



ASAM OpenDRIVE

Open Dynamic Road Information for Vehicle Environment

Version 1.6

Date: 2020-03-12

Base Standard

© by ASAM e.V., 2020

Disclaimer

This document is the copyrighted property of ASAM e.V.

Any use is limited to the scope described in the license terms. The license terms can be viewed at www.asam.net/license.

In alteration to the regular license terms, ASAM allows unrestricted distribution of this standard. §2 (1) of ASAM's regular license terms is to be substituted by the following clause: "The licensor grants everyone a basic, non-exclusive and unlimited license to use the standard ASAM OpenDRIVE".



Table of Contents

1 Foreword	7
1.1 Deliverables of the OpenDRIVE specification	7
2 Introduction	8
2.1 Overview	8
2.2 Normative and non-normative statements and deliverables.....	8
2.3 Conventions.....	9
2.3.1 Naming conventions.....	9
2.3.2 Units	9
2.3.3 Modal verbs	10
2.3.4 Typographic conventions	10
2.3.5 UML notation.....	11
2.3.6 Use of IDs	12
2.3.7 Curvature.....	12
3 Relations to other standards (preliminary)	13
3.1 Positioning of ASAM OpenDRIVE within ASAM activities.....	13
3.2 Relation of ASAM OpenDRIVE to OpenCRG and OpenSCENARIO.....	13
3.3 Backward compatibility to earlier releases.....	13
3.4 References to other standards	14
4 General architecture of OpenDRIVE	15
4.1 File structure.....	15
4.2 Combining files	15
4.3 Attributes used in the file.....	16
4.3.1 Enclosing element.....	16
4.3.2 Header.....	17
4.4 General rules and assumptions	17
4.4.1 Traffic direction	17
5 Additional data	18
5.1 User Data	18
5.2 Including data	19
5.3 Using different layout types	20
5.4 Description of the data quality	20
6 Coordinate systems	21
6.1 Coordinate systems overview	21
6.2 Inertial coordinate system	22
6.3 Reference line coordinate system.....	23



6.4 Local coordinate system.....	25
6.5 Summary of all available coordinate systems.....	27
6.6 Georeferencing in OpenDRIVE	28
6.6.1 Offset	28
7 Geometry	30
7.1 Road reference line	30
7.2 Straight line.....	34
7.3 Spirals	34
7.4 Arc	36
7.5 Generating arbitrary road courses from geometry elements	37
7.6 Cubic polynom (deprecated)	37
7.6.1 Background information on cubic polynomials	38
7.6.2 Creating roads using cubic polynomials.....	38
7.7 Parametric cubic curve	41
7.7.1 Generating roads using parametric cubic curves.....	42
8 Roads	46
8.1 Properties for road sections and cross section	48
8.2 Road linkage.....	48
8.3 Road type.....	52
8.3.1 Speed limits for road types.....	53
8.4 Methods of Elevation	54
8.4.1 Road elevation	55
8.4.2 Superelevation	56
8.4.3 Shape definition	58
8.5 Road surface.....	61
8.6 Use cases for roads	67
8.6.1 Modelling a road shape with a linear crossfall	67
9 Lanes	69
9.1 Lane grouping within lane sections	72
9.2 Lane sections	74
9.3 Lane offset	77
9.4 Lane linkage.....	78
9.5 Lane properties	81
9.5.1 Lane width	81
9.5.2 Lane borders.....	84
9.5.3 Lane type	86
9.5.4 Lane material	90
9.5.5 Lane speed limit.....	91
9.5.6 Lane access.....	93
9.5.7 Lane height	95

9.5.8 Excluding lanes from road superelevation	97
9.6 Road markings	98
9.6.1 Road marking types and lines	100
9.6.2 Explicit road marking types and lines	101
9.6.3 Offsets in road markings	102
9.7 Specific lane rules	104
10 Junctions	106
10.1 Incoming roads.....	108
10.2 Connecting roads	109
10.2.1 Priorities of connecting roads within a junction	111
10.2.2 Direction of connecting roads.....	111
10.3 Road surface within junctions.....	112
10.4 Virtual junctions	114
10.4.1 Virtual connections.....	116
10.5 Junction groups	119
10.5.1 Controllers for junctions	120
11 Objects	123
11.1 Repeating objects.....	127
11.2 Object outline	130
11.2.1 Corner roads	132
11.2.2 Corner local.....	133
11.3 Object material	135
11.4 Lane validity for objects.....	136
11.5 Access rules to parking spaces	136
11.6 Object marking	138
11.7 Object borders.....	141
11.8 Object reference	142
11.9 Tunnels	143
11.10 Bridges	144
12 Signals	146
12.1 Lane validity for signals.....	150
12.2 Signal dependency.....	152
12.3 Links between signals and objects	154
12.4 Signal positioning	155
12.5 Reuse of signal information	158
12.6 Controllers	160
13 Railroads	162



13.1 Railroad tracks	162
13.2 Switches.....	163
13.2.1 Main track	165
Related topics	165
13.2.2 Side track.....	166
13.2.3 Partner switches	166
13.3 Stations.....	168
13.3.1 Platforms.....	170
13.3.2 Segments.....	170
14 Bibliography	172
Figure Directory	173
Table Directory	176

1 Foreword

ASAM OpenDRIVE provides the exchange format specification to describe static road networks for driving simulation applications. The primary task of ASAM OpenDRIVE is the road description including objects along the road. The OpenDRIVE Specification covers the description on how to model e.g. roads, lanes, junctions. Dynamic content is not covered by ASAM OpenDRIVE.

1.1 Deliverables of the OpenDRIVE specification

The following deliverables are provided for OpenDRIVE:

- File format specification
- XML schemas
- UML model
- Sample files (UseCases and Examples)
- Example implementations
- Reference implementations spiral
- Signal Base catalog for OpenDRIVE

2 Introduction

2.1 Overview

The OpenDRIVE format provides a common base for describing road networks with Extensible Markup Language (XML) syntax, with the file extension xodr. The data that is stored in an OpenDRIVE file describes the geometry of roads as well as features along the roads that influence the logics, for example, lanes and signals. The road networks that are described in the OpenDRIVE file can either be synthetic or real. The main purpose of OpenDRIVE is to provide a road network description that can be fed into simulations and to make these road network descriptions exchangeable.

The format is organized in nodes that can be extended with user-defined data. This facilitates a high degree of specialization for individual applications (usually simulations) while maintaining the interoperability that is required for the exchange of data between different applications.

2.2 Normative and non-normative statements and deliverables

This specification uses a standard information structure. The following rules apply regarding normativity of sections:

Statements expressed as requirements, permissions, or prohibitions according to the use of modal verbs, as defined in “0”

- are normative.
- UML diagrams from the OpenDRIVE UML model are normative.
- Rules for OpenDRIVE data structures in “Rules” sections are normative.
- XML examples and use case descriptions are non-normative.

2.3 Conventions

2.3.1 Naming conventions

In this document, the following conventions apply:

data types are given according to IEEE standard

2.3.2 Units

Unless stated otherwise, all numeric values within this specification are in SI units, for example:

- position/distance in [m]
- angles in [rad]
- time in [s]
- speed in [m/s]

Geographic positions are stated in the unit defined by the spatial coordinate system, for example, in accordance with WGS 84 – EPSG 4326 [1].

Some data elements allow you to explicitly state the unit of a given value. If the unit is not given or if there is no means to state the unit, SI units apply. The following units may be used in explicit assignments:

Category	Description	Identifier
distance	meter	m
	kilometer	km
	feet	ft
	land mile	mile
Speed	meters per second	m/s
	miles per hour	mph
	kilometers per hour	km/h
Mass	kilogram	kg
	metric tons	t
Slope	percent	%

Table 1: Units

These optional units shall be used for purposes of signage and speed indication only. They shall not be used as general units, for example, to define road geometry, etc.

2.3.3 Modal verbs

To ensure compliance with the OpenDRIVE standard, users need to be able to distinguish between mandatory requirements, recommendations, permissions, as well as possibilities and capabilities.

The following rules for using modal verbs apply:

Provision	Verbal form
Requirement Requirements shall be followed strictly in order to be conform to the standard. Deviations are not allowed.	shall shall not
Recommendation Recommendations indicate that among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others.	should should not
Permission Permissions indicate a course of action permissible within the limits of OpenDRIVE deliverables.	may need not
Possibility and capability These verbal forms are used for stating possibilities or capabilities, being technical, material, physical, or others.	can can not
Obligation and necessity These verbal forms are used to describe legal, organizational, or technical obligations and necessities that are not regulated or enforced by the OpenDRIVE standard.	must must not

Table 2: Rules for using modal verbs

2.3.4 Typographic conventions

This documentation uses the following typographical conventions:

Mark-up	Definition
Code elements	This format is used for code elements, such as technical names of classes and attributes, as well as attribute values.
Code snippets	This format is used for excerpts of code that serve

	as an example for implementation.
Terms	This format is used to introduce glossary terms, new terms and to emphasize terms.
Mathematical elements	This format is used for calculations and mathematical elements.
<element>	This describes a tag for an element within the XML specification
@attribute	The “@” identifies an attribute of any OpenDRIVE element

Table 3: Typographical conventions

2.3.5 UML notation

The OpenDRIVE structure diagrams are modeled according to the Unified Modeling Language (UML). For detailed information on UML, see Booch et al. (1997).

Figure 1 describes the UML notations that are used in this standard.

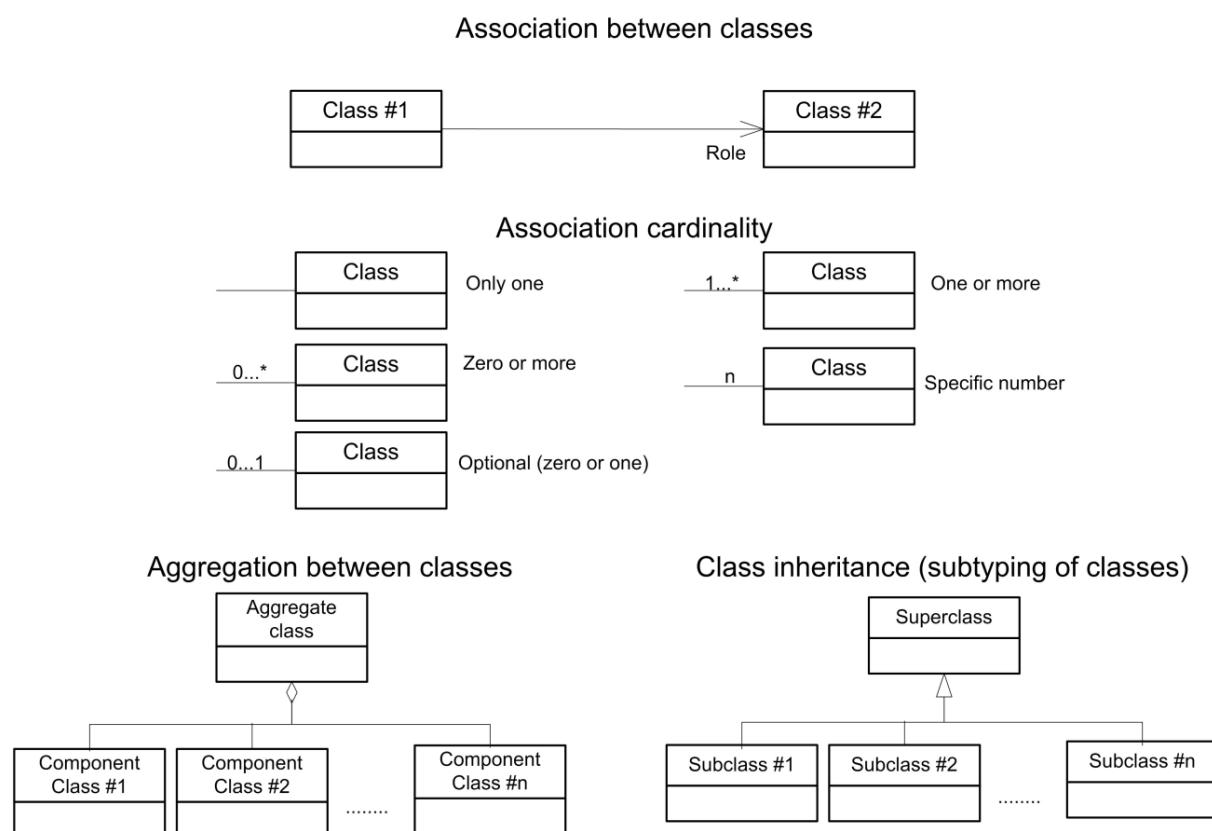


Figure 1: UML notation (see ISO TS 19103, Geographic information - Conceptual schema language)

The context that an element takes within an association is indicated by its role. The role is given near the target of the association.

