
**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)



PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

STANDARDSISO.COM : Click to view the full PDF of ISO 17572-2:2008



COPYRIGHT PROTECTED DOCUMENT

© ISO 2008

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword	iv
Introduction.....	v
1 Scope	1
2 Normative references	1
3 Terms and definitions	2
4 Abbreviated terms	2
5 Requirements for a location referencing standard	2
6 Conceptual data model for location referencing methods	3
7 Specification of pre-coded location references	3
7.1 General concept	3
7.2 Location database creation and updating	4
7.3 Location database provision.....	4
7.4 Location database usage.....	4
8 Implementations at present.....	4
8.1 Introduction.....	4
8.2 Traffic Message Channel (TMC) / Alert-C Specification	4
8.3 Vehicle Information and Communication System (VICS).....	5
8.4 Korean node link ID system	5
Annex A (informative) Logical format for VICS link location.....	7
Annex B (informative) TPEG physical format for ALERT-C-location references	26
Annex C (informative) TPEG physical format for Korean node-link ID references	30
Bibliography.....	33

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

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.

ISO 17572-2 was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*.

ISO 17572 consists of the following parts, under the general title *Intelligent transport systems (ITS) — Location referencing for geographic databases*:

- *Part 1: General requirements and conceptual model*
- *Part 2: Pre-coded location references (pre-coded profile)*
- *Part 3: Dynamic location references (dynamic profile)*

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 Location Reference 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, Location References are often coded. This is especially significant if the Location Reference 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 will be addressed. Amongst these, Location Referencing of the road network, or components thereof, is a particular focus.

Communication of a Location Reference for specific geographic phenomena, corresponding to objects in geographic databases, in a standard, unambiguous manner is a vital part of an integrated ITS system in which different applications and sources of geographic data will be 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 Location Referencing Method 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. 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 Location Referencing Methods.

This International Standard provides specifications for location referencing for ITS systems (although other committees or standardization bodies may subsequently consider extending it to a more generic context). In addition, this edition does not deal with public transport location referencing; this issue will be dealt with in a later edition.

STANDARDSISO.COM : Click to view the full PDF of ISO 17572-2:2008

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

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

1 Scope

This International Standard specifies Location Referencing Methods (LRM) that describe locations in the context of geographic databases and will be used to locate transport-related phenomena in an encoder system as well as in the decoder side. This International Standard 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.

This International Standard specifies two different LRMs:

- pre-coded location references (pre-coded profile);
- dynamic location references (dynamic profile).

This International Standard does not define a physical format for implementing the LRM. However, the requirements for physical formats are defined.

This part of ISO 17572 specifies the pre-coded location referencing method, comprising:

- specification of pre-coded location references (pre-coded profile);
- logical format for VICS link location;
- TPEG physical format for ALERT-C-location references;
- TPEG physical format for Korean node-link ID references.

It is consistent with other International Standards developed by ISO/TC 204 such as ISO 14825, *Intelligent transport systems — Geographic Data Files (GDF) — Overall data specification*.

2 Normative references

The following referenced documents are indispensable for the application 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.

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)
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
SOEI	System Operating and Exchanging Information
TMC	Traffic Message Channel
TPEG	Transport Protocol Expert Group
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 Location Referencing Methods, see ISO 17572-1:2008, Annex A.

6 Conceptual data model for location referencing methods

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 a location database (LDB) that is exactly the same as the corresponding location database used by a service provider of a particular message being exchanged. All pre-coded location referencing methods 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 location referencing method here is divided into three steps performed to implement the location referencing system. 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 (location database creation). In the second step, this database is provisioned via various means into the service providers database as well as into all receiving systems (location database 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 (location database usage). See Figure 1 illustrating this concept.

