

OpenLR™

White Paper

Version: 1.2



An open standard for encoding, transmitting and decoding location references in digital maps

Copyright © 2009, 2010 TomTom International B.V.

OpenLR™ is a trade mark of TomTom International B.V.

This work is licensed under the Creative Commons Attribution-No Derivative Works 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nd/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.





License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

- a. **"Adaptation"** means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.
- b. **"Collection"** means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.
- c. **"Distribute"** means to make available to the public the original and copies of the Work through sale or other transfer of ownership.
- d. **"Licensor"** means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.
- e. **"Original Author"** means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.
- f. **"Work"** means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed



by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

- g. **"You"** means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.
- h. **"Publicly Perform"** means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.
- i. **"Reproduce"** means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

- a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections; and,
- b. to Distribute and Publicly Perform the Work including as incorporated in Collections.
- c. For the avoidance of doubt:
 - i. **Non-waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;
 - ii. **Waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,
 - iii. **Voluntary License Schemes.** The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats, but otherwise you have no rights to make Adaptations. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

- a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to



exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested.

- b. If You Distribute, or Publicly Perform the Work or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work. The credit required by this Section 4(b) may be implemented in any reasonable manner; provided, however, that in the case of a Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.
- c. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination



- a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.
- b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

- a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
- b. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
- c. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
- d. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.
- e. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.



Version information

Version 1.0

Changes:

- Initial version

Author: TomTom International B.V

Date: 2009-09-08

Version 1.1

Changes:

- Errata for v1.0 included (sections 4.4.4, 4.6.2, and 9.1)
- Note added for physical format version number (section 4.5.1)
- Section 3.4, Rule-4: U-turns nodes are also valid nodes; invalid nodes might be chosen exceptionally if no other possibility exists
- Encoding steps, Step 1: turn restrictions are optional (section 7.1)
- Encoding steps, Step 9: explanation how to deal with lines longer than 15km (section 7.1)
- Decoding steps, Step 3: find candidate lines directly if no candidate nodes can be found (section 8.1)
- Decoding steps, Step 4: LRP attributes may vary between encoder and decoder map so that values don't need to match exactly (section 8.1)
- Decoding steps, Step 5: explanation how to deal with a situation where start and end line are equal; retrying other candidate line pairs if no valid shortest-path can be determined (section 8.1)
- Explanation added for handling of prohibited turns in decoding example (section 9.2)

Author: TomTom International B.V

Date: 2009-12-11

Version 1.2

Changes:

- Description of the XML format for OpenLR™ location references (new section 7)
- Schema definition of the XML format (new section 8)
- Old sections 7 – 10 have moved to sections 9 – 12
- Encoding and decoding example extended with the XML format (section 11)
- Encoding process (section 9) and decoding process (section 10) now handle (abstract) physical formats instead of the concrete binary format.

Author: TomTom International B.V.

Date: 2010-02-01



Content

1.	Introduction	9
1.1	Common terms and abbreviations.....	10
2.	Concepts	12
2.1	Location properties	12
2.2	Map requirements.....	13
3.	OpenLR™ - Logical Data format specification	14
3.1	Introduction	14
3.1.1.	Location Reference Point.....	14
3.1.2.	Topological connection of LR-points	14
3.2	Components of LR-points	14
3.2.1.	Coordinate pair	15
3.2.2.	Functional Road Class	15
3.2.3.	Form of way.....	15
3.2.4.	Bearing	16
3.2.5.	Distance to next LR-point.....	17
3.2.6.	Lowest FRC to next LR-point	17
3.2.7.	Offsets	17
3.2.7.1	Positive offset	17
3.2.7.2	Negative offset.....	17
3.3	Relationship Attributes – LR-point	18
3.4	Data format rules	18
3.4.1.	Overview of the data format rules	19
4.	OpenLR™ - Binary representation.....	20
4.1	Introduction	20
4.2	Data types.....	20
4.3	Coordinates (COORD).....	20
4.3.1.	Absolute format	21
4.3.2.	Relative format	21
4.4	Attribute values	21
4.4.1.	Functional Road Class (FRC)	21
4.4.2.	Form of way (FOW).....	22
4.4.3.	Bearing (BEAR).....	22
4.4.4.	Distance to next LR-point (DNP).....	23
4.4.5.	Lowest FRC to next point (LFRCNP)	23
4.5	Location Reference header	23
4.5.1.	Version (VER)	24
4.5.2.	Attribute flag (AF)	24
4.5.3.	Area flag (ArF).....	24
4.6	Offsets	25
4.6.1.	Offset flags	25
4.6.2.	Offset values	25
5.	OpenLR™ - Physical data format specification.....	26
5.1	Overview	26
5.2	Status byte.....	26
5.3	First LR-point coordinates	27
5.4	Following LR-point coordinates	27
5.5	Attributes.....	27
5.5.1.	First attribute byte (attr. 1)	27
5.5.2.	Second attribute byte (attr. 2).....	27
5.5.3.	Third attribute byte (attr. 3).....	27
5.5.4.	Fourth attribute byte (attr. 4)	28
5.6	Offset	28



6.	Code size calculation	29
7.	OpenLR™ - XML representation.....	30
7.1	Introduction	30
7.2	Data types.....	30
7.3	Coordinates	30
7.4	Line attribute values.....	30
7.4.1.	Functional Road Class (FRC)	31
7.4.2.	Form of way (FOW).....	31
7.4.3.	Bearing (BEAR).....	31
7.5	Path attribute values.....	31
7.5.1.	Distance to next LR-point (DNP).....	32
7.5.2.	Lowest FRC to next point (LFRCNP)	32
7.6	Line location reference	32
7.6.1.	Location reference point.....	32
7.6.2.	Last location reference point.....	32
7.6.3.	Offsets	33
7.7	XML location reference.....	33
7.8	Binary location references	33
7.8.1.	Binary location reference.....	33
7.9	Additional information	33
7.9.1.	Bounding box	34
7.9.2.	Map database.....	34
7.9.3.	Additional data.....	34
7.10	Root tag: OpenLR.....	34
7.10.1.	Location ID.....	35
8.	OpenLR™ - Physical data format specification (XML)	36
9.	Encoding locations	42
9.1	Encoding steps	42
9.1.1.	Overview of the encoding steps	45
10.	Decoding location references.....	47
10.1	Decoding steps.....	47
10.1.1.	Overview of the decoding steps.....	49
11.	OpenLR™ example.....	51
11.1	Example setup	51
11.2	Encoding.....	52
11.3	Binary location reference.....	53
11.4	XML location reference.....	55
11.5	Decoding.....	57
12.	Appendix	61



1. Introduction

Communication of spatial information involves the communication of location. The communication chain of a machine readable location can be described as encoding the location at the sender side, transfer of the code to the receiving system and decoding the code at the receiver side. The process of encoding a location is also called Location Referencing. This White Paper assumes a map on the sender side from which the location is encoded and a map on the receiver side in which the decoded location is found back. An obvious way of Location Referencing is using geographic coordinates. One important disadvantage of using coordinates is that it assumes identical maps at both sides of the communication chain which often is not the case. As a consequence, the decoded location may not be found back in the receiver map, or decoding (e.g. map-matching) may be inaccurate or ambiguous. OpenLR™ is a method for location referencing which does not have this disadvantage. It accommodates requirements of communication of location between systems which have dissimilar maps. OpenLR™ is communication channel independent. It takes bandwidth requirements into account in the sense that OpenLR™ requires minimal bandwidth.

OpenLR™ has been designed for the use case of transferring traffic information from a centre to in-vehicle systems, built-in or used as an add-on (PND, Smart Phone). The information transferred can consist of the current traffic situation at a certain location, a traffic forecast or special alerts. The corresponding locations are roads or a list of connected roads.

The most well-known and most used method to transfer traffic information today is called RDS-TMC. The Location Referencing used in RDS-TMC makes use of pre-coded locations. These pre-coded locations are added to the corresponding locations in the map by the map providers of the sending and the receiving map. The process of encoding is looking up the location code in the map belonging to the relevant location. The process of decoding is finding back the location code in the map and looking up the corresponding location.

From the fact that RDS-TMC makes use of pre-coded location it follows that the amount of locations fit to be transferred is limited. OpenLR™ does not have that restriction. With OpenLR™ every location in a map can be transferred.

The technical goal of OpenLR™ is to specify a location referencing system with the following objectives:

- *dynamic generation of a location reference, including urban and low level roads*
- *compact format for transmission*
- *handling map differences caused by different map vendors or versions*

The industry goal of OpenLR™ is wide-scale adoption by the industry at large. OpenLR™ is therefore proposed as an open standard in an Open Source framework. It shall be usable for anyone dealing with locations and transmitting these between systems having dissimilar maps. The current version of OpenLR™ focuses on line locations but can easily be extended to handle point and area locations.

While the standard is developed and maintained by TomTom International B.V., everyone is invited to contribute to its further development.

This document describes the basic concepts of the OpenLR™ standard in section 2. It defines a logical data format for location references in section 3. Section 4 defines a binary representation of location references and the physical representation is given in section 5. Section 6 deals with the corresponding code size. Section 7 defines a XML representation of location references and the schema definition is given in section 8. Encoding locations and decoding location references will be outlined in section 9 and section 10. Finally an example in section 11 will explain all relevant steps.

1.1 Common terms and abbreviations

The following tables explain common terms and abbreviations used in this document and in the context of location referencing

Term	Description
Bit	A bit is a binary digit, taking a logical value of either "1" or "0" (also referred to as "true" or "false" respectively).
Byte	A byte is a basic unit of measurement of information storage in computer science. There is no standard but a byte most often consists of eight bits.
Candidate line	A line object in the network of a decoder which might be referenced by the Location Code (as generated by the encoder).
Candidate node	A node object in the network of a decoder which might be referenced by the Location code (as generated by the encoder).
Decoder	A software component which decodes a Location Code and finds the corresponding location back in a map.
Encoder	A software component which generates a Location code for location in a map.
Intermediate	An internal Location Reference Point (LRP); not the start LRP nor the end LRP (destination).
Intermediate location reference point	see "Intermediate"
Line	A one-dimensional representation of a road or part of road in a road network. A line starts and ends at a node. It is directed, this means two-way traffic flow is represented by two (directed) lines, one per direction.
Location	A specification of the position on the earth surface of an object in a digital map.
Location reference	Location Code, created according to a specific set of rules, used to reference a location.
Location reference path	A route in a road network in a digital map which is referenced by the location reference. This path might be longer than the location itself.
Location Reference Point (LRP)	A description of an object in a digital map which consists of a coordinate and additional information about a line in the map.
Map	A Map is the geospatial representation of an area on the earth surface.
Node	A zero-dimensional object in the road network. A node acts as start and end for lines.
Offset	The specification of a position along a path to indicate the start or the end of a location.
Route	A route is a collection of line objects in a digital map connecting a departure location and a destination location, defined according to certain criteria which might include time distance or cost.
TMC	Traffic Message Channel; public or commercial traffic information broadcast channel using RDS (Radio-Data-System).
XML	Extensible Markup Language

Table 1: Explanation of common terms



Abbreviation	Description
AF	Attribute Flag – a flag which indicates that the binary representation of the location reference includes attribute information
ArF	Area Flag – a flag which indicates that the location reference describes an area
BEAR	Bearing – angle between the direction to a point in the network and a reference direction (here: the true North)
COORD	Coordinates – a pair of two values (longitude and latitude) representing a position in a two-dimensional network
DNP	Distance to Next Point – the length in meter to the next location reference point (measured along the location reference path between these two LRP)
FOW	Form Of Way - Certain aspects of the physical form that a line takes. It is based on a number of certain physical and traffic properties.
FRC	Functional Road Class - A classification based on the importance of the role that the line performs in the connectivity of the total road network.
lat	Latitude – geographic coordinate used for north-south measurement
LFRCNP	Lowest Functional Road Class to Next Point
lon	Longitude – geographic coordinate used for east-west measurement
LRP	Location Reference Point – a point of the location which holds relevant information for a map-independent location reference
NOFF	Negative Offset – distance in meter along the location reference path between the real end of the location and the end of the location reference path
NOffF	Negative Offset Flag – a flag which indicates that a negative offset is included in the location reference
POFF	Positive Offset – distance in meter along the location reference path between the start of the location reference path and the real start of the location
POffF	Positive Offset Flag – a flag which indicates that a negative offset is included in the location reference
RFU	Reserved for future use – a bit in a binary stream which does not have a use yet
TMC	Traffic Message Channel - is a technology for delivering traffic and travel information to drivers using pre-defined points and segments in a network
VER	Version – Version information

Table 2: Explanation of common abbreviations



2. Concepts

OpenLR™ describes a method and a format for encoding, transmitting and decoding (map-independent) references for locations. Locations are objects in a digital map, like points, paths and areas. The method makes it possible to encode a location in a map, send it to a system having another (possibly different) map and find the location back on this receiving map. Provided that both, encoder map and decoder map are meeting “navigable map” standards (in terms of accuracy and content), the encoder does not need to know about the decoder map and the decoder also does not have to care about the map used for encoding the location. The format to transmit such location reference is compact so that it can be used in systems having bandwidth restrictions.

The current OpenLR™ standard focuses on line locations (paths) in a digital map. OpenLR™ can be used for transmitting locations between different maps and one use case is the distribution of traffic information from a traffic information centre to several customers, like car drivers.

The main idea of OpenLR™ is describing the location completely with a concatenation of (several) shortest-paths. The concatenation of such shortest-paths shall cover the location completely. Each shortest-path is specified by information about its start and its end. This information is combined in so called location reference points (LRPs). The LRPs are ordered from the start of the location to the end of the location and the shortest-path between two subsequent LRPs covers a part of the location. The concatenation of all these shortest-paths will cover the location completely and this path is called the location reference path. The location reference path may be longer than the original location and offsets will trim this path down to the size of the location path.

The two location reference points describing the start and end of the location reference path are mandatory components in any OpenLR™ Location Reference. Between these two LRPs may exist several additional location reference points (intermediates) which are added to the list of LRPs if a shortest-path does not cover the location anymore. The role of the encoder is to determine the number and the positions of the location reference point to describe the location uniquely. The role of the decoder is to resolve the received location reference points to a location in its own map.

The format for transmitting location reference is also described in the OpenLR™ standard. A binary format is compact and can be used for the transmission of location reference data where the data size is important to be small. A XML format can be used if the need of having a compact format is less important and the existing environment for data transmission is already based on XML.

The data used for describing each location reference point relies on map data common for navigable digital road maps. This common data includes a geographic position and line attributes like functional road class and form of way.

The location properties and the map requirements are outlined in the following sections. The OpenLR™ standard is developed and maintained by TomTom International B.V. but it is intended as an open standard that can be used by other parties.

2.1 Location properties

The OpenLR™ standard focuses on creating map independent references for line locations. Line locations are two-dimensional objects in a network and normally consist of a list of line elements. The OpenLR™ approach is also prepared for handling point (one-dimensional objects in a network) and area locations (closed, two-dimensional objects which have an inside and an outside) but this version of OpenLR™ does not make use of this opportunity.

The line locations which can be encoded must fulfill the following requirements:

- connected and ordered list of line elements
 - the list of line elements needs to be ordered from the start of a location to the end of a location
 - two subsequent lines in the location must also be connected and adjacent in the underlying network



- if a driving direction is available then the location shall be traversable from its start to its end
- all lines in the location must have a functional road class value which has a corresponding FRC value as defined in the logical data format
 - There needs to be a proper mapping between the functional road class values in the encoder map, or the decoder map respectively, and the FRC values covering the range of navigable roads from highest to lowest importance, as defined in the logical data format. Lines which have a FRC value which is not mapped to such a value cannot be encoded.

Line locations which do not fulfill all requirements cannot be encoded using OpenLR™.

2.2 Map requirements

The encoder map and decoder map might differ but nevertheless the OpenLR™ standard provides a method to reference to the same location represented in both maps. For being able to generate a map-independent location reference and also being able to resolve locations properly a map should contain information about the following data:

- Functional road class (FRC)
 - every line in the network should have a functional road class value indicating its importance in the network
- Form of way (FOW)
 - every line in the network should have a form of way value indicating its physical properties
- Geometry
 - every line should know about its real geometry in the real world
 - lines shall not be abstracted by the airline
- Coordinates in WGS84
 - every node in the network should have coordinates in the WGS84 format
 - the preferable accuracy is decamicrodegrees for each value
- Length in meter
 - every line should have a length value in meter indicating its real dimension along the geometry

If a map provides less information than required the encoding and decoding does generally still succeed but the error rate may increase considerably.