For better readability, the OpenDRIVE class diagrams use a color scheme:

- The top-level element of a diagram is marked orange. This helps finding the entry point when reading a diagram top-down.
- Classes that are marked yellow belong to the UML package that is discussed in the chapter of the specification, where the UML diagram is given.
- Classes that are marked blue belong to an OpenDRIVE package that is different from the package that is associated with the yellow color.
- Classes that are marked green are classes that contain geometry information.

Mandatory and optional attributes

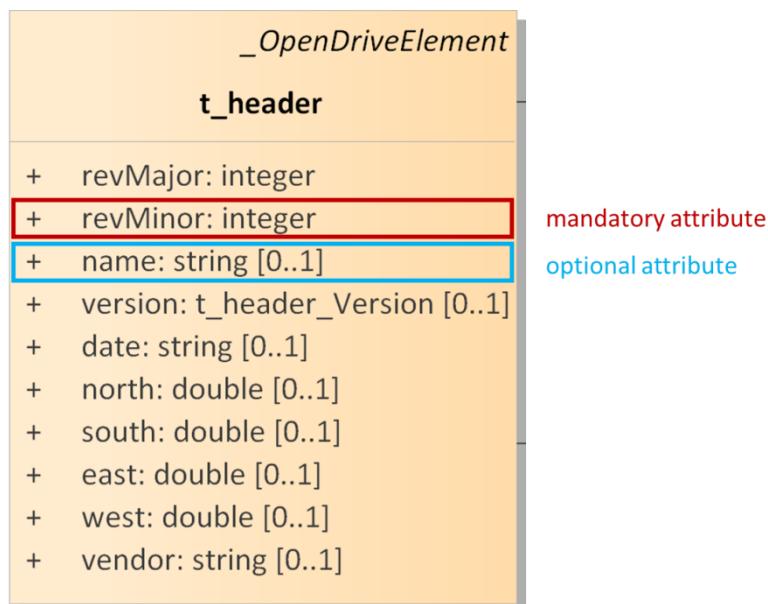


Figure 2: UML attribute notation

In the UML Model attributes are marked as mandatory and optional. In the above figure the marking of the attributes can be seen. Optional attributes have the UML specific notation [0..1], mandatory attributes do not have any notation.

2.3.6 Use of IDs

The following rules apply to the use of IDs in OpenDRIVE:

- IDs shall be unique within a class.
- Lane IDs shall be unique within a lane section.
- Only defined IDs may be referenced.

2.3.7 Curvature

For curvature indications, the following convention applies:

- Positive curvature: left curve (counter-clockwise motion)
- Negative curvature: right curve (clockwise motion)

Curvature == 1/radius

3 Relations to other standards (preliminary)

3.1 Positioning of ASAM OpenDRIVE within ASAM activities

ASAM OpenDRIVE is part of the ASAM simulation standards that focus on simulation data for the automotive environment. Next to ASAM OpenDRIVE, ASAM Provides other standards for the simulation domain, like ASAM OpenSCENARIO and ASAM openCRG.

3.2 Relation of ASAM OpenDRIVE to OpenCRG and OpenSCENARIO

ASAM OpenDRIVE defines a storage format for the static description of road networks. In combination with ASAM OpenCRG it is possible to add very detailed road surface descriptions to the road network. ASAM OpenDRIVE and ASAM OpenCRG only contains static content. To add dynamic content ASAM OpenSCENARIO is needed. Combined all three standards provide a scenario-driven description of traffic simulation that contains static and dynamic content.

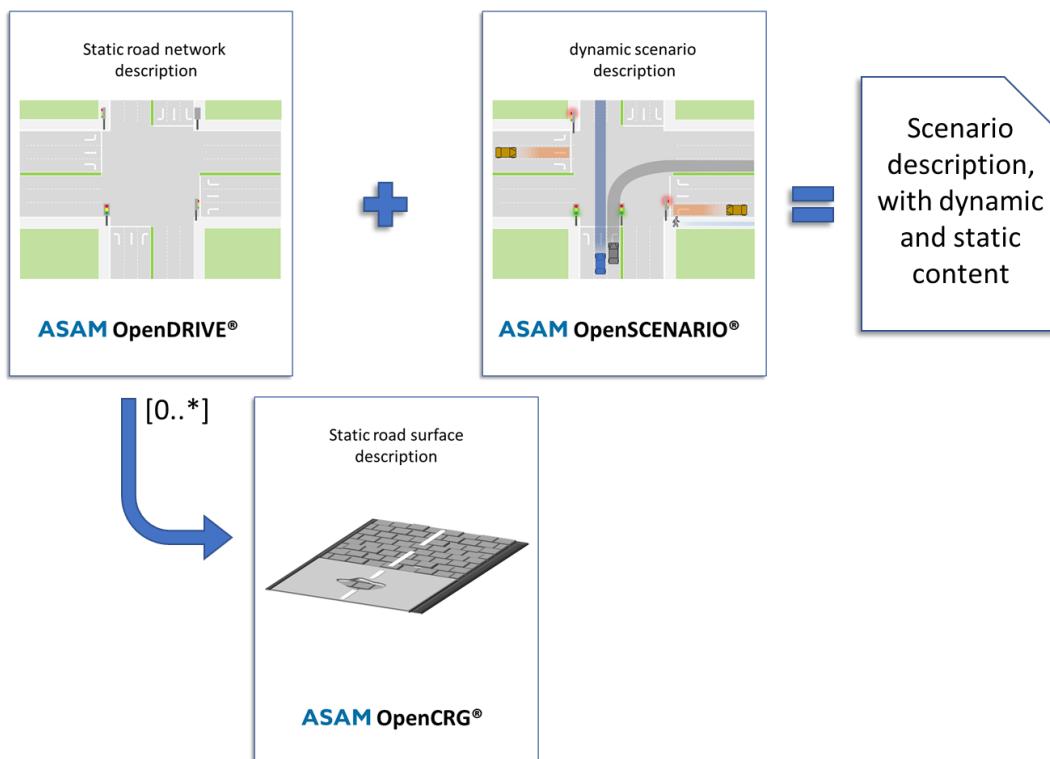


Figure 3: Relation between OpenDRIVE, OpenCRG, and OpenSCENARIO

3.3 Backward compatibility to earlier releases

OpenDRIVE 1.6 contains elements that were introduced in version 1.5 but are not compatible with version 1.4. To ensure compatibility with versions 1.4 and 1.5, these

elements are technically defined as optional in the XML schema of version 1.6. They are marked as “Optional for backwards compatibility” in the annotations of the UML model.

3.4 References to other standards

- XML 1.0 Schema [4]
- UML 2.5.1 Standard
- ISO 3166-2 for country codes [8]
- ISO 8601 for time / date [7]
- Georeferencing (ISO DIN 19111)
- ASAM OpenCRG [3]

4 General architecture of OpenDRIVE

4.1 File structure

OpenDRIVE data is stored in XML files with the extension .xodr. Compressed OpenDRIVE files have the extension “xodrz” (compression format: gzip).

The OpenDRIVE file structure conforms to XML rules; the associated schema file is referenced in the XML. The schema file for the OpenDRIVE® format can be retrieved from <https://www.asam.net/standards/detail/opendrive/>

Elements are organized in levels. Elements with a level that is greater than zero (0) are children of the preceding level. Elements with a level of one (1) are called primary elements.

Each element can be extended with user-defined data. This data is stored in so-called user data elements.

All floating-point numbers used in OpenDRIVE are IEEE 754 double precision floating-point numbers. In order to ensure accurate representation of floating-point numbers in the XML representation, implementations SHOULD use a known correct accuracy preserving minimal floating-point printing algorithm (e.g. [11], [12]), or ensure that 17 significand decimal digits are always produced (e.g. using the "%."17g" ISO C printf modifier). Importing implementations should use a known correct accuracy preserving floating-point reading algorithm (e.g. [13]).

4.2 Combining files

Multiple files can be combined with an <include> tag at the appropriate locations. Upon parsing this tag, OpenDRIVE readers shall immediately start reading the file specified as attribute of the tag. It is the user's responsibility to make sure that contents read from an include file are consistent with the context from which the inclusion starts.

The parent tag under which the <include> tag occurs must be present in both, the parent file and the included file.

XML Example:

Original File

```
<planView>
  <include file="planview.xml"/>
</planView>
```

Included File

```
<planView>
  <geometry x="-0.014" y="-0.055" hdg="2.88" length="95.89" s="0.0">
    <arc curvature="-0.000490572"/>
  </geometry>
  <geometry x="-92.10" y="26.64" hdg="2.84" length="46.65" s="95.89">
    <spiral curvStart="-0.000490572" curvEnd="-0.004661241"/>
  </geometry>
</planView>
```

4.3 Attributes used in the file

All attributes that can be used in an OpenDRIVE file are fully annotated in the UML model:

- If **units** are applicable to an attribute, these are stated according to 2.3.2 Units.
- **Type:** Describes the data type of an attribute. It can be either a primitive data type, for example, string, double, float, or a complex data type that refers to an object described within this specification.
- **Value:** Value determines the value range of the given attribute relative to the specified type

4.3.1 Enclosing element

The overall enclosing element of the file is:

delimiters	<OpenDRIVE > . . . </OpenDRIVE>
parent	none
instances	1
attributes	xmlns="http://www.opendrive.org"

Table 4: Attributes of the OpenDRIVE element

4.3.2 Header

The <header> element is the very first element within the <OpenDRIVE> element.

delimiters	<header>...</header>			
parent	<OpenDRIVE>			
instances	1			
attributes				
name	type	unit	value	description
revMajor	ushort	-	1	Major revision number of OpenDRIVE format
revMinor	ushort	-		Minor revision number of OpenDRIVE format; 6 for OpenDrive 1.6
name	string	-	-	Database name
version	string	-	-	Version of this road network
date	string	-	-	Time/date of database creation according to ISO 8601 (preference: YYYY-MM-DDThh:mm:ss)
north	double	m	-	Maximum inertial y value
south	double	m	-	Minimum inertial y value
east	double	m	-	Maximum inertial x value
west	double	m	-	Minimum inertial x value
vendor	string	-	-	Vendor name

Table 5: Attributes of the header element

4.4 General rules and assumptions

4.4.1 Traffic direction

Unless stated otherwise, all examples, figures, and descriptions in this specification assume right-hand traffic.

5 Additional data

OpenDRIVE offers the possibility to include external data. The processing of this data depends on the application.

