INTERNATIONAL STANDARD

ISO 17572-1

First edition
2008-12-15

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

Part 1
General requirements and conceptual model

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

Partie 1: Exigences générales et modèle conceptuel

Reference number
ISO 17572-1:2008(E)

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

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 17572-1 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.

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

Part 1: General requirements and conceptual model

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 International Standard does not define details of the Location Referencing System (LRS), i.e. how the LRMs are to be implemented in software, hardware, or processes.

This part of ISO 17572 specifies the following general LRM related sections:

— requirements of a Location Referencing Method;

— conceptual Data Model for Location Referencing Methods;

— inventory of Location Referencing Methods;

— examples of Conceptual Data Model Use;

— description of selected UML Elements;

— comparison of Definitions with ISO/TC 211;

— introduction to the TPEG Physical Format.

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 Terms and definitions

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

2.1 General terms [1)]

2.1.1
accuracy
measure of closeness of results of observations, computations or estimates to the true values or the values accepted as being true

2.1.2
area
two-dimensional, geographical region on the surface of the earth

NOTE An area can be represented as an implicit area or an explicit area.

2.1.3
area location
two-dimensional location, representing a geographical region on the surface of the earth

2.1.4
attribute
characteristic property of an entity like a real-world feature

NOTE It allows the identification of that feature by the sum of its attributes. An attribute has a defined type and contains a value. Attributes can be either simple, i.e. consisting of one atomic value, or composite (see composite attribute).

2.1.5
coordinate
one of an ordered set of N numbers designating the position of a point in N-dimensional space

2.1.6
complex intersection
intersection that consists at least of two or more junctions and one or more road elements

2.1.7
composite attribute
complex attribute
attribute consisting of two or more atomic values and/or attributes

2.1.8
datum
set of parameters and control points used to accurately define the three-dimensional shape of the earth

NOTE The corresponding datum is the basis for a planar coordinate reference system.

2.1.9
descriptor
characteristic of a geographic object, usually stored in an attribute

EXAMPLE Road names or road numbers.

---

1) As part of the general intent to harmonize this International Standard with the ISO/TC 211 family of Geographic Information Systems standards, a comparison of terms and definitions between this International Standard and ISO/TC 211 standards is included as Annex D.

2.1.10
digital map database
structured set of digital and alphanumeric data portraying geographic locations and relationships of spatial features

NOTE Typically, such structures represent, but are not limited to, the digital form of hard copy maps. For example, drawings may be imported into a Geographic Information System (GIS) and considered as a form of digital map.

2.1.11
dynamic location reference
location reference generated on-the-fly based on geographic properties in a digital map database

2.1.12
explicit area
two-dimensional face on the surface of the earth, with a specified outline either being a simple geometric figure or an irregular outline/polygon

2.1.13
face
two-dimensional element bounded by a closed sequence of edges not intersecting themselves

NOTE The face is the atomic two-dimensional element.

2.1.14
implicit area
selection of road segments to be referenced belonging to a certain area (subnetwork)

NOTE One implicit area can be built up of multiple subnetworks that are geographically connected.

2.1.15
international terrestrial reference frame
ITRF
realization of the ITRS

NOTE The ITRF94 reference frame is consistent with WGS84 at the 5 cm level, and therefore is equivalent to WGS84 for ITS applications.

2.1.16
international terrestrial reference system
ITRS
reference system for the earth derived from precise and accurate space geodesy measurements, not restricted to GPS Doppler measurements, which is periodically tracked and revised by the international earth rotation service

2.1.17
intersection
crossing and/or connection of two or more roads

NOTE 1 In GDF, an intersection is a Level 2 representation of a junction which bounds a road or a ferry. It is a complex feature, composed of one or more Level 1 junctions, road elements and enclosed traffic areas. The definition is different from GDF because the location referencing system refers to real-world objects rather than a database definition as defined in GDF.

NOTE 2 Crossings can be at-grade or grade-separated. Crossings that are grade-separated where no connection between the road segments exist, are excluded from this definition.

2.1.18
junction
elementary element in the road network, connecting two or more road elements

NOTE In GDF terms, it is a Level 1 feature that bounds a road element or ferry connection. Junctions that represent real crossings are at least trivalent (having three roads connected). A bivalent junction may only be defined in case an attribute change occurs along the road (e.g. road name change). A junction is also coded at the end of a dead-end road, to terminate it.

2.1.19
linear location
location that has a one-dimensional character

EXAMPLE A road segment.

2.1.20
link
edge
direct topological connection between two nodes that has a unique link id in a given digital map database

NOTE A link may contain additional intermediate coordinates (shape points) to better represent the shape of curved features. A link may be directed or undirected.

2.1.21
link identifier
link id
identifier that is uniquely assigned to a link

NOTE A link identifier may be arbitrary or may be assigned by convention, to assure that no multiple occurrences of the same identifier will be used within one instance of a network or map database.

2.1.22
link location
location identifiable by a part of the road network database having one identifier or having a uniquely identifiable combination of attributes throughout the continuous stretch

NOTE One link location can consist of multiple links.

2.1.23
location
simple or compound geographic object to be referenced by a location reference

NOTE A location is matched to database objects by location definitions, which specify what is meant by a particular Location. Without any explicit remark it is meant to be a linear stretch in terms of topology in the database network without any loops or discontinuities in between (linear location). It might also be only a point in the network as a specialization of a linear stretch with length zero. In addition to that, a location can also be a set of road elements representing an area. This area is e.g. expressible by a polygon or a list of linear locations. For further description of different categories of locations, refer to 5.4.

2.1.24
location definition
actual delineation of exactly what is meant (and, therefore, what is not meant) by a particular location within a specific database

NOTE It is the precise location definition of the database object, or set of database objects, which is referenced.

EXAMPLE The GDF road elements that make up a particular instance of an ALERT-C Location.

2.1.25
location reference
reference
label which is assigned to a location

NOTE With a single LRM, one reference shall define unambiguously and exactly one location in the location referencing system. The reference is the string of data which is passed between different implementations of a location referencing system to identify the location.

2.1.26
location referencing method
LRM
methodology of assigning location references to locations

2.1.27
location referencing system
LRS
complete system by which location references are generated, according to a location referencing method, and communicated, including standards, definitions, software, hardware, and databases

2.1.28
matching
translating a location reference to a specific object in a given map database to attempt recognition of the same identified object in both the sender's and the receiver's map database

NOTE Matching is seen as a subsequent part to the method of decoding a location reference adhering to the defined LRM.

2.1.29
node
zero-dimensional element that is a topological junction of two or more edges, or an end point of an edge

NOTE A node is created for topologically significant points, such as simple intersections of roads or other linear features including boundaries but also for locations such as electric beacons, kilometre-posts or sensors detecting traffic flows, being significant points specified in a map.

2.1.30
node identifier
identifier assigned to a node

NOTE A node identifier may be arbitrary, or may be assigned by convention, to ensure that multiple occurrences of the same identifier will not occur within one network or within the universe of similar networks or databases.

2.1.31
outlined area
explicit area with an outline defined by segments being either polylines or linear locations

2.1.32
point
zero-dimensional element that specifies geometric location

NOTE One coordinate pair or triplet specifies the location.

2.1.33
point location
location that has a zero-dimensional character

EXAMPLE A simple crossing.

2.1.34
precision
exactness of the measurement of a data value, or of the storage allocated to a measured data value

NOTE Alternatively, the closeness of measurements of the same phenomenon repeated under exactly the same conditions and using the same techniques.

2.1.35
pre-coded location reference
location reference using a unique identifier that is agreed upon in both sender and receiver system to select a location from a set of pre-coded locations

2.1.36
quad tree
hierarchical data structure which on a next lower level subdivides a given area into four quadrants of the same size where any level has knowledge of its four sublevels and its parent level

2.1.37
relationship
semantic or topological interrelation or dependency between locations in the LRS

NOTE Relationships can exist between locations in the LRS. These relationships will generally be structured to allow more sophisticated use of the location reference, such as a topological or hierarchical structure. For example, a county location may be defined as an aggregate of several city locations or a long stretch of road may be an aggregate of several smaller road segments. Referencing the county may be easier than referencing all the cities which make up the county. This allows scalability and ease of use in the LRSs using the LRM.

2.1.38
resolution
smallest unit which can be represented fixing a limit to precision and accuracy

2.1.39
road
part of the road network which is generally considered as a whole and which can be addressed by a single identification like a road name or road number throughout

NOTE 1 In general, it is a connection within the road network, with or without crossings, which functionally can be considered as a unity. A road with multiple (associated) carriageways can be considered as one road. (Note that, in the context of this part of ISO 17572, the term also covers the natural language term street).

NOTE 2 The subsequent parts of this International Standard intentionally do not make direct use of this term because under different circumstances it may be not possible to define exactly where a road ends. For this reason, reference will be made to artificial but more precisely definable road elements or road sections of the road network.

2.1.40
road crossing
location where two or more roads connect or intersect

NOTE A road crossing may be 'simple', corresponding to one junction, or 'complex', including internal road elements and junctions.

2.1.41
road element
linear section of the road network which is designed for vehicular movement having a junction at each end

NOTE It serves as the smallest unit of the road network at GDF Level 1 that is independent.

2.1.42
road section
road segment that is bounded by two intersections and has the same attributes throughout

NOTE Generally the two intersections are different, only in some specific cases are the intersections the same, e.g. a tear-drop street or slip roads inside of complex intersections.

2.1.43
road segment
part of a road, having its start and end along that road

NOTE Important difference between a road section and road segment is that the segment does not necessarily end at intersections.

2.1.44
shape point
intermediate coordinate pair to represent the shape of curved features

2.1.45
simple geometric area
explicit area with an outline defined by a simple geometric figure

2.1.46
simple object access protocol
SOAP
protocol providing a platform-independent way for applications to communicate with each other over the internet

NOTE SOAP technology relies on XML to define the format of the information and then adds the necessary HTTP headers to send it. Standardization is done within IETF: http://www.ietf.org/rfc.

2.1.47
subnetwork
plurality of road segments lying in geographical or topological conjunction to each other

2.1.48
synchronisation markup language
SyncML
data synchronisation protocol

NOTE A data synchronization protocol defines the workflow for communication during a data synchronization session when the mobile device is connected to the network. The protocol supports naming and identification of records, common protocol commands to synchronize local and network data, and it can support identification and resolution of synchronization conflicts.

2.1.49
topology
properties of spatial configuration invariant under continuous transformation

NOTE In a digital map database this means the logical relationships among map features. It can be used to characterize spatial relationships such as connectivity and adjacency.

2.1.50
world geodetic system of 1984
WGS84
earth-centred global reference frame, including an earth model, based on satellite and terrestrial data

NOTE It contains primary parameters that define the shape, angular velocity, and the earth mass of an earth ellipsoid, and secondary parameters that define a gravity model of the earth. Primary parameters are used to derive latitude-longitude coordinates (horizontal datum).

2.2 UML expressions for diagrams

This International Standard uses UML to express specific circumstances. As such, the graphical elements are used to express specific constraints and structural relationships. A full definition can be found in the UML Standard ISO/IEC 19501. However, a short introduction of used elements is given in Annex C.