3. OpenLR™ - Logical Data format specification

3.1 Introduction

A location reference is a description of a designated part of a digital map or a sequence of geographical positions. For this description the model of location reference points is used (LR-points, see 3.1.1).

A location reference for line locations contains at least two LR-points but there is no maximum number of LR-points defined. The location reference path is the path in the digital map described by the LR-points and can be found by a shortest-path calculation between each consecutive pair of LR-points.

The logical data format describes the logical model for location references according to the OpenLR™ standard.

3.1.1. Location Reference Point

The basis of a location reference is a sequence of location reference points (LR-points). Such a LR-point contains a coordinate pair, specified in WGS84 longitude and latitude values and additionally several attributes.

The coordinate pair (see 3.2.1) represents a geographical position within a map/network and is mandatory for a LR-point. The coordinate pair belongs to a “real” node within a network (in most cases these nodes represent junctions in the real world).

The attributes (see section 3.2.2 to 3.2.6) describe values of a line within a network at which the line is connected to the node described by the coordinate pair. In this context it is not defined if the attributes refer to an incoming or outgoing line regarding the node. This will be specified in section 3.3.

3.1.2. Topological connection of LR-points

The location reference points shall be stored in a topological order so that a point A will directly follow a point B if B also comes after A in the location reference path and there is no other LR-point in between. This topological order defines a “next point”-relationship of successive LR-points. The last point in this order will have no next point in this relationship.

Figure 1 shows an example of this relationship. The LR-points are indicated by A, B and C and the black lines and arrows indicate the order of the points from A to C in the location reference path. In this example the LR-point A will have B as next point, B will have C as next point and C will have no next point.

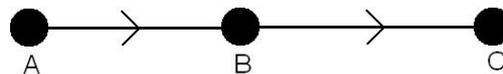


Figure 1: Connection of LR-points

3.2 Components of LR-points

This section describes the components of a location reference point.

3.2.1. Coordinate pair

Coordinate pair stands for a pair of WGS84 longitude (lon) and latitude (lat) values. This coordinate pair specifies a geometric point in a digital map. The lon and lat values are stored in a decamicrodegrees resolution (10^{-5} , five decimals).

Abbreviation: COORD

Type: (float, float)

3.2.2. Functional Road Class

The functional road class (FRC) is a road classification based on the importance of a road. The possible values of the FRC attribute are shown in Table 3, covering the range of navigable roads from highest to lowest importance. If there are fewer or more FRC values defined in the encoder map, or decoder map respectively, than these 8 values used for location referencing, then a proper mapping needs to be done or less important classes needs to be ignored.

FRC
FRC 0 – Main road
FRC 1 – First class road
FRC 2 – Second class road
FRC 3 – Third class road
FRC 4 – Fourth class road
FRC 5 – Fifth class road
FRC 6 – Sixth class road
FRC 7 – Other class road

Table 3: Logical format: Functional road class

Abbreviation: FRC

Type: integer

3.2.3. Form of way

The form of way (FOW) describes the physical road type. The possible values of the FOW attribute are shown in Table 4.

FOW	Description
UNDEFINED	The physical road type is unknown.
MOTORWAY	A Motorway is defined as a road permitted for motorized vehicles only in combination with a prescribed minimum speed. It has two or more physically separated carriageways and no single level-crossings.
MULTIPLE_CARRIAGEWAY	A multiple carriageway is defined as a road with physically separated carriageways regardless of the number of lanes. If a road is also a motorway, it should be coded as such and not as a multiple carriageway.
SINGLE_CARRIAGEWAY	All roads without separate carriageways are considered as roads with a single carriageway.
ROUNDABOUT	A Roundabout is a road which forms a ring on which traffic traveling in only one direction is allowed.
TRAFFICSQUARE	A Traffic Square is an open area (partly) enclosed by roads which is used for non-traffic purposes and which is not a Roundabout.
SLIPROAD	A Slip Road is a road especially designed to enter or leave a line.
OTHER	The physical road type is known but does not fit into one of the other categories.

Table 4: Logical Format: Form of way

Abbreviation: FOW

Type: integer

3.2.4. Bearing

The bearing (BEAR) describes the angle between the true North and a line which is defined by the coordinate of the LR-point and a coordinate which is BEARDIST along the line defined by the LR-point attributes. If the line length is less than BEARDIST then the opposite point of the line is used (regardless of BEARDIST). The bearing is measured in degrees and always positive (measuring clockwise from North). The parameter BEARDIST is defined in Table 5.

Abbreviation: BEAR

Type: integer

Abbreviation	Description	Value	Unit
BEARDIST	distance between two coordinates which form a line for the calculation of the bearing value	20	meters

Table 5: Logical format: Parameter BEARDIST

Figure 2 shows how the second point for the bearing calculation is determined. The figure shows a line from A to B which is longer than BEARDIST. The shaded part of this line is exactly BEARDIST meters long so that the point marked with B' is BEARDIST meters away from A when we walk along the line from A to B. The straight line from A to B' is now considered for the calculation of the bearing value. Note, this is different to the angle we would have calculated if we always take the opposite node of line (in this case, this would be B).

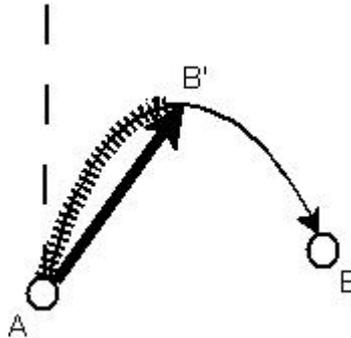


Figure 2: Logical format: Bearing point

Figure 3 shows two examples of the bearing value calculation. There are two lines, one from A to B and one from A to C. For both lines the arcs indicate the angles to the North.

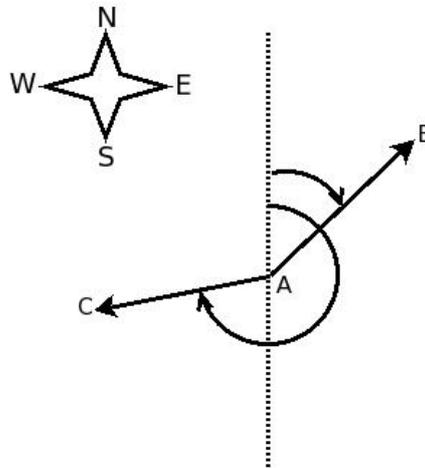


Figure 3: Logical format: Bearing

3.2.5. Distance to next LR-point

This DNP field describes the distance to the next LR-point in the topological connection of the LR-points. The distance is measured in meters and is calculated along the location reference path. The last LR-point will have the distance value 0.

Abbreviation: DNP

Type: integer

Figure 4 shows an example of the distance calculation and assignment. The three LR-points are in a sequence from A over B to C. Therefore the distance between A and B along the location reference path will be assigned to A. The LR-point B will hold the distance between B and C and the LR-point C will have a distance value of 0.

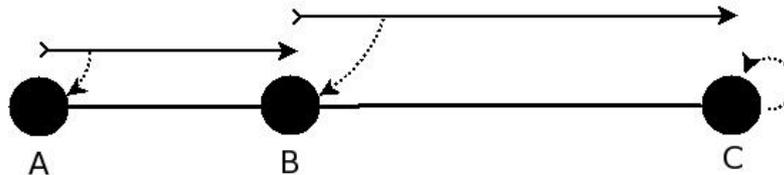


Figure 4: Logical format: Distance to next point

3.2.6. Lowest FRC to next LR-point

The lowest FRC (LFRCNP) is the lowest FRC value which appears in the location reference path between two consecutive LR-points. This information could be used to limit the number of road classes which need to be scanned during the decoding. The highest FRC value is 0 and the lowest possible FRC value is valued with 7.

Abbreviation: LFRCNP

Type: integer

3.2.7. Offsets

Offsets are used to shorten the location reference path at its start and end. The new positions along the location reference path indicate the real start and end of the location.

3.2.7.1 Positive offset

The positive offset (POFF) is the difference of the start point of the location reference and the start point of the desired location along the location reference path. The value is measured in meters. Figure 5 shows an example for the calculation of the positive and negative offset. The lines are indicating the location reference path and the hatching indicates the desired location.

Abbreviation: POFF

Type: integer

3.2.7.2 Negative offset

The negative offset (NOFF) is the difference of the end point of the desired location and the end point of the location reference along the location reference path. The value is measured in meters. Figure 5 shows an example for the calculation of the positive and negative offset. The lines are indicating the location reference path and the hatching indicates the desired location.

Abbreviation: NOFF

Type: integer



Figure 5: Logical format: Positive and negative offset

3.3 Relationship Attributes – LR-point

All attributes are linked to a LR-point. For all LR-points (except that last LR-point) the attributes describe an outgoing line of the node at the LR-point coordinate. The attributes of the last LR-point direct to an incoming line of the node at the LR-point coordinate.

Figure 6 shows an example for the relationship between a LR-point and the attributes. The lines indicate the location reference path and the nodes A, B and C are the LR-points. Note that there is also a line whose start and end node is not a LR-point (the third line in the sequence). This line does not need to be referenced because it is covered by the shortest path between the LR-points B and C.

The LR-points A and B direct to an outgoing line and the last LR-point C directs to an incoming line.

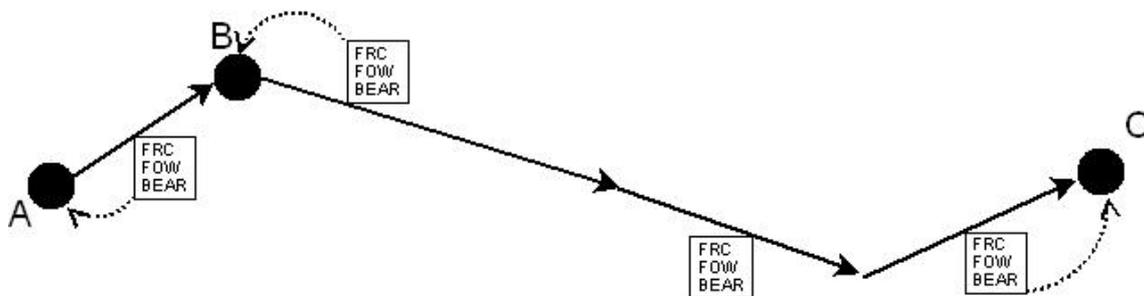


Figure 6: Relationship: Attributes – LR-point

3.4 Data format rules

The OpenLR™ rules describe additional regulations for OpenLR™ location references, supplementary to the general “shortest-path” rule for sub-dividing the location reference path. These rules are used to simplify the encoding and decoding process and to increase the accuracy of the results.

Rule – 1 *The maximum distance between two location reference points shall not exceed 15km. The distance is measured along the location reference path. If this condition is not fulfilled for a location reference then a sufficient number of additional LR-points shall be inserted.*

The maximum distance between two consecutive location reference points is restricted in order to speed up shortest-path computation because several short routes can be computed quicker than one large route if the routing algorithm has to take the whole network into account. The restriction also provides the opportunity to form a compact physical format with an acceptable accuracy.

Rule – 2 *All lengths are integer values. If there are float values available then we will round these values to get an integer representation.*

Different maps might store the length values in different formats and also with different precision and the uniform basis for all is the usage of integer values.

Rule – 3 Two LR-points are mandatory and the number of intermediate LR-points is not limited.

A line location reference must always have at least two location reference points indicating the start and the end of the location. In addition, further (mandatory) intermediate location reference points are to be included if conditions of shortest-path or Rule 1 are met. If the encoder detects critical situations where the decoder (on a different map) might get into trouble, the location reference might be enhanced with additional (optional) intermediate LR-points.

Rule – 4 The coordinates of the LR-points shall be chosen on valid network nodes.

These valid network nodes shall be junctions in the real world and it is expected that these junctions can be found in different maps with a higher probability than positions somewhere on a line. Additional nodes other than junctions (invalid nodes) shall be avoided which can be easily skipped during a route search. At these invalid nodes it is not possible to deviate from a route.

Nodes having only one incoming and one outgoing line (as directed connectivity) shall be invalid and therefore avoided since these nodes are not related to junctions (see Figure 7) and can be stepped over during route search. Nodes which have two incoming and two outgoing lines and there are only two adjacent nodes shall also be invalid, unless a U-turn is possible at that node (see Figure 8).

If one of these nodes is selected for a LR-point then this LR-point should be shifted along the location reference path in order to find a suitable node. This can be done since a route calculation will step over such invalid nodes without leaving the desired path.



Figure 7: Invalid node (1)



Figure 8: Invalid node (2)

If the start or the end of a location is placed on invalid nodes then the encoder should expand the location uniquely and should find a suitable node outside of the location. This expansion must never go into the location because this will shorten the location's spatial extent.

If the encoder cannot determine a valid node at all, then exceptionally an invalid node may be chosen for a LRP. Examples for such a case are lines which are longer than 15 km (see Rule – 1).

3.4.1. Overview of the data format rules

The following Table 6 summarizes the data format rules.

Rule	Description	Value
Rule 1	max distance between two consecutive LR-points	15000 m
Rule 2	road length values	treated as integer values
Rule 3	number of LR-points	at least two LR-points
Rule 4	invalid nodes	LR-points shall be placed on valid network nodes (also valid for start and end of a location)

Table 6: Data format rules overview

4. OpenLR™ - Binary representation

4.1 Introduction

The physical data format describes a byte-oriented stream format for the OpenLR™ data format. It uses the components described in the logical data format in section 3.

4.2 Data types

The physical data format uses the following data types. Table 7 gives an overview of all available data types and specifies the name, the type and the designated size of each data type. In the following sections the data type names are used to indicate the size and type for each data component.

Data type name	Type	Size	Range
Boolean	flag with true=1, false=0	1 bit	0 – 1
uByte	unsigned integer	1 byte	0 – 255
uShort	unsigned integer	2 bytes	0 – 65535
uSmallInt	unsigned integer	3 bytes	0 – 16777215
uInteger	unsigned integer	4 bytes	0 – 4294967295
sByte	signed integer	1 byte	-128 – 127
sShort	signed integer	2 bytes	-32768 – 32767
sSmallInt	signed integer	3 bytes	-8388608 – 8388607
sInteger	signed integer	4 bytes	-2147483648 – 2147483647
String[n]	array of n characters	n bytes	variable size
BitField[n]	array of n bits	n bits	variable size

Table 7: Physical format: Data types (Binary)

Negative integer values are stored in the two's complement format.

4.3 Coordinates (COORD)

Each point in a map consists of a coordinate pair “longitude” (lon) and “latitude” (lat) represented in WGS84 coordinates. The directions north and east are represented by positive values (longitude and latitude respectively). The lon and lat values are stored in a decamicrodegrees resolution (10^{-5} , five decimals).

The coordinate values will be transmitted as integer values. These values will be generated using Equation 1 which calculates a 24-bit integer representation. The resolution parameter is set to 24. This translation leads to a coordinate resolution (“error”) of about 2.4 meter at most. The backward translation is described in Equation 2. Both equations make use of the signum function which is -1 for negative values, 1 for positive values and 0 otherwise.

$$\text{int} = \left(\text{sgn}(\text{deg}) * 0.5 + \frac{\text{deg} * 2^{\text{Re solution}}}{360^\circ} \right)$$

Equation 1: Transformation from decimal coordinates into integer values (Binary)

$$\text{deg} = \left(\frac{(\text{int} - \text{sgn}(\text{int}) * 0.5) * 360^\circ}{2^{\text{Re solution}}} \right)$$

Equation 2: Transformation from integer values into decimal coordinates (Binary)

The physical format makes use of an absolute and a relative coordinate format. The absolute format represents the designated values of the geographical position and the relative value is the offset the coordinates relative to the preceding coordinate.

4.3.1. Absolute format

The absolute format describes geographical position in a 24-bit resolution. Table 8 shows the data type used for the absolute format.