Additional data may be placed at any position in OpenDRIVE.

5.1 User Data

Ancillary data should be described near the element it refers to. Ancillary data contains data that are not yet described in OpenDRIVE, or data that is needed by an application for a specific reason. Examples are different road textures.

In OpenDRIVE, ancillary data is represented by `<userData>` elements. They may be stored at any element in OpenDRIVE.

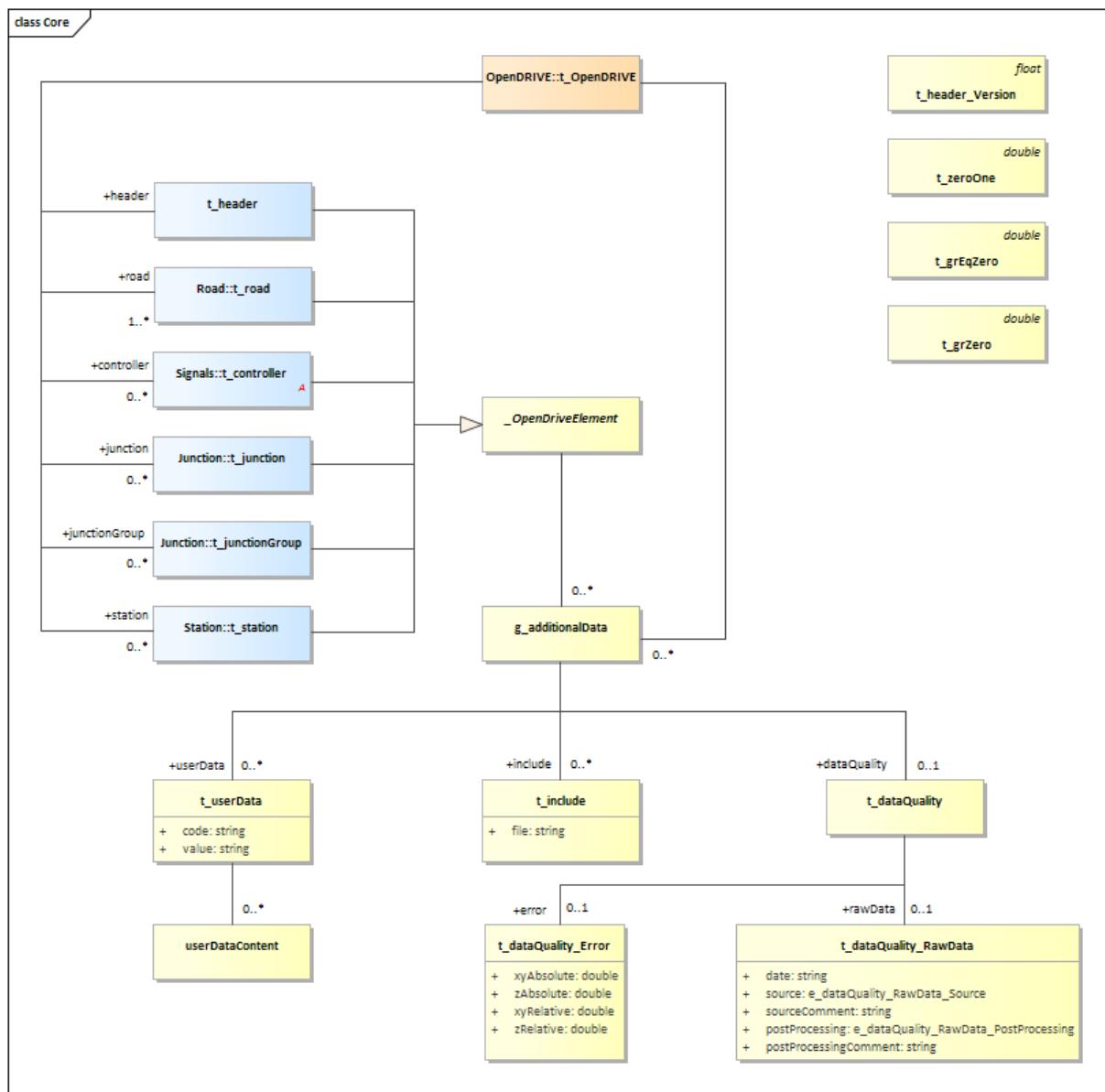


Figure 4: UML model OpenDRIVE Core

5.2 Including data

OpenDRIVE allows including external files into the OpenDRIVE file. The processing of the files depends on the application.

Included data is represented by <include> elements. They may be stored at any position in OpenDRIVE.

5.3 Using different layout types

OpenDRIVE offers the possibility to integrate user-generated layouts for elements like road marks or signals. The design of these alternative layouts is not stored in OpenDRIVE, but in the application.

In OpenDRIVE, different layout types are represented by `<set>` elements. They may be stored at any position in OpenDRIVE.

Every `<set>` element may be followed by one or more `<instance>` element that specifies the layout.

5.4 Description of the data quality

Raw data or data from external sources that is integrated in OpenDRIVE may be of varying quality. It is possible to describe quality and accuracy of external data in OpenDRIVE.

The description of the data quality is represented by `<dataQuality>` elements. They may be stored at any position in OpenDRIVE.

Measurement data derived from external sources like GPS that is integrated in OpenDRIVE may be inaccurate. The error range, given in [m], may be listed in the application.

The absolute or relative errors of road data are described by `<error>` elements within the `<dataQuality>` element.

Some basic metadata containing information about raw data included in OpenDRIVE is described by the `<rawData>` element within the `<dataQuality>` element.

6 Coordinate systems

6.1 Coordinate systems overview

OpenDRIVE uses three types of coordinate systems, as shown in Figure 5:

- The inertial x/y/z coordinate system
- The reference line s/t/h coordinate system
- The local u/v/z coordinate system

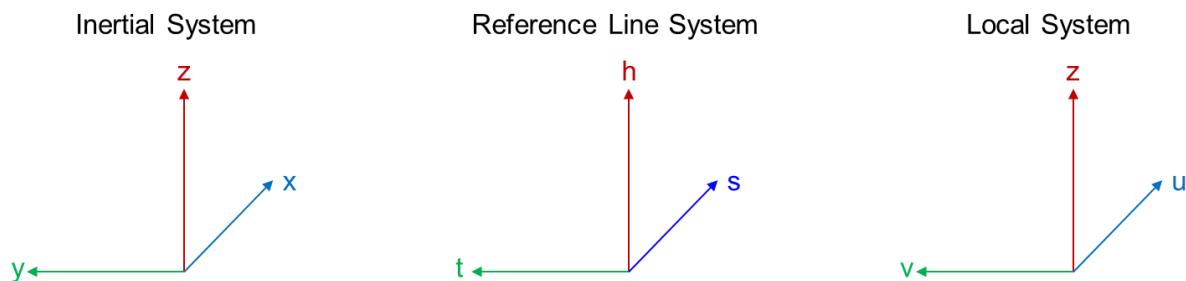


Figure 5: available coordinate systems in OpenDRIVE

If not indicated otherwise, the local coordinate system is located and oriented relative to the reference line coordinate system. The reference line coordinate system is located and oriented relative to the inertial coordinate system by specifying the origin, as well as the heading, roll, and pitch rotation angles of the origins with respect to each other.

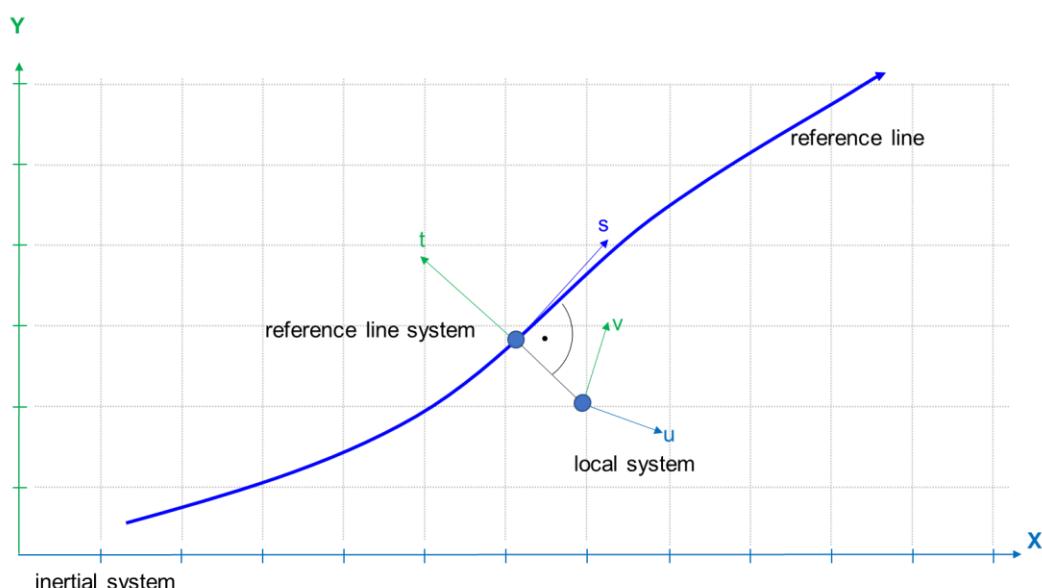


Figure 6: Coordinate systems in OpenDRIVE interacting with another

6.2 Inertial coordinate system

The inertial system is a right-handed coordinate system according to ISO 8855 [2] with the axes pointing to the following directions (see Figure 7):

- x right
- y up
- z coming out of drawing plane

For geographic reference, the following convention applies:

- x east
- y north
- z up

Elements like objects and signals can be placed within the inertial coordinate system by applying a heading, followed by pitch, followed by roll:

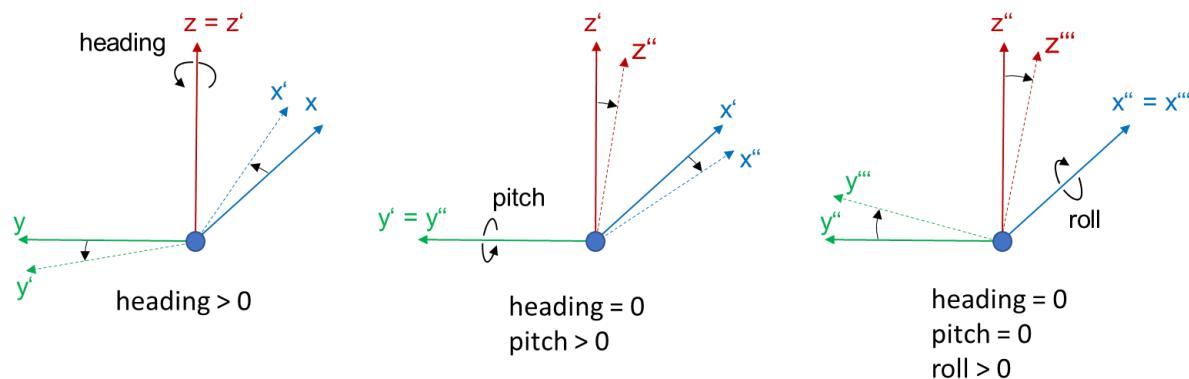


Figure 7: Inertial coordinate system with defined rotations

Figure 7 shows the positive axes and positive directions of the corresponding angles.

