>
            <xsd:element name="numrd" type="xsd:integer" minOccurs="0"/>
            <xsd:element name="rds" type="Rd" minOccurs="0" maxOccurs="255"/>
      </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Rd">
      <xsd:sequence>
            <xsd:element name="rname" type="UTF8String" minOccurs="0"/>
            <xsd:element name="rnum" type="xsd:integer" minOccurs="0"/>
            <xsd:element name="linkid" type="xsd:integer" minOccurs="0"/>
            <xsd:element name="intersectname" type="UTF8String" minOccurs="0"/>
            <xsd:element name="intersectid" type="xsd:integer" minOccurs="0"/>
            <xsd:element name="nodeid" type="xsd:integer" minOccurs="0"/>
```

```
                        <xsd:element name="brunnelname" type="UTF8String" minOccurs="0"/>
                        <xsd:element name="offsecthdr" type="Offsecthdr" minOccurs="0"/>
                        <xsd:element name="offsectinf" type="Offsectinf" minOccurs="0"
                                                                        maxOccurs="unbounded"/>
                </xsd:sequence>
        </xsd:complexType>
        <xsd:complexType name="Offsecthdr">
                <xsd:sequence>
                        <xsd:element name="offsett" type="Offsett" minOccurs="0"/>
                        <xsd:element name="defoffsetorg" type="Defoffsetorg" minOccurs="0"/>
                        <xsd:element name="offsetdu" type="Offsetdu" minOccurs="0"/>
                        <xsd:element name="offsetau" type="Offsetau" minOccurs="0"/>
                        <xsd:element name="numoffinf" type="xsd:integer" minOccurs="0"/>
                </xsd:sequence>
        </xsd:complexType>
        <xsd:complexType name="Offsectinf">
                <xsd:sequence>
                        <xsd:element name="noffset" type="xsd:integer" minOccurs="0"/>
                        <xsd:element name="offsets" type="Offsets" minOccurs="0"
maxOccurs="255"/>
                </xsd:sequence>
        </xsd:complexType>
        <xsd:complexType name="Offsets">
                <xsd:sequence>
                        <xsd:element name="airlinedfo" type="xsd:integer" minOccurs="0"/>
                        <xsd:element name="angle" type="Angletype" minOccurs="0"/>
                        <xsd:element name="vehicledrivingd" type="xsd:integer" minOccurs="0"/>
                        <xsd:element name="direction" type="Directiontype" minOccurs="0"/>
                </xsd:sequence>
        </xsd:complexType>
        <xsd:complexType name="Angletype">
                <xsd:choice>
                        <xsd:element name="angint" type="xsd:integer"/>
                        <xsd:element name="angreal" type="xsd:float"/>
                </xsd:choice>
        </xsd:complexType>
        <xsd:complexType name="Directiontype">
                <xsd:choice>
                        <xsd:element name="direcint" type="xsd:integer"/>
                        <xsd:element name="direcreal" type="xsd:float"/>
                </xsd:choice>
        </xsd:complexType>
        <xsd:complexType name="UDExt">
                <xsd:sequence>
                        <xsd:element name="UDEExtHeader" type="UDExtHeader" minOccurs="0"/>
                        <xsd:element name="UDExtinf" type="UDExtinf"/>
                </xsd:sequence>
        </xsd:complexType>
        <xsd:complexType name="UDExtHeader">
                <xsd:sequence>
                        <xsd:element name="extID" type="ExtID" minOccurs="0"/>
                        <xsd:element name="extName" type="UTF8String" minOccurs="0"/>
                        <xsd:element name="extType" type="UTF8String" minOccurs="0">
                                <xsd:annotation>
                                        <xsd:documentation>Extension_Type</xsd:documentation>
                                </xsd:annotation>
                        </xsd:element>
                </xsd:sequence>
        </xsd:complexType>
        <xsd:complexType name="UDExtinf">
                <xsd:sequence>
                        <xsd:element name="byte" type="xsd:byte"/>
                </xsd:sequence>
        </xsd:complexType>
        <xsd:complexType name="ExtID">
                <xsd:choice>
                        <xsd:element name="extIDint" type="xsd:integer"/>
                        <xsd:element name="extIDstr" type="UTF8String"/>
                </xsd:choice>
        </xsd:complexType>
        <xsd:complexType name="Longitude">
```

```
        <xsd:choice>
                <xsd:element name="lonint">
                        <xsd:annotation>
                                <xsd:documentation>microdegrees</xsd:documentation>
                        </xsd:annotation>
                        <xsd:simpleType>
                                <xsd:restriction base="xsd:integer">
                                        <xsd:minInclusive value="-180000000"/>
                                        <xsd:maxInclusive value="180000000"/>
                                </xsd:restriction>
                        </xsd:simpleType>
                </xsd:element>
                <xsd:element name="lonreal">
                        <xsd:annotation>
                                <xsd:documentation>degrees</xsd:documentation>
                        </xsd:annotation>
                        <xsd:simpleType>
                                <xsd:restriction base="xsd:float">
                                        <xsd:maxInclusive value="180"/>
                                        <xsd:minInclusive value="-180"/>
                                </xsd:restriction>
                        </xsd:simpleType>
                </xsd:element>
        </xsd:choice>
</xsd:complexType>
<xsd:complexType name="Latitude">
        <xsd:choice>
                <xsd:element name="latint">
                        <xsd:annotation>
                                <xsd:documentation>microdegrees</xsd:documentation>
                        </xsd:annotation>
                        <xsd:simpleType>
                                <xsd:restriction base="xsd:integer">
                                        <xsd:maxInclusive value="90000000"/>
                                        <xsd:minInclusive value="-90000000"/>
                                </xsd:restriction>
                        </xsd:simpleType>
                </xsd:element>
                <xsd:element name="latreal">
                        <xsd:annotation>
                                <xsd:documentation>degrees</xsd:documentation>
                        </xsd:annotation>
                        <xsd:simpleType>
                                <xsd:restriction base="xsd:float">
                                        <xsd:maxInclusive value="90"/>
                                        <xsd:minInclusive value="-90"/>
                                </xsd:restriction>
                        </xsd:simpleType>
                </xsd:element>
        </xsd:choice>
</xsd:complexType>
<xsd:complexType name="Altitude">
        <xsd:choice>
                <xsd:element name="altint">
                        <xsd:annotation>
                                <xsd:documentation>meters</xsd:documentation>
                        </xsd:annotation>
                        <xsd:simpleType>
                                <xsd:restriction base="xsd:integer">
                                        <xsd:maxInclusive value="10000"/>
                                        <xsd:minInclusive value="-1000"/>
                                </xsd:restriction>
                        </xsd:simpleType>
                </xsd:element>
                <xsd:element name="altreal">
                        <xsd:annotation>
                                <xsd:documentation>kilometers</xsd:documentation>
                        </xsd:annotation>
                        <xsd:simpleType>
                                <xsd:restriction base="xsd:float">
                                        <xsd:maxInclusive value="10"/>
```