Data type	Value	Description
sSmallInt	-8388608 – +8388607	24 bit representation

Table 8: Physical format: Coordinate format (absolute) (Binary)

4.3.2. Relative format

The relative format is used to describe differences between two consecutive coordinates. The difference is calculated for each value (lon/lat) separately as shown in Equation 3. The current and previous values represent the latitude (longitude) value in degrees. The difference between these two values is multiplied with 100000 in order to resolve an integer value.

$$relative = round(100000 * (currentPoint - previousPoint))$$

Equation 3: Relative coordinates calculation (Binary)

Table 9 shows the maximum distances which are possible using a 16-bit representation. For illustration, the ranges for relative coordinates are calculated for a given coordinate in the Netherlands at lon = 5° and lat = 52° (location in the Netherlands).

byte	latitude		longitude	
	lower bound	upper bound	lower bound	upper bound
2	-36459 m	36460 m	-22504 m	22504 m

Table 9: Physical format: Longitude/Latitude ranges for relative coordinates (Example)

Table 10 shows the data type for 2 bytes offsets.

Data type	Value	Description
sShort	-32768 – +32767	2 bytes relative coordinates

Table 10: Physical format: Coordinate format (relative) (Binary)

4.4 Attribute values

The binary format of the attributes will follow in this section.

4.4.1. Functional Road Class (FRC)

The functional road class (FRC) can hold eight different values as described in the logical format. These eight values are represented by 3 bits and the mapping is shown in Table 11.

Data type	Value (integer)	Value (binary)	Description
BitField[3]	0	000	FRC 0 – Main road, highest importance
	1	001	FRC 1 – First class road
	2	010	FRC 2 – Second class road
	3	011	FRC 3 – Third class road
	4	100	FRC 4 – Fourth class road
	5	101	FRC 5 – Fifth class road
	6	110	FRC 6 – Sixth class road
	7	111	FRC 7 – Other class road, lowest importance

Table 11: Physical format: Functional road class (Binary)

4.4.2. Form of way (FOW)

The form of way (FOW) can hold eight different values as described in the logical format. These eight values are represented by 3 bits and the mapping is shown in Table 12.

Data type	Value (integer)	Value (binary)	Description
BitField[3]	0	000	UNDEFINED
	1	001	MOTORWAY
	2	010	MULTIPLE_CARRIAGEWAY
	3	011	SINGLE_CARRIAGEWAY
	4	100	ROUNDAABOUT
	5	101	TRAFFICSQUARE
	6	110	SLIPROAD
	7	111	OTHER

Table 12: Physical format: Form of way (Binary)

4.4.3. Bearing (BEAR)

The bearing describes the angle between the true North and the road as described in the logical format. The physical data format defines 32 sectors whereby each sector covers 11.25° of the circle. These 32 sectors are represented by 5 bits. Table 13 shows the data type for the bearing attribute and Table 14 shows the mapping from the sectors to the concrete value.

Data type	Value	Description
BitField[5]	0-31	number of the sector in which the angle between the true North and the line specified in the logical data format is located; the full circle is divided into 32 sectors each covering an angle of 11.25°.

Table 13: Physical format: Bearing (Binary)

Value	Sector	Value	Sector
0	000.00° ≤ x < 011.25°	16	180.00° ≤ x < 191.25°
1	011.25° ≤ x < 022.50°	17	191.25° ≤ x < 202.50°
2	022.50° ≤ x < 033.75°	18	202.50° ≤ x < 213.75°
3	033.75° ≤ x < 045.00°	19	213.75° ≤ x < 225.00°
4	045.00° ≤ x < 056.25°	20	225.00° ≤ x < 236.25°
5	056.25° ≤ x < 067.50°	21	236.25° ≤ x < 247.50°
6	067.50° ≤ x < 078.75°	22	247.50° ≤ x < 258.75°
7	078.75° ≤ x < 090.00°	23	258.75° ≤ x < 270.00°
8	090.00° ≤ x < 101.25°	24	270.00° ≤ x < 281.25°
9	101.25° ≤ x < 112.50°	25	281.25° ≤ x < 292.50°
10	112.50° ≤ x < 123.75°	26	292.50° ≤ x < 303.75°
11	123.75° ≤ x < 135.00°	27	303.75° ≤ x < 315.00°
12	135.00° ≤ x < 146.25°	28	315.00° ≤ x < 326.25°
13	146.25° ≤ x < 157.50°	29	326.25° ≤ x < 337.50°
14	157.50° ≤ x < 168.75°	30	337.50° ≤ x < 348.75°
15	168.75° ≤ x < 180.00°	31	348.75° ≤ x < 360.00°

Table 14: Physical format: Bearing value definition (Binary)

Equation 4 outlines the calculation of the bearing value and Figure 9 provides a graphical overview of the sectors.

$$value = \left\lfloor \frac{angle}{11.25^\circ} \right\rfloor, 0^\circ \leq angle < 360^\circ$$

Equation 4: Calculation of the bearing value (Binary)

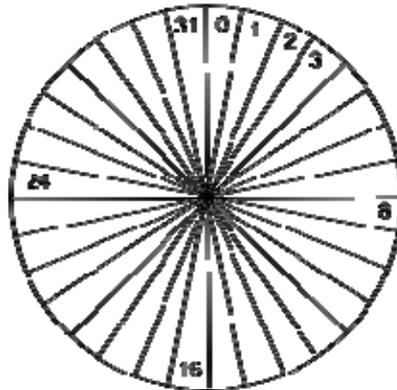


Figure 9: Physical format: Bearing sectors (Binary)

4.4.4. Distance to next LR-point (DNP)

The DNP attribute measures the distance between two consecutive LR-points along the location reference path as described in the logical format.

The physical data format defines an 8-bit representation and Table 15 shows the data type used for DNP. This representation defines 256 intervals and in combination with Rule – 1 of the data format rules (maximum length between two consecutive LR-points is limited by 15000m) each interval will have a length of 58.6 meters.

Data type	Value	Description
BitField[8]	0-255	distance interval according to Equation 5

Table 15: Physical format: Distance to next point (Binary)

Equation 5 shows how the DNP values can be calculated.

$$value = \left\lfloor \frac{length}{58.6m} \right\rfloor$$

Equation 5: Calculation of the DNP value (Binary)

4.4.5. Lowest FRC to next point (LFR CNP)

The lowest FRC to the next point indicates the lowest functional road class used in the location reference path to the next LR-point. This information could be used to limit the number of road classes which need to be scanned during the decoding. See Table 16 for a definition of the data type.

Data type	Value	Description
BitField[3]	0-7	holds the same values as described in Table 11

Table 16: Physical format: Lowest FRC to next point (Binary)

4.5 Location Reference header

The Location Reference header contains general information about the reference.

4.5.1. Version (VER)

The version is used to distinguish between several physical and data formats for location references. The version number is represented by 3 bits and the data type is shown in Table 17.

Data type	Value	Description
BitField[3]	0-7	current version number

Table 17: Physical format: Version (Binary)

Note:

The actual version of the (binary) physical data format is 2 so that the VER field is constantly set to binary 010.

Note:

The version number of the physical format is independently controlled from the version of the OpenLR™ method. An implementation of a given version of the OpenLR™ method may support multiple versions of the physical format. Likewise, a particular version of the physical format may be used throughout the evolution of the OpenLR™ across several versions of the method.

4.5.2. Attribute flag (AF)

The attribute flag indicates whether there are attributes appended to each LR-point or not. The AF value is 0 if no attributes are appended and therefore the location reference only consists of coordinates. Otherwise a value of 1 indicates that attributes are appended to each LR-point. The data type for the AF is shown in Table 18 and Table 19.

Data type	Value	Description
Boolean	0, 1	flag, indicating whether attributes are appended to each LR-point or not

Table 18: Physical format: Attribute flag (Binary)

Value	Description
0	no attributes are appended
1	for each LR-point a set of attributes is appended

Table 19: Physical format: Attribute flag values (Binary)

Note:

Since the current version of the physical format supports only location references including attributes the AF flag will constantly be set to 1.

4.5.3. Area flag (ArF)

The area flag indicates whether the location reference describes an area or not. If this flag is set then the location shall be connected and we describe an area.

Data type	Value	Description
Boolean	0, 1	flag, indicating whether the location reference describes an area or not

Table 20: Physical format: Area flag (Binary)

Value	Description
0	location reference describes no area
1	location reference describes an area

Table 21: Physical format: Area flag values (Binary)

Note:

The current physical data format supports only line locations so that the ArF is constantly set to 0.

4.6 Offsets

Offsets are used to locate the start and end of a location more precisely than bounding to the nodes in a network. The logical format defines two offsets, one at the start of the location and one at the end of the location and both offsets operate along the lines of the location and are measured in meters. The offset values are not mandatory and a missing offset value means an offset of 0 meters. Offsets are also only valid for line locations which have attributes included.

4.6.1. Offset flags

Offset flags indicate whether the data includes a specific offset information or not. The physical data format deals with two flags corresponding to the two different offset values. The positive offset flag (PoffF) and the negative offset flag (NoffF) are described in Table 22 and Table 23.

Data type	Value	Description
Boolean	0, 1	flag, indicating whether the corresponding offset value is included in the data or not

Table 22: Physical format: Offset flag (Binary)

Value	Description
0	location reference data does NOT include the corresponding offset information
1	location reference data includes the corresponding offset information

Table 23: Physical format: Offset flag values (Binary)

4.6.2. Offset values

The offset values (positive and negative, POFF and NOFF) indicate the distance between the start (end) of the location reference path and the “real” start (end) of the location.

The physical data format defines an 8-bit representation for each offset value. Table 24 shows the data type used for POFF and NOFF. This representation allows us to define 256 intervals with a length of each interval of 58.6 meters, as for Distance to next LR-Point. The interval number calculation for offsets is outlined in Equation 6.

Data type	Value	Description
BitField[8]	0-255	offset length interval according to Equation 6

Table 24: Physical format: Offset (Binary)

$$value = \left\lfloor \frac{offset\ length}{58.6m} \right\rfloor$$

Equation 6: Calculation of offset values (Binary)

5. OpenLR™ - Physical data format specification

This section describes the arrangement of the data fields in a byte stream. It is assumed that a byte-oriented stream applies and 8 bits per byte are used. The byte order being used is the big endian format. This means that the most significant byte (MSB) comes first in the sequence of data.

5.1 Overview

The main structure of the binary format is:

Header, First LR-point, following LR-points, Last LR-point, and offsets

The Header, the first LR-point and the last LR-point are mandatory and the number of following LR-points is not limited. The Last LR-point has its own structure due to a different information level. Offsets are optional and the existence will be indicated by flags in the attributes of the last LR-point.

Table 25 gives an overview of the main structure. The stream can be read from the left to the right, so that the first received byte will be the status byte. For each coordinate the first received value will be the longitude value followed by the latitude value.

The calculation of message sizes depending on the number of LR-points can be found in section 6.

Structure	Header	First LR-point					following LR-point					...
Name	Status	absolute Longitude	absolute Latitude	attr. 1	attr. 2	attr. 3	relative Longitude	relative Latitude	attr. 1	attr. 2	attr. 3	...
# bytes	1	3	3	1	1	1	2	2	1	1	1	...
description	section 5.2	section 5.3	section 5.3	section 5.5.1	section 5.5.2	section 5.5.3	section 5.4	section 5.4	section 5.5.1	section 5.5.2	section 5.5.3	...

Structure	...	last LR-point				positive offset	negative offset
Name	...	relative Longitude	relative Latitude	attr. 1	attr. 4	offset	offset
# bytes	...	2	2	1	1	1	1
description	...	section 5.3	section 5.3	section 5.5.1	section 5.5.4	section 5.6	section 5.6

Table 25: Binary format overview

5.2 Status byte

The status byte is transmitted once for every location reference and contains the area flag (ArF, section 4.5.3), attribute flag (AF, section 4.5.2) and the version information (VER, section 4.5.1). The bits 7, 6 and 5 are reserved for future use (RFU) and shall be 0. Table 26 gives an overview of the usage of each bit in the status byte.

Bit	7	6	5	4	3	2	1	0
used for	RFU	RFU	RFU	Arf	AF	VER		

Table 26: Status byte

Note:

The current format adds attributes to each LR-point and will not describe an area. The current version is 2, so that the status byte will have the value shown in Table 27:

Bit	7	6	5	4	3	2	1	0
value	0	0	0	0	1	010		

Table 27: Status byte value

5.3 First LR-point coordinates

The coordinates of the first LR-point are transmitted in an absolute format (see section 4.3.1) and therefore each value (lon and lat) will use 3 bytes. Table 28 shows for longitude and latitude values.

Bit	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
used for	highest byte									middle byte									lowest byte					

Table 28: First LR-point coordinates

5.4 Following LR-point coordinates

The coordinates of the following LR-points and the last LR-point are transmitted in a relative format (see section 4.3.2) and therefore each value (lon and lat) will use 2 bytes. Table 29 shows the byte order for longitude and latitude values.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
used for	highest byte								lowest byte							

Table 29: Following LR-points coordinates

5.5 Attributes

Attributes are added to each LR-point. There are 4 different types of attributes depending on the position of a LR-point in the location reference.

5.5.1. First attribute byte (attr. 1)

The first attribute byte contains the attributes FRC (see section 4.4.1) and FOW (see section 4.4.2) and two bits are reserved for future use. Table 30 shows the usage of each bit.

Bit	7	6	5	4	3	2	1	0
used for	RFU	RFU	FRC			FOW		

Table 30: First attribute byte – valid for all LR-points

5.5.2. Second attribute byte (attr. 2)

The second attribute byte contains the attributes LFRCNP (see section 4.4.5) and BEAR (see section 4.4.3). Table 31 shows the usage of each bit. This attribute is not valid for the last LR-point since there is no LFRCNP information available.

Bit	7	6	5	4	3	2	1	0
used for	LFRCNP				BEAR			

Table 31: Second attribute byte – valid for all LR-points, except the last LR-point

5.5.3. Third attribute byte (attr. 3)

The third attribute byte contains the attribute DNP (see section 4.4.4). This attribute is not valid for the last LR-point since there is no DNP information available.

Bit	7	6	5	4	3	2	1	0
used for	DNP							

Table 32: Third attribute byte – valid for all LR-points, except the last LR-point

5.5.4. Fourth attribute byte (attr. 4)

The attribute 4 contains the BEAR information, the positive and negative offset flags (see section 4.6.1) and one bit is reserved for future use. This attribute is used for the last LR-point.

Bit	7	6	5	4	3	2	1	0
used for	RFU	POFF	NOFF	BEAR				

Table 33: Fourth attribute bytes – valid only for the last LR-point

5.6 Offset

The positive offset (POFF) and negative offset (NOFF) are only included if the corresponding flags in attribute 4 indicate their existence. Absent offset values indicate an offset of 0 meters. The use of offset information must be handled consistently; so that the offset flags in attribute 4 are only set if the corresponding offset values exist and therefore the additional byte or bytes appear in the data stream. The offset values are calculated according to section 4.6.

Bit	7	6	5	4	3	2	1	0
used for	POFF							

Table 34: Positive offset value

Bit	7	6	5	4	3	2	1	0
used for	NOFF							

Table 35: Negative offset value

6. Code size calculation

The code size of a location reference depends on the number of LR-points included in the location reference. There must be at least two LR-points in the location reference. Also mandatory is the header with the status information. The following calculation and Table 36 show message sizes depending on the number of LR-points.

- *Header*
1 byte status
Total: 1 byte
- *First LR-Point*
6 bytes COORD (3 bytes each for lon / lat)
3 bytes Attributes
Total: 9 bytes
- *Following LR-points*
4 bytes COORD (2 bytes each for lon / lat)
3 bytes Attributes
Total: 7 bytes
- *Last LR-point*
4 bytes COORD (2 bytes each for lon / lat)
2 bytes Attributes
Total: 6 bytes
- *Offset (if included)*
1 byte positive offset (if included)
1 byte negative offset (if included)
Total: 0 – 2 bytes

# LR-points	Message size
2	16 bytes (+1 or +2 bytes offset, if included)
3	23 bytes (+1 or +2 bytes offset, if included)
4	30 bytes (+1 or +2 bytes offset, if included)
5	37 bytes (+1 or +2 bytes offset, if included)
6	44 bytes (+1 or +2 bytes offset, if included)
7	51 bytes (+1 or +2 bytes offset, if included)
8	58 bytes (+1 or +2 bytes offset, if included)
...	...
n (n>1)	1 + 9 + (n-2)*7 + 6 bytes (+1 or +2 bytes offset, if included)

Table 36: Message sizes depending on the number of LR-points

7. OpenLR™ - XML representation

7.1 Introduction

This physical data format describes a XML format for the OpenLR™ data format. It uses the components described in the logical data format in section 3.