- *heading* around z-axis, where
heading = 0.0: x' points into direction of x-axis / east
heading = + $\pi/2$: x' points into direction of y-axis / north
- *pitch* around y'-axis, where
pitch = 0.0: x''/y''-plane = x'/y'-plane
pitch = + $\pi/2$: direction x'' = -z' = -z
- *roll* around x''-axis, where
roll = 0.0: x''/y''-plane = x''/y''-plane
roll = + $\pi/2$: direction z''' = -y''

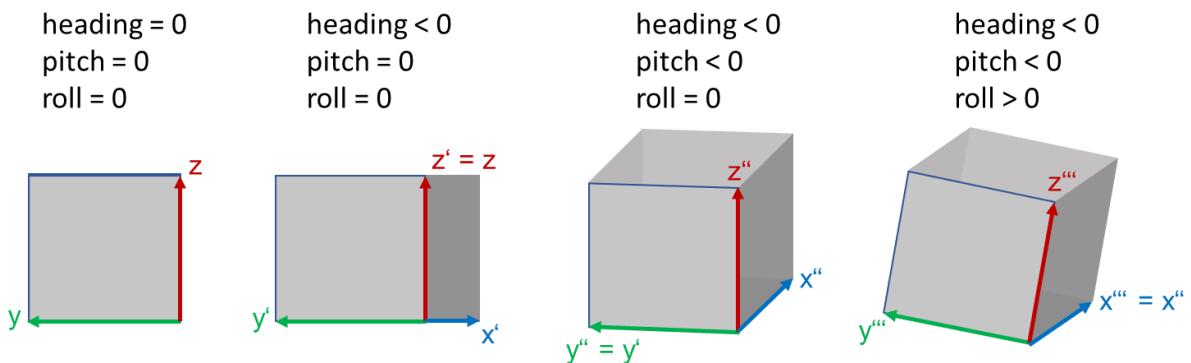


Figure 8: combined heading, pitch, roll on a cube

$x'/y'/(z'=z)$ denotes the coordinate system after rotating $x/y/z$ with the heading angle around the z -axis. The coordinate system $x''/(y''=y')/z''$ denotes the coordinate system after rotating $x'/y'/z'$ with the pitch angle around the y' -axis. The final rotated coordinate system ($x'''=x''$)/ y'''/z''' is obtained after rotating system $x''/y''/z''$ with roll angle.

6.3 Reference line coordinate system

The reference line coordinate system applies along the reference line of a road. It is a right-handed coordinate system. The s -direction follows the tangent of the reference line. It has to be noted that the reference line is always located within the x/y -plane defined by the inertial coordinate system. The t -direction is orthogonal to the s -direction. The right-hand system is completed by defining the up-direction h orthogonal to x -axis and y -axis. The following degrees of freedom are defined:

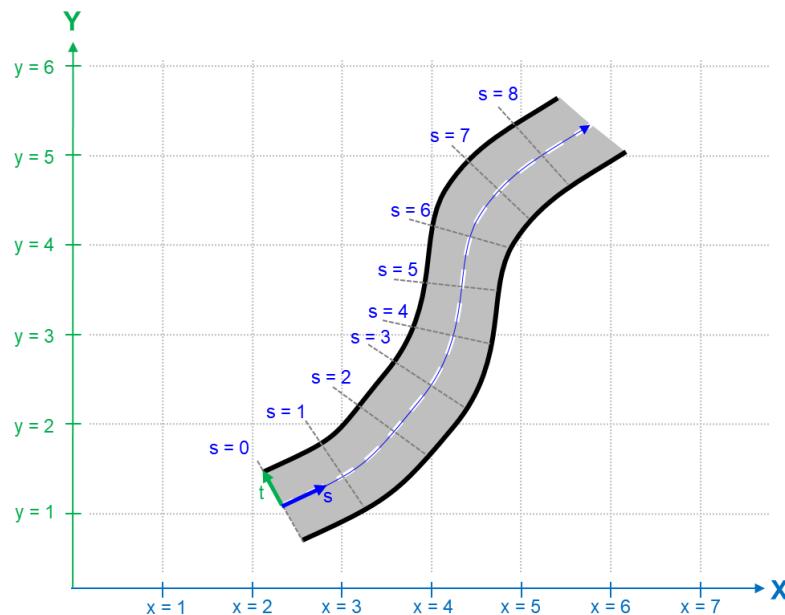


Figure 9: Reference line coordinate system

- s coordinate along reference line, measured in [m] from the beginning of the road reference line, calculated in the xy -plane (that is, not taking into account the elevation profile of the road)
- t lateral position, positive to the left within the inertial x/y plane
- h orthogonal to st plane in a right-handed coordinate system

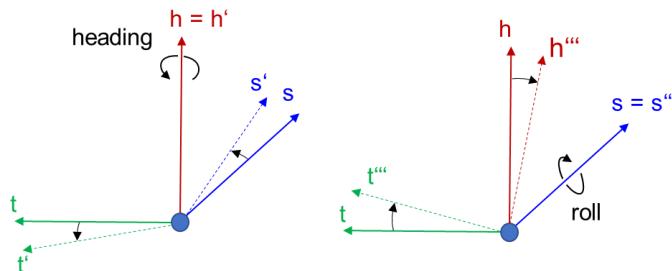


Figure 10: Reference line system with defined rotations

- *heading* around h -axis, where
 $heading = 0.0$: s' is directed forward along s -direction
 $heading = +\pi/2$: s' is directed to t
- *roll* around s' -axis, where
 $roll = 0.0$: s'''/t''' -plane = s'/t' -plane

Similar to the inertial system, the $s'/t'/h'$ and $s'''/t'''/h'''$ denote the rotated systems around heading and roll angle.

The reference line coordinate system can be positioned in the inertial space by providing the origin's coordinates and the orientation (heading) of the origin with respect to the inertial system, as shown in Figure 11.

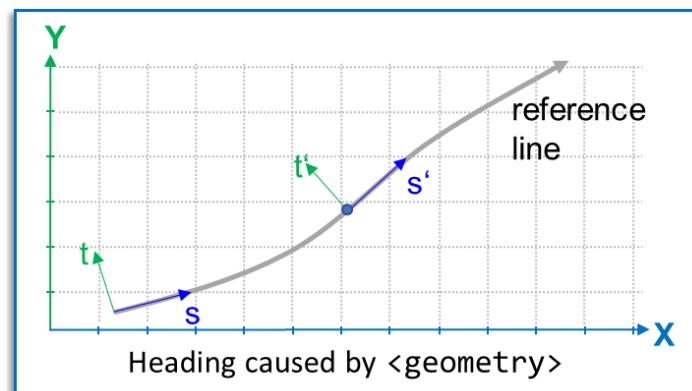


Figure 11: Heading in reference line

Superelevation causes a roll in the reference line.

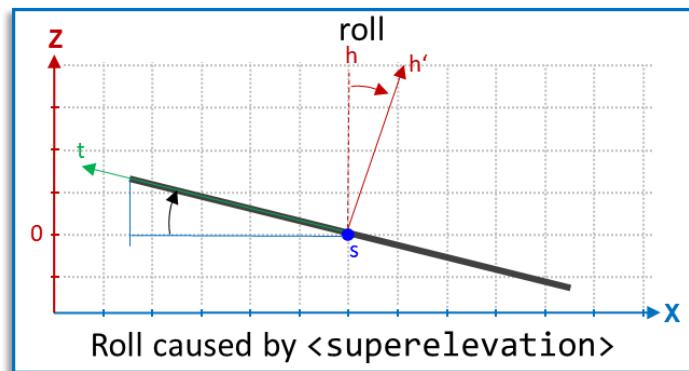


Figure 12: Roll in reference line

For the s/t/h coordinate system no pitch is possible, the elevation of the reference line is as follows. Elevation has no effect on the length of s.

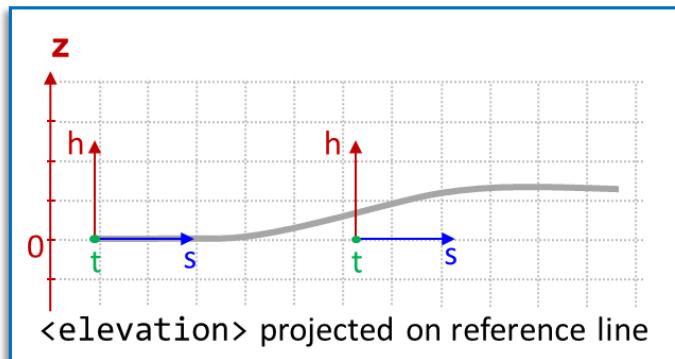


Figure 13: Elevation of reference line

6.4 Local coordinate system

The local system is a right-handed coordinate system according to ISO 8855 with the axes pointing to the following directions. For a non-rotated coordinate system the following applies:

- u Forward matches s
- v Left matches t
- z Up matches h

Elements like objects can be placed within the local coordinate system by applying a heading, followed by pitch, followed by roll:

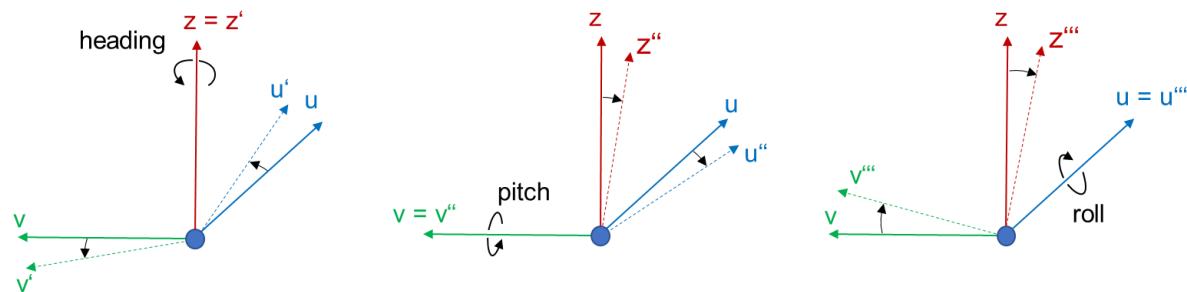


Figure 14: Local system with defined rotations

Within the local system, the following angles are defined:

- *heading* around z-axis, 0.0 = east
heading = 0.0:
heading = $+\pi/2$:
u' is directed forward along u-direction
u' is directed to t
- *pitch* around v'-axis, where
pitch = 0.0:
u''/v''-plane = u'/v'-plane
- *roll* around u''-axis, where
roll = 0.0:
u'''/v'''-plane = u''/v''-plane

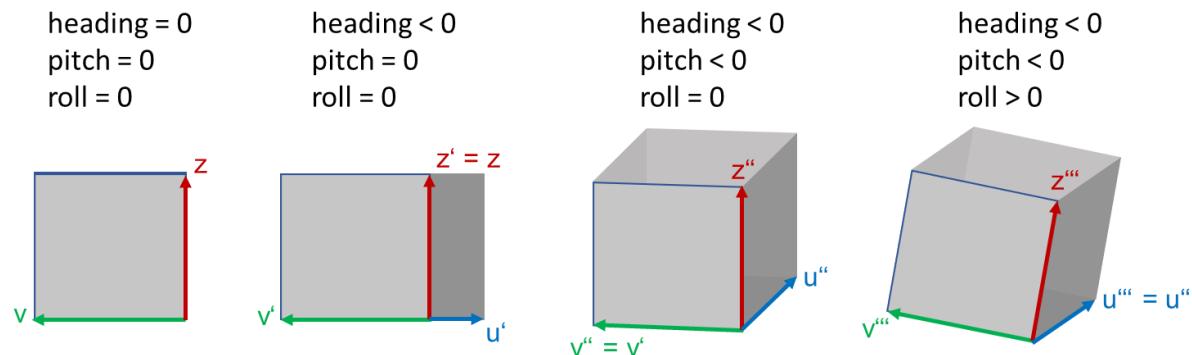


Figure 15: Local coordinate systems with heading, pitch, roll

Figure 14 shows the positive axes and positive directions of the corresponding angles. The local system can only be positioned in reference line space by providing the origin of the local coordinate system within the reference line system and the orientation (heading) of the local system with respect to the reference line system, as shown in figure 16.