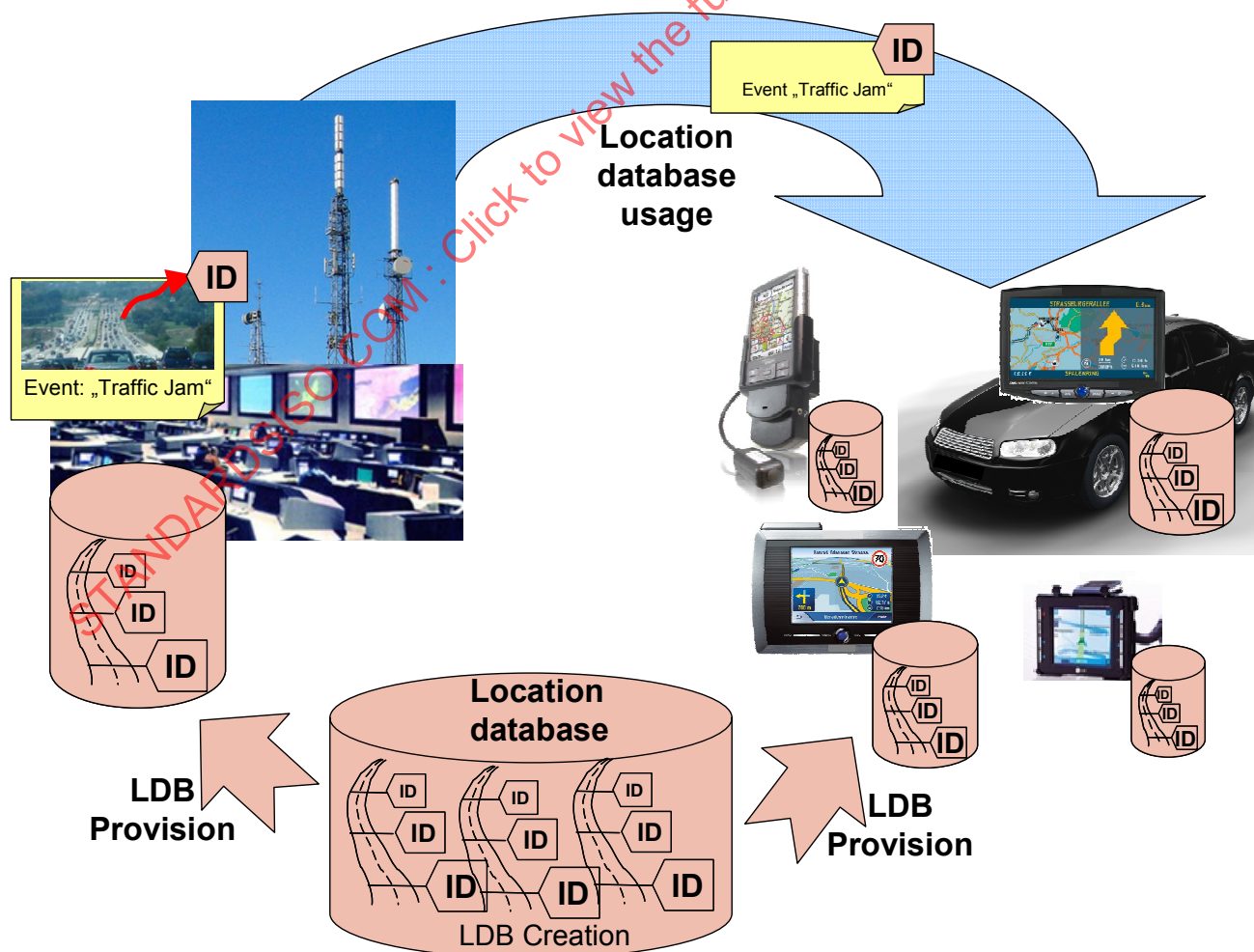


Figure 1 — General concept of pre-coded location referencing system

7.2 Location database creation and updating

The different location referencing systems more or less support standardized ways to create a new release of a location database. 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 location database.

7.3 Location database provision

After finalization of the creation process, the newly created location database is provisioned into the devices with maintenance service agreements. This is mostly done on a regular based map release update. The location referencing system 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 Location database usage

A service provider, using the current release dataset, now creates messages with location references according to specified rules a location reference 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 Introduction

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 location referencing methods and introduces them shortly. It also refers the reader to the different documents needed to fully apply to the different implementations.

8.2 Traffic Message Channel (TMC) / Alert-C Specification

8.2.1 General

The location referencing rules defined in ISO 14819-3 ^[11] address the specific requirements of Traffic Message Channel (TMC) systems, which use abbreviated coding formats to provide TTI messages over mobile bearers (e.g. GSM, DAB) or via exchange protocols like DATEX. In particular, the rules address the Radio Data System Traffic Message Channel (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].

8.2.2 Location database 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].

8.2.3 Location database usage

A location ID in such a message refers and serves as a tabular 'address' of the pre-stored location details in the location database 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 location database.