```
                                        <xsd:minInclusive value="-1"/>
                                    </xsd:restriction>
                                </xsd:simpleType>
                            </xsd:element>
                    </xsd:choice>
            </xsd:complexType>
            <xsd:element name="Lrp1linf">
                    <xsd:complexType>
                            <xsd:sequence>
                                    <xsd:element name="lsheader" type="Lsheader" minOccurs="0"/>
                                    <xsd:element name="lsinf" type="Lsinf"/>
                            </xsd:sequence>
                    </xsd:complexType>
            </xsd:element>
</xsd:schema>
```

Annex B
(informative)

ALERT-C location reference, TPEG2 logical structure

B.1 Description of the logical structure

B.1.1 General

TPEG and the TPEG Automotive Profile (TAP) service profile were conceived to foster provisioning of traffic information using TPEG technology. To allow easy adoption and migration of RDS-TMC based systems and services, reuse of TMC location referencing functionality from this existing solution is made possible. The ALERT-C (TLR) location referencing method allows TPEG application to continue to use ALERT-C Location tables already in place, in addition to new dynamic LRMs. This presents TPEG services the option either to code very efficiently, using already existing ALERT-C location codes but also to allow any location to be coded. The duality of requirements has been adhered in such, that this data format specification defines data elements for both methodologies including necessary selection means.

B.1.2 Pre-coded ALERT-C location referencing

This annex defines the TPEG logical datastructure for ALERT-C location referencing to be used in pre-coded LRC container for TPEG services. This logical datastructure conforms to the TPEG modelling rules[16].

The method of ALERT-C location referencing is defined in ISO 14819-3:2013[12]. ALERT-C defines a hierarchical structure of areas, lines and points. The points are connected to predecessors and successors and belong to lines and areas. All elements in the structure are defined by one unique number which is used to refer to that particular part of the structure. In addition to the normal ALERT-C LRM, a precise LRM is also foreseen here.

The proposed method defined in the precise location referencing specification[5] defines attributes to code an arbitrary position in between two points being successors in the structure mentioned before. These attributes are made optional because not every location will want to make use of it. B.3.1 specifies the precise location referencing sub elements as defined for ALERT-C Location.

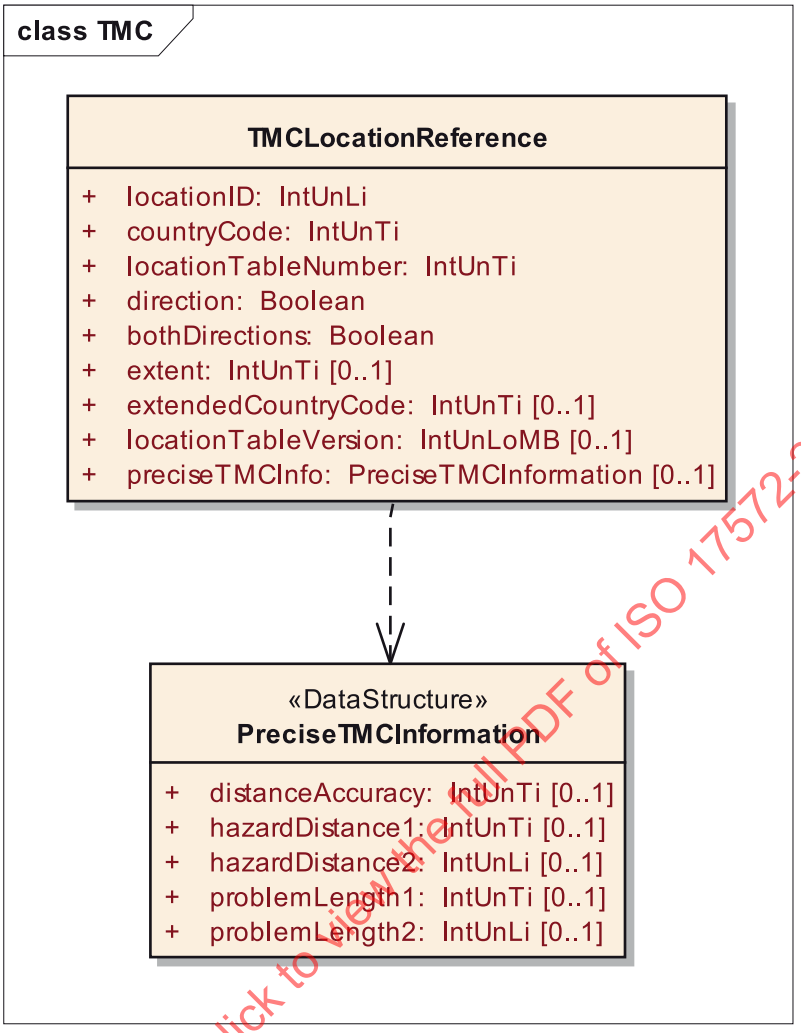

Figure B.1 — TPEG2 UML structure[16] of TLR (Alert-C)

B.2 TLR message components

B.2.1 TMCLocationReference

The TMC location stores a pre-coded location description including the possibility to code precisely a part of a road section in-between TMC locations.

Name	Type	Multiplicity	Description
locationID	IntUnLi	1	The actual location number (primary location).
countryCode	IntUnTi	1	Defines the different countries being responsible for the database creation. The table of country codes is specified in ISO 14819-3:2013.
locationTableNumber	IntUnTi	1	The value is specified per country and may differ over time.
direction	Boolean	1	If the direction flag is true it means positive direction i.e. following to the successor in the location list.

Name	Type	Multiplicity	Description
bothDirections	Boolean	1	If set to true the location shall be treated as being applicable for both directions. NOTE Be aware that ISO 18234 defines this information in two pieces: 1. implicitly per event code and 2. as inversion bit in the control code.
extent	IntUnTi	0..1	Number of hops in the linked list of location codes. The direction is defined by the direction attribute. Absent value is defined 0 and means inside the current location.
extendedCountryCode	IntUnTi	0..1	In non-European countries, the full range of location tables per country code can be used provided that the Extended Country Code associated with the ALERT-C location table is defined and provided (Extended Country Codes are defined in IEC 62106:2000).
locationTableVersion	IntUnLoMB	0..1	If 1 byte, 4 bits for major and 3 bits for minor. If 2 bytes, 7 bits for major and 7 bits for minor. 3 bytes not possible because TMC restricts to 0-99 for both.
preciseTMCInfo	PreciseTMCInformation	0..1	n.a.

B.3 TLR datatypes

B.3.1 PreciseTMCInformation

PreciseTMCInformation defines the collection of optional precise TMC location coding.

Name	Type	Multiplicity	Description
distanceAccuracy	IntUnTi	0..1	0 = 100 m, 1 = 500 m, 2 = 1 km, 3 = more than 1 km 100 m if attribute is absent.
hazardDistance1	IntUnTi	0..1	Upstream distance from primary location (against the direction of traffic), also known as D1. Range up to 25 km calculated in steps of 100 m.
hazardDistance2	IntUnLi	0..1	Same as hazardDistance1 but range up to 6 553 km.
problemLength1	IntUnTi	0..1	Distance from the start of the problem to the end of the problem. Which is not necessarily the primary location. Length up to 25 km.
problemLength2	IntUnLi	0..1	Same as problemLength1 but length up to 6 553 km.

Annex C
(normative)

ALERT-C location reference, TPEG2 binary representation

C.1 General

This annex defines the TPEG2 binary representation[17] of TLR container for use with TPEG services.

C.2 TLR container components

C.2.1 List of Generic Component Ids

The GCId of root element is defined by the Location Referencing Container specification[19].

Name	Id
TMCLocationReference	defined by LRC

C.2.2 TMCLocationReference

<TMCLocationReference(x)>:=	
<IntUnTi>(x),	: Id of this component.
<IntUnLoMB>(lengthComp),	: Number of bytes in component.
<IntUnLoMB>(lengthAttr),	: Number of bytes in attributes.
<IntUnLi>(locationID),	: The actual location number (primary location).
<IntUnTi>(countryCode),	: Defines the different countries being responsible for the database creation. The table of country codes is specified in ISO 14819-3:2013.
<IntUnTi>(locationTableNumber),	: The value is specified per country and may differ over time.
BitArray(selector),	
if (bit 0 of selector is set)	
<Boolean>(direction),	: If the direction flag is true it means positive direction i.e. following to the successor in the location list.
if (bit 1 of selector is set)	
<Boolean>(bothDirections),	: If set to true the location shall be treated as being applicable for both directions. NOTE Be aware that ISO 18234 defines this information in two pieces: 1. implicitly per event code and 2. as inversion bit in the control code.
if (bit 2 of selector is set)	
<IntUnTi>(extent),	: Number of hops in the linked list of location codes. The direction is defined by the direction attribute. Absent value is defined 0 and means inside the current location.
if (bit 3 of selector is set)	

<IntUnTi>(extendedCountryCode),	: In non-European countries, the full range of location tables per country code can be used provided that the Extended Country Code associated with the ALERT-C location table is defined and provided (Extended Country Codes are defined in IEC 62106:2000).
if (bit 4 of selector is set)	
<IntUnLoMB>(locationTableVersion),	: If 1 byte, 4 bits for major and 3 bits for minor. If 2 bytes, 7 bits for major and 7 bits for minor. 3 bytes not possible because TMC restricts to 0-99 for both.
if (bit 5 of selector is set)	
<PreciseTMCInformation>(preciseTMCInfo);	

C.3 TLR datatypes

C.3.1 PreciseTMCInformation

<PreciseTMCInformation>:=	
BitArray(selector),	
if (bit 0 of selector is set)	
<IntUnTi>(distanceAccuracy),	: 0 = 100 m, 1 = 500 m, 2 = 1 km, 3 = more than 1 km 100 m if attribute is absent.
if (bit 1 of selector is set)	
<IntUnTi>(hazardDistance1),	: Upstream distance from primary location (against the direction of traffic), also known as D1. Range up to 25 km calculated in steps of 100 m.
if (bit 2 of selector is set)	
<IntUnLi>(hazardDistance2),	: Same as hazardDistance1 but range up to 6 553 km.
if (bit 3 of selector is set)	
<IntUnTi>(problemLength1),	: Distance from the start of the problem to the end of the problem. Which is not necessarily the primary location. Length up to 25 km.
if (bit 4 of selector is set)	
<IntUnLi>(problemLength2);	: Same as problemLength1 but length up to 6 553 km.

Annex D
(normative)

ALERT-C location reference, TPEG2 XML representation

D.1 General

This annex defines the TPEG2 XML representation[18] of the TLR container for use with TPEG services.

D.2 TLR container components

D.2.1 TMCLocationReference