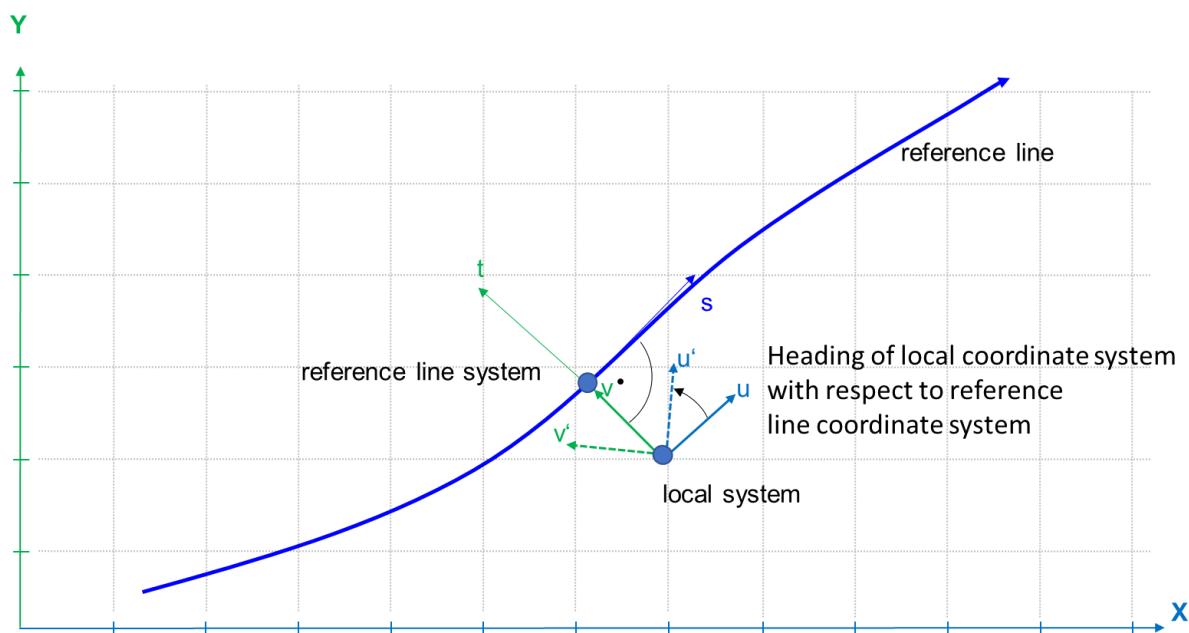


Figure 16: Local coordinate system with respect to reference line system

6.5 Summary of all available coordinate systems

Inertial, reference line and local system are used in OpenDRIVE at the same time. An example for positioning and orientation of the different coordinate systems relative to each other is depicted in Figure 17.

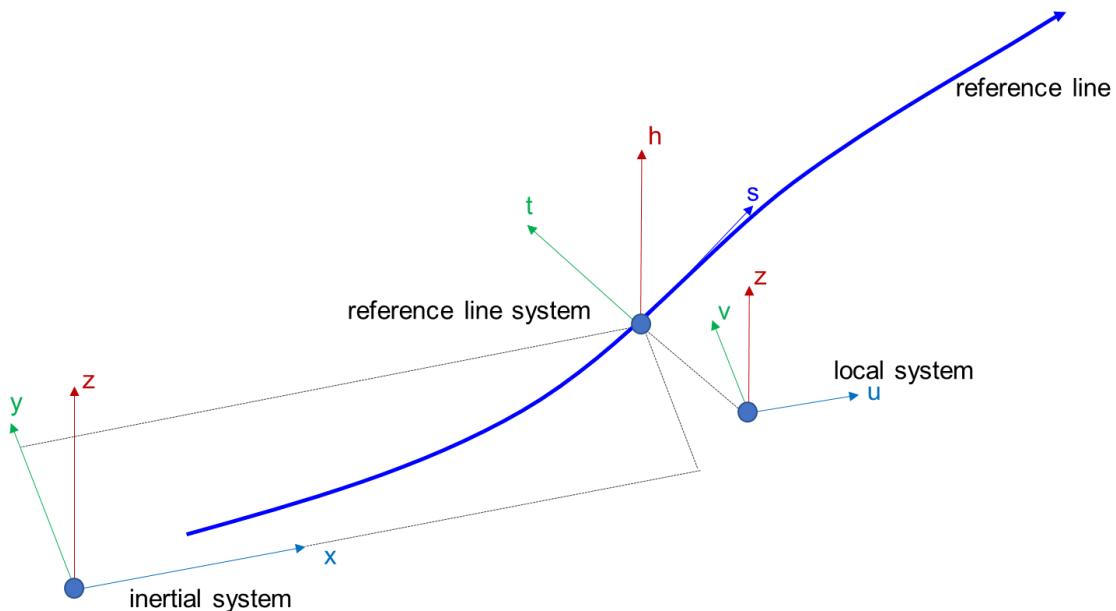


Figure 17: Summary of coordinate system in OpenDRIVE

6.6 Georeferencing in OpenDRIVE

Spatial reference systems are standardized by the European Petroleum Survey Group Geodesy (EPSG [5]) and are defined by parameters describing the geodetic datum. A geodetic datum is a coordinate reference system for a collection of positions that are relative to an ellipsoid model of the earth.

A geodetic datum is described by a projection string according to PROJ, that is, a format for the exchange of data between two coordinate systems. This data shall be marked as CDATA, because it may contain characters that interfere with the XML syntax of an element's attribute.

In OpenDRIVE, the information about the geographic reference of an OpenDRIVE dataset is represented by the `<geoReference>` element within the `<header>` element.

proj-strings, as shown in the below XML Example, contain all parameters that define the used spatial reference system:

For detailed information on proj-strings, see <https://proj.org/usage/projections.html>

There shall be no more than one definition of the projection. If the definition is missing, a local Cartesian coordinate system is assumed.

It is highly recommended to use official parameter sets for proj-strings, which can be found at <https://epsg.io/>. Parameters should not be changed. Some spatial reference systems, for example, UTM, have implicit false easting and northing, which are defined using the parameters `+x_0` and `+y_0`.

If you want to apply an offset, use the `<offset>` element instead of changing all parameter values.

XML example:

```
<geoReference>
    <! [CDATA[+proj=utm    +zone=32    +ellps=GRS80    +towgs84=0,0,0,0,0,0,0
          +units=m +no_defs ]]>
</geoReference>
```

6.6.1 Offset

To avoid large coordinates, an offset of the whole dataset may be applied using the `<offset>` element. It enables inertial relocation and re-orientation of datasets.

The dataset is first translated by `@x`, `@y`, and `@z`. Afterwards, it is rotated by `@hdg` around the new origin. Rotation around the z-axis should be avoided.

In OpenDRIVE, the offset of a database is represented by the `<offset>` element within the `<header>` element.

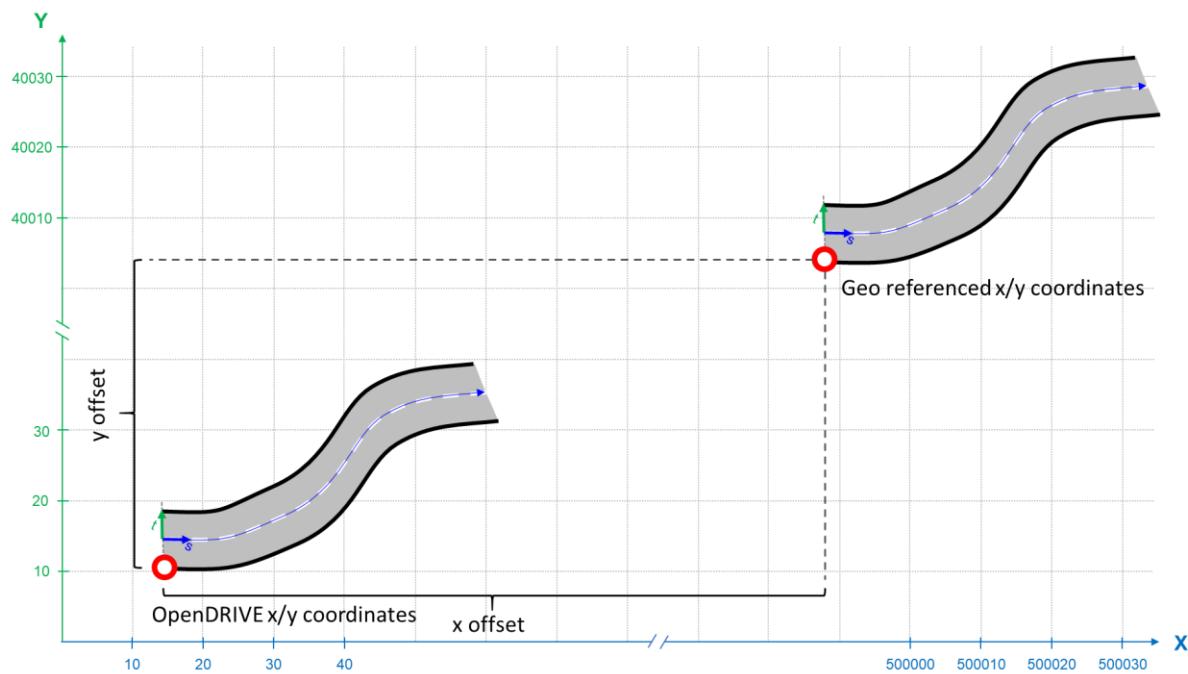


Figure 18: Offset for the geoReference

Rules

The following rule applies to the georeferenced element:

- The `<offset>` should be such that the *x* and *y* coordinates of OpenDRIVE are approximately centered around (0;0). If the *x* and *y* coordinates are too large, applications using float coordinates internally might not be able to process them accurately enough due to the limited precision of IEEE 754 double precision floating point numbers

7 Geometry

Road courses can have many different shapes. There are long, straight roads on open ground, elongated curves on motorways, or narrow turns in the mountains. In order to model all these road courses in a mathematically correct way, OpenDRIVE provides a variety of geometry elements.

Figure 19 shows the five possible ways to define the geometry of a road's reference line:

- A straight line
- A spiral or a clothoid that has a linearly changing curvature
- An arc that has a constant curvature
- Cubic polynomials
- Parametric cubic polynomials

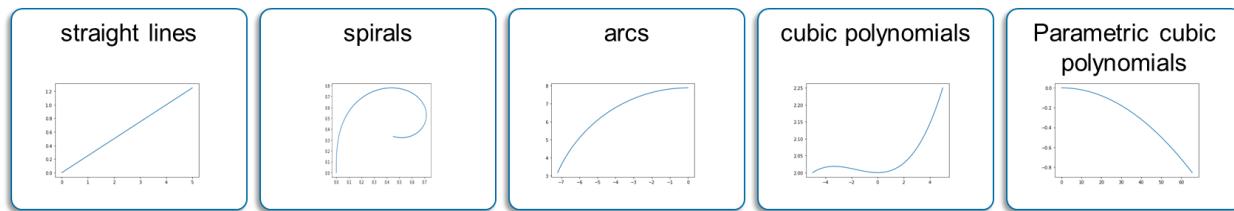


Figure 19: Geometry elements in OpenDRIVE

7.1 Road reference line

The basic element of every road in OpenDRIVE is the road reference line. All geometry elements that describe the road shape and further properties of the road are defined along the reference line. These properties include lanes and signals.