7.2 Data types

The physical data format uses the following data types. Table 37 gives an overview of all available data types and specifies the name and the type each data type. In the following sections the data type names are used to indicate the type for each data component.

Data type name	Type
integer	The integer data type is used to specify a numeric value without a fractional component.
double	The double data type is used to specify a numeric value with a fractional component.
string	The string data type can contain characters, line feeds, carriage returns, space and tab characters.
DATA	The data type can hold any kind of XML data not specified by the schema. It allows documents to contain additional elements.
Base64Data	Binary data type to express binary-formatted data (Base64-encoded).

Table 37: Physical format: Data types (XML)

7.3 Coordinates

Each point in a map consists of a coordinate pair “longitude” (lon) and “latitude” (lat) represented in WGS84 coordinates. The directions north and east are represented by positive values (longitude and latitude respectively).

The coordinate values will be transmitted as double values. The physical format makes use of absolute coordinates which represents the designated values of the geographical position. The longitude and latitude fields are mandatory.

```
<openlr:Coordinates>
  <openlr:Longitude> [double] </openlr:Longitude>
  <openlr:Latitude> [double] </openlr:Latitude>
</openlr:Coordinates>
```

7.4 Line attribute values

Line attributes are part of a location reference point and consists of functional road class (FRC), form of way (FOW) and bearing (BEAR) data. All components are mandatory fields.

```
<openlr:LineAttributes>
  <openlr:FRC> [FRC value] </openlr:FRC>
  <openlr:FOW> [FOW value] </openlr:FOW>
  <openlr:BEAR> [integer] </openlr:BEAR>
</openlr:LineAttributes>
```

7.4.1. Functional Road Class (FRC)

The functional road class (FRC) can hold eight different values as described in the logical format. These eight values are shown in Table 38.

```
<openlr:FRC> [FRC value] </openlr:FRC>
```

Data type	Value	Description
FRC value	FRC0	Main road, highest importance
	FRC 1	First class road
	FRC 2	Second class road
	FRC 3	Third class road
	FRC 4	Fourth class road
	FRC 5	Fifth class road
	FRC 6	Sixth class road
	FRC 7	Other class road, lowest importance

Table 38: Physical format: Functional road class (XML)

7.4.2. Form of way (FOW)

The form of way (FOW) can hold eight different values as described in the logical format. These eight values are shown in Table 39.

```
<openlr:FOW> [FOW value] </openlr:FOW>
```

Data type	Value
FOW value	UNDEFINED
	MOTORWAY
	MULTIPLE_CARRIAGEWAY
	SINGLE_CARRIAGEWAY
	ROUNDBOUT
	TRAFFICSQUARE
	SLIPROAD
	OTHER

Table 39: Physical format: Form of way (XML)

7.4.3. Bearing (BEAR)

The bearing describes the angle between the true North and the road as described in the logical format. The physical data format defines the bearing field as an integer value between 0 and 360 whereby "0" is included and "360" is excluded from that range. The values should be rounded if not available as an integer.

```
<openlr:BEAR> [integer] </openlr:BEAR>
```

7.5 Path attribute values

The field path attributes is part of a location reference point (except for the last location reference point) and consists of lowest functional road class (LFRCNP) and distance to next point (DNP) data. All components are mandatory fields.

```
<openlr:PathAttributes>
  <openlr:LFRCNP> [FRC value] </openlr:LFRCNP>
```

```
<openlr:DNP> [integer] </openlr:DNP>
</openlr:PathAttributes>
```

7.5.1. Distance to next LR-point (DNP)

The DNP attribute measures the distance in meters between two consecutive LR-points along the location reference path as described in the logical format.

```
<openlr:DNP> [integer] </openlr:DNP>
```

7.5.2. Lowest FRC to next point (LFRCNP)

The lowest FRC to the next point indicates the lowest functional road class used in the location reference path to the next LR-point. This information could be used to limit the number of road classes which need to be scanned during the decoding. See Table 38 for a definition of the data type.

```
<openlr:LFRCNP> [FRC value] </openlr:LFRCNP>
```

7.6 Line location reference

A line location reference describes a path within a map and consists of location reference point(s), a last location reference point and offset data. There must be at least one location reference point and exactly one last location reference point. The offset field is optional.

```
<openlr:LineLocationReference>
  [LocationReferencePoint(s)]
  [LastLocationReferencePoint]
  [Offsets]
</openlr:LineLocationReference>
```

7.6.1. Location reference point

A location reference point consists of coordinate, line attribute and path attribute data. All fields are mandatory.

```
<openlr:LocationReferencePoint>
  [Coordinate]
  [LineAttributes]
  [PathAttributes]
</openlr:LocationReferencePoint>
```

7.6.2. Last location reference point

A last location reference point consists of coordinate and line attribute data. All fields are mandatory.

```
<openlr>LastLocationReferencePoint>
  [Coordinate]
  [LineAttributes]
</openlr>LastLocationReferencePoint>
```



7.6.3. Offsets

Offsets are used to locate the start and end of a location more precisely than bounding to the nodes in a network. The logical format defines two offsets, one at the start of the location and one at the end of the location and both offsets operate along the lines of the location and are measured in meters. The offset values are optional and a missing offset value means an offset of 0 meters. If there are no offsets at all the offset field can be omitted.

```
<openlr:Offsets>
  <openlr:PosOff> [integer] </openlr:PosOff>
  <openlr:NegOff> [integer] </openlr:NegOff>
</openlr:Offsets>
```

7.7 XML location reference

A XML location reference currently consists of a line location reference. This inner field is mandatory.

```
<openlr:XMLLocationReference>
  <openlr:LineLocationReference> [LineLocationReference]
</openlr:LineLocationReference>
</openlr:XMLLocationReference>
```

7.8 Binary location references

The field binary location references holds the binary representation of the same location reference. There may exist several different versions of the binary format but at least one binary location reference is mandatory.

```
<openlr:BinaryLocationReferences>
  [BinaryLocationReference(s)]
</openlr:BinaryLocationReferences>
```

7.8.1. Binary location reference

A binary location reference represents the binary format of the same location reference as described in this XML representation. This field contains the Base64-encoded binary data. The attributes id and version are mandatory whereby the id identifies this specific binary location reference in the list of possibly several binary location references and the version expresses the binary format version being used.

```
<openlr:BinaryLocationReference id="[name]" version="[version]">
  [Base64Data]
</openlr:BinaryLocationReference>
```

7.9 Additional information

The additional information field consists of extra information not being relevant for the OpenLR™ location reference data. This field is optional and also all inner fields are optional. If no additional data is specified this field can be omitted. This field consists of bounding box, map database and any other kind of XML data.

```
<openlr:AdditionalInformation>
  <openlr:BoundingBox> [BoundingBox] </openlr:BoundingBox>
  <openlr:MapDatabase> [string] </openlr:MapDatabase>
  <openlr:AdditionalData> [DATA] </openlr:AdditionalData>
```



```
</openlr:AdditionalInformation>
```

7.9.1. Bounding box

The (optional) bounding box describes a rectangle which covers the location reference path completely. The location reference path is the path defined by the location reference points so that this path might be longer than the location. The bounding box is defined by the coordinates of its lower left and upper right corner. Both coordinates are mandatory fields and consists of a longitude and latitude pair.

```
<openlr:BoundingBox>
  <openlr:LowerLeft>
    <openlr:Longitude> [double] </openlr:Longitude>
    <openlr:Latitude> [double] </openlr:Latitude>
  </openlr:LowerLeft>
  <openlr:UpperRight>
    <openlr:Longitude> [double] </openlr:Longitude>
    <openlr:Latitude> [double] </openlr:Latitude>
  </openlr:UpperRight>
</openlr:BoundingBox>
```

7.9.2. Map database

The (optional) map database field provides information about the map being used during decoding. The information can be of any type but must be expressible as a string.

```
<openlr:MapDatabase> [string] </openlr:MapDatabase>
```

7.9.3. Additional data

The (optional) additional data field makes the XML format extendable with any kind of other XML data not specified in the XML format schema definition. This field may be used to transmit additional data like debug data or decoding hints.

```
<openlr:AdditionalData> [DATA] </openlr:AdditionalData>
```

7.10 Root tag: OpenLR

The root tag of the XML format is OpenLR. This tag includes the namespace definitions and schema locations. The tag consists of a location id and XML location reference, binary location reference and additional information data. The location id is mandatory; all other data fields are optional. But there should be at least one of the data fields XML location reference or binary location references.

```
<?xml version="1.0" encoding="UTF-8"?>
<openlr:OpenLR xmlns:openlr=http://www.openlr.org/openlr
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openlr.org/openlr openlr.xsd ">
  [LocationID]
  [XMLLocationReference]
  [BinaryLocationReferences]
  [AdditionalInformation]
</openlr:OpenLR>
```



7.10.1. Location ID

The location id identifies this location reference uniquely. The identifier can be of any type but must be expressible as a string.

```
<openlr:LocationID> [string] </openlr:LocationID>
```

8. OpenLR™ - Physical data format specification (XML)

The XML format for OpenLR™ location references is defined by the following XML schema definition. The schema includes additional documentation.

Figure 10 shows the structure of such XML documents in a graphical way.

```

<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:openlr="http://www.openlr.org/openlr" attributeFormDefault="unqualified"
  elementFormDefault="qualified" targetNamespace="http://www.openlr.org/openlr">

  <xs:annotation>
    <xs:documentation xml:lang="en">
      This schema describes the XML format for OpenLR location references.
      This format includes a XMLLocationReference but also supports
      BinaryLocationReference. The XMLLocationReference shall make use of the
      raw data for any type so that no compression techniques are used (e.g.
      use of intervals for the description of length values). The BinaryLocationReference
      shall represent the compressed format of the location reference.
      For further information on the relevant fields see the OpenLR
      white paper at http://www.openlr.org.

      OpenLR is a trade mark of TomTom International B.V.
      author: TomTom International B.V.
    </xs:documentation>
  </xs:annotation>

  <xs:simpleType name="FRC_Type">
    <xs:annotation>
      <xs:documentation xml:lang="en">
        The functional road class is a road classification based
        on the importance of a road.
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="FRC0" />
      <xs:enumeration value="FRC1" />
      <xs:enumeration value="FRC2" />
      <xs:enumeration value="FRC3" />
      <xs:enumeration value="FRC4" />
      <xs:enumeration value="FRC5" />
      <xs:enumeration value="FRC6" />
      <xs:enumeration value="FRC7" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="FOW_Type">
    <xs:annotation>
      <xs:documentation xml:lang="en">
        The form of way describes the physical road type.
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="UNDEFINED" />
      <xs:enumeration value="MOTORWAY" />
      <xs:enumeration value="MULTIPLE_CARRIAGEWAY" />
      <xs:enumeration value="SINGLE_CARRIAGEWAY" />
      <xs:enumeration value="ROUNDAABOUT" />
      <xs:enumeration value="TRAFFICSQUARE" />
      <xs:enumeration value="SLIPROAD" />
      <xs:enumeration value="OTHER" />
    </xs:restriction>
  </xs:simpleType>

```

```

<xs:simpleType name="Bearing_Type">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      The bearing describes the angle between the true North
      and a line. The values range from 0 to 359 degree.
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0" />
    <xs:maxExclusive value="360" />
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="LineAttributes">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Line attributes describe features of a line in a network. The
      attributes the functional road class, form of way and the bearing
      of a line.
    </xs:documentation>
  </xs:annotation>
  <xs:all>
    <xs:element name="FRC" type="openlr:FRC_Type" />
    <xs:element name="FOW" type="openlr:FOW_Type" />
    <xs:element name="BEAR" type="openlr:Bearing_Type" />
  </xs:all>
</xs:complexType>

<xs:complexType name="PathAttributes">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Path attributes describes features of a path. The LFRCNP attribute
      is the lowest functional road class value which appears in the path.
      The highest FRC value is FRC0 and the lowest possible FRC value
      is FRC7.
    </xs:documentation>
  </xs:annotation>
  <xs:all>
    <xs:element name="LFRCNP" type="openlr:FRC_Type" />
    <xs:element name="DNP" type="xs:integer" />
  </xs:all>
</xs:complexType>

<xs:complexType name="Coordinates">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Coordinates stands for a pair of WGS84 longitude and latitude
      values. This coordinate pair specifies a geometric point in a
      digital map.
    </xs:documentation>
  </xs:annotation>
  <xs:all>
    <xs:element name="Longitude" type="xs:double" />
    <xs:element name="Latitude" type="xs:double" />
  </xs:all>
</xs:complexType>

<xs:complexType name="BoundingBox">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      The bounding box describes a rectangle which covers the location
      completely. The rectangle is defined by the coordinates of the
      lower left and upper right corners.
    </xs:documentation>
  </xs:annotation>
  <xs:all>
    <xs:element name="LowerLeft" type="openlr:Coordinates" />
    <xs:element name="UpperRight" type="openlr:Coordinates" />
  </xs:all>
</xs:complexType>

```



```

</xs:all>
</xs:complexType>

<xs:complexType name="Offsets">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      Offsets are used to shorten the location reference path at its
      start and end. The new positions along the location reference
      path indicate the real start and end of the location. The positive
      offset is the difference of the start point of the location
      reference and the start point of the desired location along the
      location reference path. The negative offset is the difference
      of the end point of the desired location and the end point of the
      location reference along the location reference path. Both values
      are measured in meter.
    </xs:documentation>
  </xs:annotation>
  <xs:all>
    <xs:element minOccurs="0" name="PosOff" type="xs:integer" />
    <xs:element minOccurs="0" name="NegOff" type="xs:integer" />
  </xs:all>
</xs:complexType>

<xs:complexType name="AdditionalData">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      AdditionalData can be used to add any other kind of information
      needed like debug information or additional encoder/decoder flags.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:any minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="AdditionalInformation">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      AdditionalInformation can be added if necessary. This includes
      the bounding box covering the location and information on the map
      used for encoding. The AdditionalData section also provides the freedom
      to add any other information needed.
    </xs:documentation>
  </xs:annotation>
  <xs:all>
    <xs:element minOccurs="0" name="BoundingBox" type="openlr:BoundingBox" />
    <xs:element minOccurs="0" name="MapDatabase" type="xs:string" />
    <xs:element minOccurs="0" name="AdditionalData" type="openlr:AdditionalData" />
  </xs:all>
</xs:complexType>

<xs:complexType name="LocationReferencePoint">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      The basis of a location reference is a sequence of location
      reference points (LRPs). Such a LRP contains a coordinate pair,
      specified in WGS84 longitude and latitude values and additionally
      several line and path attributes.
    </xs:documentation>
  </xs:annotation>
  <xs:all>
    <xs:element minOccurs="1" name="Coordinates" type="openlr:Coordinates" />
    <xs:element minOccurs="1" name="LineAttributes" type="openlr:LineAttributes" />
    <xs:element minOccurs="1" name="PathAttributes" type="openlr:PathAttributes" />
  </xs:all>
</xs:complexType>

<xs:complexType name="LastLocationReferencePoint">

```

```

<xs:annotation>
  <xs:documentation xml:lang="en">
    The sequence of location reference points is terminated by a last
    location reference point. This LRP does not need to have any
    path attributes.
  </xs:documentation>
</xs:annotation>
<xs:all>
  <xs:element minOccurs="1" name="Coordinates" type="openlr:Coordinates" />
  <xs:element minOccurs="1" name="LineAttributes" type="openlr:LineAttributes" />
</xs:all>
</xs:complexType>

<xs:complexType name="BinaryLocationReference">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      The BinaryLocationReference is the Base64-encoded binary representation
      of the location reference. Each binary location reference needs
      to have an identifier and a version. The version is defined by the
      version of the binary physical format.
    </xs:documentation>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="xs:base64Binary">
      <xs:attribute name="id" type="xs:string" use="required" />
      <xs:attribute name="version" type="xs:string" use="required" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:complexType name="LineLocationReference">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      A LineLocationReference is defined by an ordered sequence of
      location reference points and a terminating last location
      reference point.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded"
      name="LocationReferencePoint" type="openlr:LocationReferencePoint" />
    <xs:element minOccurs="1" maxOccurs="1"
      name="LastLocationReferencePoint" type="openlr:LastLocationReferencePoint" />
    <xs:element minOccurs="0" name="Offsets" type="openlr:Offsets" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="BinaryLocationReferences">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      It may exist several different formats for binary location references
      and all formats or a subset can be included.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="unbounded"
      name="BinaryLocationReference" type="openlr:BinaryLocationReference" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="XMLLocationReference">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      A XMLLocationReference describes an OpenLR location reference.
      Currently only line locations are defined but other location types
      may follow.
    </xs:documentation>
  </xs:annotation>