```
<xs:element name="TMCLocationReference" type="TMCLocationReference"/>
<xs:complexType name="TMCLocationReference">
 <xs:complexContent>
 <xs:extension base="tsf:ApplicationRootMessageML">
 <xs:sequence>
 <xs:element name="locationID" type="tdt:IntUnLi"/>
 <xs:element name="countryCode" type="tdt:IntUnTi"/>
 <xs:element name="locationTableNumber" type="tdt:IntUnTi"/>
 <xs:element name="direction" type="tdt:Boolean"/>
 <xs:element name="bothDirections" type="tdt:Boolean"/>
 <xs:element name="extent" type="tdt:IntUnTi" minOccurs="0"/>
 <xs:element name="extendedCountryCode" type="tdt:IntUnTi" minOccurs="0"/>
 <xs:element name="locationTableVersion" type="tdt:IntUnLoMB" minOccurs="0"/>
 <xs:element name="preciseTMCInfo" type="PreciseTMCInformation" minOccurs="0"/>
 </xs:sequence>
 </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

D.3 TLR datatypes

D.3.1 PreciseTMCInformation

```
<xs:complexType name="PreciseTMCInformation">
 <xs:sequence>
 <xs:element name="distanceAccuracy" type="tdt:IntUnTi" minOccurs="0"/>
 <xs:element name="hazardDistance1" type="tdt:IntUnTi" minOccurs="0"/>
 <xs:element name="hazardDistance2" type="tdt:IntUnLi" minOccurs="0"/>
 <xs:element name="problemLength1" type="tdt:IntUnTi" minOccurs="0"/>
 <xs:element name="problemLength2" type="tdt:IntUnLi" minOccurs="0"/>
 </xs:sequence>
</xs:complexType>
```

D.4 Full TLR schema definition

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- This XML schema is generated with tpegUMLconverter V2.3 -->
<xs:schema xmlns="http://www.tisa.org/TPEG/TLR_2_0"
   targetNamespace="http://www.tisa.org/TPEG/TLR_2_0"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:tsf="http://www.tisa.org/TPEG/SFW_1_1"
   xmlns:tdt="http://www.tisa.org/TPEG/TPEGDataTypes_2_0"
   elementFormDefault="qualified"
   attributeFormDefault="qualified">
   <xs:import namespace="http://www.tisa.org/TPEG/SFW_1_1" schemaLocation="SFW_1_1.xsd"/>
   <xs:import namespace="http://www.tisa.org/TPEG/
TPEGDataTypes_2_0" schemaLocation="TDT_2_0.xsd"/>
```