3 Abbreviated terms

AGORA	Name of a European project 2000-2002 implementAtion of Global lOcation Referencing Approach
ALERT-C	Advice and problem Location for European Road Traffic-Compact
CAD	Computer Aided Design
EVIDENCE	name of a European project 1998-1999 Extensive Validation of IDENtification Concepts in Europe
GDF	Geographic Data File
GIS	Geographic Information System
GPS	Global Positioning System
IETF	Internet Engineering Task Force
ILOC	Intersection LOCation
ITS	Intelligent Transport Systems
LR	Location Referencing (or Reference)
LRC	Location Reference Container
POI	Point Of Interest
RDS	Radio Data System
TPEG	Transport Protocol Expert Group
TMC	Traffic Message Channel
TTI	Traffic and Traveller Information
UML	Unified Modeling Language
UTM	Universal Transverse Mercator
VICS	Vehicle Information and Communication System

4 Objectives and requirements for a location referencing method

4.1 Objectives for an optimal location referencing method

ITS applications have different objectives regarding location referencing, which from their contradictory nature, cannot be fulfilled completely. In theory a best location referencing method would require every LRS to have at a given time the same, completely accurate map and all locations would be identifiable without any additional computational effort. Even though this is not achievable, the following goals should guide the definition and optimization of a location referencing method. The circumstances of the specific location referencing system may give different weight to the following goals:

The first goal therefore states that processing power in any case is a cost factor to be minimized.

O-1. The LRM should be simple enough to be implemented in a resource and performance efficient way.

Secondly, location referencing implies at least two systems communicating with each other. Communication also causes costs and therefore needs to be minimized.

O-2. The LRM should not unduly add to the volume of data to be transferred.

The aim to use the exact location, both in the sender and the receiver system, is the reason of referring to it. In many cases it will be up to the receiver to decode the location reference as well as possible. To help the receiver to do so it shall be implied that the sending system sends the location reference as accurately as possible.

O-3. The LRM should provide location references with the highest accuracy possible.

4.2 Requirements of the location referencing method

In addition to the goals, some minimal requirements shall make the different location referencing methods feasible for the foreseen categories of locations (see clause 5.4).

One of the most important data characteristics for ITS applications is spatial accuracy. Spatial accuracy is an aspect of data quality and is described in GDF in the following way: the shape of a level 0 edge including all positions on the segment as a whole shall not have any position that diverges from the real shape more than an allowed error. Spatial data accuracy requirements for ITS vary according to the application. This means not just categories of applications, but how an application works operationally. Some applications, notably those for advanced vehicle safety systems, would require very accurate data. Even within a particular application, requirements for different levels of data accuracy may exist, and will be subject to change as applications and products evolve. The spatial data accuracy requirements impact on the location referencing method chosen for an application.

One fundamental requirement across all methods is that, whichever method is used, use of that method does not result in additional spatial location error beyond that already present in the data. However, for the location referencing of area information, for example weather information or information specifying a zone of environmental contamination, some positional error is permitted due to the imprecise ("fuzzy") nature of such information. The key requirement for such references is that they be made with sufficient precision to allow the user to avoid the area or take other appropriate action. Specifically, it is a requirement that:

R-1. An LRM and the process of its operation shall not introduce a supplementary position error relative to that specified in the database from which the reference is generated. The location reference should be conveyed with spatial and temporal accuracy sufficient to enable the vehicle or user to identify the (spatial) extent of the location.

For certain locations, knowing the side of the road or block on which the location is found can be very important to the user. For example, crossing a road to a location on the opposite side may not be possible for vehicles or pedestrians in some road lane configurations without additional routing, whereas turning into a location on the same side may be easily accomplished. Therefore it is a particular requirement that:

R-2. The LRM shall enable referencing of the relative spatial relationship of objects.

In addition to spatial accuracy requirements, location referencing methods have functional requirements regarding topological relationships, for example that a point is on one side of an object or that many points shall be ordered in a certain way along a road. For example, for locations referenced by positions along a logical or physical route, the ordering of points shall be preserved by the reference. For location referencing the requirement is that:

R-3. The LRM shall not change the topological relationships within a set of point data by its own action, limitation, or deficiency. For referencing by geographic coordinates, spatial relationships between locations should not be confused by lack of precision or by any other attribute of the referencing system or its operation.

This International Standard is foreseen to be used in a variety of Location Referencing Systems. However, the restriction to some specified categories of locations is understood as a first step of optimisation of the defined Location Referencing Methods. For this it shall at least hold:

R-4. The definition of an LRM shall adhere to the common terminology and conceptual model defined in this part of ISO 17572.

R-5. The LRM shall provide means to refer only to the categories of Locations explicitly defined by that LRM.

R-6. The LRM shall in principle allow addressing of every location on the road network.

Location referencing is an important technology for traffic telematics applications in particular and in general for any location based service. It provides an understandable reference for a location about which information needs to be provided. Understandable generally means machine-readable, i.e. the location information (as well as the message content) can be understood by equipment that translates the complete message to human understandable information e.g. position on a map or a descriptive way to reach the location. Therefore:

R-1. The location reference shall be machine-readable.

5 Conceptual data model for location referencing methods

5.1 Role of conceptual model

The conceptual model provides a framework to describe and define an LRM in generic, conceptual terms. The conceptual model is generally valid, i.e. not limited to the LRMs defined in this part of ISO 17572. Therefore, examples of other LRMs are mentioned, to illustrate the underlying conceptual view. See also Annex B.

5.2 Components of conceptual model

All location referencing methods have some form of the following components. The detailed definition of the terms is left to Clause 2.

Attributes
An attribute allows the LRS to process or evaluate the information about the location. Attributes discriminate the reference in such way that the receiver system can identify the location correctly.

Location
A location is a part of the road network that is intended to be identified. The sender system aims to refer to it, the receiver system aims to find it in its map database.

Location Definition
The location definition is the defined delineation in a digital map database of exactly what is meant. The location definition in a sender system may be different from the one in a receiver system even that the location is the same.

Reference
The reference is the label which is assigned to the location.

Relationship
Relationships to other locations in the LRS are also used to support the use of locations in an LRM.

NOTE Not all LRM require relationships and attributes. For example, an LRS which uses arbitrary numbers assigned to each location might never need to define how any of the locations relate to each other. However, in an LRS which meets the requirements which have been identified (such as flexibility, extensibility, compactness, etc.), relationships may be inherent in the referencing procedures. For example, pre-coded locations used in RDS-TMC systems arbitrarily number locations, but since the referencing system allows the use of an “extent,” each location may carry the definitions of which location is “before” and “after” it.

5.3 Description of the conceptual model

A reference is the label which is assigned to a location. In a single LRS, a reference shall be defined or definable which unambiguously identifies each location in this LRS. A location is a simple or compound geographic object, which is matched to database objects by location definitions, which specify what is meant by the particular location. A location can aggregate different attributes, which allow for the identification of the methods to process or evaluate the information about the location. Relationships associate different locations e.g. to allow more sophisticated use of the location reference, such as a topological, or hierarchical structure, they are intended to reflect the possibility of two or more Locations in a relationship, either ordered or unordered.

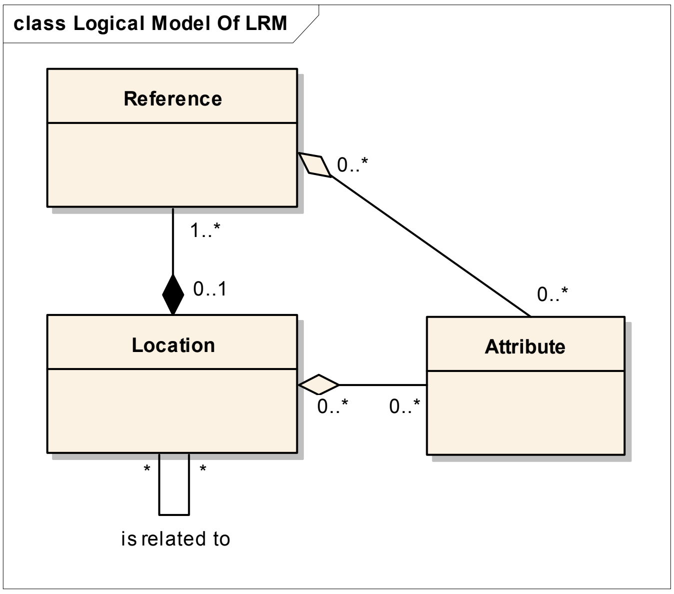


Figure 1 — Diagram of conceptual model of LRM

5.4 Location categories

Locations shall be categorised as Point Locations, Linear Locations and Area Locations. These Location Categories represent real world objects which can be described as follows:

— Existing at a single position (Point Location)

— Between two positions within intersections as road section (Linear Location, in terms of one or more Link Locations)

— Between two positions as road segment (Linear Location)

— Consisting of two or more Link Locations (Linear Location)

— A selection of road segments belonging to a defined collection (subnetwork or collection of subnetworks) (Implicit Area Location)

— Within the boundaries of a defined area (Explicit Area Location)

Point Locations can be described as existing at a single position. Point Locations include, for example, points of interest, public service facilities, commercial establishments, etc. Link Locations are linear objects bound by two Point Locations. Linear Locations are two or more consecutive Link Locations, bounded therefore by 3 or more consecutive Point Locations that define a connected linear stretch in the road network. Implicit Area Locations are more than one Linear Location of a certain area put together in one package. Explicit Area Locations are two-dimensional features such as governmental administrative area, postal district, telephone exchange district, etc. or just defined outlines as faces at a given place on the map.

Specific reference Object Instance Classes within these Categories are:

— General points - points that may or may not lie on a road network, including points where a road crosses administrative boundaries or borders of map grid cells

— Points at nodes in a topological network representation of roads and their intersections

— Links defined by two consecutive intersections of roads (road sections)

- Points along links bounded by intersections of roads
- Manoeuvres defined by two consecutive links (therefore, three intersections)
- Areas defined by a sequence of Points
- Areas defined by a sequence of link locations
- Areas defined by an origin point and attributes such as the radius of a circle around the point or offsets defining a bounding box

Generally, important location categories for ITS databases are man made structures like road crossings, road sections or segments, as well as sequences of road sections necessary for describing manoeuvres. Location categories can be arranged in class/type hierarchies to aid in decoding between dissimilar receiver / sender systems.

5.5 Conceptual model of road network

One purpose of Location Referencing is to refer to parts of the road network. The conceptual model of the road network is therefore depicted in Figure 2 and described here to give a clear understanding of the different terms and their relationships. This is especially needed, because the sophisticated definition of roads and intersections in GDF does not meet the requirements of a conceptual model for location referencing.

In general the road network consists of roads. A road is generally represented by a one name (or number) throughout as a whole and consists of a set of road sections. On a road an indefinite number of road segments may be defined (and referenced). A road section consists of nodes and edges, is bounded by intersections and may have intermediate intersections (where the road name does not change). An intersection is a connection or crossing of roads. The simplest intersection consists of just one node (i.e. junction). If an intersection has two or more nodes and one or more edges, it is considered to be a complex intersection.