```



```

<xs:all>
  <xs:element minOccurs="1" maxOccurs="1"
    name="LineLocationReference" type="openlr:LineLocationReference" />
</xs:all>
</xs:complexType>

<xs:element name="OpenLR">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      An OpenLR location reference consists of a unique identifier and
      a location reference as a XMLLocationReference, BinaryLocationReference
      or both. The schema accepts OpenLR tags without any location reference
      but these tags will be ignored.
      The XMLLocationReference shall make use of the
      raw data for any type so that no compression techniques are used (e.g.
      use of intervals for the description of length values). The
      BinaryLocationReference shall represent the compressed format of
      the location reference. If both types are used the location
      reference must represent the same location.
      Additional information may be added if necessary.
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="LocationID"
        type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1"
        name="XMLLocationReference" type="openlr:XMLLocationReference" />
      <xs:element minOccurs="0" maxOccurs="1"
        name="BinaryLocationReferences" type="openlr:BinaryLocationReferences" />
      <xs:element minOccurs="0" maxOccurs="1"
        name="AdditionalInformation" type="openlr:AdditionalInformation" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>

```

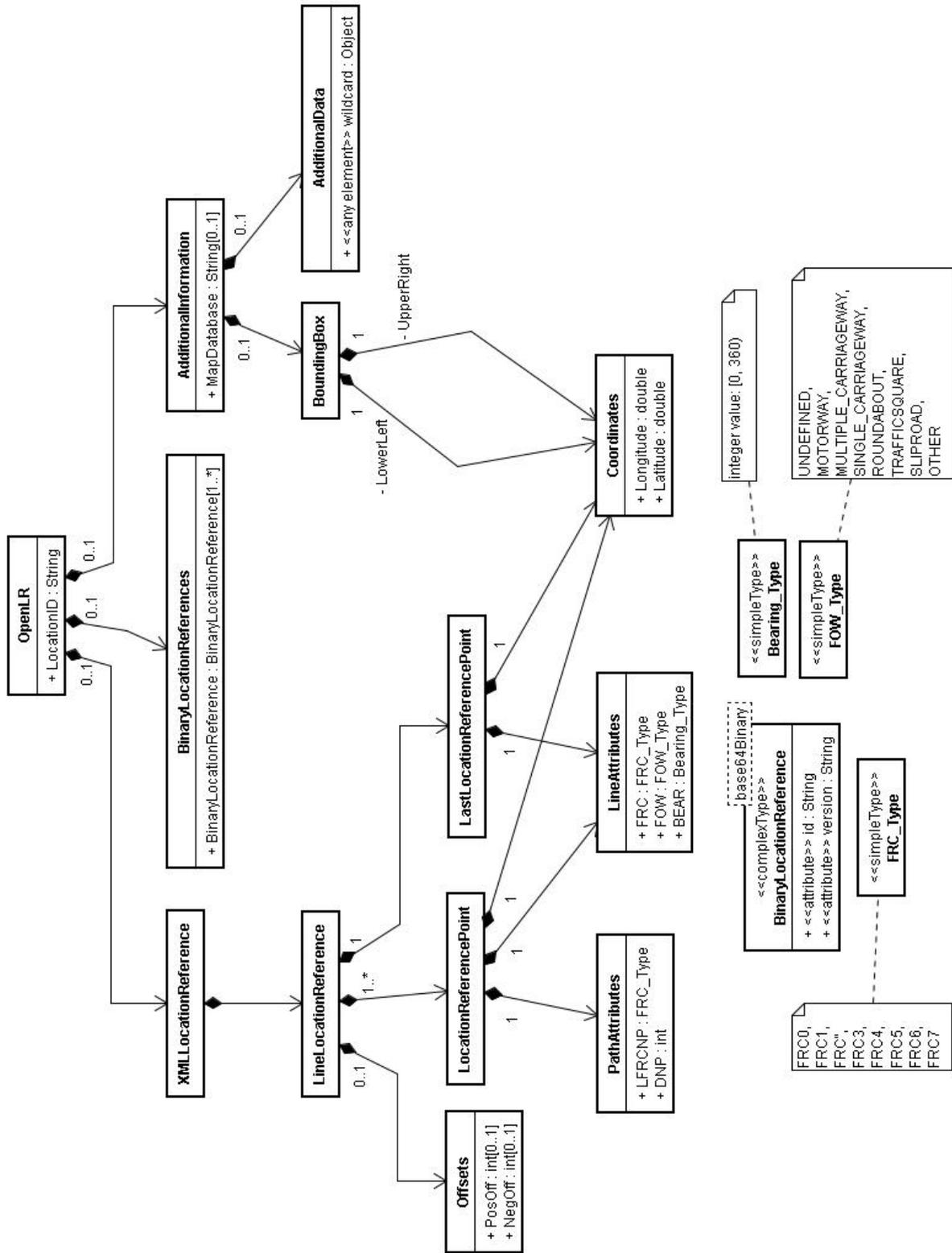


Figure 10: XML structure overview

9. Encoding locations

The OpenLR™ encoder generates a map-independent location reference for a (map-dependent) location. The output of this encoder can be used to distribute location information to other parties and the decoder side will be able to resolve the location on its own map.

9.1 Encoding steps

The encoding steps 1 to 10 describe the process to generate a location reference for a location. The location consists of an ordered list of lines and it will be checked whether this location is valid for encoding or not. The following steps act as a guideline.

Step – 1 *Check validity of the location to be encoded.*

A location is valid if the following constraints are fulfilled:

- The location is a connected path.
- The location is traversable from its start to its end.
- All lines of the location have a functional road class between 0 and 7 (including 0 and 7).

Optionally the location can be checked against turn restrictions. This check can only be made if turn restriction information is supported by the map and if this check is enabled. Checking of turn restrictions is only made in the location validation check and does not affect the location generation phase. If the location is not valid the encoder should fail.

Step – 2 *Adjust start and end node of the location to represent valid map nodes.*

OpenLR™ places location reference points on valid nodes. Valid nodes are such nodes where a shortest-path calculation needs to decide between several different ways. Otherwise invalid nodes are such nodes where a shortest path calculation can step over (see also section 3.4).

Since the start and end of a location will become a location reference point these nodes need to be adjusted to valid nodes if necessary (expansion of location). The real start of the location can then be referenced using offsets (positive offset for the start node and negative offset for the end node, respectively).

If adjusting the nodes to valid nodes fails then the encoder could proceed with the next steps and does not need to fail.

Step – 3 *Determine coverage of the location by a shortest-path.*

Calculate a shortest-path between the current start line and the end line of the location. If no intermediate location reference point was detected so far, the current start line is identical to the start line of the location. If an intermediate location reference point was detected in a previous step, then the line corresponding to the intermediate location reference point acts as current start line. The start line is always part of the location.

The shortest path algorithm should take the whole network into account in order to calculate a shortest path between the current start and end. Additionally it should fulfill the following constraints:

- All lengths of the lines should be measured in meter and should also be converted to integer values, so that float values need to be rounded correctly.
- The search is node based and will start at the start node of the first line and will end at the end node of the last line.
- The algorithm shall return an ordered list of lines representing the calculated shortest-path.

If no shortest-path can be calculated the encoding should fail.

Step – 4 *Check whether the calculated shortest-path covers the location completely or not. Go to step 5 if the location is not covered completely, go to step 7 if location is covered.*

Compare the calculated shortest-path against the part of the location which is currently not covered by a shortest path in a previous step. The shortest-path covers the location if the following constraints are fulfilled:

- Every line of the location (between current start and end) needs to be part of the shortest-path.
- The order of the lines in the location (between current start and end) needs to be the same order as the lines in the shortest-path.

If one of the constraints is not fulfilled go to step 5 in order to determine a proper intermediate location reference point. If all constraints are fulfilled then go to step 7.

Step – 5 *Determine position of a new intermediate location reference point so that the part of the location between the start of the shortest-path calculation and the new intermediate is covered completely by a shortest-path.*

If the location (between current start and end) is not fully part of the shortest-path or the order of the lines is mixed up then a proper intermediate location reference point needs to be determined. This intermediate must fulfill the following constraints:

- The shortest-path between the current start and the line indicated by the intermediate location reference point must cover the corresponding location completely.
- The start node of the line indicated by the intermediate location reference point shall be positioned on a valid node (if no valid node can be determined, an invalid node can be chosen).

Step – 6 *Go to step 3 and restart shortest path calculation between the new intermediate location reference point and the end of the location.*

The remaining part of the location still needs to be covered by a shortest-path. Go to step 3 and calculate a shortest-path between the last intermediate found and the end of the location.

Step – 7 *Concatenate the calculated shortest-path(s) for a complete coverage of the location and form an ordered list of location reference points (from the start to the end of the location).*

The calculated shortest-paths(s) need to be concatenated so that the resulting location reference path covers the location completely. According to the concatenation of the path an ordered list of the location reference points needs to be formed, starting with the start of the location, followed by all intermediates, ending with the end of the location.

All relevant attributes for a location reference point need to be collected and calculated. This includes the functional road class, the form of way, the bearing of the corresponding line, and the distances and lowest functional road class between two subsequent location reference points.

Step – 8 *Check validity of the location reference path. If the location reference path is invalid then go to step 9, if the location reference path is valid then go to 10.*

The location reference as a list of ordered location reference points will be checked in order to fulfill the following constraints:

- The distance between two subsequent location reference points shall not exceed the maximum distance value as defined in the data format rules (see 3.4).

If the location reference does not fulfill all constraints then go to step 9, otherwise go to step 10.



The location reference may contain additional location reference points if necessary. The encoder may detect situations where it might be difficult for a decoder to choose the correct path. In this case additional intermediate location reference points are allowed. The intermediate must fulfill the same constraints as defined in step 5.

Step – 9 *Add a sufficient number of additional intermediate location reference points if the distance between two location reference points exceeds the maximum distance.*

In order to fulfill the constraints in step 8 additional intermediate location reference points need to be added. If the maximum distance between two subsequent location reference points is exceeded additional location reference points shall be placed at valid nodes along the location reference path between these two location reference points. If placing on valid nodes is not possible an invalid node shall be used. Examples for such a case are lines which are longer than 15 km. In such a case the coordinates of additional LRP may be resolved from a point being less than 15 km away from the start node of that line. Such a point is directly on the line and is not represented by a node. All LRP attributes need to be calculated relative to this point.

The calculated attributes need to be updated and the new intermediate location reference point must be added to the list without breaking the order from start to end. The process of adding intermediates needs to be repeated until all constraints in step 8 are fulfilled.

Step – 10 *Create physical representation of the location reference.*

The ordered list of location reference points needs to be transformed into a physical representation (see section 4 and section 7).

After processing all necessary steps the location reference defines a location reference path which covers the location completely. If the encoding of such a location fails, no location reference can be created and the encoder should report on this.

9.1.1. Overview of the encoding steps

The following Table 40 summarizes the encoding rules and Figure 11 outlines the workflow for OpenLR™ encoding.

Step	Action
Step – 1	<i>Check validity of the location to be encoded</i>
Step – 2	<i>Adjust start and end node of the location to represent valid map nodes</i>
Step – 3	<i>Determine coverage of the location by a shortest-path</i>
Step – 4	<i>Check whether the calculated shortest-path covers the location completely or not. Go to step 5 if the location is not covered completely, go to step 7 if location is covered</i>
Step – 5	<i>Determine position of a new intermediate location reference point so that the part of the location between the start of the shortest-path calculation and the new intermediate is covered completely by a shortest-path.</i>
Step – 6	<i>Go to step 3 and restart shortest path calculation between the new intermediate location reference point and the end of the location.</i>
Step – 7	<i>Concatenate the calculated shortest-path(s) for a complete coverage of the location and form an ordered list of location reference points (from the start to the end of the location)</i>
Step – 8	<i>Check validity of the location reference path. If location reference path is invalid then go to step 9, if location reference path is valid then go to 10.</i>
Step – 9	<i>Add a sufficient number of additional intermediate location reference points if the distance between two location reference points exceeds the maximum distance.</i>
Step – 10	<i>Create physical representation of the location reference</i>

Table 40: Encoding steps overview

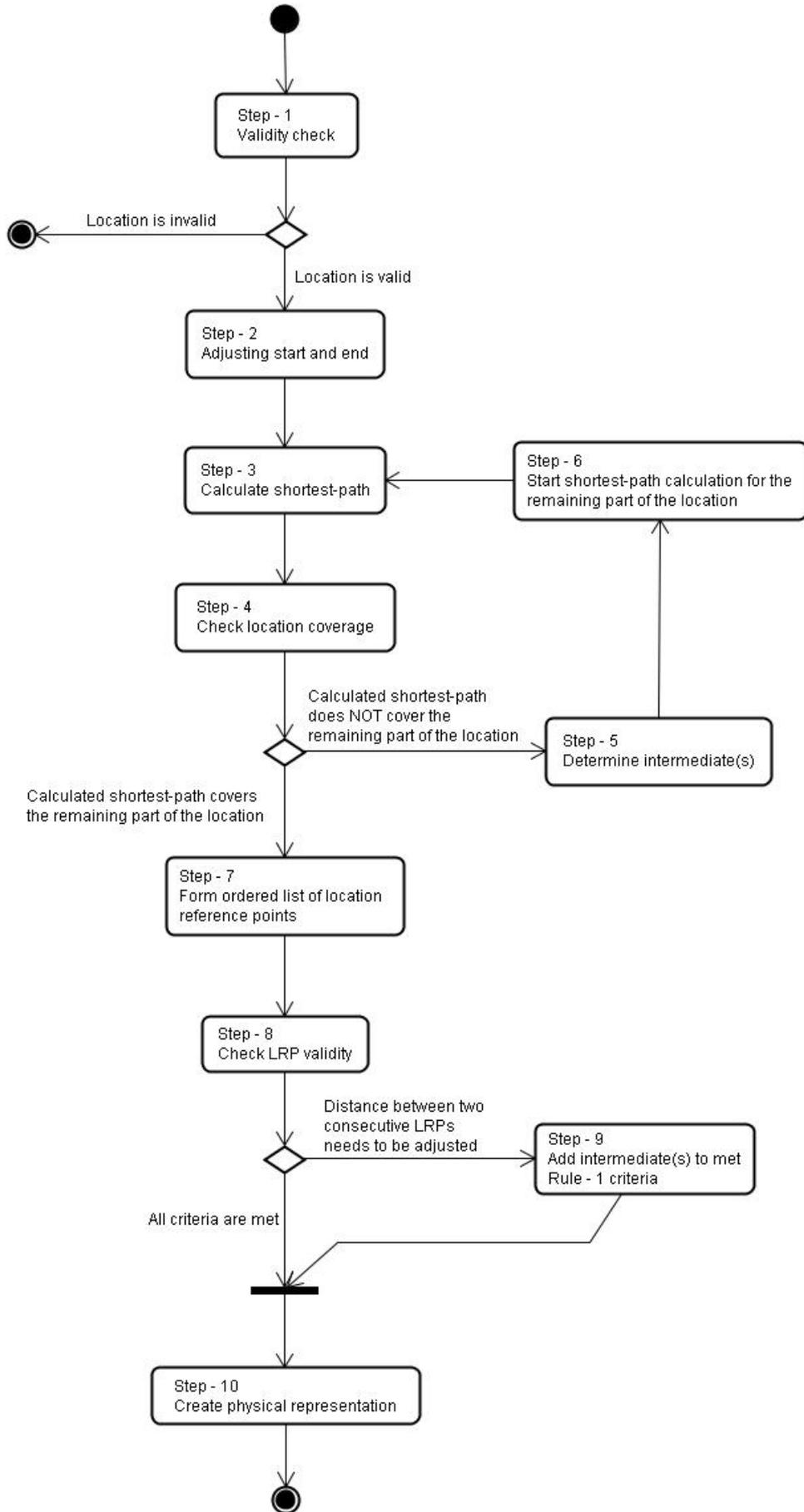


Figure 11: Workflow – OpenLR™ encoding



10. Decoding location references

The OpenLR™ decoder retrieves a map-independent location reference which was encoded by an OpenLR™ encoder described in the previous section. The decoder resolves a (map-dependent) location using its own map. This map might differ to the one used during encoding.