```
   <xs:element name="TMCLocationReference" type="TMCLocationReference"/>
   <xs:complexType name="TMCLocationReference">
      <xs:complexContent>
         <xs:extension base="tsf:ApplicationRootMessageML">
            <xs:sequence>
               <xs:element name="locationID" type="tdt:IntUnLi"/>
               <xs:element name="countryCode" type="tdt:IntUnTi"/>
               <xs:element name="locationTableNumber" type="tdt:IntUnTi"/>
               <xs:element name="direction" type="tdt:Boolean"/>
               <xs:element name="bothDirections" type="tdt:Boolean"/>
               <xs:element name="extent" type="tdt:IntUnTi" minOccurs="0"/>
               <xs:element name="extendedCountryCode" type="tdt:IntUnTi" minOccurs="0"/>
               <xs:element name="locationTableVersion" type="tdt:IntUnLoMB"
minOccurs="0"/>
               <xs:element name="preciseTMCInfo" type="PreciseTMCInformation"
minOccurs="0"/>
            </xs:sequence>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
   <xs:complexType name="PreciseTMCInformation">
      <xs:sequence>
         <xs:element name="distanceAccuracy" type="tdt:IntUnTi" minOccurs="0"/>
         <xs:element name="hazardDistance1" type="tdt:IntUnTi" minOccurs="0"/>
         <xs:element name="hazardDistance2" type="tdt:IntUnLi" minOccurs="0"/>
         <xs:element name="problemLength1" type="tdt:IntUnTi" minOccurs="0"/>
         <xs:element name="problemLength2" type="tdt:IntUnLi" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
</xs:schema>
```

Annex E
(informative)

Extended TMC Location reference, TPEG2 logical structure

E.1 Description of the logical structure

E.1.1 General

The RDS-TMC protocol (i.e. ALERT-C protocol) is specified in ISO 14819-1:2013. This protocol is designed to provide pre-coded information messages using pre-coded location references to end-users on inter-urban road networks. Both messages and locations are required to be stored in all client devices. The TMC system, developed for FM transmission in the RDS sub-channel, is limited to a code-base of <64 000 locations per location table. Some Events in RDS-TMC protocol address only the exits and entries of the point location defined by the location code (e.g. event "exit closed").

In TPEG, as a design principle, locations and events are strictly separated. The location (the "exit") shall be signalled in the Location referencing container, the event ("closed") in the application container. Thus, to allow with TMC locations also addressing the exit and entry roads of a TMC location, the location container shall contain the additional information for referencing precisely these entries and exits.

The ETL extends the ALERT-C location reference defined in Annex B to allow specifying exit and entry roads of a defined TMC point location, as well as to more precisely indicate whether the junction internal (through) road elements of the primary and secondary location are affected by the application contents or not.

This annex defines the TPEG logical data structure for the ETL. This logical datastructure conforms to the TPEG modelling rules[16]. Although both ALERT-C location referencing and ETL share the basic TMC location concept of Reference [12], ETL has a separate logical datastructure and a separate corresponding binary and XML representation, since in TPEG services the ETL location reference is signalled separately in the TPEG Location Referencing Container[19].

E.1.2 ETL specific constraints

E.1.2.1 ETL specification version number signalling

Version numbering is used to track the separate versions of a location referencing toolkit through its development and deployment. The differences between these versions may have an impact on client devices.

Table E.1 shows the current version numbers for this specification version of ETL (to be signalled in attribute specVersionID).

Table E.1 — Current version numbers for signalling ETL within TPEG2-XML and TPEG2-binary

Major version number	1
Minor version number	0

E.1.2.2 Definition of TMC internal, TMC external and TMC direction

Figure E.1 shows the definition of the TMC internal road segments (shown for primary and secondary location) and TMC external road segments (shown for the primary location). In the TMClocationReference (see E.3.6) the Boolean 'useInternalPrimaryLocation' indicates whether the TMC internal (through)

road segments of the primary location shall be considered part of the location to be referenced or not. The Boolean 'useInternalSecondaryLocation' in this TMClocationReference (also see E.3.6) indicates whether the TMC internal (through) road segments of the secondary location shall be considered part of the location to be referenced or not.

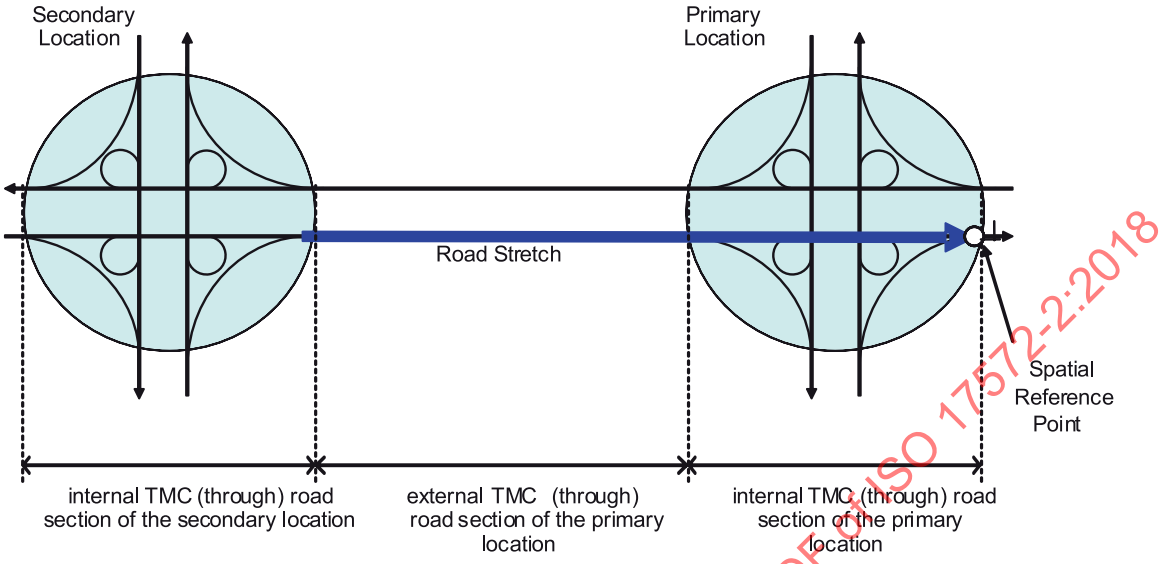

Figure E.1 — internal and external TMC through road section of the primary and secondary location.

The direction Boolean in the TMClocationReference (see E.3.6) defines the direction from the primary to the secondary location (against driving direction). Positive (true/1) means following the successor in the TMC table. The following example (see Table E.2) on the A9 Germany shows the correct use of the direction Boolean. For the driving direction München-Freimarn (12724) to München Frankfurter Ring (12723) the direction Boolan has to be coded as true/1.

Table E.2 — Extract out of the German TMC Location Table for the A9 motorway

→ positive direction following the succesor	Location code	first name	positive offset
	12722	München-Schwabing	12723
	12723	München-Frankfurter Ring	12724
	12724	München-Freimann	12725
	12725	München-Fröttmaning Süd	12726

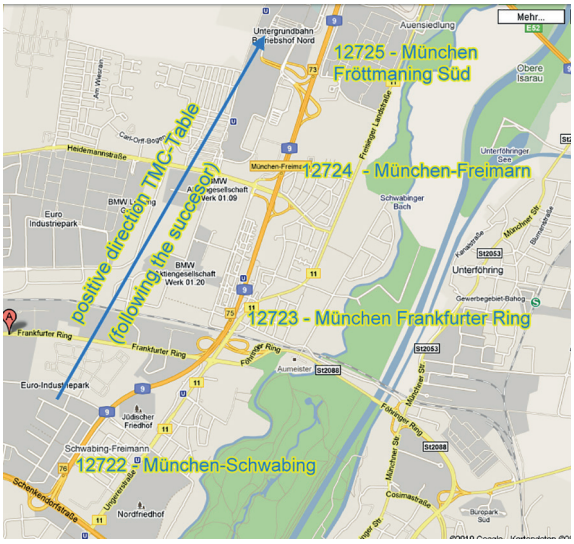

Figure E.2 — TMC-Locations in the digital road network map

E.2 ETL Structure

E.2.1 General

Figure E.3 shows the logical structure of the ETL container. The specific components and data types are described in E.2.2 and E.3 respectively.

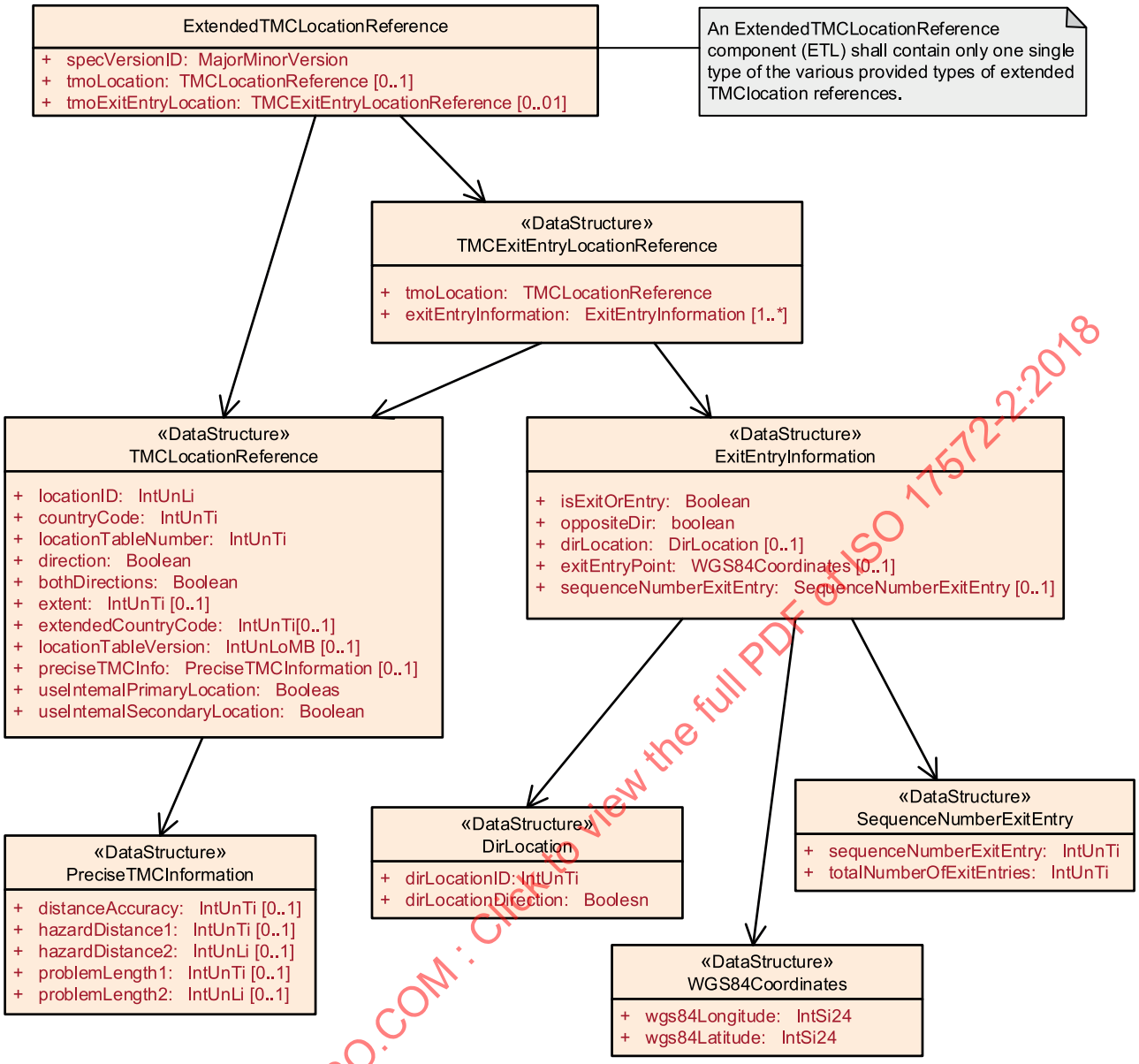

Figure E.3 — Structure of the ETL

E.2.2 ETL container components