By definition, the reference line runs in s-direction, while the lateral deviation of objects from the reference line runs in t-direction.

Figure 20 shows the different parts of a road in OpenDRIVE.

- The road reference line
- The individual lanes of a road
- Features like signals that are placed along the road

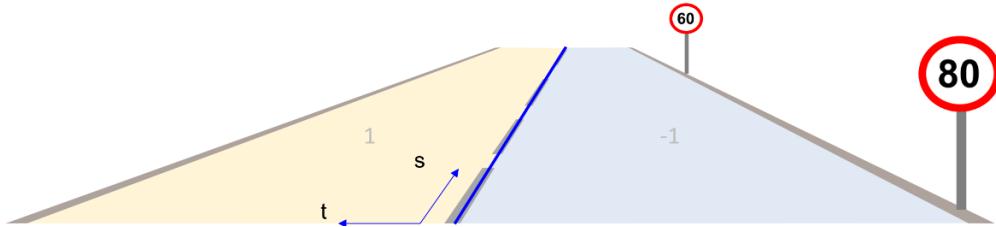


Figure 20: The individual parts of a road

In OpenDRIVE, the geometry of a reference line is represented by the `<geometry>` element within the `<planView>` element.

The `<planView>` element is a mandatory element in every `<road>` element.

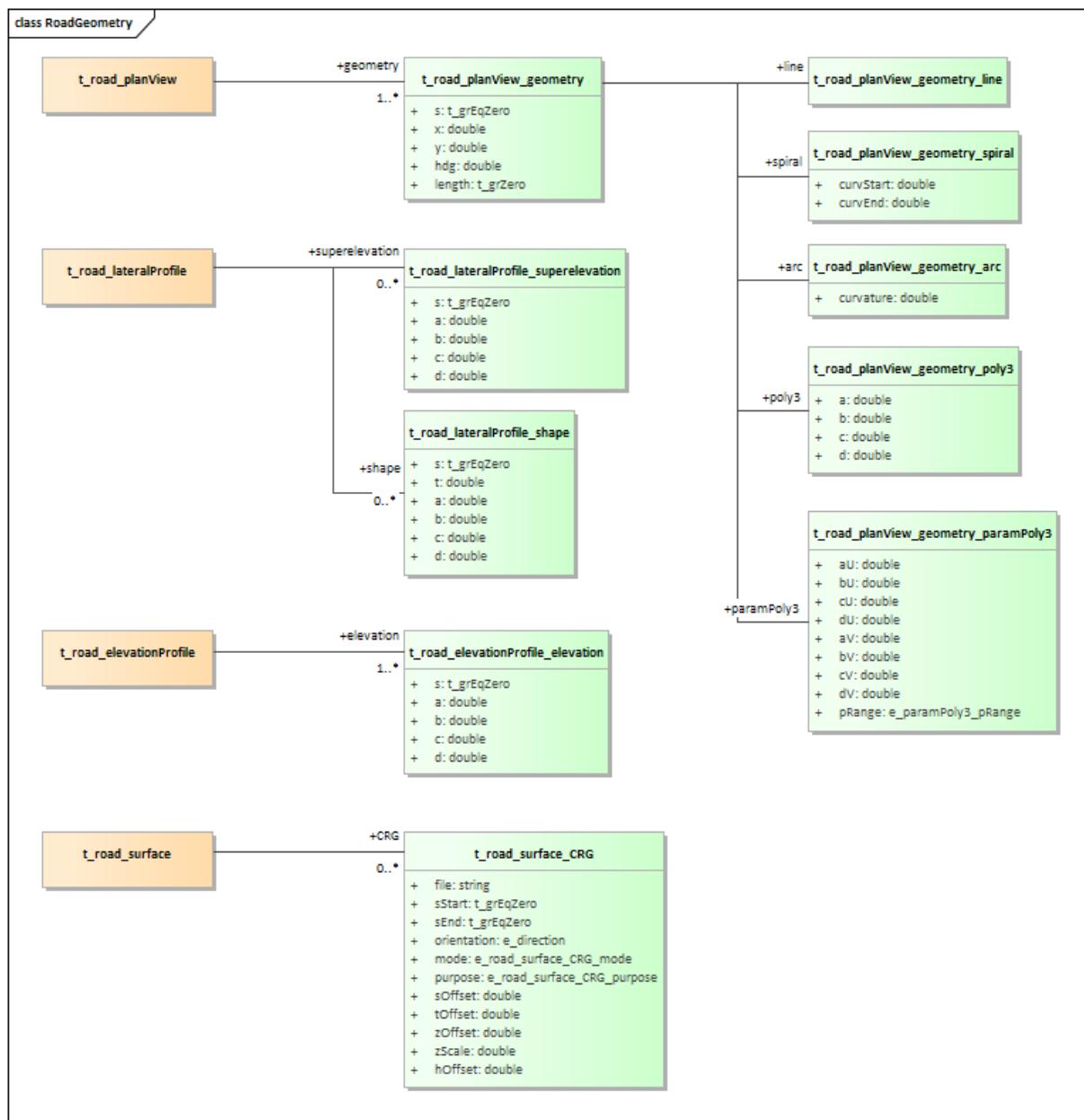


Figure 21: UML model Road Geometry including the Reference Line elements

Attributes

t_road_planView_geometry

attributes

name	type	unit	value	Description
s	t_grEqZero	m	[0;∞[s-coordinate of start position
x	double	m]∞;∞[Start position (x inertial)
y	double	m]∞;∞[Start position (y inertial)
hdg	double	rad]∞;∞[Start orientation (inertial heading)
length	t_grZero	m	[0;∞[Length of the element's reference line

Table 6: Attributes of the planView element

Rules

The following rules apply to road reference lines:

- Each road shall have a reference line.
- There shall be only one reference line per road.
- The reference line usually runs in the center of the road but may be laterally offset.
- Geometry elements shall be listed in ascending order along the reference line, that is, in increasing s-position.
- One `<geometry>` element shall contain only one element that further specifies the geometry of the road.
- If two roads are connected without a junction, the reference line of a new road shall always begin at the `<contactPoint>` of its successor or predecessor road. The reference lines may be directed in opposite directions.
- A reference line shall have no leaps
- A reference line should have no kinks

Related topics

- Straight line
- Spirals
- Arc
- Roads
- Lanes

7.2 Straight line

A straight line, as shown in figure 22, is the simplest geometry element. It contains no further attributes.

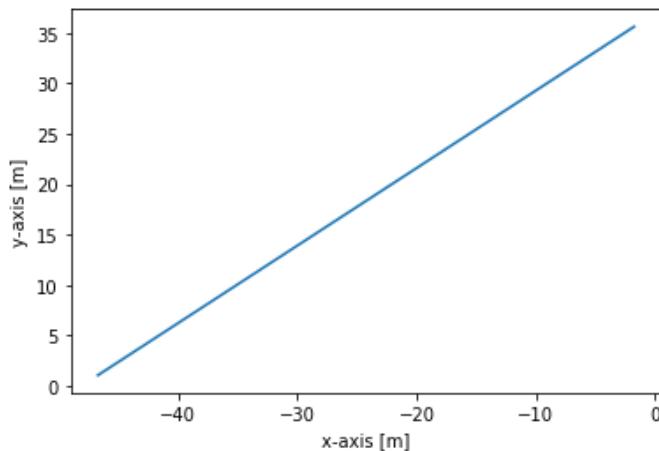


Figure 22: A straight line

In OpenDRIVE, a straight line is represented by a `<line>` element within the `<geometry>` element.

XML example

```
<planView>
  <geometry>
    <straightLine s="0.000000000000000e+00"
      x="-4.717075271170401e+01"
      y="7.2847983820912710e-01"
      hdg="6.5477882613167993e-01"
      length="5.728000000000000e+01">
      </straightLine>
    </geometry>
  </planView>
```

Related topics

- Road reference line
- Road

7.3 Spirals

A spiral is a clothoid that describes a changing curvature of the reference line, as shown in figure 23. Spirals may be used to describe the transition from a `<line>` to an `<arc>` without causing leaps in the curvature.

A spiral is characterized by the curvature at its start position (`@curvStart`) and the curvature at its end position (`@curvEnd`). Along the arc length of the spiral (see `@length` of the `<geometry>` element), the curvature is linear from the start to the end.

It is also possible to arrange several `<line>`, `<spiral>`, and `<arc>` elements in a sequence in order to describe complex curvatures.

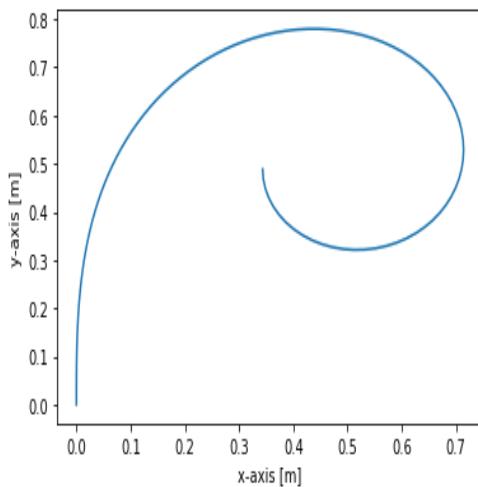


Figure 23: Road geometry described by a spiral

In OpenDRIVE, a spiral is represented by a `<spiral>` element within the `<geometry>` element.

Attributes

`t_road_planView_geometry_spiral`

attributes

name	type	unit	value	description
curvStart	double	1/m	$]-\infty; \infty[$	Curvature at the start of the element
curvEnd	double	1/m	$]-\infty; \infty[$	Curvature at the end of the element

Table 7: Attributes of the spiral element

XML example

```
<geometry s="100.0" x="38.00" y="-1.81" hdg="0.33" length="30.00">
  <spiral curvStart="0.0" curvEnd="0.013"/>
</geometry>
```

Rules

The following rules apply to spirals:

- `@curvStart` and `@curvEnd` should not be the same.

Related topics

- Arc
- Reference line
- Generating arbitrary road courses from geometry elements
- Road

7.4 Arc

An arc, as shown in figure 24, describes a road reference line with constant curvature.

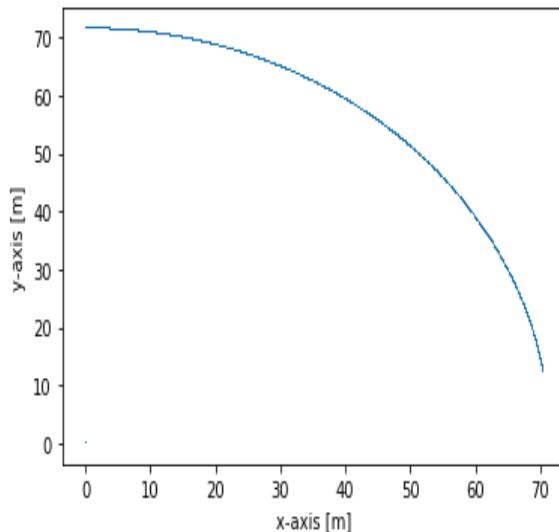


Figure 24: Road geometry described by an arc

In OpenDRIVE, an arc is represented by an `<arc>` element within the `<geometry>` element.

Attributes

t_road_planView_geometry_arc

attributes					
	name	type	unit	value	description
	curvature	double	1/m]-∞;∞[Constant curvature throughout the element

Table 8: Attributes of the arc element

XML example

```
<planView>
  <geometry
    s="3.6612031746270386e+00"
    x="-4.6416930098385274e+00"
    y="4.3409250448366459e+00"
    hdg="5.2962250374496271e+00"
    length="9.1954178989066371e+00">
    <arc curvature="-1.2698412698412698e-01"/>
  </geometry>
</planView>
```

Rules

The following rules apply to arcs:

- The curvature should not be zero.