8.3 Vehicle Information and Communication System (VICS)

8.3.1 Location database creation

Vehicle Information and Communication System specifies in bibliography item [2] a digital map database as the basis for other map provider 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. See Figure 2 which defines the conceptual data model for this map.

8.3.2 Location database 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 bibliography item [1] defines how the digitized location IDs has to be coded to build up a more sophisticated location.

8.4 Korean node link ID system

8.4.1 General

The Ministry of Construction and Transportation (MOCT) of Korea has developed 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 Location database creation

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

8.4.3 Location database usage

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

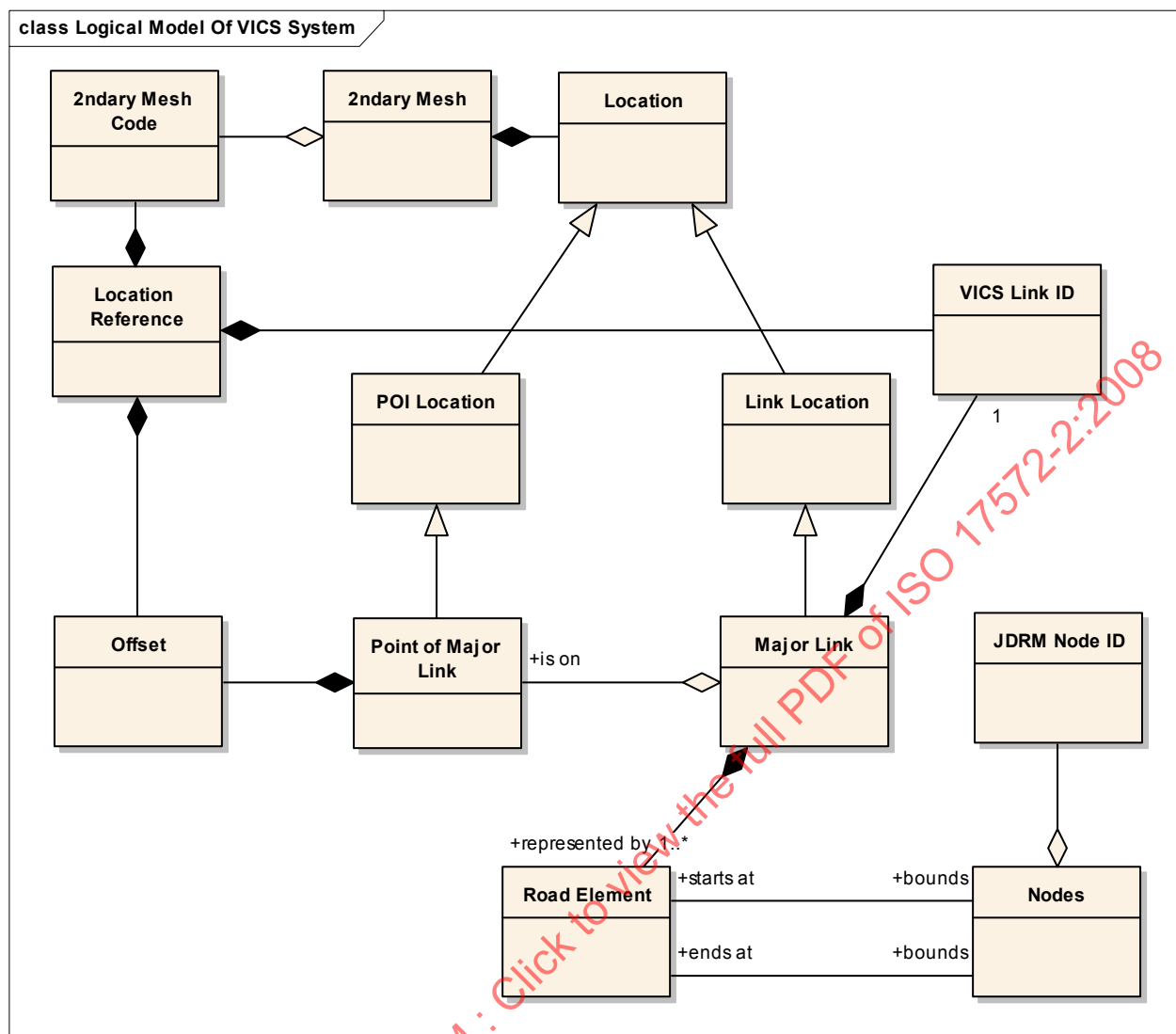


Figure 2 — Logical data model of VICS digital base map

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 Data structure helps to understand the concept. It consists of a Location Information (LI) Header and Location Content as shown in Figure A.1, with the latter further subdivided functionally into Coordinates, Descriptors and Offset information. The Figure A.2 describes the structure of the LI main in form of a UML Diagram. Clauses A.3 and A.4 do 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.