E.2.2.1 ExtendedTMCLocationReference

This Extended TMC Location referencing container includes a number of enhancements with respect to the ALERT-C Location referencing container, in specific for exits and entries, internal/external location parts and version field to support future improvements. An ETL component shall contain only one single type of the various provided types of ETLs, for example the tmcLocation XOR the tmcExitEntryLocation.

Table E.3 — ExtendedTMCLocationReference

Name	Type	Multiplicity	Description
specVersionID	MajorMinorVersion	1	Major and Minor version of the ETL specification used to construct this location reference.
tmcLocation	TMCLocationReference	0..1	Enhanced ALERT-C location referencing container containing extra fields to specify the inclusion of TMC internal map segments of the primary and secondary location, as well as version information of the TMC location table used.
tmcExitEntryLocation	TMCExitEntryLocationReference	0..1	Extension to TMC-Location to address entries and/or exits of a given point location.

E.3 ETL Datatypes

E.3.1 DirLocation

The DirLocation datastructure in Table E.4 specifies a TMC location where the referenced exit road leads to or the referenced entry road originates from.

Table E.4 — DirLocation

Name	Type	Multiplicity	Description
dirLocationID	IntUnLi	1	The number of the TMC location where the exit leads to or the entry originates from.
dirLocationDirection	Boolean	1	The direction of the dirLocation where the exit leads to or the entry originates from.

E.3.2 PreciseTMCInformation

The PreciseTMCInformation datastructure in Table E.5 specifies additional fields in order to reference a more precise location in between successive TMC locations by means of a metric distance (hazard distance), measured from the primary location in accordance with ISO 14819-1:2013.

Table E.5 — PreciseTMCInformation

Name	Type	Multiplicity	Description
distanceAccuracy	IntUnTi	0..1	0 = 100 m, 1 = 500 m, 2 = 1 km, 3 = more than 1 km 100 m if attribute is absent.
hazardDistance1	IntUnTi	0..1	Upstream distance from primary location (against the direction of traffic), also known as D1. Range up to 25 km calculated in steps of 100 m.
hazardDistance2	IntUnLi	0..1	Same as hazardDistance1 but range up to 6 553 km.
problemLength1	IntUnTi	0..1	Distance from the start of the problem to the end of the problem. Which is not necessarily the primary location. Length up to 25 km.
problemLength2	IntUnLi	0..1	Same as problemLength1 but length up to 6 553 km.

Figure E.4 shows the definition of hazard Distance (D1) and Problem length (L).

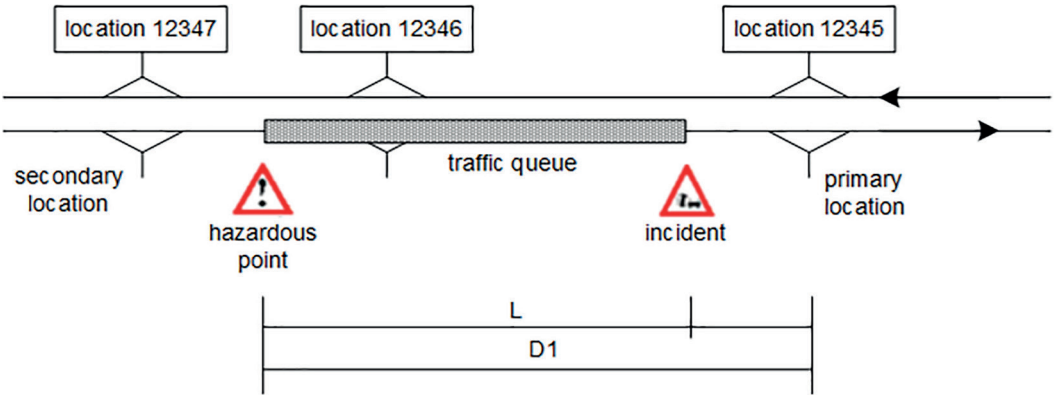

Figure E.4 — Definition of Hazard Distance (D1) and relation to Problem Length (L) (L may be signalled in an application container)

E.3.3 SequenceNumberExitEntry

The datastructure SequenceNumberExitEntry in Table E.6 specifies the sequence number of the exit or entry (e.g. exit 2 out of 5).

Table E.6 — SequenceNumberExitEntry

Name	Type	Multiplicity	Description
sequenceNumberExitEntry	IntUnTi	1	SequenceNumber of exit vs total number of exits on one side of the road, or SequenceNumber of entry vs total number of entries on one side of the road. SequenceNumber shall start at 1 up to total number of exits, respectively entries. First exit or entry encountered in driving direction shall be numbered 1. E.g. on one side of the road the following numbering can occur:Exit 1 Entry 1 Entry 2 Exit 2 Entry 3
totalNumberOfExitEntries	IntUnTi	1	Total number of exits or entries.

E.3.4 TMCExitEntryLocationReference

The TMCExitEntryLocation reference datastructure in Table E.7 specifies the applicable entries and/or exits of a TMC location where the content of the application content is located (e.g. to signal "first exit closed", where the TMCExitEntryLocationReference references the "first exit" and the application container signals the "closed" event).

Table E.7 — TMCExitEntryLocation

Name	Type	Multiplicity	Description
tmcLocation	TMCLocationReference	1	TMC location container indicating the main (junction) location for which the entries and/or exits are referenced in the datastructure exitEntryInformation.
exitEntryInformation	ExitEntryInformation	1..*	The specific entries or exits of the main TMC location to be referenced.

E.3.5 WGS84Coordinates

The datastructure WGS84 coordinates in Table E.8 specifies a standard WGS84 coordinate pair with 24 bit resolution.

Table E.8 — WGS84

Name	Type	Multiplicity	Description
wgs84Longitude	IntSi24	1	Longitude in standard 24 bit accuracy stores coordinates in order of magnitude of 10 micro degrees resolution (five decimals). Longitude (unit degrees) is encoded as follows: WGS84longitude = int (sign(longitude)*0,5 + (longitude * (224))/360)
wgs84Latitude	IntSi24	1	WGS84 latitude in standard 24 bit accuracy stores coordinates in order of magnitude of 10 micro degrees resolution (five decimals). Latitude (unit degrees) is encoded as follows: WGS84latitude = int (sign(latitude)*0,5 + (latitude * (224))/360)

E.3.6 TMCLocationReference

The datastructure TMC Location Reference in Table E.9 extends the ALERT-C Location referencing container to specify whether the TMC internal road segments of the primary and secondary location are affected (i.e. part of the location) or not (i.e. not part of the location to be referenced).

Table E.9 — TMCLocationReference

Name	Type	Multiplicity	Description
locationID	IntUnLi	1	The actual location number.
countryCode	IntUnTi	1	Defines the different countries being responsible for the database creation. The table of country codes is specified in ISO 14819-3:2013.
locationTableNumber	IntUnTi	1	The number of the location table. The value is specified per country and may differ over time.
direction	Boolean	1	The direction bit defines the direction from the primary to the secondary location (against driving direction). If the direction flag is true/1 it means positive direction i.e. following to the successor in the location list.

Table E.9 *(continued)*