10.1 Decoding steps

The decoding steps 1 to 7 describe the process of resolving a location from a location reference. The location reference comes in a physical representation as described in section 4 or section 7. The decoder resolves a location in the decoder map or it reports on errors if no location can be resolved. The following steps act as a guideline.

Step – 1 *Decode physical data and check its validity*

The decoder receives a physical representation of a location reference and it needs to check whether the data is valid. If a check fails the decoder should report an error and stop the decoding process.

The decoder should also extract the physical data including offset information and resolve the interval information into lower and upper boundaries. The relative coordinates should be converted into absolute coordinates.

Step – 2 *For each location reference point find candidate nodes*

Each location reference point contains coordinates specifying a node in the encoder map. The decoder should try to find so called candidate nodes in the decoder map whereby the coordinates of the candidate nodes are close by the coordinates of the location reference point coordinates. The distance should be the airline distance between the coordinates.

It might happen that several candidate nodes for one location reference point exist. Nodes in the decoder map which are far away from the location reference point coordinates should not be considered as candidate nodes in the further processing.

If no candidate node can be determined for a location reference point the decoder should try to determine a candidate line directly. The LRP coordinate can be projected onto lines which are not far away from that coordinate. The following steps should take the projection points into account instead of one of the start or end node of the determined lines.

Step – 3 *For each location reference point find candidate lines*

For each location reference point the decoder tries to determine lines which should fulfill the following constraints:

- The start node, end node for the last location reference point or projection point shall be close by the coordinates of the location reference point.
- The candidate lines should be outgoing lines (incoming lines for the last location reference point) of the candidate nodes or projection points determined in the previous step
- The candidate lines should match the attributes functional road class, form of way and bearing as being extracted from the physical data. Slight variances are allowed and shall be taken into account in step 4.

If no candidate line can be found for a location reference point, the decoder should report an error and stop further processing.

Step – 4 *Rate candidate lines for each location reference point*

All candidate lines for a location reference point shall be rated according to the following criteria:

- The start node, end node for the last location reference point or projection point shall be as close by as possible to the coordinates of the location reference point
- The functional road class of the candidate line should match the functional road class of the location reference point
- The form of way of the candidate line should match the form of way of the location reference point
- The bearing of the candidate line should match indicated bearing angles of the location reference point
- Slight variances in the concrete values are allowed and shall be considered in the rating function.

The candidate lines should be ordered in a way that the best matching line comes first.

Step – 5 *Determine shortest-path(s) between two subsequent location reference points*

The decoder needs to compute a shortest-path between two subsequent location reference points. For each location reference point a suitable candidate line must be chosen and the candidate line of the preceding location reference point acts as start for the shortest-path calculation. The candidate line of the second location reference point is the end of the shortest-path calculation. If the lines being chosen are equal, then no shortest-path calculation needs to be started.

The shortest path algorithm should take the part of the network into account which contains of all lines having a functional road class better or equal than the lowest functional road class of the preceding location reference point. This value might be altered if the decoder anticipates having different functional road class values than the encoder map.

Additionally the shortest-path algorithm should fulfill the following constraints:

- All lengths of the lines should be measured in meter and should also be converted to integer values, so that float values need to be rounded correctly
- The search is node based and will start at the start node of the first line and will end at the end node of the last line
- The algorithm shall return an ordered list of lines representing the calculated shortest-path

If no shortest-path can be calculated for two subsequent location reference points, the decoder might try a different pair of candidate lines or finally fail and report an error. If trying a different pair of candidate lines it might happen that the start line needs to be changed. In such a case this might also affect the end line of the previous shortest-path and this path also needs to be re-calculated and checked again. The number of retries of shortest-path calculations should be limited in order to guarantee a fast decoding process.

Step – 6 *Check validity of the calculated shortest-path(s)*

After the shortest-path calculation the length of such a path should be checked against the distance to next point information of the preceding location reference point. If the length information differ to much the decoder could decide to try a different pair of candidate lines (see also Step – 5) or to fail and report an error.

Step – 7 *Concatenate shortest-path(s) to form the location and trim path according to the offsets*

If all shortest-paths are calculated and checked the decoder concatenates the shortest-path(s) according to the order of the location reference points. This concatenation shall be trimmed using the positive and negative offset of the physical data. The resulting path forms the decoder location.

10.1.1. Overview of the decoding steps

The following Table 41 summarizes the decoding steps and Figure 12 outlines the workflow of OpenLR™ decoding.

Step	Action
Step – 1	<i>Decode physical data and check its validity</i>
Step – 2	<i>For each location reference point find candidate nodes</i>
Step – 3	<i>For each location reference point find candidate lines</i>
Step – 4	<i>Rate candidate lines for each location reference point</i>
Step – 5	<i>Determine shortest-path(s) between two subsequent location reference points</i>
Step – 6	<i>Check validity of the calculated shortest-path(s)</i>
Step – 7	<i>Concatenate shortest-path(s) to form the location and trim path according to the offsets</i>

Table 41: Decoding steps overview

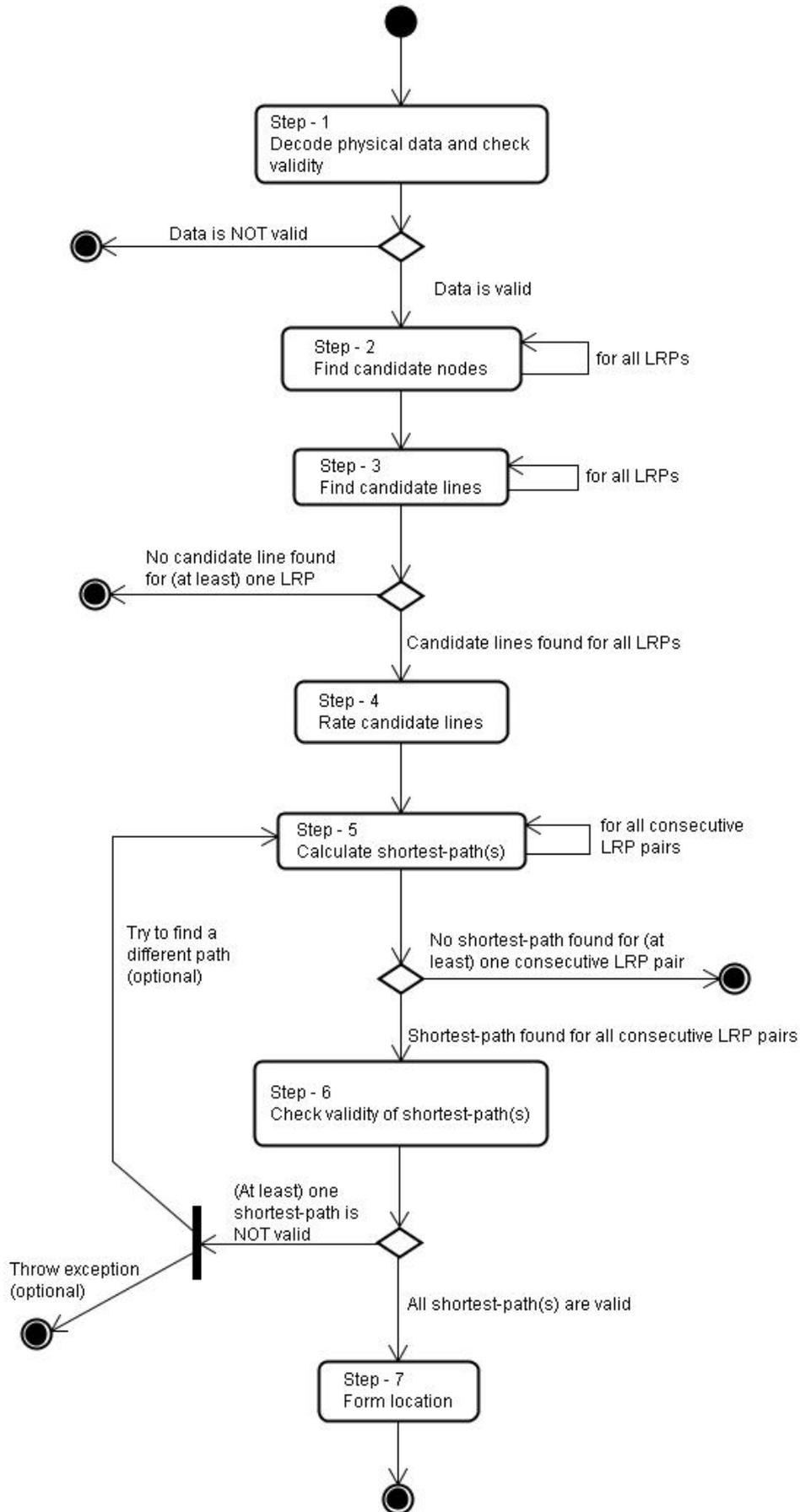


Figure 12: Workflow – OpenLR™ decoding



11. OpenLR™ example

OpenLR™ shall now be explained using an abstract example. The example defines two different maps (encoder map in Figure 13 and decoder map in Figure 17) and it executes the encoder steps as described in section 9.1, outlines the binary and XML representation as defined in section 4 and section 7 and it executes the decoder steps as described in section 10.1.

11.1 Example setup

The encoder map is shown in Figure 13 and consists of 15 nodes and 23 lines (two-way lines are counted twice). The nodes are numbered from 1 to 15. The necessary line attributes are shown beside every line using the format: <FRC>, <FOW>, <Length in meter>. The arrows indicate the driving direction for each line. The coordinates chosen for the OpenLR example are exemplary and do not correspond to any real world objects!

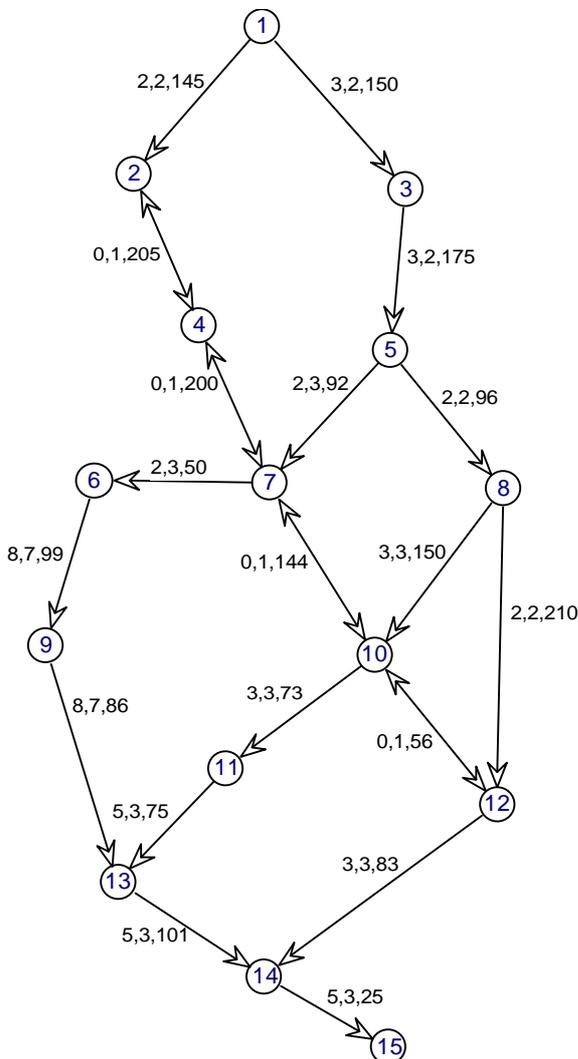


Figure 13: Example network

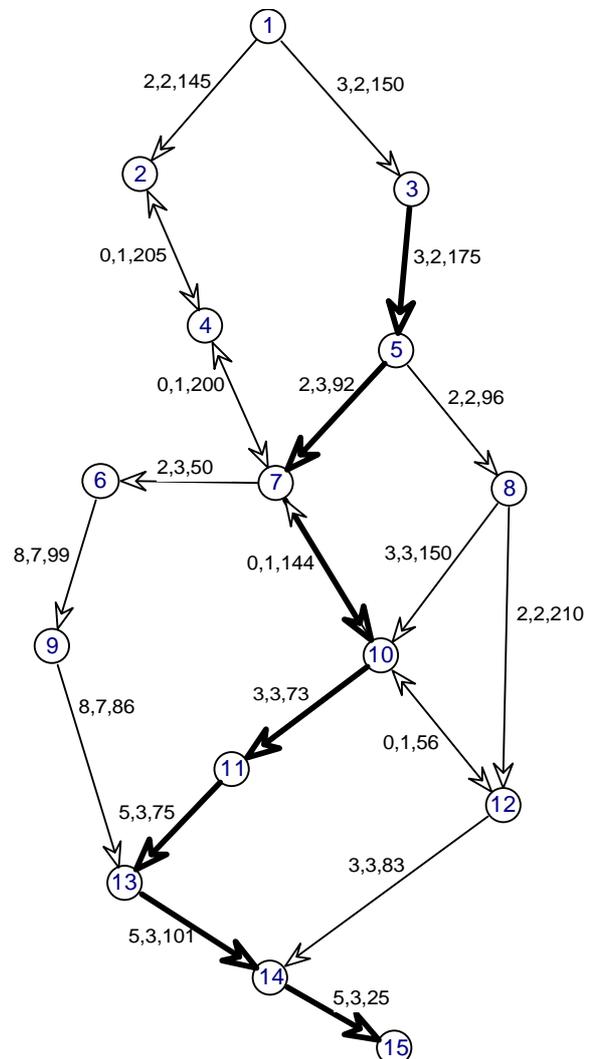


Figure 14: Location

The location to be encoded is shown in Figure 14 using bold lines. The location starts at node 3 and continues over the nodes 5, 7, 10, 11, 13, 14, and ends at node 15. Its total length in the encoder map is 685 meters. The ordered list of lines and the map to be used during encoding serves as input for the OpenLR™ encoder.

11.2 Encoding

In step 1 of the encoding process the location will first be checked for validity. Since the location is connected and drivable and all functional road classes along the location are between 0 and 7, this location is valid. Given that turn restrictions are omitted from the map data used for this example, the encoder can ignore this check.

The encoder step 2 checks the start and end node of the location being valid nodes according to the data format rules in section 3.4. The end node 15 has only one incoming line and is therefore valid.

The start node 3 also has two incident lines but here it is one outgoing and one incoming line. Therefore this node is not valid and the encoder searches for a valid node outside the location. The encoder will find node 1 to be a valid node and it also expands the location uniquely. Node 1 is chosen as the new start node for the location reference and there will be a positive offset of 150 meters. The total length of the location reference path results in 835 meters.