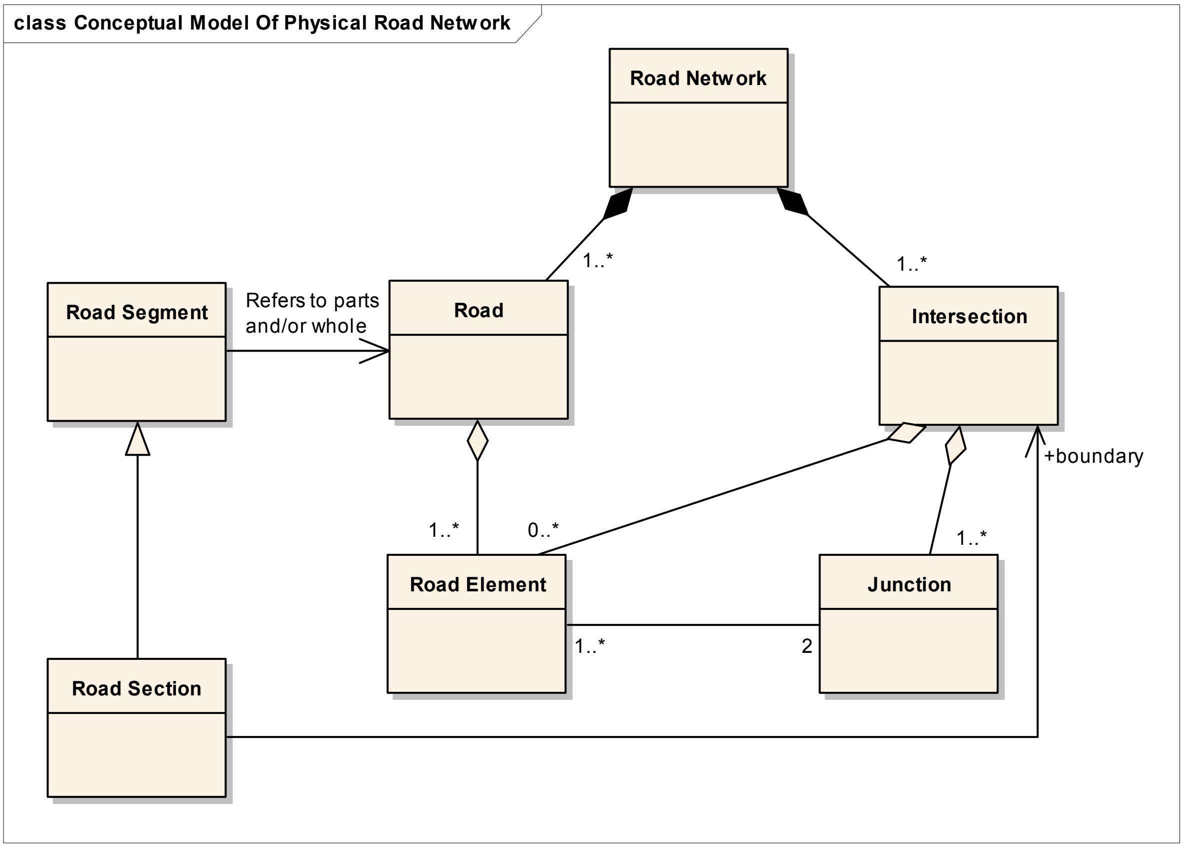


Figure 2 — Conceptual model for the physical road network

5.6 Conceptual model of area locations

An area is considered to be a two-dimensional, geographical region bounded on the surface of the Earth. Area locations have specific constraints on location referencing due to requirements of the application. The different requirements of the applications determine if the full geographical figure shall be mentioned or a defined number of roads inside this region. To enable an LRM handling and defining rules for explicit and implicit area types, the conceptual model defines terms for it and describes conceptually their containment.

In one case it might be feasible to define the area by a geometric figure (explicitly). In other cases it may be necessary to select a list of road elements spanning an area (implicitly).

An implicit area consists of one or more subnetwork locations which each consist of at least two road segments being aggregated to the subnetwork. The explicit area specifies a part of the earth surface being the area by means of a geometric (regular) function is defined as region, meanwhile an area with a shape freely defining the shape of it is called outline. Both of the area types have at least one referenced connection to the surface allowing the decoding system to precisely position the given area.

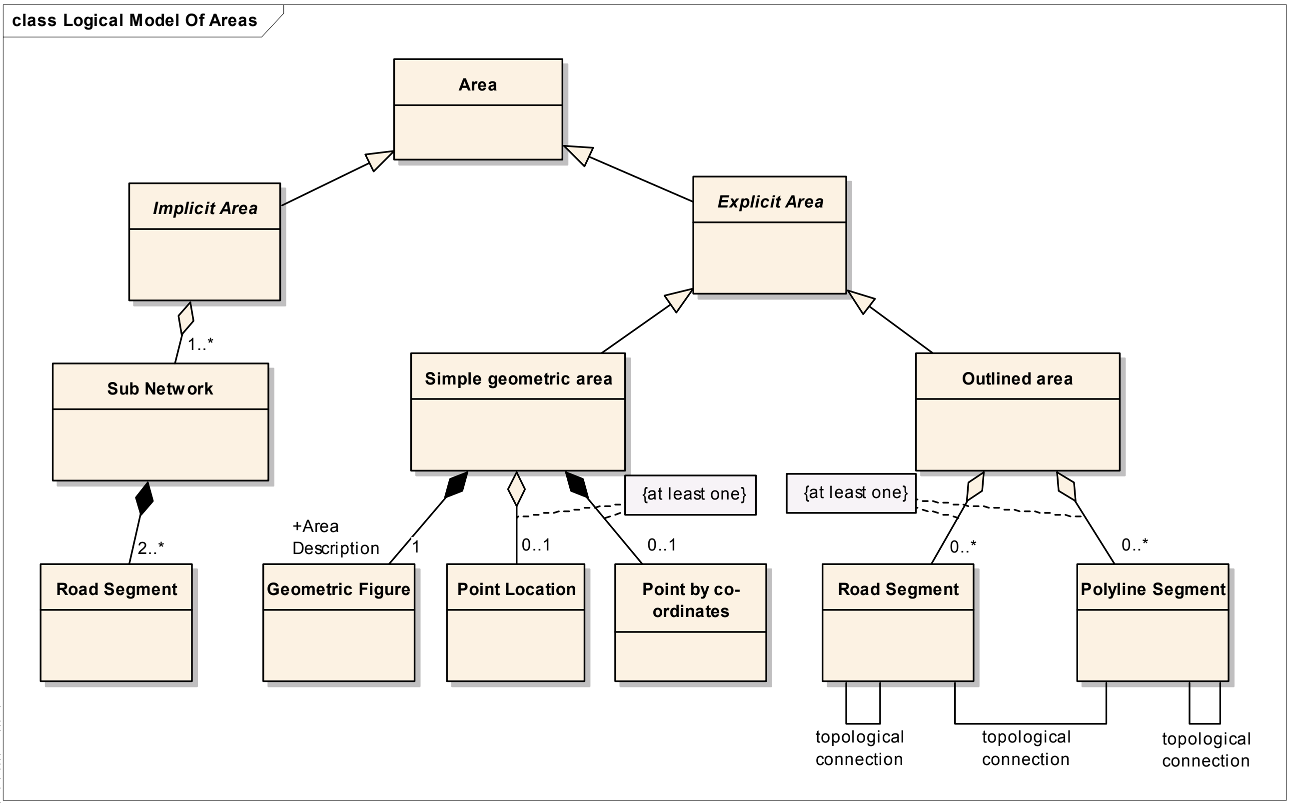


Figure 3 — Conceptual model of areas

Annex A
(informative)

Inventory of location referencing methods

A.1 General

There are two essentially different ways of referencing road locations: with reference to attributes of the road network itself, including linear distances or street addresses, or with reference to a regular or irregular division of space which exists independent of the representation of the road network. The former kind of reference is sometimes called an indirect reference. The latter kind involves coordinate systems, for example US UTM coordinates or geographic latitude/longitude coordinates.

Some indirect referencing methods stress the topology of networks (for example, Link ID and Linear Referencing), and some stress the attributes of the features that comprise the network (for example, Cross Street Offset Matching and street addresses). Geographic coordinates express location directly in terms of the geodetic reference framework itself; they are simply measurements on the framework, although they can be expressed in terms of continuous vector fields (e.g. longitude/latitude) or as quantized, regular subdivisions (e.g. quad trees). Figure A.1 illustrates indirect (specifically the Link-ID method) and coordinate referencing methods (specifically the method of Geographic Coordinates).

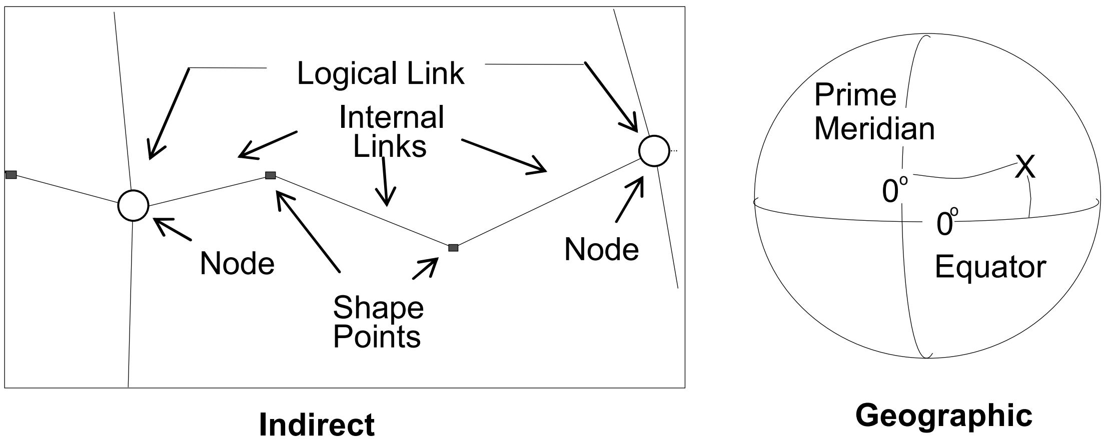


Figure A.1 — Indirect (Link-ID) vs. Geographic Coordinate Schemes

The following list summarizes location referencing methods in use today, which will be discussed briefly in the following sections:

- Referencing by link ID or node ID
- Referencing by geographic coordinates
- Referencing by grid
- Linear referencing
- Referencing by cross-streets
- Referencing by address information
- Combinations

A.2 Referencing by link ID or node ID

Logical links represent topological connectivity between nodes corresponding to real world points like intersections. The real world network may be represented within the data set by a planar or non-planar graph. The difference is that containing links do or do not cross without nodes. Shape points are typically inserted between nodes to represent road geometry between intersections. A link identifier (link ID) is a usually a numeric identifier assigned to each link in the network. Link ID references can be passed in more than one way. The link ID may be:

— unique, or

— non-unique within a hierarchical scheme, or

— derived from some manipulation of location, such as the bit interleaving of end-node coordinates.

Two modes of referencing are:

— Unique Link ID - The link itself has one identifier (possibly corresponding to only one direction of the link and optionally complemented by a second Identifier for the reverse direction, i.e. the link may either be directed or undirected.

— End Node IDs - The link is then identified by two identifiers, those of the link start node and end node.

Within a Link ID reference, additional information may be specified, such as offsets from start and/or end nodes, an indicator for the side of the street or road on which a point-of-interest or linear segment-of-interest resides, or implied directionality for a unique Link-ID reference. The link ID LRM refers to a previously defined data base of identifiers and is therefore categorized as a pre-coded LRM.

A.3 Referencing by geographic coordinates

A location on the earth's surface is often expressed in terms of coordinates defined by a coordinate system (axes, origin, and values) and a geodetic datum, the set of geodetic parameters defining the space with respect to which location is to be referenced (see ITRF). Coordinate systems may be earth-centred or local, geodetic or planar, and may allow position specification horizontally, vertically, or both. Geodetic parameters may include:

— National Geodetic Datum

— Reference Ellipsoid

— Projection Method

— National Map Grid

— Geoid Ondulation

— Magnetic Declination

Referencing by geographic coordinates is defined in ISO 19111:2007, Geographic information – Spatial referencing by coordinates [16], and the new version of ISO 6709, Standard representation of latitude, longitude and altitude for geographic point locations [19].