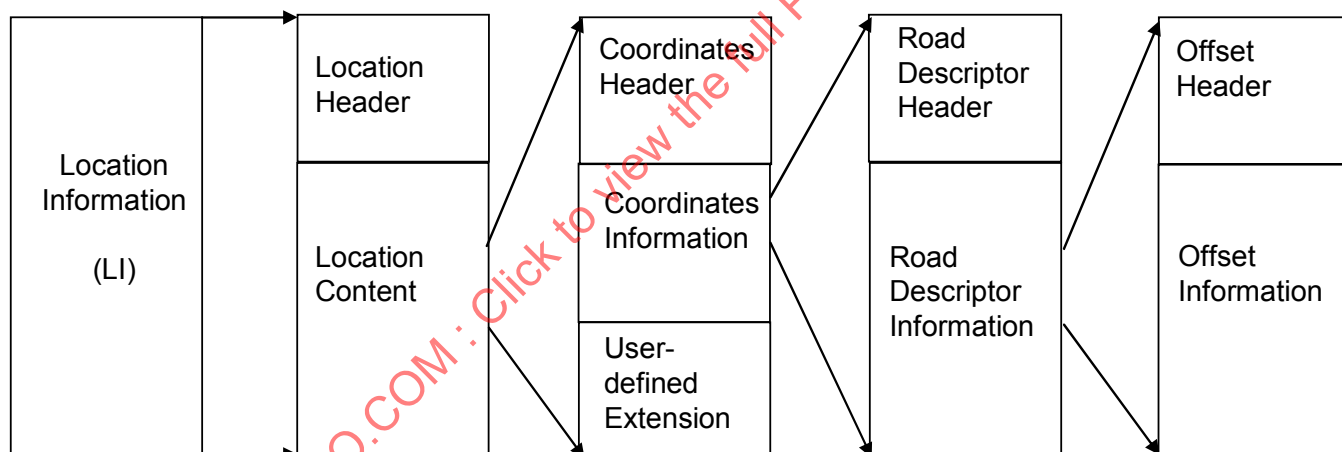


Figure A.1 — Outline diagram of logical structure

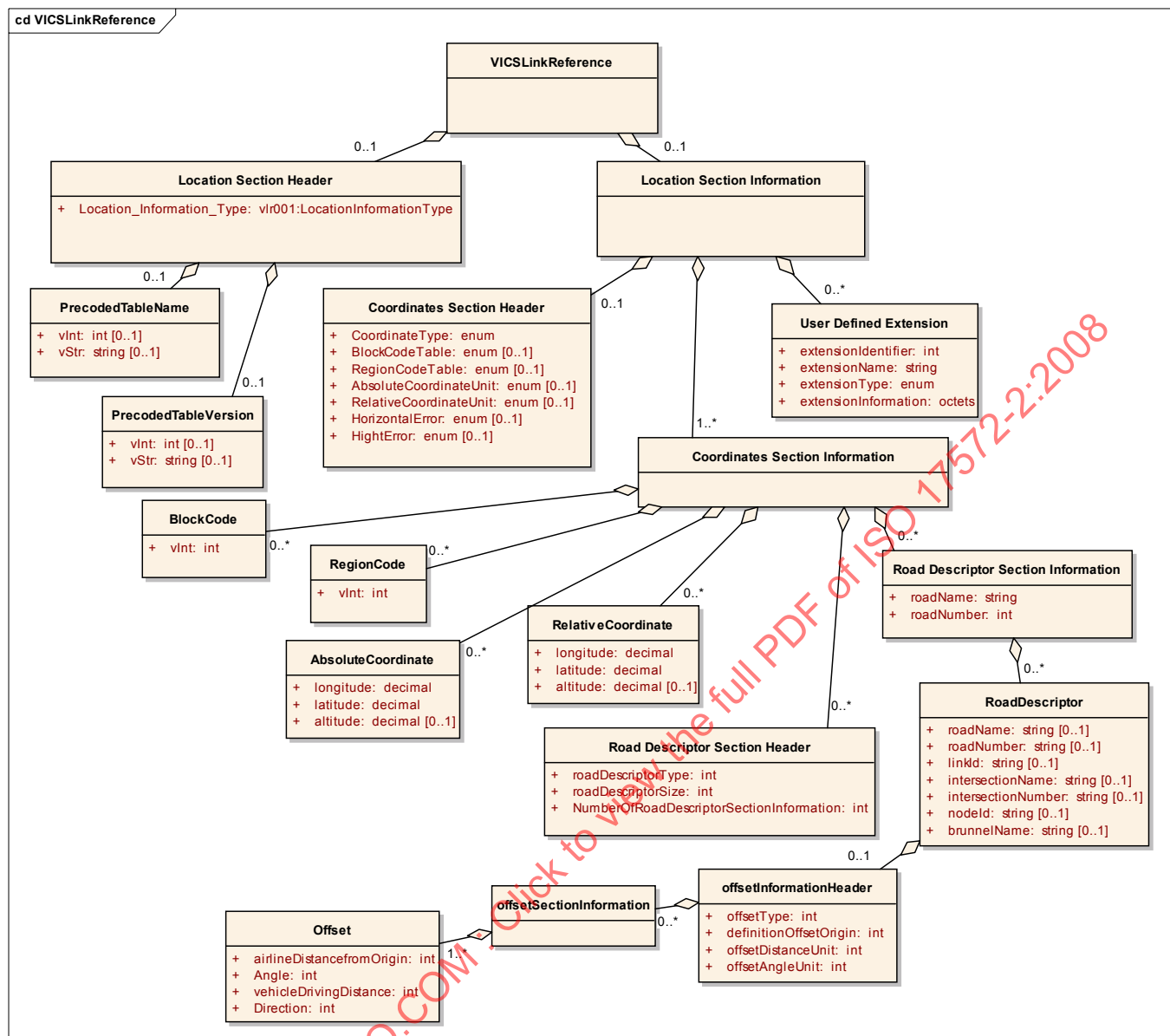


Figure A.2 — Outline diagram of logical structure in UML

A.1.2 Data values

Table A.1 lists all 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 1/2500~1/10,000 scale
basemap2	A parameter specifying that the location is digitized on a map of 1/25000~1/50,000 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 coordinates 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 one meter
error2	A parameter specifying that a height error is less than ten meters
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

<u>Data Element Name</u>	<u>Definition</u>	<u>Data Type</u>	<u>Valid Value Rule</u>
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}
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}

Table A.2 (continued)

<u>Data Element Name</u>	<u>Definition</u>	<u>Data Type</u>	<u>Valid Value Rule</u>
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)
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)

Table A.2 (continued)

<u>Data Element Name</u>	<u>Definition</u>	<u>Data Type</u>	<u>Valid Value Rule</u>
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 a meter or ten meters.	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)