Name	Type	Multiplicity	Description
bothDirections	Boolean	1	If set to true the location shall be treated as being applicable for both directions. NOTE In contrast to ISO 14819-1:2013 and ISO 14819-2:2013, which specifies a default directionality for each event code, in TPEG the directionality has to be set explicitly.
extent	IntUnTi	0..1	Number of hops in the linked list of location codes. The direction is defined by the direction attribute. Absent value is defined 0 and means inside the current location.
extendedCountryCode	IntUnTi	0..1	In non-European countries, the full range of location tables per country code can be used provided that the Extended Country Code associated with the ALERT-C location table is defined and provided (Extended Country Codes are defined in IEC 62106:2000). Even if this field is optional this field shall be sent mandatorily.
locationTableVersion	IntUnLoMB	0..1	If 1 byte, 4 bits for major and 3 bits for minor. If 2 bytes, 7 bits for major and 7 bits for minor. 4 bytes not possible because ALERT-C restricts to 0-99 for both. Letters in some old versions are interpreted as minor increments to the given number.
preciseTMCInfo	PreciseTMCInformation	0..1	Precise ALERT-C Information follows.
useInternalPrimaryLocation	Boolean	1	true/1 if the internal TMC through road section of the primary location is also affected. False/0 otherwise.
useInternalSecondaryLocation	Boolean	1	true/1 if the internal TMC through road section of the secondary location is also affected. False/0 otherwise.

E.3.7 ExitEntryInformation

The datastructure ExitEntryInformation in Table E.10 specifies detailed information as to which exits and/or entries are affected by the application contents.

Table E.10 — ExitEntryInformation

Name	Type	Multiplicity	Description
isExitOrEntry	Boolean	1	true or 1 for exit , false or 0 for entry.
oppositeDir	Boolean	1	If true or 1, the entry or exit is on the road side opposite to the direction of the main TMC location reference (This is to be used if the main tmc location indicates 'both directions').
dirLocation	DirLocation	0..1	The TMC location where the exit leads to or entry originates from.
exitEntryPoint	WGS84Coordinates	0..1	WGS84Coordinates of the specified exit entry. May be used advantageously if there is more than one Exit or Entry for the specified side and not all of them are closed.
sequenceNumberExitEntry	SequenceNumberExitEntry	0..1	Sequence number of exit or entry follows (e.g. exit 4 of 5).

Annex F
(normative)

Extended TMC Location reference, TPEG2 binary representation

F.1 General

This annex defines the TPEG2 binary representation[17] of the ETL container for use with TPEG services.

F.2 ETL container components

F.2.1 List of Generic Component Ids

The GCId of root element is defined by Location Referencing Container specification[19].

Name	Id
ExtendedTMCLocationReference	defined by LRC

F.2.2 ExtendedTMCLocationReference

This ExtendedTMC Location referencing container includes a number of enhancements with respect to the ALERT-C Location referencing container, in specific for exit and entries, internal/external location parts and version fields to support future improvements. An ETL component (ETL) shall contain only one single type of the various provided types of ETLs, for example the tmcLocation XOR the tmcExitEntryLocation.

<ExtendedTMCLocationReference(x)>:=	
<**IntUnTi**>(x),	: id of this component
<**IntUnLoMB**>(lengthComp),	: number of bytes in component
<**IntUnLoMB**>(lengthAttr),	: number of bytes in attributes
<**MajorMinorVersion**>(specVersionID),	: Major and Minor version of the ETL specification used to construct this location reference.
<**BitArray**>(selector),	
if (bit 0 of selector is set)	
<**TMCLocationReference**>(tmcLocation),	: Enhanced ALERT-C location referencing container containing extra fields to specify the inclusion of TMC internal map segments of the primary and secondary location, as well as version information of the TMC location table used.
if (bit 1 of selector is set)	
<**TMCExitEntryLocationReference**>(tmcExitEntryLocation);	: extension to TMC-Location to address entries and/or exits of a given point location

F.3 ETL Datatypes

F.3.1 DirLocation

The DirLocation datastructure specifies a TMC location where the referenced exit road leads to or the referenced entry road originates from.

<DirLocation()>:=	
<**IntUnLi**>(dirLocationID),	: The number of the TMC location where the exit leads to or the entry originates from.
<**BitArray**>(selector),	
if (bit 0 of selector is set)	
<**Boolean**>(dirLocationDirection);	: The direction of the dirLocation. where the exit leads to or the entry originates from.

F.3.2 PreciseTMCInformation

The PreciseTMCInformation datastructure specifies additional fields in order to reference a more precise location in between successive TMC location by means of a metric distance (hazard distance), measured from the primary location in accordance with ISO 14819-1:2013.

<PreciseTMCInformation()>:=	
<**BitArray**>(selector),	
if (bit 0 of selector is set)	
<**IntUnTi**>(distanceAccuracy),	: 0 = 100 m, 1 = 500 m, 2 = 1 km, 3 = more than 1 km 100 m if attribute is absent.
if (bit 1 of selector is set)	
<**IntUnTi**>(hazardDistance1),	: Upstream distance from primary location (against the direction of traffic), also known as D1. Range up to 25 km calculated in steps of 100 m
if (bit 2 of selector is set)	
<**IntUnLi**>(hazardDistance2),	: Same as hazardDistance1 but range up to 6 553 km
if (bit 3 of selector is set)	
<**IntUnTi**>(problemLength1),	: Distance from the start of the problem to the end of the problem. Which is not necessarily the primary location. Length up to 25 km
if (bit 4 of selector is set)	
<**IntUnLi**>(problemLength2);	: Same as problemLength1 but length up to 6 553 km

F.3.3 SequenceNumberExitEntry

The datastructure SequenceNumberExitEntry specifies the sequence number of the exit or entry (e.g. exit 2 out of 5).

<SequenceNumberExitEntry()>:=	