This LRM is called dynamic LRM because the nature of the defined coordinate system is independent of the road network's locations so that a location's code is produced on demand, for example while determining the position of the real world object in the geodetic system.

A.4 Referencing by grid

The common element of a grid (or raster) scheme is a regular subdivision of a surface into finite shapes, typically rectangular, and the assignment of coordinates in some regular way (e.g. letters A-Z for columns and numbers 1-10 for rows). To minimize data set size and for efficient manipulation and search based on divide-and-conquer algorithms, hierarchical tessellations recursively subdivide a surface into regular groupings of shapes numbered hierarchically. Data sets built on hierarchical tessellations preserve information where and only where there is information. Such methods are therefore hybrid between continuous field representations and grids. In terms of the location referencing method here again the grid is defined independent of the road network and is therefore categorized as dynamic LRM.

A.5 Linear referencing

A linear (1-dimensional) referencing method is a method of identifying a location on a network or part of a network by reference to known positions of (spatial) objects. If space is constrained to the road network itself, distances along roads from established nodes (or even topologically non-significant points) can be used to specify location. Mile point or reference point sub-methods use a road label and distance measure, and mile marker, reference marker and addressing sub-methods use physical features inserted into the digital base map. Because of the reliance on predefined identifiers this method is categorized as pre-coded LRM even if parts of the reference e.g. a given offset do vary inside of different references.

Linear Referencing is also addressed in the ISO/TC 211 family of Geographic Information Standards in Section 6.6, Package: Linear Reference Systems, in ISO 19133, *Geographic information — Location-based services — Tracking and navigation* [13], which supplies classes and types for the definition of linear reference systems generally. This work is itself related to that developed by the U.S. National Cooperative Highway Research Program (NCHRP) as reported in NCHRP Report 460, Guidelines for the Implementation of Multimodal Transportation Location Referencing Systems (2001).

A.6 Referencing by cross-streets

The Cross Streets Method uses the names of intersecting (cross) streets to identify nodes. This method relies on passing three street names and offset(s) from the first intersection to identify location on a link. Inclusion of the names of two streets that intersect identifies an intersection. Adding a third street name denotes two intersections along the same road, which identifies a specific road segment or link. Offsets specify position along the link for POI or sub-link references. Adding a third street (i.e., two intersections along the same road) identifies the specific road segment and extent of a position along a road. Coordinates of the centre-line intersection of streets can be used to resolve ambiguities found when using street names alone. This LRM is the predecessor of the dynamic location referencing method defined in part 3 of this International Standard. It is in such dynamic, as the rules to create the reference are independent of the real world information, although it relies very much in the fact that the street names do exist as pre-defined information on encoders and decoders side. Especially this deficiency has been reason to enrich the method with more types of information to ensure independence of such names as much as needed. See part 3 for more information about dynamic location referencing.

A.7 Referencing by address information

An address is a value unambiguously associated with a known location controlled by an all side accepted authority like a governmental mail system. The most common form is the street mail address (combinations of street names and numbers indicating location along the street). Given a scheme to prevent ambiguity problems between domains and the existence of consistent naming and numbering conventions, addresses are efficient references within larger domains such as nations. In addition, they are potentially useful adjuncts to other location referencing methods. ISO/TC 211 addresses this referencing method in ISO 19112, *Geographic information — Spatial referencing by geographic identifiers* [17] which provides essential elements for gazetteering. From the nature of the method it clearly relies on pre-defined identifiers and therefore it is categorized as pre-coded location referencing.

A.8 Combinations

Elements of different methods can be combined to form others or to improve a given method's performance. For example, a street or road name can be used to reduce ambiguity in a location reference by geographic coordinates. This is a typical way to make one LRM more robust against mismatches caused by deficiency of it standalone.

A.9 TPEG location referencing

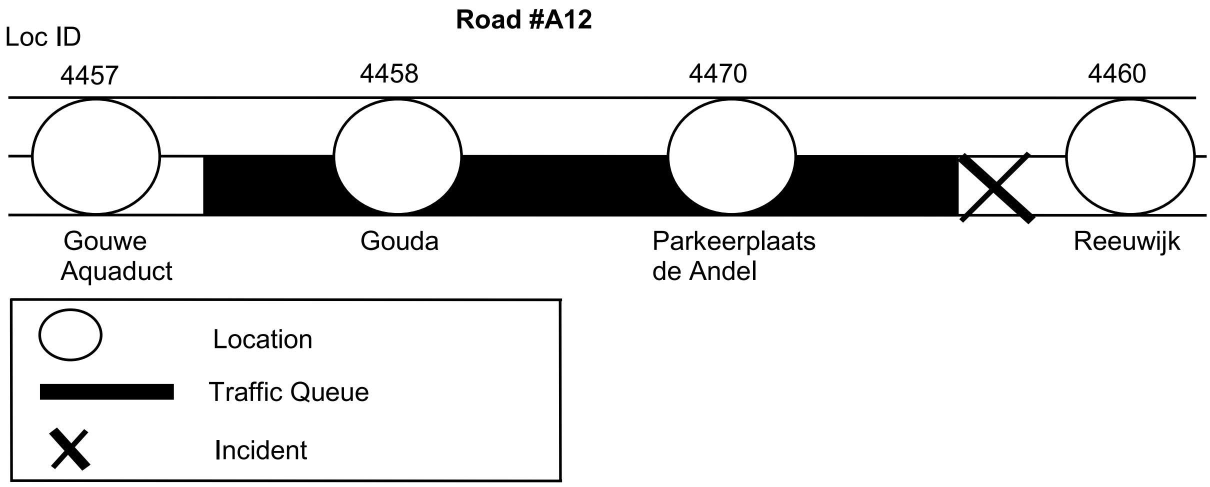
The TPEG Location Referencing method is categorized as a dynamic LRM as it allows collection of different attributes on request. The TPEG technology makes use of several aforesaid methods. It combines coordinates, location codes, street names, intersection information, and makes it possible to build connectivity trees. With this wide range of choices the service provider can proprietarily choose which information he wants to send out and receivers may or may not be able to decode automatically the location information. More focus here lies on the idea to transport efficiently a human readable sentence in a compact way which then the client user can read and recognize the position on its own. It is specified in bibliography items [3] and [4].

Annex B
(informative)

Examples of location referencing methods in use (mapping to conceptual data model for location referencing systems)

B.1 Example of RDS-TMC system (Alert-C)

In ISO 14819-3 the location referencing method is specified and more information about that can be found in ISO 17572-2. The following example shows a location on one side of a street going over a number of pre-defined location points, which normally represent intersections.



Location number	**Type**	**Road / junction number**	**First name**	**Second name**	**Reference to area**	**Reference to linear location**	**Negative offset**	**Positive offset**
4457	Aqua.	A12	Gouwe Aquaduct		2009	949	4456	4458
4458	Exit	A12	Gouda	N207	30089	949	4457	4470
4470	Park.	A12	Parkeerplaats de Andel		30089	949	4458	4460
4460	Exit	A12	Reeuwijk		2009	949	4470	4461
Reference	*Attribute*	*Definition*			*Relationships*			

Figure B.1 — Example of the Alert-C location referencing system

In relation to the general logical data model in Figure 1 this translates to:

Definition = Each Point Location instance is defined by at least one of the four items Road/Junction number', 'Road Name', 'First Name' or 'Second Name'.. Other Location Types may be defined using different information.

Reference = Location Number.

Attributes = Type.

Relationships = Reference to Area, Reference to Linear Location, Positive Offset and Negative Offset.

B.2 Example of VICS system

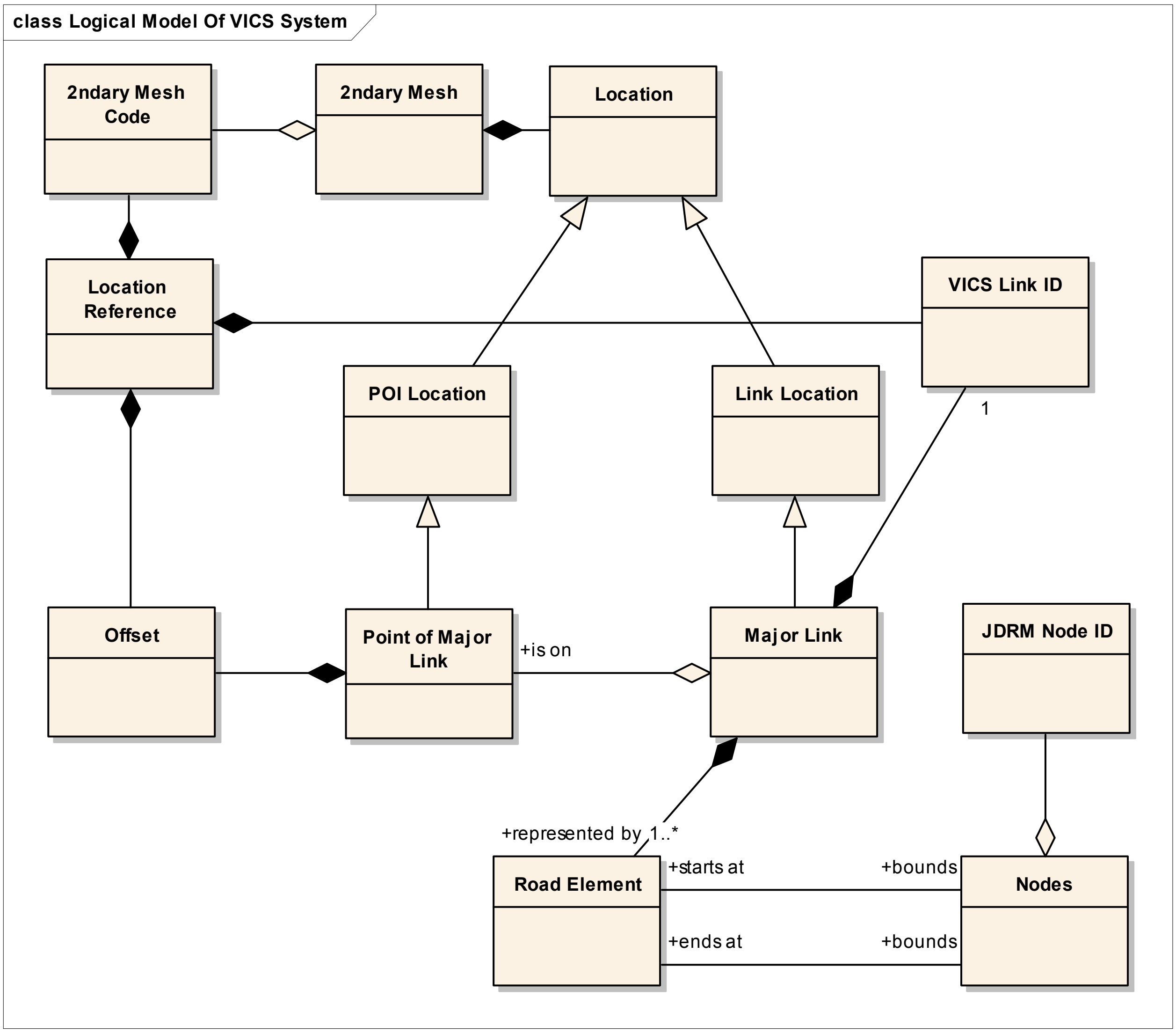


Figure B.2 — Example of Conceptual Model applied to Link ID references of the VICS System

In relation to the general logical data model in Figure 1 this translates to:

Definition = JDRM Basic Roads.