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 expressway, 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

In Table A.3 “*” indicates that a Data Frame is repeatable; Data Frame ‘X’ occurs ‘Number_of_X’ times.

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

<u>Data Frame Name</u>	<u>Definition</u>	<u>Data Elements/Nested Frames</u>
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*
Coordinates_Section_Header	A Data Frame containing header information for coordinate information	User_Defined_Extension Coordinate_Type Block_Code_Table Region_Code_Table Absolute_Coordinate_Unit Relative_Coordinate_Unit Horizontal_Error Height_Error Number_of_Coordinates_Section_Information Number_of_Blocks Number_of_Regions Number_of_Absolute_Coordinates Number_of_Relative_Coordinates Road_Descriptor_Section_Header Road_Descriptor_Section_Information*
Coordinates_Section_Information*	A Data Frame containing block, region, absolute, and relative coordinate information	Block_Code Region_Code Latitude Longitude Altitude
Blocks*	A Data Frame containing Block_Codes	Block_Code
Regions*	A Data Frame containing Region_Codes	Region_Code
Absolute_Coordinates*	A Data Frame of ITRF geographic coordinates.	Latitude Longitude Altitude
Relative_Coordinates*	A Data Frame containing relative x, y, and z coordinate information.	Relative_X Relative_Y Relative_Z

Table A.3 (continued)