Related topics

- Reference line
- Road

7.5 Generating arbitrary road courses from geometry elements

The combination of all geometry elements available in OpenDRIVE allows for the creation of a great variety of road courses, as shown in figure 25.

To avoid leaps in the curvature, it is recommended to use spirals to combine lines with arcs and other elements with different curvatures.

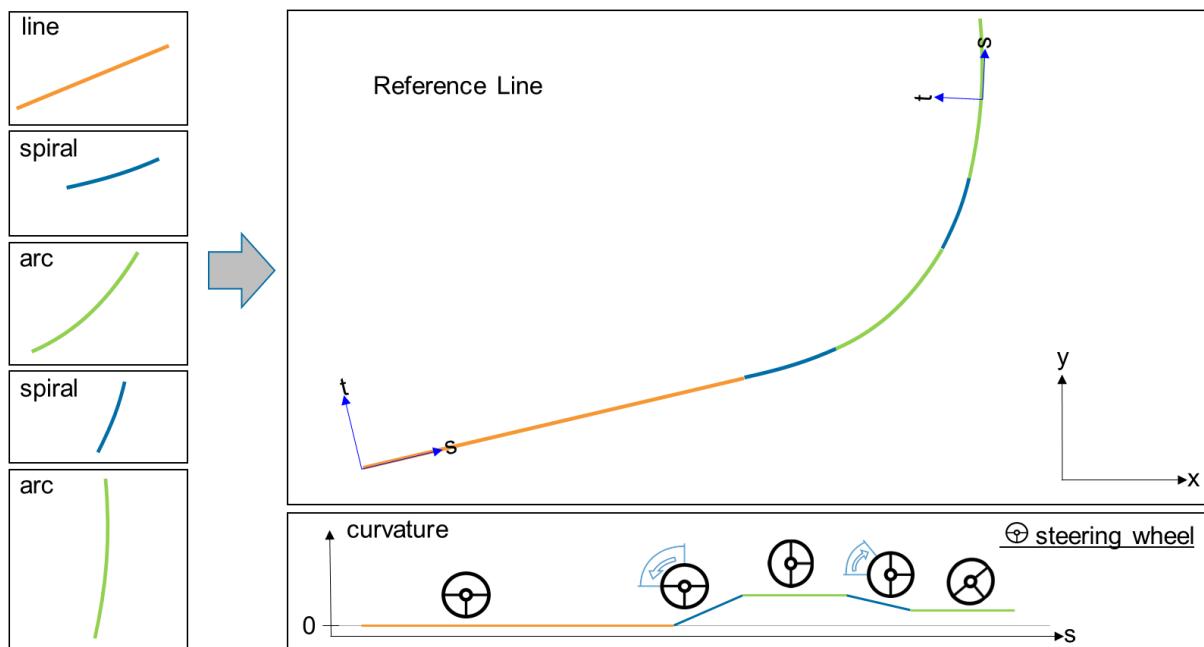


Figure 25: Creating a reference line from geometry elements

XML example:

See the sample file Ex_Line-Spiral-Arc.xodr

7.6 Cubic polynom (deprecated)

Cubic polynomials may be used to generate complex road courses that are derived from measurement data. For a given sequence of measured coordinates along the reference line in the x/y-coordinate system, measurement pairs define the polynom limits of the segment.

The reference line of the road is described by a local cubic polynom. Specifying continuity conditions, for example segment continuity, tangent and/or curvature continuity, at the limits of the segment allows to merge several cubic polynom segments and to form a global cubic spline interpolation curve for the entire course of the road.

As an additional advantage, routing along polynoms can be realized more efficiently than along clothoids.

7.6.1 Background information on cubic polynomials

The interpolation of a cubic polynom in the x/y-coordinate system is described with the following formula:

$$y(x) = a + b*x + c*x^2 + d*x^3$$

The polynom parameters a, b, c, d in the calculation are used to define the course of the roads. With the help of the parameters a-d, the y-coordinate can be calculated from the x-coordinate at every point in the coordinate system.

Figure 26 shows a cubic polynom in the x/y-coordinate system with the following values:

$$a = 20$$

$$b = 0$$

$$c = 0.0005$$

$$d = 0.0001$$

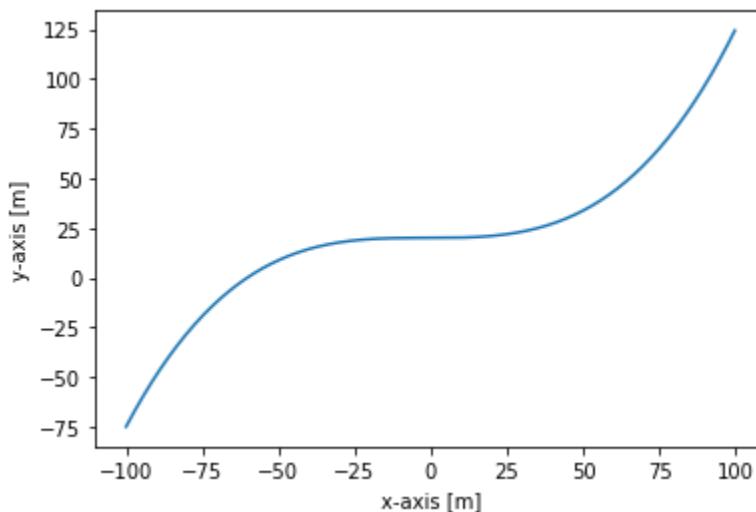


Figure 26: A cubic polynom

7.6.2 Creating roads using cubic polynomials

A cubic polynom described in the x/y-coordinate system is not suitable to describe curved segments with an arbitrary orientation, as shown in figure 27. To handle curved segments with two or more y-coordinates at a given x-coordinate, cubic polynom segments may be defined with respect to a local u/v-coordinate system. Using the local u/v-coordinate system increases flexibility in the curve definition. The following formula is used:

$$v(u) = a + b*u + c*u^2 + d*u^3$$

The orientation of the local u/v-coordinate system should be chosen in such a way that the curve is expressed as a function v(u) at increasing u-coordinates.



Figure 27: A curve which can't be represented by a cubic polynom w.r.t x parameter

Usually, the u/v-coordinate system is aligned with the s/t-coordinate system at the segment's start position (@x,@y) and start orientation @hdg, specified in the <geometry> element. This choice results in polynom parameters a=b=0 (see figure 28). As an additional option, the local u/v-coordinate system may be rotated relative to the start point (@x,@y) by specifying a polynom parameter @b that is unequal to zero. Here, the arctan (@b) defines the start heading of the polynom curve with respect to the local u/v-coordinate system. An additional shift of the u/v-coordinate origin along the v-coordinate axis, while (@x,@y) shall be located at u=0, may be achieved by setting the polynom parameter @a unequal to zero (see figure 29). The parameter u may be varied within 0 and the projection of the end point of the curve onto the u-coordinate axis.

For the given parameter u, the local coordinate v(u) defines the point on the curve in the local u/v-coordinate system.

$$v(u) = a + b*u + c*u^2 + d*u^3$$

Taking into account shift and rotation parameters @a and @b and the (@x,@y) and @hdg specified in the <geometry> element, the final x/y-curve position at a given u-coordinate, as shown in figure 29

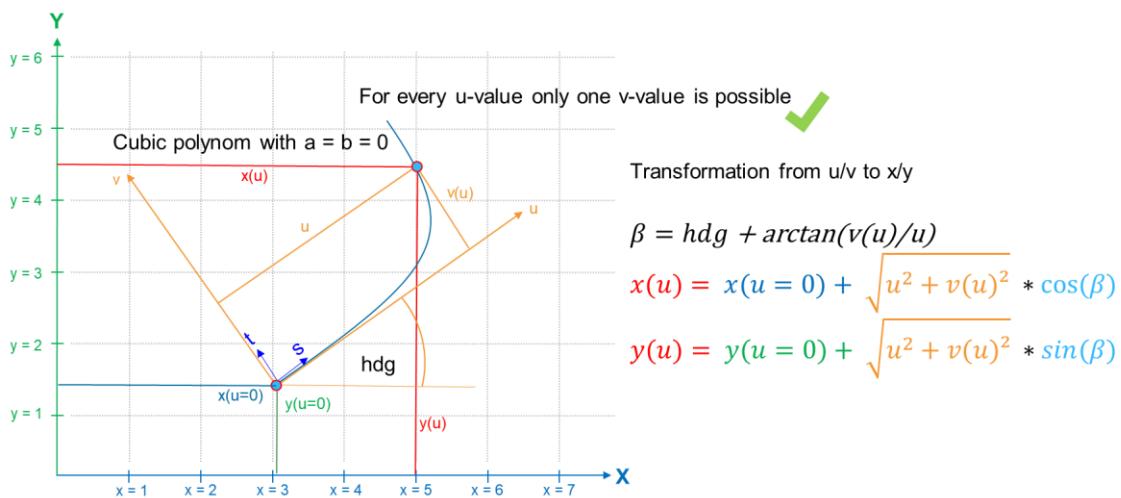


Figure 28: Transformation from a u/v- to a x/y coordinate system with $a=b=0$

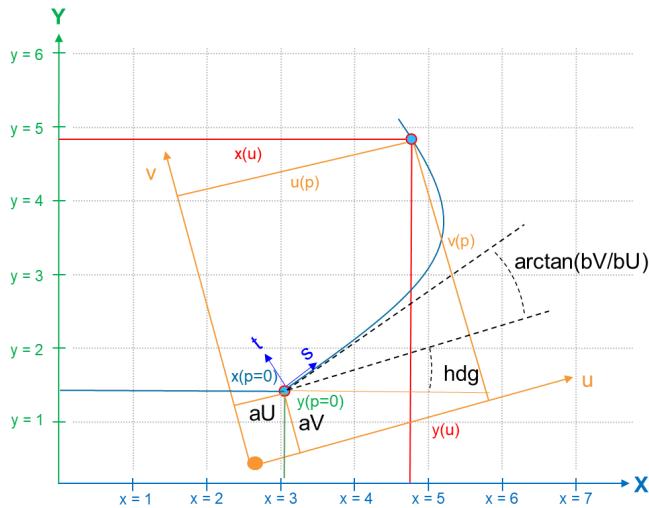


Figure 29: Transformation from a u/v- to a x/y coordinate system with $a \neq 0$ and $b \neq 0$

In OpenDRIVE, a cubic polynom is represented by a `<poly3>` element within the `<geometry>` element.

Attributes

`t_road_planView_geometry_poly3`

attributes					
	name	type	unit	value	description
	a	double	m	$]^{-\infty}; \infty[$	Polynom parameter a
	b	double	1	$]^{-\infty}; \infty[$	Polynom parameter b
	c	double	1/m	$]^{-\infty}; \infty[$	Polynom parameter c
	d	double	1/m ²	$]^{-\infty}; \infty[$	Polynom parameter d

Table 9: Attributes of the poly3 element

XML example

```
<geometry
    s="0.000000000000000e+00"
    x="-6.8858131487889267e+01"
    y="4.1522491349480972e-01"
    hdg="6.5004409066736524e-01"
    length="2.5615689718113455e+01">
<poly3
    a="0.000000000000000e+00"
    b="0.000000000000000e+00"
    c="1.4658732624442020e-02"
    d="-5.7746497381565959e-04"/>
</geometry>
<geometry
    s="2.5615689718113455e+01"
    x="-4.8650519031141869e+01"
    y="1.5778546712802767e+01"
    hdg="2.9381264033570398e-01"
    length="3.1394863696852912e+01">
<poly3
    a="0.000000000000000e+00"
    b="0.000000000000000e+00"
    c="-1.9578575382799307e-02"
    d="2.3347864348004167e-04"/>
</geometry>
```

Rules

The following rules apply to cubic polynomials:

- A cubic polynomial may be used to describe the course of a road for which measurement data is available
- If the local u/v-coordinate system is aligned with the s/t-coordinate system of the start point, the polynomial parameter coefficients are $a=b=0$.
- The starting point (@x, @y) of the <geometry> element is located on the v-coordinate axis of the local u/v-coordinate system.
- The polynomial parameters a and b should be 0 for a smooth reference line.