References = VICS Link ID.

Attributes = None.

Relationships = Topological connections to other JDRM Basic Roads in the JDRM database.

Annex C
(informative)

Description of UML expression elements

This International Standard makes use of a newly developed methodology to express structural circumstances called UML. The following table shows a short description of UML diagram elements used to ensure that no misinterpretation may occur caused from further development of UML1.4 is standardized in ISO/IEC 19501. However, for UML2 the standardization is to be completed by the Object Management Group http://www.omg.org.

In different class diagrams light or dark colouring is used, to express the intent of a particular diagram. The light colour implies that the diagram is of logical/explanatory nature; the dark colour implies that a particular instantiation will be introduced afterwards. The dark colouring here is used for the description of the structure of the proposed physical format.

Table C.1 — Description of UML expression elements

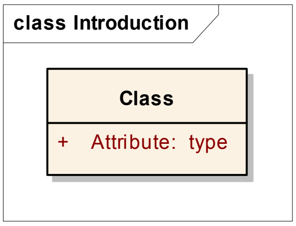
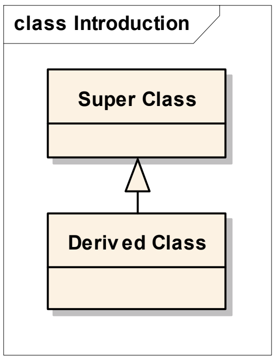
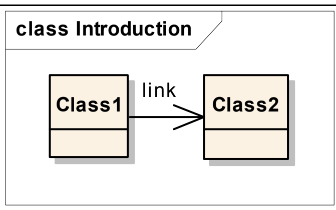
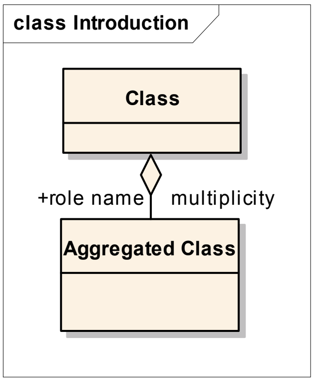
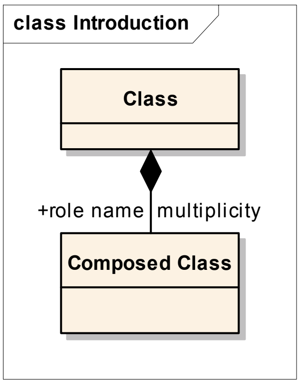
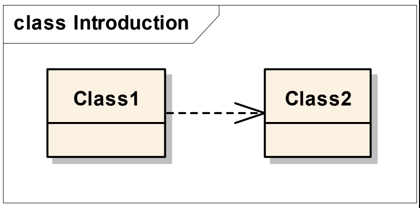
Element Name	Element	Description
Class		A class is a template for a given data element which can contain attributes. It is a rectangle divided into three compartments. The topmost compartment contains the name of the class. The middle compartment contains a list of attributes owned by that class and the bottom compartment contains a list of operations which is not shown here because operations are not used in this International Standard. In some diagrams, the bottom compartment of Attributes may be omitted for clarity reason. An attribute line has a specifier "+, # or –" for the visibility (not used in this International Standard) a name of the attribute and after a colon a data type and in squared brackets the multiplicity which is described in aggregation hereunder.
Specialisation		A Specialisation (i.e. Inheritance) defines a general class (super class) which properties are inherited from the derived class. In data structures that imply that the derived class has at least the same attributes as the super class and normally will define more attributes to it. Reason for using a inheritance in general is the capability of having different specialisations from one super class.
Association		The association shows that two classes do have a connection in between. Associations are used in this International Standard to express a loosely linkage having the type of that linkage as a name of the link. An arrow at the head expresses the direction of the association which means only in direction of the arrow the association applies. In the small example the class 1 is linked (with a link) to class 2 but class 2 does not know anything about class 1. The association has no direct counterpart in data structures, but will indirectly be visible somehow.

Table C.1 (*continued*)

Element Name	**Element**	**Description**
Aggregation		The aggregation is a more explicit design element for describing attributes. It is a more strong association telling that the class on the side of the diamond “has” a instance of the aggregated class. The name of that instance is given on the left side of the connection and starts again with the “+” as specifier of visibility. On the right side the multiplicity of that instance is given as a range of the allowed count of occurrences. An aggregation does let open if the aggregated element has the same lifetime as the aggregating class. In data structures the aggregation can be a reference to another data structure or a embedded data element.
Composition		The composition strengthens the type of aggregation in that way that the lifetime of the composed element is the same as the composing class, i.e. the structure can be seen as a “composition”. In data structures composition is normally seen as an embedded data element.
Dependency		The dependency is a unspecific type of relationship between two classes.

Annex D
(informative)

Comparison of definitions with TC 211

D.1 Introduction

This Annex compares terms and definitions from this part of ISO 17572, with terms and definitions from TC 211 standards, in order to establish a mapping of terminology between the two Technical Comittees' approaches. The TC 211 definitions are taken from the ISO/TC 211 Terminology spread sheet, edition 9, and from Annex E (Crosswalk between common terminology in ISO/TC 211 and ISO/TC 204) of the document ISO/TC 211 CD 19132 Geographic information - Location Based Services - Reference Model. In the table below, ISO 17572-1 terms and definitions are listed in conjunction with definitions for equivalent, or near-equivalent terms from ISO/TC 211.

Table D.1 allows side-by-side comparisons. TC 204/WG 3 terms tend to have more ‘practical’ detail and are more verbose, whereas TC 211 terms are more conceptual and succinct. For example, compare definitions for Junction and Link, below, and the 125 words in the ISO 17572 definition of ‘location’ versus the 3 words in the TC 211 definition. In the latter case, reusing the simpler TC 211 definition would be semantically inaccurate for the purposes of ISO 17572. In many cases, ISO 17572 terms are not used in TC 211, and vice-versa. This can be because of the difference in conceptual level between the groups, or just because different terms are used for the same concepts, with the same or similar definitions.

From the Crosswalk in Annex E of ISO/TC 211 CD 19132, “All of the differences between ISO/TC 211 and ISO/TC 204 encountered do not constitute a genuine variation of usages, vision or concept. In general they represent a variation on choices in description of surprisingly similar technical approaches. In general, GDF is an application schema based on ISO 19109 and ISO 19110. This essentially makes it a profile of the ISO/TC 211 standards. A profile (ISO 10000) is allowed to choose options and parameter values set forth in a base standard. This would include the application schema specification as defined in ISO 19110.”

D.2 Table of compared terms

Table D.1 — Comparison of TC 204 location referencing and TC 211 terms

Term	TC 204 Term's Definition	TC 211 Term's Definition	Comment
Accuracy	Measure of closeness of results of observations, computations or estimates to the true values or the values as accepted as being true.	Closeness of agreement between a test result and the accepted reference value	Same meaning
Attribute	Characteristic property of an entity like a real world feature. It allows the identification of that feature by the sum of its attributes. An attribute has a defined type and contains a value. Attributes can be either simple, i.e. consisting of one atomic value, or composite, i.e. consisting of a number of values, each represented by separate, subsequent attributes. Composite attributes are also called complex.	Named property of an entity. Alt: Characteristic of a feature	Same meaning

Table D.1 (*continued*)

Term	TC 204 Term's Definition	TC 211 Term's Definition	Comment
Coordinate	One of an ordered set of N numbers designating the position of a point in N-dimensional space.	One of a sequence of N-numbers designating the position of a point in N-dimensional space	Same
Datum	A set of parameters and control points used to accurately define the three-dimensional shape of the Earth (e.g. as an ellipsoid). The corresponding datum is the basis for a planar coordinate system.	Parameter or set of parameters that serve as a reference or basis for the calculation of other parameters. Geodetic Datum: datum describing the relationship of a coordinate system to the Earth	Similar meaning. Interchangeable in practice.
Face	A two-dimensional element bounded by a closed sequence of edges not intersecting themselves. The face is the atomic two dimensional element.	2-dimensional topological primitive	Same meaning
Junction	Elementary crossing in the road network, connecting 2 or more road elements. In GDF terms, it is a Level 1 Feature that bounds a road element or ferry connection. Junctions that represent real crossings are at least trivalent (having three roads connected). A bivalent junction may only be defined in case an attribute change occurs along the road (e.g. road name change). A junction is also coded at the end of a dead end road, to terminate that.	Single topological node in a network with its associated collection of turns, incoming and outgoing links	Same meaning
Link	A topological connection between two nodes that has a unique Link Id in a given map database. A link may contain additional intermediate coordinates (shape points) to better represent the shape of curved features. A link may be directed or undirected.	Directed topological connection between two nodes (junctions), consisting of an edge and a direction.	TC 211 meaning differs in directionality. Direction is mandatory for TC 211, optional for TC 204. A TC 204 Link also carries other information; it is a more application oriented meaning.
Link Location	A Link Location is a location identifiable by a part of the road network database having e.g. one Id or having the same attributes throughout the not disjoined stretch.	Link Position: position within a network on a link defined by some strictly monotonic measure associated with that link	TC 211 Link Position assumes a linear referencing context; the meaning is position along a link. Not commensurable terms.

Table D.1 (*continued*)

Term	TC 204 Term's Definition	TC 211 Term's Definition	Comment
Location	A simple or compound geographic object to be referenced by a Location Reference. A Location is matched to database objects by Location Definitions, which specify what is meant by a particular Location. Without any explicit remark it is meant to be a linear stretch in terms of topology in the database network without any loops or discontinuities in between (Linear Location). It might also be only a point in the Network as a specialization of a linear stretch with length zero. In addition to that a location can also be a set of network database elements representing an area. This area is e.g. expressible by a polygon or a list of linear locations.	Identifiable geographic place	TC 204 definition applies within the context of the Location Referencing standard only.
Node	A zero-dimensional element that is a topological junction of two or more edges, or an end point of an edge.	0-dimensional topological primitive	Same meaning
Point	A zero-dimensional element that specifies geometric location. One coordinate pair or triplet specifies the location.	0-dimensional geometric primitive, representing a position	Same meaning
Precision	The exactness of the measurement of a data value, or of the storage allocated to a measured data value. Alternatively, the closeness of measurements of the same phenomenon repeated under exactly the same conditions and using the same techniques.	Measure of the repeatability of a set of measurements	Same meaning
Resolution	The smallest unit which can be detected. It fixes a limit to precision and accuracy.	Size of a pixel	Same meaning, but the TC 211 definition is from imagery, and is only appropriate only in that context.
Road	A road is a part of the road network which is generally considered as a whole and which can be addressed by a single identification like a road name or road number throughout. In general, it is a connection within the road network, with or without crossings, which functionally can be considered as a unity. Associated carriageways shall be considered as one road. (Note that, in the context of this International Standard, the term also covers the natural language term street).	Route: sequence of links, and / or partial links, that describe a path, usually between two positions, within a network	

Table D.1 (*continued*)

Term	TC 204 Term's Definition	TC 211 Term's Definition	Comment
Topology	Properties of spatial configuration invariant under continuous transformation. In a digital map database this means the logical relationships among map features. It can be used to characterize spatial relationships such as connectivity and adjacency.	Topological object: spatial object representing spatial characteristics that are invariant under continuous transformations	Same meaning

Annex E
(informative)

Introduction to the TPEG physical format

E.1 Introduction

One part of any ITS application's message is the location reference. While the ITS applications are specified within other groups like the traveller information services association (TISA), this International Standard concentrates on the location referencing only. The Annexes of part 2 and 3 do give the opportunity of a first standardized way to exchange location references over different links. The location reference container specified here can be inserted at the appropriate positions specified by different ITS application specifications. For example the set of ISO 18234 and ISO 24530 specifications are defined by TISA with which the physical format proposed here is very much synchronised. The physical format, described in the different annexes, is the comprehensive physical representation of the different LRMs in a structural, compact binary and internet based XML form. ITS applications may use the complete protocol specified by TPEG or can just use the formats specified in this part of ISO 17572 on their own.