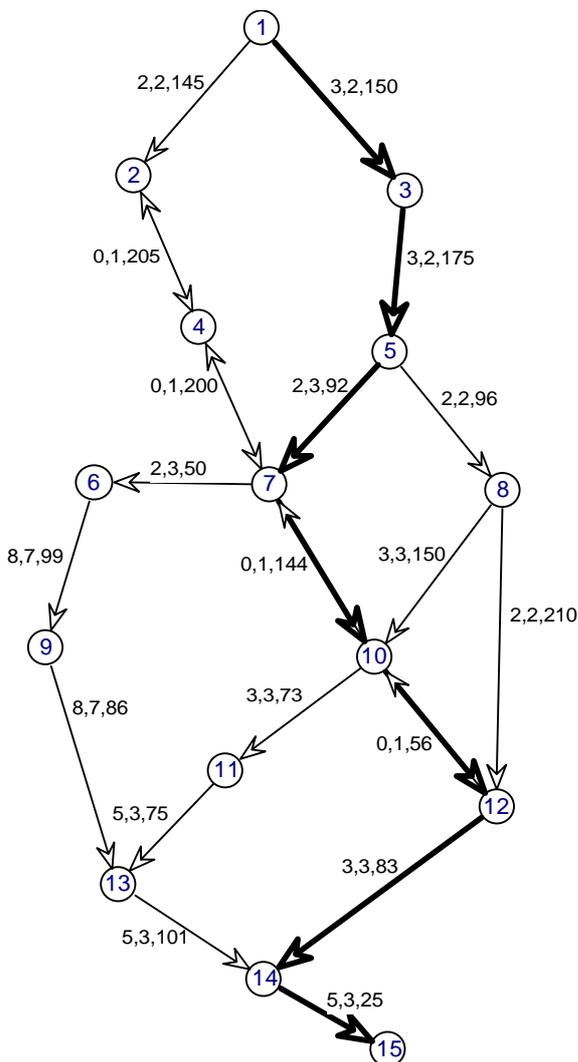


Figure 15: Shortest path

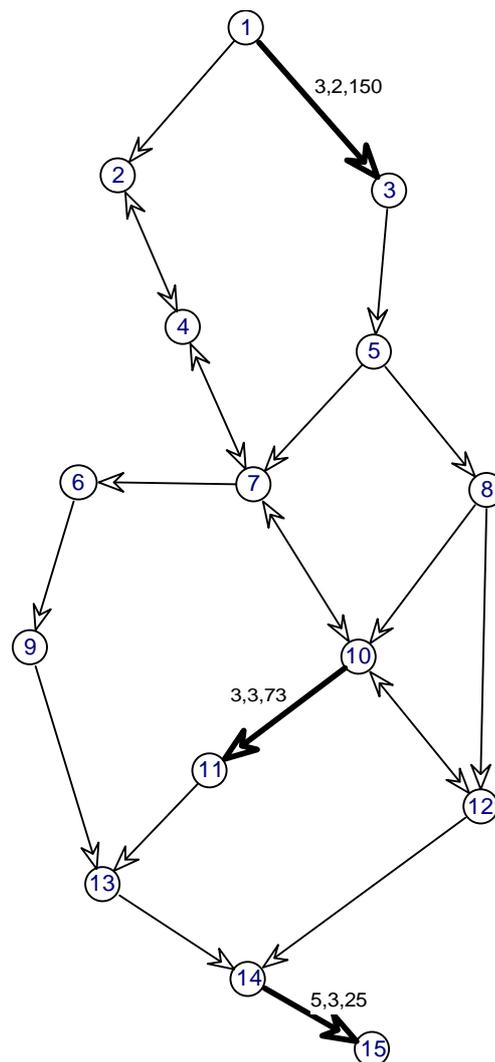


Figure 16: Location reference points

The encoder proceeds with step 3 and calculates a shortest-path between the start line (line between nodes 3 and 5) and the end line (line between nodes 14 and 15) of the location. The line between the nodes 1 and 3 may be ignored because this line was added during the extension phase. The resulting shortest-path (including the extended line) is outlined in Figure 15 using bold lines. The shortest-path has a length of 725 meters (without the extended line: 675 meters).



Step 4 of the encoding process now check whether the location is covered by the calculated shortest-path. It will determine that this is not the case and there is a deviation after node 10. Proceed with step 5 the encoder will determine the line from node 10 to 11 as becoming a new intermediate location reference point. Node 10 is a valid node since it cannot be stepped over during route search and the shortest-path to this line covers the corresponding part of the location completely. The length of the location being covered after this first shortest-path calculation is 561 meters.

The following step 6 prepares the route calculation in order to determine a shortest-path for the remaining part of the location (from node 10 over 11 and 14 to 15). The shortest-path calculation will therefore start at the line from 10 to 11 and ends at the line from 14 to 15.

Step 3 will determine a shortest path (length: 274 meters) between 10 and 15 and step 4 will return that the location is now completely covered by the calculated shortest paths.

In step 7 the location reference path will be composed of the two shortest-paths and the ordered list of location reference points will be formed. Figure 16 shows the lines being selected for the location reference points. The first location reference point points to the line from node 1 to 3 and indicates the start of the location reference path, the second location reference point points to the line from node 10 to 11 and this line was necessary to avoid the deviation from the location. The last location reference point points to the line from node 14 to 15 and indicates the end of the location reference path.

The following step 8 checks the validity of the location reference. Since all lengths between two subsequent location reference points are less than the maximum distance, the location reference emphasizes as being valid.

The last step of the encoding process (step 10) will generate the binary and XML representation of the location reference.

11.3 Binary location reference

The location reference consists of three location reference points and Table 42 shows the coordinates for the nodes 1, 10 and 15. These nodes are the corresponding nodes to the location reference points. In preparation of the binary format this table also shows the relative coordinates. The node 1 corresponds to the location reference point 1 and will have coordinates in absolute format. Node 10 corresponding to location reference point 2 will have relative coordinates to the location reference point 1. Node 15 corresponding to location reference point 2 will also have relative coordinates but now referencing to location reference point 2.

Node ID	LRP index	Longitude	Latitude	Relative longitude	Relative latitude
1	1	6.12683°	49.60851°	--	--
10	2	6.12838°	49.60398°	155	-453
15	3	6.12817°	49.60305°	-21	-93

Table 42: Example coordinates

The offsets being calculated in step 2 of the encoding process are shown in Table 43. In the binary data only the positive offset will appear because the negative offset is 0 and a missing offset will be treated as 0.

Field	Value
positive Offset	150
negative Offset	0

Table 43: Example offset values

Table 44 collects the relevant data for each location reference point. This includes the functional road class, the form of way and the bearing of the corresponding line. The needed information about the path between two subsequent location reference points is also shown (lowest functional road class and distance to the next location reference point).

LRP index	FRC	FOW	BEAR	LFRCNP	DNP
1	FRC3	MULTIPLE_CARRIAGEWAY	135°	FRC3	561
2	FRC3	SINGLE_CARRIAGEWAY	227°	FRC5	274
3	FRC5	SINGLE_CARRIAGEWAY	290°	--	--

Table 44: Location reference points determined during encoding

These tables above hold all relevant information for creating the binary data. The following tables outline the binary data according to section 5:

- Status byte: see Table 45
- LRP 1: see Table 46 to Table 50
- LRP 2: see Table 51 to Table 55
- LRP 3: see Table 56 to Table 59
- Offset: see Table 60

Bit	7	6	5	4	3	2	1	0
Description	RFU	RFU	RFU	ArF	AF	Version		
Value	0	0	0	0	1	0	1	0

Table 45: Binary example: status byte

Byte	First								Second								Third							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Value	0	0	0	0	0	1	0	0	0	1	0	1	1	0	1	1	0	1	0	1	1	0	1	1

Table 46: Binary example: LRP 1 – absolute longitude

Byte	First								Second								Third							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Value	0	0	1	0	0	0	1	1	0	1	0	0	0	1	1	0	1	1	1	1	0	1	0	

Table 47: Binary example: LRP1 – absolute latitude

Bit	7	6	5	4	3	2	1	0
Description	RFU	RFU	FRC			FOW		
Value	0	0	0	1	1	0	1	0

Table 48: Binary example: LRP1 – attribute 1

Bit	7	6	5	4	3	2	1	0
Description	LFRCNP					Bearing		
Value	0	1	1	0	1	1	0	0

Table 49: Binary example: LRP1 – attribute 2

Bit	7	6	5	4	3	2	1	0
Description	DNP							
Value	0	0	0	0	1	0	0	1

Table 50: Binary example: LRP1 – attribute 3

Byte	First								Second							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Value	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	1

Table 51: Binary example: LRP2 – relative longitude

Byte	First								Second							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Value	1	1	1	1	1	1	1	0	0	0	0	1	1	1	0	1

Table 52: Binary example: LRP2 – relative latitude

Bit	7	6	5	4	3	2	1	0
Description	RFU	RFU	FRC			FOW		
Value	0	0	0	1	1	0	1	1

Table 53: Binary example: LRP2 – attribute 1

Bit	7	6	5	4	3	2	1	0
Description	LFRCNP				Bearing			
Value	1	0	1	1	0	1	0	0

Table 54: Binary example: LRP2 – attribute 2

Bit	7	6	5	4	3	2	1	0
Description	DNP							
Value	0	0	0	0	0	1	0	0

Table 55: Binary example: LRP2 – attribute 3

Byte	First								Second							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Value	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1

Table 56: Binary example: LRP3 – relative longitude

Byte	First								Second							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Value	1	1	1	1	1	1	1	1	1	0	1	0	0	0	1	1

Table 57: Binary example: LRP3 – relative latitude

Bit	7	6	5	4	3	2	1	0
Description	RFU	RFU	FRC			FOW		
Value	0	0	1	0	1	0	1	1

Table 58: Binary example: LRP3 – attribute 1

Bit	7	6	5	4	3	2	1	0
Description	RFU	PoffF	NoffF	Bearing				
Value	0	1	0	1	1	0	0	1

Table 59: Binary example: LRP3 – attribute 4

Bit	7	6	5	4	3	2	1	0
Description	POFF							
Value	0	0	0	0	0	0	1	0

Table 60: Binary example: positive Offset

The full binary data stream will have a length of 24 bytes and consists of the following (ordered as bytes from left to right and top to down, see integer values in parentheses):

```

00001010(10)  00000100(4)   01011011(91)  01011011(91)  00100011(35)  01000110(70)
11110100(-12) 00011010(26)  01101100(108) 00001001(9)   00000000(0)   10011011(-101)
11111110(-2)  00111011(59)  00011011(27)  10110100(-76) 00000100(4)   11111111(-1)
11101011(-21) 11111111(-1)  10100011(-93) 00101011(43)  01011001(89)  00000010(2)

```

11.4 XML location reference

The location reference consists of three location reference points in total. The XML format therefore contains two location reference points and a last location reference points. The offset field contains only the positive offset value as there is no negative offset.

The XML representation also contains the Base64-encoded binary representation of the location reference.



```

<?xml version="1.0" encoding="UTF-8"?>
<openlr:OpenLR xmlns:openlr="http://www.openlr.org/openlr"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openlr.org/openlr openlr.xsd ">
  <openlr:LocationID>ExampleLocation</openlr:LocationID>
  <openlr:XMLLocationReference>
    <openlr:LineLocationReference>
      <openlr:LocationReferencePoint>
        <openlr:Coordinates>
          <openlr:Longitude>6.12683</openlr:Longitude>
          <openlr:Latitude>49.60851</openlr:Latitude>
        </openlr:Coordinates>
        <openlr:LineAttributes>
          <openlr:FRC>FRC3</openlr:FRC>
          <openlr:FOW>MULTIPLE_CARRIAGEWAY</openlr:FOW>
          <openlr:BEAR>135</openlr:BEAR>
        </openlr:LineAttributes>
        <openlr:PathAttributes>
          <openlr:LFR CNP>FRC3</openlr:LFR CNP>
          <openlr:DNP>561</openlr:DNP>
        </openlr:PathAttributes>
      </openlr:LocationReferencePoint>
      <openlr:LocationReferencePoint>
        <openlr:Coordinates>
          <openlr:Longitude>6.12838</openlr:Longitude>
          <openlr:Latitude>49.60398</openlr:Latitude>
        </openlr:Coordinates>
        <openlr:LineAttributes>
          <openlr:FRC>FRC3</openlr:FRC>
          <openlr:FOW>SINGLE_CARRIAGEWAY</openlr:FOW>
          <openlr:BEAR>227</openlr:BEAR>
        </openlr:LineAttributes>
        <openlr:PathAttributes>
          <openlr:LFR CNP>FRC5</openlr:LFR CNP>
          <openlr:DNP>274</openlr:DNP>
        </openlr:PathAttributes>
      </openlr:LocationReferencePoint>
      <openlr>LastLocationReferencePoint>
        <openlr:Coordinates>
          <openlr:Longitude>6.12817</openlr:Longitude>
          <openlr:Latitude>49.60305</openlr:Latitude>
        </openlr:Coordinates>
        <openlr:LineAttributes>
          <openlr:FRC>FRC5</openlr:FRC>
          <openlr:FOW>SINGLE_CARRIAGEWAY</openlr:FOW>
          <openlr:BEAR>290</openlr:BEAR>
        </openlr:LineAttributes>
      </openlr>LastLocationReferencePoint>
      <openlr:Offsets>
        <openlr:PosOff>150</openlr:PosOff>
      </openlr:Offsets>
    </openlr:LineLocationReference>
  </openlr:XMLLocationReference>
  <openlr:BinaryLocationReferences>
    <openlr:BinaryLocationReference id="BinaryExample" version="2">
      CgRbWyNG9BpsCQCb/jsbtAT/6/+jK1kC
    </openlr:BinaryLocationReference>
  </openlr:BinaryLocationReferences>
  <openlr:AdditionalInformation>
    <openlr:BoundingBox>
      <openlr:LowerLeft>
        <openlr:Longitude>6.12601</openlr:Longitude>
        <openlr:Latitude>49.60243</openlr:Latitude>
      </openlr:LowerLeft>
      <openlr:UpperRight>
        <openlr:Longitude>6.12951</openlr:Longitude>
        <openlr:Latitude>49.60891</openlr:Latitude>
      </openlr:UpperRight>
    </openlr:BoundingBox>
  </openlr:AdditionalInformation>

```



```

</openlr:BoundingBox>
<openlr:MapDatabase>ExampleMap Version 0.1</openlr:MapDatabase>
<openlr:AdditionalData/>
</openlr:AdditionalInformation>
</openlr:OpenLR>

```

11.5 Decoding

This section outlines the decoding of the location reference being prepared in the sections above. The decoding walks along the steps presented in section 10.1.

The step 1 of the decoding process extracts the physical data (binary format) and checks the validity of the location reference. The information extracted from the binary data will not be as accurate as before creating the binary stream. Due to the usage of intervals for the bearing and the distance to next point the concrete value cannot be extracted but a small interval containing the concrete value.

The information being extracted from the binary data example is shown in Table 61, Table 62 and Table 63.

LRP index	Longitude	Latitude
1	6.12682°	49.60850°
2	6.12838°	49.60397°
3	6.12817°	49.60304°

Table 61: Decoded coordinates

LRP index	FRC	FOW	Bearing	LFRCNP	DNP
1	FRC3	MULTIPLE_CARRIAGEWAY	135.00° - 146.25°	FRC3	527.4m – 586.0m
2	FRC3	SINGLE_CARRIAGEWAY	225.00° - 236.25°	FRC5	234.4m – 293.0m
3	FRC5	SINGLE_CARRIAGEWAY	281.25° - 292.50°	--	0m

Table 62: Decoded LRP information

Offset	Value
Positive offset	117.2m – 175.8m
Negative offset	- no offset available -

Table 63: Decoded offset information

This information is sufficient to resolve the location on the decoder map shown in Figure 17. The map consists of 17 nodes and 26 lines (two-way lines are counted twice).

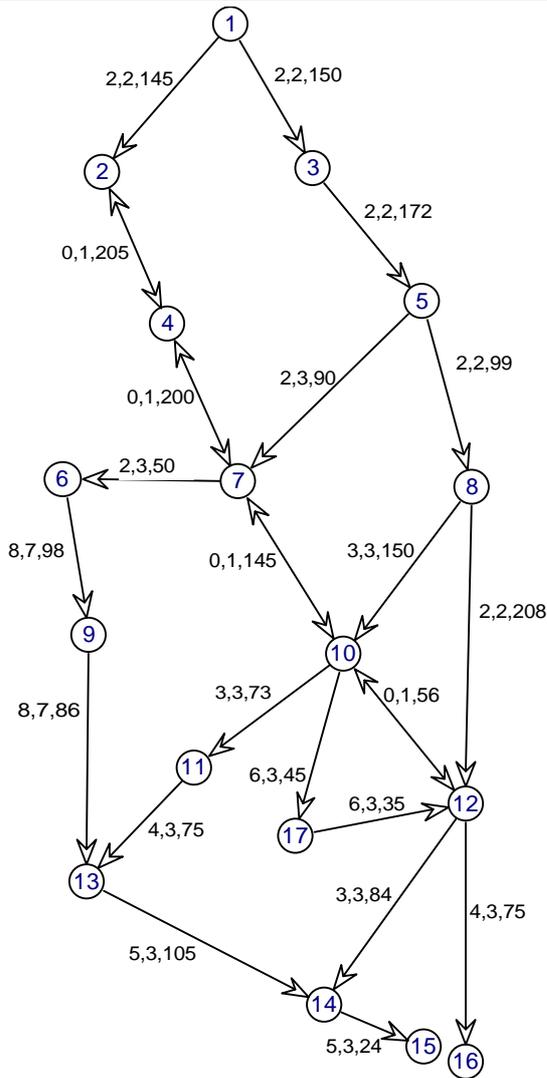


Figure 17: Decoder map

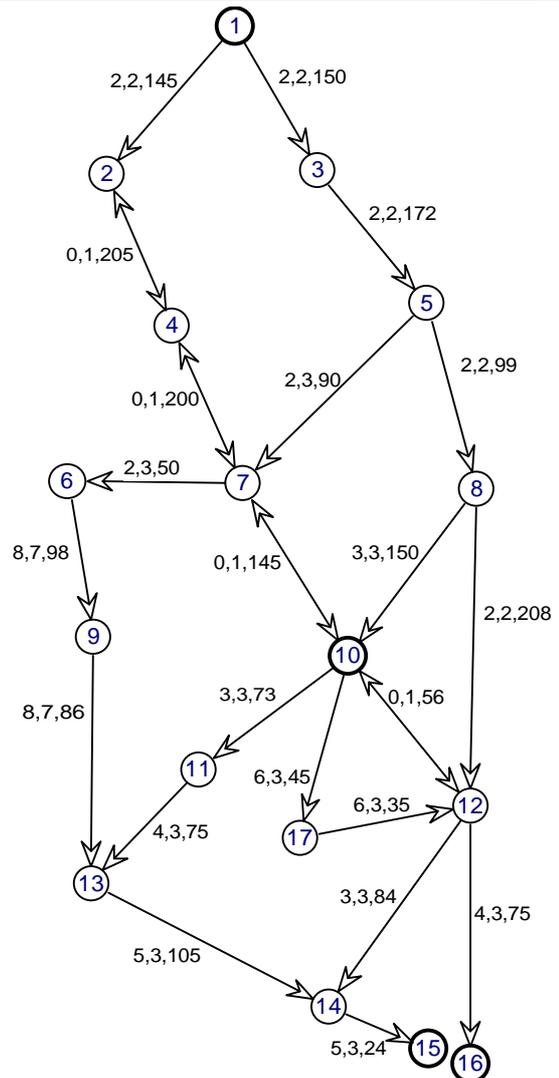


Figure 18: Candidate nodes

This map differs from the encoder map (see Figure 13) in several ways. Some length values are different (e.g. line from node 3 to 5), some functional road class values have changed (e.g. line from node 3 to 5) and there are two more nodes 16 and 17 and also additional lines connecting these new nodes. The challenge of the decoder is to resolve the location in this different map.