<**IntUnTi**>(sequenceNumberExitEntry),	: SequenceNumber of exit vs total number of exits on one side of the road, or SequenceNumber of entry vs total number of entries on one side of the road. SequenceNumber shall start at 1 up to total number of exits, respectively entries. : First exit or entry encountered in driving direction shall be numbered 1. E.g. on one side of the road the following numbering can occur:Exit 1 Entry 1 Entry 2 Exit 2 Entry 3
<**IntUnTi**>(totalNumberOfExitEntries);	: Total number of exit or entries

F.3.4 TMCExitEntryLocationReference

The TMCExitEntryLocation reference datastructure specifies the applicable entries and/or exits of a TMC location where the content of the application content is located (e.g. to signal 'first exit closed', where the TMCExitEntryLocationReference references the 'first exit' and the application container signals the 'closed' event.

<TMCExitEntryLocationReference()>:=	
<**TMCLocationReference**>(tmcLocation),	: TMC location container indicating the main TMC location container indicating the main (junction) location for which the entries and/or exits are referenced in the datastructure exitEntryInformation.
<**IntUnLoMB**>(n),	
n * <**ExitEntryInformation**>(exitEntryInformation);	: The specific entries or exits of the main TMC location to be referenced.

F.3.5 WGS84Coordinates

The datastructure WGS84coordinates specifies a standard WGS84 coordinate pair with 24 bit resolution.

<WGS84Coordinates()>:=	
<**IntSi24**>(wgs84Longitude),	: Longitude in standard 24 bit accuracy stores co-ordinates in order of magnitude of 10 micro degrees resolution (five decimals). Longitude (unit degrees) is encoded as follows: WGS84longitude = int (sign(longitude)*0,5 + (longitude * (2^{24}))/360)
<**IntSi24**>(wgs84Latitude);	: WGS84 latitude in standard 24 bit accuracy stores co-ordinates in order of magnitude of 10 micro degrees resolution (five decimals). Latitude (unit degrees) is encoded as follows: WGS84latitude = int (sign(latitude)*0,5 + (latitude * (2^{24}))/ 360)

F.3.6 TMCLocationReference

The datastructure TMC Location Reference extends the ALERT-C Location referencing container to specify whether the TMC internal road segments of primary and secondary location are affected/part of the location or not affected/not part of the location to be referenced.

<TMCLocationReference()>:=	
<**IntUnLi**>(locationID),	: The actual location number.
<**IntUnTi**>(countryCode),	: Defines the different countries being responsible for the database creation. The table of country codes is specified in ISO 14819-3:2013.
<**IntUnTi**>(locationTableNumber),	: The number of the location table. The value is specified per country and may differ over time.
<**BitArray**>(selector),	
if (bit 0 of selector is set)	
<**Boolean**>(direction),	: The direction bit defines the direction from the primary to the secondary location. If the direction flag is true/1 it means positive direction i.e. following to the successor in the location list.
if (bit 1 of selector is set)	
<**Boolean**>(bothDirections),	: If set to true the location shall be treated as being applicable for both directions. NOTE In contrast to ISO 14819-1:2013 and ISO 14819-2:2013, which specifies a default directionality for each event code, in TPEG the directionality has to be set explicitly.
if (bit 2 of selector is set)	
<**IntUnTi**>(extent),	: Number of hops in the linked list of location codes. The direction is defined by the direction attribute. Absent value is defined 0 and means inside the current location.
if (bit 3 of selector is set)	
<**IntUnTi**>(extendedCountryCode),	: In non-European countries, the full range of location tables per country code can be used provided that the Extended Country Code associated with the ALERT-C location table is defined and provided (Extended Country Codes are defined in IEC 62106:2000). Even if this field is optional this field shall be sent mandatorily.
if (bit 4 of selector is set)	
<**IntUnLoMB**>(locationTableVersion),	: If 1 byte, 4 bits for major and 3 bits for minor. If 2 bytes 7 bits for major and 7 bits for minor. 3 bytes not possible because ALERT-C restricts to 0-99 for both. Letters in some old versions are interpreted as minor increments to the given number.
if (bit 5 of selector is set)	
<**PreciseTMCInformation**>(preciseTMCInfo);	: Precise ALERT-C Information follows.
if (bit 6 of selector is set)	

<**Boolean**>(useInternalPrimaryLocation),	: true/1 if the internal TMC through road section of the primary location is also affected. False/0 otherwise.
if (bit 7 of selector is set)	
<**Boolean**>(useInternalSecondaryLocation),	: true/1 if the internal TMC through road section of the secondary location is also affected. False/0 otherwise.

F.3.7 ExitEntryInformation

The datastructure ExitEntryInformation specifies detailed information as to which exits and/or entries are affected by the application contents.

<ExitEntryInformation()>:=	
<**BitArray**>(selector),	
if (bit 0 of selector is set)	
<**Boolean**>(isExitOrEntry),	: true or 1 for exit , false or 0 for entry.
if (bit 1 of selector is set)	
<**Boolean**>(oppositeDir),	: If true or 1, the entry or exit is on the road side opposite to the direction of the main TMC location reference (This is to be used if the main tmc location indicates 'both directions').
if (bit 2 of selector is set)	
<**DirLocation**>(dirLocation),	: The TMC location where the exit leads to, or entry originates from.
if (bit 3 of selector is set)	
<**WGS84Coordinates**>(exitEntryPoint),	: WGS84Coordinates of the specified exit entry. May be used advantageously if there is more than one Exit or Entry for the specified side and not all of them are closed.
if (bit 4 of selector is set)	
<**SequenceNumberExitEntry**>(sequenceNumberExitEntry);	

Annex G
(normative)

Extended TMC Location reference, TPEG2 XML representation

G.1 General

This annex defines the TPEG2 XML representation[18] of the ETL container for use with TPEG services.

G.2 ETL container components

G.2.1 ExtendedTMCLocationReference