The TPEG technology uses a byte-oriented stream format, which may be carried on almost any digital bearer with an appropriate adaptation layer. TPEG messages are used to transfer information from the database of a service provider to other service providers or to an end-user's equipment. Specifically, it is focused on optimising TPEG exchanges for the traveller, transport and automotive environment. The protocol is intended to be bearer neutral. Actually two different physical implementations do exist: The first is a binary implementation especially designed to save bandwidth and to be most possible compact; the second is XML-based allowing the internet based applications to efficiently use existing technology for transporting of messages.

TPEG knows two different types of generalized components in any TPEG Application (message): the Message Management Container [1] where message management information is uniquely put at the beginning of a message and the Location Reference Container [2], where different TPEG Application specifications can make use of it several times and at self-specified positions in the data-stream. The Location Reference Container (LRC) is designed to store the location references in one or more different formats. The different formats are specified in parts 2 and 3 and TPEG-LOC is specified by the TPEG Forum in bibliography item [2], which is addressing the possibility to present a location both graphically on a map and as a spoken message. Bibliography item [2] is intended to become publicly available as future release of ISO/TS 24530-2 and ISO/TS 18234-6.

This annex explains the concept of bibliography item [2] as far as it is needed to understand the physical format implementations of part 2 and 3.

NOTE 1 The next release of ISO/TS 24530-2 and ISO/TS 18234-6 may have some differences from this concept. In this case precedence shall be given to the relevant ISO-Standards.

A simple TPEG-Message may have only one location reference container in the message, but a more complicated one may also have different location containers. This specification applies to all of the occurrences of the location reference container. The following figure depicts how a location refits within TPEG and how it is mapped to other TPEG applications.

Figure E.1 shows a TPEG Location Reference Container and its possible optional formats for the different corresponding methods. At least one location reference must be available in a location container and all location references shall refer as much as possible to the same location.

NOTE 2 In theory the location references are required to be exactly the same, but with the fact that the different LRSs are not completely identical in their resolution, this might not be reachable in any case.

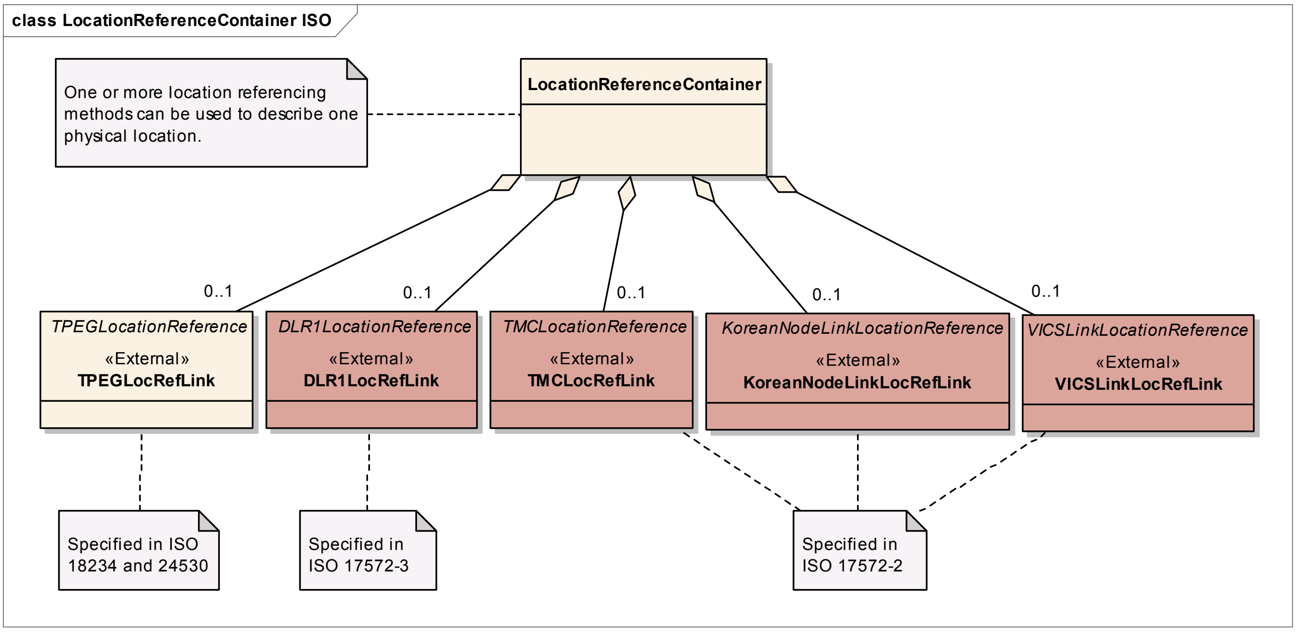


Figure E.1 — Overview of TPEG Location Reference Container concept

The stereotype "external" in the diagram designates, that the content of this classes are specified in other documents. ISO 17572-2 and -3 contain annexes proposing the different physical formats for the series of according location referencing methods.

The Instantiations of the different methods are differentiated by generic component ids or by the name of the method. The following list applies to the methods defined in Figure E.1:

— TPEGLocationReference = 0

— DLR1LocationReference = 1

— TMCLocationReference = 2

— KoreanNodeLinkLocationReference = 3

— VICSLinkLocationReference = 4

E.2 Data types

E.2.1 General

This chapter summarizes all data types specified in bibliography items [3] and [4] as far as it is needed, to be able to implement physical format encodings of part 2 and part 3 of this International Standard.

NOTE The next release of ISO/TS 24530-x and ISO/TS 18234-2 may have some differences from this chapter. In this case precedence shall be given to the relevant ISO-Standards.

E.2.2 Primitive data types

E.2.2.1 Base data types

In case that an attribute shall be stored in a fixed data size and with a given type TPEG standard SSF [3] prepares the following names:

Table E.1 — TPEG location reference physical format

Data Type Name	Type	Size	Comment
IntUnTi	unsigned integer	1 byte	corresponds to intunti in [12]
IntUnLi	unsigned integer	2 byte	corresponds to intunli in [12]
IntUnLo	unsigned integer	4 byte	corresponds to intunlo in [12]
IntSiTi	signed integer	1 byte	corresponds to intsiti in [12]
IntSiLi	signed integer	2 byte	corresponds to intsili in [12]
IntSi24	signed integer	3 byte	see clause E.2.2.3
IntSiLo	signed integer	4 byte	corresponds to intsilo in [12]
IntUnLoMB	unsigned integer	variable	see clause 0
IntSiLoMB	signed integer	variable	see clause 0
Float	floating point num.	4 byte	corresponds to float in [12]
ShortString	array of byte	variable	corresponds to short_string in [12]
BitArray	array of bytes	variable	Container for a list of defined Boolean values.
Boolean	one bit with 1 = true	1 bit	A flag in the defined BitArray flagging the Boolean statement.
TableNumber	unsigned integer	1 byte	used to identify a specific table
TableEntry	unsigned integer	1 byte	used to identify a specific table entry
TimeCount	unsigned integer	4 byte	corresponds to time_t in [12]

E.2.2.2 Multi-byte integer

A multi-byte integer consists of a series of octets, where the most significant bit is the continuation flag and the remaining seven bits are a scalar value. The continuation flag indicates that an octet is not the end of the multi-byte sequence. A single integer value is encoded into a sequence of N octets. The first N-1 octets have the continuation flag set to a value of one (1). The final octet in the series has a continuation flag value of zero (0). This allows to know exactly the end of a series of bytes belonging to one multi-byte, being the one with MSB=0. The bytes are encoded in “big-endian” order i.e. most significant byte first.

In this environment the maximum number of concatenated bytes is restricted to 5, so that the maximum unsigned integer, which can be encoded is $2^{(40-5)} - 1$. However, the actual specification defines the three most significant bits of the fifth byte as “reserved for future use”. This leads into the maximum number $2^{32} - 1$ which is the maximum value of a four byte unsigned integer IntUnLo.

In case of signed value interpretation, the complement on two is used on the 7 bit wide octet series. The count of bytes is then defined by the magnitude of the positive value to be stored in multi-byte. The three reserved bits in byte #5 shall be set to ‘111’ in case of negative numbers with 5 byte length and ‘000’ otherwise, to be up-ward compatible in case of introduction of a 64-bit integer value in future. Signed values from 0 to -2^6 are stored in one octet, to -2^{13} in two octets, to -2^{20} in three octets, to -2^{27} in four octets and to -2^{32} in five octets.

For example a value 0x62 (0110 0010) would be encoded with the one octet 0x62. The integer value 0xA7 (1010 0111) would be encoded with a two-octet sequence 0x8127. The signed representation of –1 is 0x7F. And –2345 is represented in two septets so that the complement on two is 0x36D7 = (110 1101.101 0111). A serialization in multi-byte then results in **1**110 1101.**0**101 0111 = 0xED57.

E.2.2.3 Three Byte Integer Value

For bandwidth saving in the binary format a special attribute type is defined. Allowing to save coordinates, which never can become bigger than three byte in a suitable data element, the three byte integer for signed values have been added. It stores three octets in big-endian order, and uses the complement on two over all three bytes in case of negative values. From that the range is defined by $[-2^{23}, 2^{23}-1]$.

E.2.2.4 BitArray

This is an encoding specific data type used for encoding an array of Booleans. The bits are encoded in a sequence of bytes, where the first bit of each byte is a continuation flag (shown as c in the figure below). If this bit is set (=1) there follows at least one more byte in this bit array. The last byte always has this bit cleared (=0). A BitArray represents a list of Boolean values which is implemented in the same way as for all lists. The first byte holds bits numbered from zero to six in that order. The second byte holds bits numbered seven to 13, again in that order, and so on.

The ordering is sequential from first to last bit. This use, to ensure consistency with other lists, differs from the encoding of numeric values which use a Big-endian bit and byte order.

Byte 0								**Byte 1**								**…**	
	Bit NR								**Bit NR**								
C	0	1	2	3	4	5	6	C	7	8	9	10	11	12	13		...

E.2.2.5 Boolean

A single true or false value stored in octets in a byte. The bit number starts at number 0 for the first Boolean attribute and increase by 1 for each successive Boolean attribute in the same data type definition. The first occurrence of a Boolean is stored as a BitArray (see above). All subsequent Booleans are grouped together in this BitArray.

E.3 Binary format description

E.3.1 Introduction

The binary format is defined by the UML diagrams specified in corresponding chapters in part 2 and 3. The coding sequence additionally follows the rules of bibliography item [3]. The rules differ from original rules in bibliography item [12] in the following way:

This section introduces the terminology and the syntax that is used to define TPEG data elements and structures.