After validating the data the decoder starts determining candidate nodes for each location reference point (step 2). Figure 18 shows the candidate nodes (bold circle) which are positioned close by the coordinates of the location reference points. For the location reference point 1 and 2 exist only one candidate node but for the last location reference point two candidate nodes are in the run.

Step 3 deals with determining candidate lines for each location reference point. The bold lines in Figure 19 are the candidate lines for this example. The location reference point 1 has two outgoing lines as candidates, location reference point 2 has three outgoing lines in the run and the last location reference point has two incoming lines (one for each candidate node).

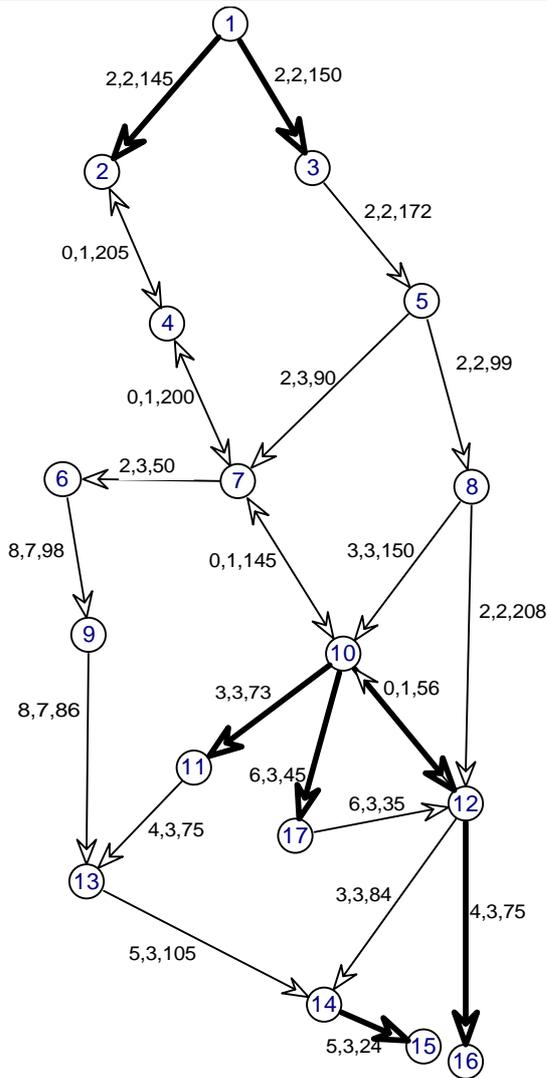


Figure 19: Candidate lines

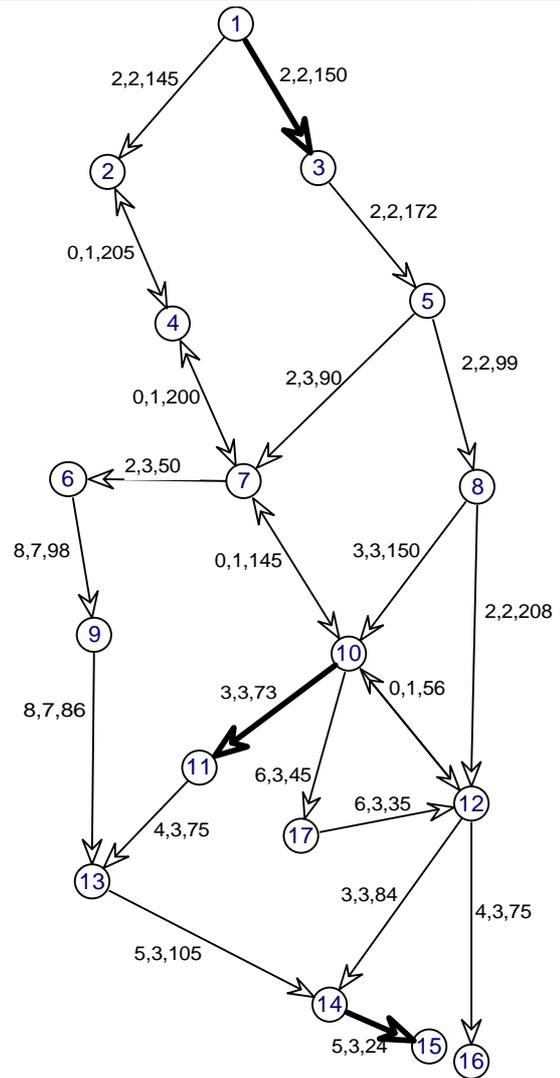


Figure 20: Location reference point - lines

Each candidate line will be rated in step 4 according to its compliance with the attributes of the location reference point and its distance to the location reference coordinate. The bold lines in Figure 20 show the lines which are rated best. These lines will be used for the following shortest-path calculation in step 5 of the decoding process.

The shortest-path calculation will run for each subsequent pair of location reference points and the concatenation of the two shortest-paths in this example are shown in Figure 21. These shortest-paths will be validated in step 6. The length of the first shortest-path (from node 1 to node 10) is 557 meters and this value fits into the distance to next point interval of the first location reference point (527.4 meters – 586.0 meters). The length of the second shortest-path (from node 10 to node 15) is 277 meters and this value also fits into the distance to next point interval of the location reference point 2 (234.4 meters – 293.0 meters). The shortest-paths are validated and the decoder does not fail.

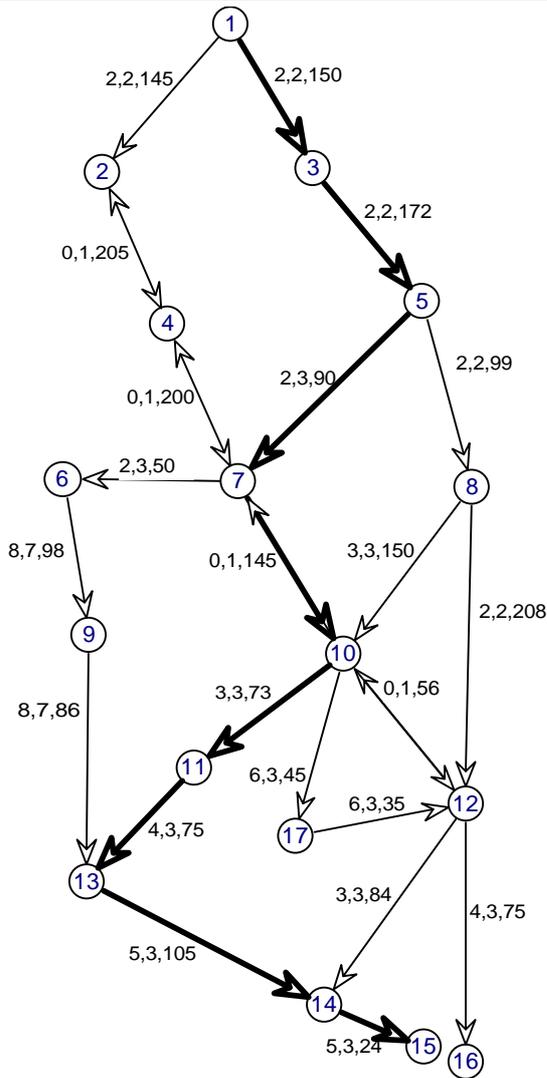


Figure 21: Decoder - shortest path

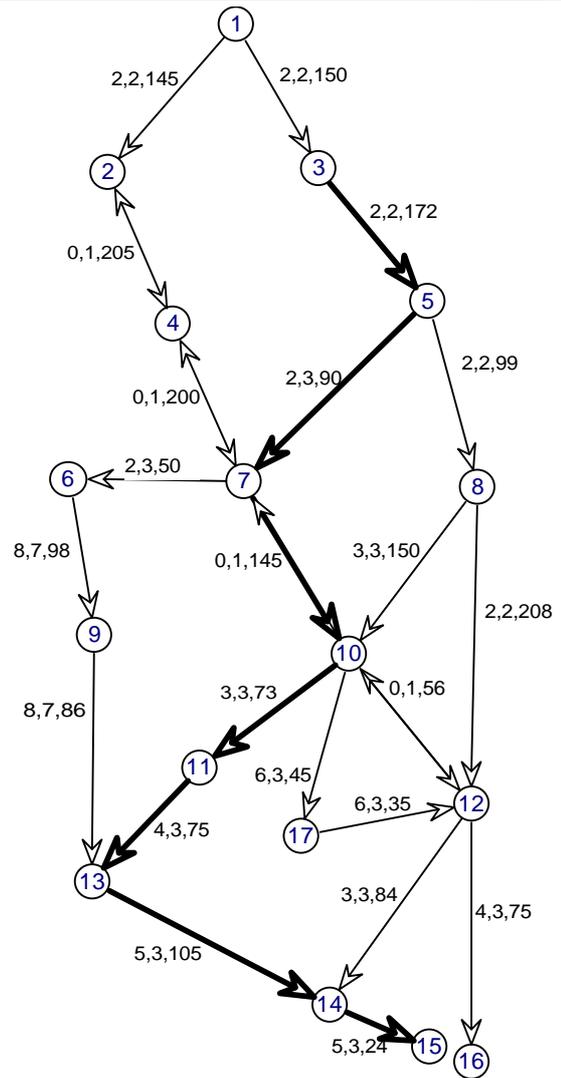


Figure 22: Decoder location

The last step of the decoding process (step 7) now trims the concatenated shortest-path according to the offsets. The example only holds a positive offset and therefore the decoder trims the shortest path at its start. The only node fitting in the positive offset interval (117.2 meters – 175.8 meters) is node 3. This node points out to be the real start of the location and the decoded location is finally shown in Figure 22.

12. Appendix

List of figures

Figure 1: Connection of LR-points	14
Figure 2: Logical format: Bearing point	16
Figure 3: Logical format: Bearing	16
Figure 4: Logical format: Distance to next point	17
Figure 5: Logical format: Positive and negative offset	18
Figure 6: Relationship: Attributes – LR-point	18
Figure 7: Invalid node (1)	19
Figure 8: Invalid node (2)	19
Figure 9: Physical format: Bearing sectors (Binary)	23
Figure 10: XML structure overview	41
Figure 11: Workflow – OpenLR™ encoding	46
Figure 12: Workflow – OpenLR™ decoding	50
Figure 13: Example network	51
Figure 14: Location	51
Figure 15: Shortest path	52
Figure 16: Location reference points	52
Figure 17: Decoder map	58
Figure 18: Candidate nodes	58
Figure 19: Candidate lines	59
Figure 20: Location reference point - lines	59
Figure 21: Decoder - shortest path	60
Figure 22: Decoder location	60

List of tables

Table 1: Explanation of common terms	10
Table 2: Explanation of common abbreviations	11
Table 3: Logical format: Functional road class	15
Table 4: Logical Format: Form of way	15
Table 5: Logical format: Parameter BEARDIST	16
Table 6: Data format rules overview	19
Table 7: Physical format: Data types (Binary)	20
Table 8: Physical format: Coordinate format (absolute) (Binary)	21
Table 9: Physical format: Longitude/Latitude ranges for relative coordinates (Example)	21
Table 10: Physical format: Coordinate format (relative) (Binary)	21
Table 11: Physical format: Functional road class (Binary)	21
Table 12: Physical format: Form of way (Binary)	22
Table 13: Physical format: Bearing (Binary)	22
Table 14: Physical format: Bearing value definition (Binary)	22
Table 15: Physical format: Distance to next point (Binary)	23
Table 16: Physical format: Lowest FRC to next point (Binary)	23
Table 17: Physical format: Version (Binary)	24
Table 18: Physical format: Attribute flag (Binary)	24
Table 19: Physical format: Attribute flag values (Binary)	24
Table 20: Physical format: Area flag (Binary)	24
Table 21: Physical format: Area flag values (Binary)	24
Table 22: Physical format: Offset flag (Binary)	25
Table 23: Physical format: Offset flag values (Binary)	25
Table 24: Physical format: Offset (Binary)	25
Table 25: Binary format overview	26
Table 26: Status byte	26
Table 27: Status byte value	26
Table 28: First LR-point coordinates	27
Table 29: Following LR-points coordinates	27



Table 30: First attribute byte – valid for all LR-points	27
Table 31: Second attribute byte – valid for all LR-points, except the last LR-point	27
Table 32: Third attribute byte – valid for all LR-points, except the last LR-point	27
Table 33: Fourth attribute bytes – valid only for the last LR-point	28
Table 34: Positive offset value	28
Table 35: Negative offset value	28
Table 36: Message sizes depending on the number of LR-points	29
Table 37: Physical format: Data types (XML)	30
Table 38: Physical format: Functional road class (XML)	31
Table 39: Physical format: Form of way (XML)	31
Table 40: Encoding steps overview	45
Table 41: Decoding steps overview	49
Table 42: Example coordinates	53
Table 43: Example offset values	53
Table 44: Location reference points determined during encoding	54
Table 45: Binary example: status byte	54
Table 46: Binary example: LRP 1 – absolute longitude	54
Table 47: Binary example: LRP1 – absolute latitude	54
Table 48: Binary example: LRP1 – attribute 1	54
Table 49: Binary example: LRP1 – attribute 2	54
Table 50: Binary example: LRP1 – attribute 3	54
Table 51: Binary example: LRP2 – relative longitude	54
Table 52: Binary example: LRP2 – relative latitude	54
Table 53: Binary example: LRP2 – attribute 1	55
Table 54: Binary example: LRP2 – attribute 2	55
Table 55: Binary example: LRP2 – attribute 3	55
Table 56: Binary example: LRP3 – relative longitude	55
Table 57: Binary example: LRP3 – relative latitude	55
Table 58: Binary example: LRP3 – attribute 1	55
Table 59: Binary example: LRP3 – attribute 4	55
Table 60: Binary example: positive Offset	55
Table 61: Decoded coordinates	57
Table 62: Decoded LRP information	57
Table 63: Decoded offset information	57

List of equations

Equation 1: Transformation from decimal coordinates into integer values (Binary)	20
Equation 2: Transformation from integer values into decimal coordinates (Binary)	20
Equation 3: Relative coordinates calculation (Binary)	21
Equation 4: Calculation of the bearing value (Binary)	23
Equation 5: Calculation of the DNP value (Binary)	23
Equation 6: Calculation of offset values (Binary)	25



An open standard for
encoding, transmitting
and decoding location
references in digital maps

Internet: <http://www.openlr.org>

Contact:

TomTom (Legal Department)
Email: legal@tomtom.com

TomTom (Technical contact)
Email: openlr@tomtom.com

Address: TomTom International B.V.
Oosterdoksstraat 114
1011DK Amsterdam
the Netherlands

Copyright © 2009, 2010 TomTom International B.V.

OpenLR™ is a trade mark of TomTom International B.V.