```
<xs:element name="ExtendedTMCLocationReference" type="ExtendedTMCLocationReference"/>
<xs:complexType name="ExtendedTMCLocationReference">
 <xs:complexContent>
 <xs:extension base="tsf:ApplicationRootMessageML">
 <xs:sequence>
 <xs:element name="specVersionID" type="tdt:MajorMinorVersion"/>
 <xs:element name="tmcLocation" type="TMCLocationReference" minOccurs="0"/>
 <xs:element name="tmcExitEntryLocation" type="TMCExitEntryLocationRefere
nce" minOccurs="0"/>
 </xs:sequence>
 </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

G.3 Datatypes

G.3.1 DirLocation

```
<xs:complexType name="DirLocation">
 <xs:sequence>
 <xs:element name="dirLocationID" type="tdt:IntUnLi"/>
 <xs:element name="dirLocationDirection" type="tdt:Boolean"/>
 </xs:sequence>
</xs:complexType>
```

G.3.2 PreciseTMCInformation

```
<xs:complexType name="PreciseTMCInformation">
 <xs:sequence>
 <xs:element name="distanceAccuracy" type="tdt:IntUnTi" minOccurs="0"/>
 <xs:element name="hazardDistance1" type="tdt:IntUnTi" minOccurs="0"/>
 <xs:element name="hazardDistance2" type="tdt:IntUnLi" minOccurs="0"/>
 <xs:element name="problemLength1" type="tdt:IntUnTi" minOccurs="0"/>
 <xs:element name="problemLength2" type="tdt:IntUnLi" minOccurs="0"/>
 </xs:sequence>
</xs:complexType>
```

G.3.3 SequenceNumberExitEntry

```
<xs:complexType name="SequenceNumberExitEntry">
 <xs:sequence>
 <xs:element name="sequenceNumberExitEntry" type="tdt:IntUnTi"/>
 <xs:element name="totalNumberOfExitEntries" type="tdt:IntUnTi"/>
 </xs:sequence>
</xs:complexType>
```

G.3.4 TMCExitEntryLocationReference

```
<xs:complexType name="TMCExitEntryLocationReference">
 <xs:sequence>
 <xs:element name="tmcLocation" type="TMCLocationReference"/>
 <xs:element name="exitEntryInformation" type="ExitEntryInformation"
maxOccurs="unbounded"/>
 </xs:sequence>
</xs:complexType>
```

G.3.5 WGS84Coordinates

```
<xs:complexType name="WGS84Coordinates">
 <xs:sequence>
 <xs:element name="wgs84Longitude" type="tdt:IntSi24"/>
 <xs:element name="wgs84Latitude" type="tdt:IntSi24"/>
 </xs:sequence>
</xs:complexType>
```

G.3.6 TMCLocationReference

```
<xs:complexType name="TMCLocationReference">
 <xs:sequence>
 <xs:element name="locationID" type="tdt:IntUnLi"/>
 <xs:element name="countryCode" type="tdt:IntUnTi"/>
 <xs:element name="locationTableNumber" type="tdt:IntUnTi"/>
 <xs:element name="direction" type="tdt:Boolean"/>
 <xs:element name="bothDirections" type="tdt:Boolean"/>
 <xs:element name="extent" type="tdt:IntUnTi" minOccurs="0"/>
 <xs:element name="extendedCountryCode" type="tdt:IntUnTi" minOccurs="0"/>
 <xs:element name="locationTableVersion" type="tdt:IntUnLoMB" minOccurs="0"/>
 <xs:element name="preciseTMCInfo" type="PreciseTMCInformation" minOccurs="0"/>
 <xs:element name="useInternalPrimaryLocation" type="tdt:Boolean"/>
 <xs:element name="useInternalSecondaryLocation" type="tdt:Boolean"/>
 </xs:sequence>
</xs:complexType>
```

G.3.7 ExitEntryInformation

```
<xs:complexType name="ExitEntryInformation">
 <xs:sequence>
 <xs:element name="isExitOrEntry" type="tdt:Boolean"/>
 <xs:element name="oppositeDir" type="tdt:Boolean"/>
 <xs:element name="dirLocation" type="DirLocation" minOccurs="0"/>
 <xs:element name="exitEntryPoint" type="WGS84Coordinates" minOccurs="0"/>
 <xs:element name="sequenceNumberExitEntry" type="SequenceNumberExitEntry" minOccurs="0"/>
 </xs:sequence>
</xs:complexType>
```

G.4 Full ETL schema definition

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- This XML schema is generated with tpegUMLconverter V2.3 -->
<xs:schema xmlns="http://www.tisa.org/TPEG/ETL_1_0"
   targetNamespace="http://www.tisa.org/TPEG/ETL_1_0"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:tsf="http://www.tisa.org/TPEG/SFW_1_1"
   xmlns:tdt="http://www.tisa.org/TPEG/TPEGDataTypes_2_0"
   elementFormDefault="qualified"
   attributeFormDefault="qualified">
   <xs:import namespace="http://www.tisa.org/TPEG/SFW_1_1" schemaLocation="SFW_1_1.xsd"/>
   <xs:import namespace="http://www.tisa.org/TPEG/
TPEGDataTypes_2_0" schemaLocation="TDT_2_0.xsd"/>
   <xs:element name="ExtendedTMCLocationReference" type="ExtendedTMCLocationReference"/>
   <xs:complexType name="ExtendedTMCLocationReference">
      <xs:complexContent>
         <xs:extension base="tsf:ApplicationRootMessageML">
            <xs:sequence>
               <xs:element name="specVersionID" type="tdt:MajorMinorVersion"/>
               <xs:element name="tmcLocation" type="TMCLocationReference" minOccurs="0"/>
               <xs:element name="tmcExitEntryLocation" type="TMCExitEntryLocationRefere
```

```
nce" minOccurs="0"/>
            </xs:sequence>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
   <xs:complexType name="DirLocation">
      <xs:sequence>
         <xs:element name="dirLocationID" type="tdt:IntUnLi"/>
         <xs:element name="dirLocationDirection" type="tdt:Boolean"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="PreciseTMCInformation">
      <xs:sequence>
         <xs:element name="distanceAccuracy" type="tdt:IntUnTi" minOccurs="0"/>
         <xs:element name="hazardDistance1" type="tdt:IntUnTi" minOccurs="0"/>
         <xs:element name="hazardDistance2" type="tdt:IntUnLi" minOccurs="0"/>
         <xs:element name="problemLength1" type="tdt:IntUnTi" minOccurs="0"/>
         <xs:element name="problemLength2" type="tdt:IntUnLi" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="SequenceNumberExitEntry">
      <xs:sequence>
         <xs:element name="sequenceNumberExitEntry" type="tdt:IntUnTi"/>
         <xs:element name="totalNumberOfExitEntries" type="tdt:IntUnTi"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="TMCExitEntryLocationReference">
      <xs:sequence>
         <xs:element name="tmcLocation" type="TMCLocationReference"/>
         <xs:element name="exitEntryInformation" type="ExitEntryInformation"
maxOccurs="unbounded"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="WGS84Coordinates">
      <xs:sequence>
         <xs:element name="wgs84Longitude" type="tdt:IntSi24"/>
         <xs:element name="wgs84Latitude" type="tdt:IntSi24"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="TMCLocationReference">
      <xs:sequence>
         <xs:element name="locationID" type="tdt:IntUnLi"/>
         <xs:element name="countryCode" type="tdt:IntUnTi"/>
         <xs:element name="locationTableNumber" type="tdt:IntUnTi"/>
         <xs:element name="direction" type="tdt:Boolean"/>
         <xs:element name="bothDirections" type="tdt:Boolean"/>
         <xs:element name="extent" type="tdt:IntUnTi" minOccurs="0"/>
         <xs:element name="extendedCountryCode" type="tdt:IntUnTi" minOccurs="0"/>
         <xs:element name="locationTableVersion" type="tdt:IntUnLoMB" minOccurs="0"/>
         <xs:element name="preciseTMCInfo" type="PreciseTMCInformation" minOccurs="0"/>
         <xs:element name="useInternalPrimaryLocation" type="tdt:Boolean"/>
         <xs:element name="useInternalSecondaryLocation" type="tdt:Boolean"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="ExitEntryInformation">
      <xs:sequence>
         <xs:element name="isExitOrEntry" type="tdt:Boolean"/>
         <xs:element name="oppositeDir" type="tdt:Boolean"/>
         <xs:element name="dirLocation" type="DirLocation" minOccurs="0"/>
         <xs:element name="exitEntryPoint" type="WGS84Coordinates" minOccurs="0"/>
         <xs:element name="sequenceNumberExitEntry" type="SequenceNumberExitEntry"
minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
</xs:schema>
```