E.3.2 Data type notation

E.3.2.1 Rules for data type definition representation

The following general rules are used for defining data types:

1) A data type is written in upper camel case letters in one single expression[2)] . The data type may letters (a-z), number (0-9), underscore "_", round brackets "()" and colon ":"; the first shall be a letter.

EXAMPLE IntUnLo stands for Integer Unsigned Long

2) A data type is framed by angle brackets “ < > ” .

3) The content of a data type is defined by a colon followed by an equal sign “ := ”.

4) The end of a data type is indicated by a semicolon “ ; ”.

5) A descriptor written in lower camel case may be added to a data type as one single expression.

6) A descriptor is framed by round brackets “ ( ) ”.

7) The descriptor contains either a value or a name of associated type.

8) Data types inside a definition of another one are separated by commas “ , ”. The order of definition is defined as the order of occurrence in a data stream.

9) Control statements (“if”, “infinite”, “unordered” or “external”) are noted in lower case letters. A control statement is followed by a block statement or only one data type.

“if” defines a condition statement. The block’s (or data type’s) occurrence is conditional to the condition statement being valid. The condition statement is framed with round brackets. This statement applies to any data type.

“infinite” defines endless repetition of the block (or data type). This is used to mark the main TPEG stream as not ending stream. This statement applies to components only.

“unordered” defines that the following block contains data types which may occur in any order, not only the one used to specify subsequent data types. This statement applies to components only.

“external” defines that the content of the data type is being defined external to the scope of given specification. The control statement “external” shall be followed by only one data. A reference to the corresponding specification should follow in the comment. All types specified in this specification are treated as being in scope of any application.

EXAMPLE
```
<MMCLink(1)>:=                                   : extern defined component
external <MessageManagementContainer(1)>;        : id = 1, see bibliography item [1]
```

10) The expression “ m * ” indicates multiplicity of occurrence of a data type . The lower and upper bound are implicitly from 0 to infinite; other bounds are described in square brackets between to points " .. " and behind the data type descriptor. The " * " stands for no limitation at upper bound.

EXAMPLE m * **<IntUnTi>**(Attribute) [1..*] , : The “Attribute” shall occur once at least and up to infinite.

11) Curly brackets (braces) “ { } ” group together a block of data types.

---

2) Camel case shall be understood as a compound word wherein any fraction is signalled by a capital letter inside the word. Upper camel case means that the word begins with capital letter, and lower camel case means, the word starts with a small letter.

12) A function “ fn ( ) ” that is calculated over a data type is indicated by italic lower case letters. The comment behind the definition of the function shall explain which function is used.

13) Any text after a colon “ : ” is regarded as a comment.

14) A data type definition can be a template (i.e. not fully defined declarative structure) having a parameter inside of round brackets "(x)" at the end of the data type name. Templates define structures, whose structural definition is included as basis in other data type definitions. To declare the given template (making it identifiable) the name of the parameter is repeated as descriptor in a nested data type of the subsequent definition list. Templates allow reading the generalised part of different instances i.e. to specify data type interfaces.

EXAMPLE **<Template(x)>** := : x defines the template parameter
<IntUnTi>(x); : descriptor x defines position of setting the parameter in the list

15) A data type can inherit a template by concatenating the data type name of the template including the square brackets to its own name. The data type itself can again be a template having the "(x)" at its end of name, or it instantiates the inherited template by defining the value of the parameter in the brackets. In latter case the brackets shall contain the decimal number of the identifier and the value shall be set in the subsequent definition list. The structural definition of the inherited template is repeated as first part of the definition list before new data types are specified.

EXAMPLE **<AnotherTemplate(x)<Template(x)>**:= : second template inherits first
<IntUnTi>(x), : repeated definition from 1st template
<IntUnLi>(n); : additional structural definition

<Instance<AnotherTemplate(1)>>:= : instantiation of the second template
<IntUnTi>(1), : definition of parameter in the stream
<IntUnLi>(n), : structural definition from template
<IntUnTi>(value); : some more definition

16) A specific instance of a template (i.e. declarative structure) is described in the definition list with out the brackets. Any inherited data type of this template may occur at that position in the data stream.

EXAMPLE **<SomeData>**:=
<AnotherTemplate>(anyAnotherTemplate); : Data stream contains **"Instance"**.

E.3.2.2 Description of data type definition syntax

A data type is an interpretation of one or more bytes. Each data type has a structure, which may describe the data type as a composition of other defined data types. The data type structure shows the composition and the position of each data element. TPEG defines data structures in the following manner:

<NewDataType>:=	: Description of data type
<DataTypeA>(descriptorA)**,**	: Description of data A
<DataTypeB>(descriptorB);	: Description of data B

This shows an example data structure, which has just two parts, one of type **<DataTypeA>** and the other of **<DataTypeB>**. A descriptor may be assigned to the data type, to relate the element to another part of the definition. Comments about the data structure are included at the right-hand side delimited by the colon “:” separator. Each of the constituent data types may be itself composed of other data types, which are defined separately. Eventually each data type is expressible as one or more bytes.

Where a data structure is repeated a number of times, this may be shown as follows:

<NewDataType>:=	: Description of data type
<DataTypeA>,	: Description of data A
m * **<DataTypeB>**[0..*];	: Description of data B

Often, in such cases it is necessary to explicitly deliver to the decoder the number of times a data type is repeated; sometimes it is not, because other means like framing or internal length coding allows knowledge of the end of the list of the repeated data type. In other cases the overall length of a data structure in bytes needs to be specified. Additionally the constraint on occurrences can be added, which tells how many instances of the data type must be expected by the decoder. The "*" as upper bound means in this case that at this place no restriction is given to the upper bound; in other words, infinite elements may follow.

Where the number of repetitions must be signalled, it may be accomplished using another data element as follows:

<NewDataType>:=	: Description of data type
<IntUnTi>(n),	: An integer representing the value of "n"
n * **<DataTypeA>**[0..255],	: Description of data A
<DataTypeB>;	: Description of data B

In the above example a decoder has to have the value of "n" in order to correctly determine the n'th position of the **<DataTypeB>** in the list. Here as consequence of data type IntUnTi not more as 255 instances of the data type can be coded.

In the following example the decoder uses the value of "n" to determine the overall length of the data structure, and the value of "m" determines that **<DataTypeB>** is repeated m times:

<NewDataType>:=	: Description of data type
<IntUnTi>(n),	: Length, n, of data structure in bytes
m * **<DataTypeA>;**	: Description of data A

This data type definition is used to describe a variable structure switched by the value of x:

<NewDataType>:=	: Description of data type
<IntUnTi>(x),	: Select parameter, x
if (x=1) then **<DataTypeA>,**	: Included if x equals 1
if (x=2) then **<DataTypeB>,**	: Included if x equals 2

E.3.2.3 Application dependent data types

This section describes the methodology and syntax by which application data types may be constructed within TPEG. Two basic forms are described: data structures (being non-declarative) and components (being declarative). Components contain an identifier which labels the structure, and which can be used by a decoder to determine the definition of content of the structure. As such, components are used where options are required, or where an application needs to build in 'future proofing'. Data structures do not contain such information, and are used in all other positions..

This document does not specify the structures, which are actually used in TPEG applications. Such specifications are made in the respective parts of the standard. However, examples are given below of how such structures may be built from the primitive elements described above.

E.3.2.4 Data structures

Data structures are built up from several (more than one) elements: primitive, compound or other structures (both non-declarative and declarative). As such, any application specific data type definition having no component identifier is per definition a data structure. The term data structure in specific is used for data type definitions having more then one sub element defined.

Examples of data structure might be:

EXAMPLE 1

<Activity>:=	: Activity
<DateTime>,	: Beginning
<DateTime>,	: End
<ShortString>;	: Text

EXAMPLE 2

<Wave>:=	: Sound sample
<IntUnLi>(n),	: Length of samples, n
n * **<IntSiTi>**(sample)[0..8000];	: between 0 and 8000 occurrences of a sample.

Another Example shows making use of a condition within a data type definition.

EXAMPLE 3 An application could use the example data types above in the following way:

<Appointment>:=	: Appointment
<IntUnTi>(at),	: Alarm type
if (at = 1)	
<WaveAlarm>,	: remind with a sound.
if (at = 2)	
<TextAlarm>,	: remind with a text.
<Activity>;	: Let some action follow.

<WaveAlarm>:=	
<DateTime>,	: Sound alarm: When to wake up
<Wave>;	: Sound to wake up to!

<TextAlarm>:=	
<DateTime>,	: Text alarm: When to display
<ShortString>;	: Text to display

E.3.2.5 Components

A component is understood as a declarative structure. A decoder of the data stream can identify the content of the structure with the help of the identifier which is unique in the scope of one TPEG Specification. In addition to the identifier a length indicator allows the decoder to step over those components whose ids are unknown to it. This enables the possibility of introducing new components in the data stream although decoders in the market do not know their content. The old decoder does expect the content of the first version of a protocol and ignores simply unrecognized data with small performance loss. The new decoder expects the second version of the protocol and can fully decode that version of the protocol. Components should be used wherever future extensions are envisioned, and where 'future proofing' is a strong requirement.

NOTE With this method even not backward compatible changes can be introduced into the existing market by having a migration period being backward compatible and then later cutting of not longer supported devices, even though it is expected that the migration will take its time.

In addition to the concept of declarative structuring a second step of improvement of size efficiency combined with the backward compatibility is specified. The first part following the header of a component in the data stream is defined as attribute block. The attribute block starts with the length of the block in bytes which again allows the decoder to step over attributes that are not specified in a first version of the protocol.

The decoder reads the attribute block length and decreases the count of bytes while reading the attributes in case that the last known attribute is read, and the attribute block count is not zero, the remaining bytes in the data stream are omitted to step over to the next well-known part of the data stream.

A component, including attributes, which is the general standard component, is defined by:

<**Component(x)**>:=	: Component template used for standard components
<IntUnTi>(x),	: id is unique within the scope of the application.
<IntUnLoMB>(compLengthInByte),	: length of the component counted in bytes.
<IntUnLoMB>(attributeBlockLengthInByte);	: length of the attribute block in bytes.

See following figure as example for jumping methodology:

1. Let C1 be a component with an attribute a1 as ShortInt and a sub component C2.
2. Let C2 be a component with an attribute a2 as one IntUnTi and a second a3 as ShortString.
3. Let C3 be a component being the successor of C1.

<C1<Component(1)>>:=	
<IntUnTi>(1),	: id = 1.
<IntUnLoMB>(compLengthInByte),	: length of the component counted in bytes.
<IntUnLoMB>(attributeBlockLengthInByte);	: length of the attribute block in bytes.
<ShortInt>(a1),	: first attribute in C1
<C2>(c2);	: sub component from C1

<C2<Component(2)>>:=	
<IntUnTi>(2),	: id = 2.
<IntUnLoMB>(compLengthInByte),	: length of the component counted in bytes.
<IntUnLoMB>(attributeBlockLengthInByte);	: length of the attribute block in bytes.
<IntUnTi>(a2),	: first attribute in C2
<ShortString>(a3);	: second attribute in C2

<C3<Component(3)>>:=	
<IntUnTi>(3),	: id = 3.
<IntUnLoMB>(compLengthInByte),	: length of the component counted in bytes.
<IntUnLoMB>(attributeBlockLengthInByte);	: length of the attribute block in bytes.