<u>Data Frame Name</u>	<u>Definition</u>	<u>Data Elements/Nested Frames</u>
Road_Descriptor_Section_Header	A Data Frame containing road descriptors header information.	Road_Descriptor_Type Road_Descriptor_Size
Road_Descriptor_Section_Information*	A Data Frame containing road descriptor section information	Number_of_Road_Descriptor_Section_Information Number_of_Road_Descriptors Road_Descriptor*
Road_Descriptor*	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*
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 Number_of_Offsets Offsets*
Offset_Section_Information*	A Data Frame containing offsets	Airline_Distance_from_Origin Angle Vehicle_Driving_Distance Direction
Offsets*	A Data Frame containing offset information on a distance from a base point to the feature	User_Defined_Extension_Header User_Defined_Extension_Information Extension_Identifier Extension_Name Extension_Type <user defined>
User_Defined_Extension	A Data Frame containing user defined extension information	
User_Defined_Extension_Header	A Data Frame containing header information for user defined information	
User_Defined_Extension_Information	A Data Frame containing data elements and/or nested data frames are user-defined and not specified	

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
intersectid	: 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 from Origin
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 ::=

procedure 1

BEGIN

EXPORTS

Lrp1-linf;

IMPORTS

Precoded-Table, Blockcode-Table;

--Lrp1-linf: Location information of Location referencing

Lrp1-linf ::= SEQUENCE {			
lsheader	Lsheader	OPTIONAL,	-Location_Section_Header
lsinf	Lsinf		-Location_Section_Information
}			
Lsheader ::= 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,
    ddmms,
    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      (-900000000..900000000), -microdegrees
    lat_real     REAL         (-90..90)                -degrees
}
```

```
Longitude ::= CHOICE {
    lon_int      INTEGER      (-1800000000..1800000000), -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
rdsiz 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 UDEExtHeader OPTIONAL, -User-Defined Extension Header
 uDExtinf UDExtinf -User-Defined Extension Information
 }

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

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

 UDExtinf ::= SEQUENCE {
 byte BYTE -user-defined set of data elements and/or nested data frames
 }
 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="Linf">
    <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:sequence>

```

```

        <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="rdsiz" 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="UDEExtHeader" minOccurs="0"/>
    <xsd:element name="UDEExtinf" type="UDEExtinf"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="UDEExtHeader">
  <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)

TPEG physical format for ALERT-C-location references

B.1 Introduction

This annex describes relevant parts for the definition of the encoding format for this part. It refers to the TPEG specifications and ISO 17572-1:2008, Annex E, in terms of the names of definitions. Specifically the naming of basic types and syntax description is intended to be changed in SSF, which makes it necessary to define the attributes being used here to be internally consistent.

TPEG intends to define a logical level of data structure definition in form of UML-Data Diagrams and to define rules, which allow deriving automatically the resulting formats in XSD and binary. This ensures that XML-messages on their way through the service chain can be converted to binary messages and vice versa. The following clause will define the container for Pre-Coded ALERT-C location referencing named TMCLocationReference

B.2 Data model of the TMC location reference container

B.2.1 Introduction

Although it is intended to replace RDS-TMC by new services like TPEG Automotive Profile (TAP), in the meantime easy adoption and migration into existing systems is needed. In addition to that reuse of potential functionality from existing solutions is intended to be made possible. Whereas e.g. TPEG Applications may replace the former ALERT-C method with its inheriting event lists, in terms of location referencing, it seems also to be a great potential to allow continuation of using ALERT-C-Location tables already in place, in addition to the new dynamic LRM. This gives the opportunity to code either very efficiently already existing 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.2.2 Pre-coded ALERT-C location referencing

B.3 defines the data structure of ALERT-C location referencing to be used in pre-coded LRC container. The method of ALERT-C location referencing is defined in the ALERT-C Standard, ISO 14819-3 [11]. 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 location referencing method, a precise location referencing method 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. This attributes are made optional because not every location will want make use of it. In the section the sub elements defined for ALERT-C Location are specified.

Another amendment to the original standard is that versioning of different location tables have been harmonized. The TMC Forum issues the versioning scheme as being separated into a major version and a minor version which further are specified in a TMC Forum internal specification. The version is introduced as a mandatory value here, to encourage future encoders to provide the database in use. However, for backward compatibility reasons the value major 0 and minor 0 is implied as being undefined (or not given) version number.