For explaining reason some padding bytes with value CD hex are added to the stream whereby a decoder can still read C1 – C3. In the following table you can see a first line with a position number a second line with the abbreviated function of that byte and a third line with sample content. The arrows under the table show the possible jumps allowing seeking over the different padding bytes.

Line function abbreviations mean:
CL : component (data) length in bytes — AL : attribute block length in bytes
P : padding bytes — A1, A2, A3 : attributes
C1, C2, C3 : component identifier, begin of the component

Pos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Func	C1	CL	AL	A1'	A1"	P	P	C2	CL	AL	A2	A3	A3	A3	A3	A3	P	C3	CL	AL
Val	1	15	4	42	12	CDh	CDh	2	8	7	3	4	'T'	'E'	'S'	'T'	CDh	3	1	0

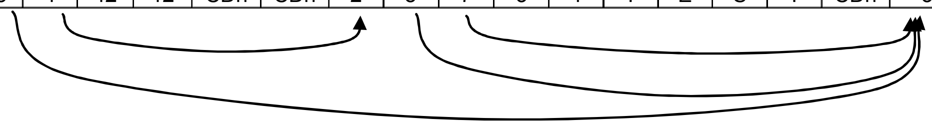


Figure E.2 — Example for jumping over unknown content with component header information

E.3.3 Application design principles

This section describes design principles that will be helpful in building TPEG applications. A fundamental assumption is that applications will develop and new features will be added. If design principles are adopted properly then older decoders will still operate properly after extending features. Correct design should permit applications to be upgraded and extended over time, providing new features to new decoders, and yet permit existing decoders to continue to operate.

E.3.3.1 Variable data structures

Switches may be included within an application, which permit variations in the subsequent data structure. However, the switch fixes the values of variations. A new type cannot be introduced without breaking backward compatibility. This may be achieved by using components. Then new features are likely to be incorporated, attention should be given to the fact that old decoders just ‘skip over’ new data fields and still expect the old components if they where mandatory.

E.3.3.2 Re-usable and extendable structures

Within an application there will be data structures, which are used repeatedly in a variety of places. There will also certainly be an ever-growing set of structures, as the application protocol develops and incorporates new features. Component templates may be used to minimize the number of occasions within the decoder’s software in which the structure needs to be defined, and to permit an increasing variety of structures to be used in a given location.

E.3.3.3 Validity of declarative structures

The Identifier of a component is uniquely defined within each application. The same number may be used in different applications for completely different purposes. Within an application one identifier designates one definition of a component. The design of an application may use components to implement placeholders or to change the composition of elements in a fixed structure.

E.4 Binary location reference container, (LRC)

E.4.1 Generic component id

Generic component	id
LocationReferenceContainer	defined by application
TPEGLocRefLink	0
DLR1LocRefLink	1
TMCLocRefLink	2
KoreanNodeLinkLocRefLink	3

E.4.2 LocationReferenceContainer component

```
<LocationReferenceContainer(x)<Component(x)>>:=                : x is defined by instantiating application
    <IntUnTi>(x),
    <IntUnLoMB>(compLengthInByte),                             : length of the component counted in bytes.
    <IntUnLoMB>(attributeBlockLengthInByte);                   : length of the attribute block in bytes.
    unordered {
            m * <TPEGLocRefLink>(tpegLoc)[0..1],
            m * <DLR1LocRefLink>(dlr1Loc)[0..1],
            m * <TMCLocRefLink>(tmcLoc)[0..1],
            m * < KoreanNodeLinkLocRefLink>(klrLoc) [0..1];
      };
```

E.4.3 TPEGLocationReference component

<TPEGLocRefLink<Component(0)>>:=	: TPEG Loc component
external <TPEGLocationReference(0)>;	: see [2]

E.4.4 DLR1LocRefLink component

<DLR1LocRefLink<Component(1)>>:=	: DLR1 Loc component template
external <DLR1LocationReference(1)>;	: see [8]

E.4.5 TMCLocRefLink component

<TMCLocRefLink<Component(2)>>:=	: TMC Loc component template
external <TMCLocationReference(2)>;	: see [7]

E.4.6 KoreanNodeLinkLocRefLink component

<KoreanNodeLinkLocRefLink<Component(3)>>:=	: KNL Loc component template
external <KoreanNodeLinkLocationReference(3)>;	: see [7]

E.5 XML schema definition for TPEG location referencing

E.5.1 General

The location references sent e.g. between traffic data provider and traffic control centre may be part of a larger database exchange done a for the internet typical xml-format like SOAP or SyncML or even to be rendered at a simple displaying device. To serve this demand this clause defines an XML Schema carrying the same data types as the binary format specified in Clause E.2. TISA currently has [4] under development which is to be served with the ISO 24530 series in future.

NOTE A new part of ISO/TS 24530 may have some differences from this chapter. In this case precedence shall be given to the relevant ISO-Standards.

To make the annexes in part 2 and 3 complete, a general type definition is needed. To make this three parts internally consistent, a part of that XSD is given here as general data type (proposed filename "tpegTYP.xsd").

E.5.2 XSD of general data types, TYP

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="TPEG" targetNamespace="TPEG" xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:simpleType name="IntUnTi">
        <xs:annotation>
            <xs:documentation>1 byte unsigned</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:unsignedByte"/>
    </xs:simpleType>
    <xs:simpleType name="IntUnLi">
        <xs:annotation>
            <xs:documentation>2 byte unsigned</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:unsignedShort"/>
    </xs:simpleType>
    <xs:simpleType name="IntUn24">
```

```xml
        <xs:annotation>
            <xs:documentation>3 byte unsigned</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:unsignedInt">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="16777215"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="IntUnLo">
        <xs:annotation>
            <xs:documentation>4 byte unsigned</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:unsignedLong"/>
    </xs:simpleType>
    <xs:simpleType name="IntUnLoMB">
        <xs:annotation>
            <xs:documentation>4 byte unsigned with variable length in binary</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:unsignedLong"/>
    </xs:simpleType>
    <xs:simpleType name="IntSiTi">
        <xs:annotation>
            <xs:documentation>1 byte signed</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:byte"/>
    </xs:simpleType>
    <xs:simpleType name="IntSiLi">
        <xs:annotation>
            <xs:documentation>2 byte signed</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:short"/>
    </xs:simpleType>
    <xs:simpleType name="IntSi24">
        <xs:annotation>
            <xs:documentation>3 byte signed</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:int">
            <xs:minInclusive value="-8388608"/>
            <xs:maxInclusive value="8388607"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="IntSiLo">
        <xs:annotation>
            <xs:documentation>4 byte signed</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:long"/>
    </xs:simpleType>
    <xs:simpleType name="IntSiLoMB">
        <xs:annotation>
            <xs:documentation>4 byte signed with varable length in binary</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:long"/>
    </xs:simpleType>
    <xs:simpleType name="BitArray">
        <xs:annotation>
            <xs:documentation>list of single bits in a string of '1' or '0'.
            The continuation flag is not coded </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
```

```
    <xs:simpleType name="Boolean">
        <xs:annotation>
            <xs:documentation>boolean</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:boolean"/>
    </xs:simpleType>
    <xs:simpleType name="ShortString">
        <xs:annotation>
            <xs:documentation>text in 8 char width</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string">
            <xs:maxLength value="255"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="TimeCount">
        <xs:annotation>
            <xs:documentation>time In seconds since 1970</xs:documentation>
        </xs:annotation>
        <xs:restriction base="IntUnLo"/>
    </xs:simpleType>
    <xs:simpleType name="DateTime">
        <xs:annotation>
            <xs:documentation>Date and time as well formed string in zulu time UTC
                    An example is:'2001-12-17T09:30:47.0+01:00'
            </xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:dateTime"/>
    </xs:simpleType>
    <xs:simpleType name="Float">
        <xs:restriction base="xs:float"/>
    </xs:simpleType>
</xs:schema>
```

E.5.3 XSD of Location Reference Container

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="TPEG" targetNamespace="TPEG"
elementFormDefault="qualified">
    <xs:include schemaLocation="tpegTYP.xsd"/>
    <xs:include schemaLocation="tpegDLR.xsd"/>
    <xs:include schemaLocation="tpegTLR.xsd"/>
    <xs:include schemaLocation="tpegKLR.xsd"/>
    <xs:include schemaLocation="tpegLOC.xsd"/>
    <xs:element name="LocationReferenceContainer" type="LocationReferenceContainer"/>
    <xs:complexType name="LocationReferenceContainer">
        <xs:sequence>
            <xs:element name="TPEGLocationReference" type="TPEGLocationReference"
minOccurs="0"/>
            <xs:element name="DLR1LocationReference" type="DLR1LocationReference" minOccurs="0"/>
            <xs:element name="TMCLocationReference" type="TMCLocationReference" minOccurs="0"/>
            <xs:element name="KoreanNodeLinkLocationReference"
type="KoreanNodeLinkLocationReference" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

Bibliography

[1] TPEG Forum, Message Management Container Specification, unpublished specification, EBU, Grand-Saconnex, 2006

[2] TPEG Forum, Location Referencing Container Specification, unpublished specification, EBU, Grand-Saconnex, 2006

[3] TPEG Forum, Binary Syntax Specification, unpublished specification, EBU, Grand-Saconnex, 2006

[4] TPEG Forum, XML Syntax Specification, unpublished specification, EBU, Grand-Saconnex, 2006

[5] SyncML, Open Mobile Alliance, 2006, http://www.openmobilealliance.org/

[6] ISO 14825:2004, *Intelligent transport systems — Geographic Data Files (GDF) — Overall data specification*

[7] ISO 17572-2, *Intelligent transport systems (ITS) — Location referencing for geographic databases — Part 2: Pre-coded location references (pre-coded profile)*

[8] ISO 17572-3, *Intelligent transport systems (ITS) — Location referencing for geographic databases — Part 3: Dynamic location references (dynamic profile)*

[9] ISO/IEC 19501:2005, *Information technology — Open Distributed Processing — Unified Modeling Language (UML) Version 1.4.2*

[10] ISO/TS 24530-2:2006, *Traffic and Travel Information (TTI) — TTI via Transport Protocol Experts Group (TPEG) Extensible Markup Language (XML) — Part 2: tpeg-locML*

[11] ISO/TS 18234-6:2006, *Traffic and Travel Information (TTI) — TTI via Transport Protocol Expert Group (TPEG) data-streams — Part 6: Location referencing applications*

[12] ISO/TS 18234-2:2006, *Traffic and Travel Information (TTI) — TTI via Transport Protocol Expert Group (TPEG) data-streams — Part 2: Syntax, Semantics and Framing Structure (SSF)*

[13] ISO 19133:2005, *Geographic information — Location-based services — Tracking and navigation*

[14] ISO 19109:2005, *Geographic information — Rules for application schema*

[15] ISO 19110:2005, *Geographic information — Methodology for feature cataloguing*

[16] ISO 19111:2007, *Geographic information — Spatial referencing by coordinates*

[17] ISO 19112:2003, *Geographic information — Spatial referencing by geographic identifiers*

[18] ISO 19132:2007, *Geographic information — Location-based services — Reference model*

[19] ISO 6709, *Standard representation of geographic point location by coordinates*

[20] ISO 14819-3, *Traffic and Travel Information (TTI) — TTI messages via traffic message coding — Part 3: Location referencing for ALERT-C*

ICS 03.220.01; 35.240.60

Price based on 41 pages