Related topics

- Parametric cubic curve
- Georeferencing
- Roads
- Generating arbitrary road courses from geometry elements

7.7 Parametric cubic curve

Parametric cubic curves are used for complex curves that are to be generated from measurement data. Parametric cubic curves are more flexible and allow a greater variety of road courses than cubic polynomials. In comparison to cubic polynomials that are defined in a x/y-coordinate system or as local u/v-coordinates, the coordinates x and y are interpolated separately by their own splines with respect to a common interpolation parameter p.

7.7.1 Generating roads using parametric cubic curves

Generating road courses with parametric cubic curves only require x- and y-coordinates. For reasons of consistency to cubic polynomials, they may be calculated simultaneously to cubic polynomials using local u- and v-coordinates.

$$u(p) = aU + bU*p + cU*p^2 + dU*p^3$$

$$v(p) = aV + bV*p + cV*p^2 + dV*p^3$$

Unless otherwise stated, the interpolation parameter p is in the range [0;1]. Alternatively, it may be given in the range [0; @length of <geometry>]. Similar to cubic polynomials, the local coordinate system with the variables u and v may be placed and oriented arbitrarily.

To simplify representation, the local coordinate system may be aligned with the s/t-coordinate system at the start point (@x,@y) and start orientation @hdg:

- u points in local s-direction, meaning along the reference line at the start point
- v points in local t-direction, meaning in lateral deviation from the reference line at the start point
- the parameters @aU, @aV and @bV shall be zero

Providing non-zero values for the parameters @aU, @aV and @bV leads to a shift and rotation of the s/t coordinates as shown Figure 26, Figure 27 and Figure 28.

After defining the points of the curve for a given parameter p, the u-values and v-values are transformed into values of the x/y-coordinate system with regard to the shifts and orientation specified by the parameters @aU, @aV, @bU, @bV, the start coordinates (@x,@y) and the start orientation @hdg.

It has to be noted that there is a non-linear relation between the interpolation parameter p and the actual length of the arc between the start point (@x,@y) in the <geometry> element and the point (x(p),y(p)) associated with the parameter p. In general, only the startpoint and endpoint parameter p=0 and p=@length (for the option @pRange=arcLength) will coincide with the actual length of the arc.

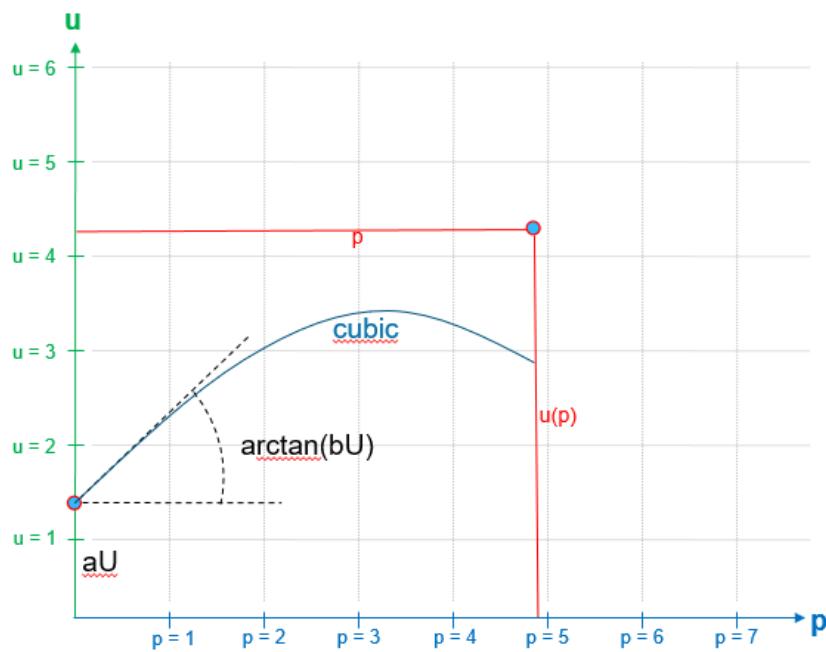


Figure 30: A parametric cubic polynomial for interpolation of u coordinate

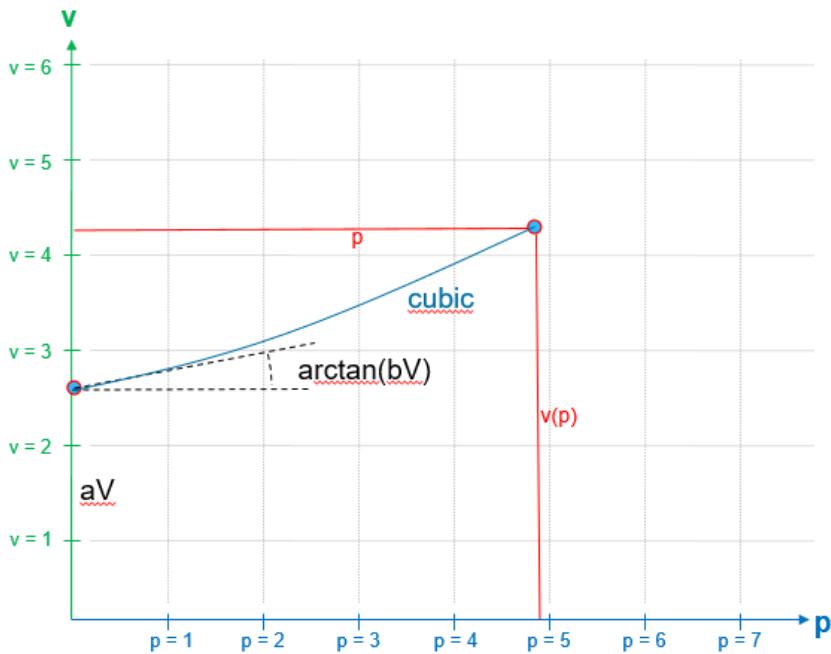


Figure 31: A parametric cubic polynom for interpolation of v coordinate

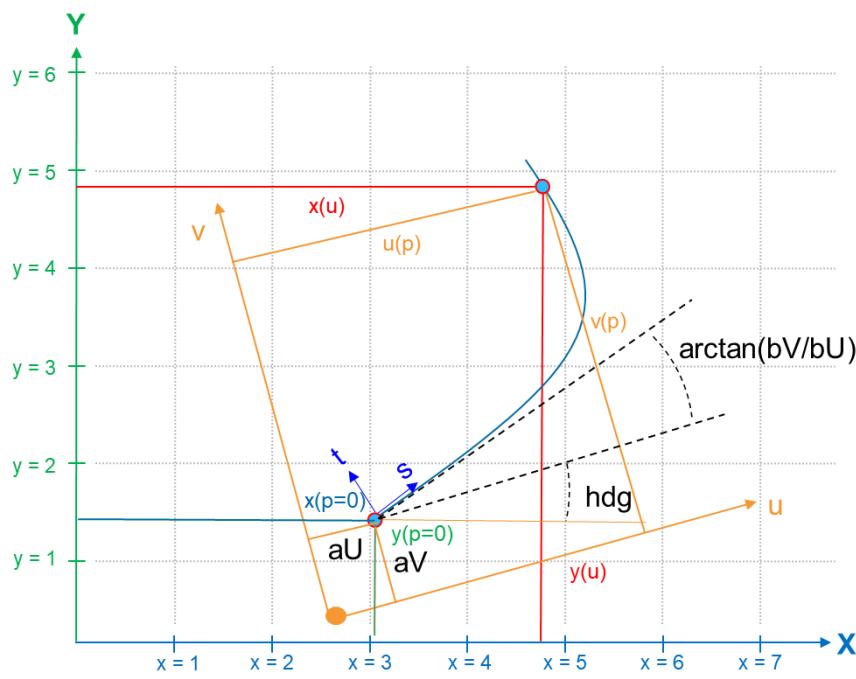


Figure 32: A parametric cubic polynomial

In OpenDRIVE, parametric cubic curves are represented by `<paramPoly3>` elements within the `<geometry>` element.

Attributes

t_road_planView_geometry_paramPoly3

attributes					
	name	type	unit	value	description
<code>aU</code>	double	m	$]^{-\infty}; \infty[$		Polynom parameter a for u
<code>bU</code>	double	1	$]^{-\infty}; \infty[$		Polynom parameter b for u
<code>cU</code>	double	$1/m$	$]^{-\infty}; \infty[$		Polynom parameter c for u
<code>dU</code>	double	$1/m^2$			Polynom parameter d for u
<code>aV</code>	double	m	$]^{-\infty}; \infty[$		Polynom parameter a for v
<code>bV</code>	double	1	$]^{-\infty}; \infty[$		Polynom parameter b for v
<code>cV</code>	double	$1/m$	$]^{-\infty}; \infty[$		Polynom parameter c for v
<code>dV</code>	double	$1/m^2$	$]^{-\infty}; \infty[$		Polynom parameter d for v
<code>pRange</code>	e_paramPoly3_pRange		arcLength ; norm alized		Range of parameter p. - Case arcLength: p in [0, @length of <geometry>] - Case normalized: p in [0, 1]

Table 10: Attributes of the paramPoly3 element

XML example

```
<planView>
  <geometry>
    s="0.000000000000e+00"
    x="6.804539427645e+05"
    y="5.422483642942e+06"
    hdg="5.287405485081e+00"
    length="6.565893957370e+01">
    <paramPoly3
      aU="0.000000000000e+00"
      bU="1.000000000000e+00"
      cU="-4.666602734948e-09"
      dU="-2.629787927644e-08"
      aV="0.000000000000e+00"
      bV="1.665334536938e-16"
      cV="-1.987729787588e-04"
      dV="-1.317158625579e-09"
      pRange="arcLength">
    </paramPoly3>
  </geometry>
</planView>
```

Rules

The following rules apply to parametric cubic curves:

- If the local u/v-coordinate system is aligned with the s/t-coordinate system of the start point, the polynom parameter coefficients are @aU=@aV=@bV=0.
- If @pRange="arcLength", p may be chosen in [0, @length from <geometry>]
- If @pRange="normalized", p may be chosen in [0, 1]
- The polynomial parameters aU, bU and aV should be 0 for a smooth reference line.

Related topics

- Cubic polynom
- Generating arbitrary road courses from geometry elements
- Roads
- Georeferencing

8 Roads

In OpenDRIVE, the road network is represented by `<road>` elements. Each road runs along one road reference line. A road shall have at least one lane with a width larger than 0.

OpenDRIVE roads may be roads in the real road network or artificial road network created for application use. Each road is described by one or more `<road>` elements. One `<road>` element may cover a long stretch of a road, shorter stretches between junctions, or even several roads. A new `<road>` element should only start if the properties of the road cannot be described within the previous `<road>` element or if a junction is required.

Related topics

- Reference line
- Road linkage
- Lanes

UML Model

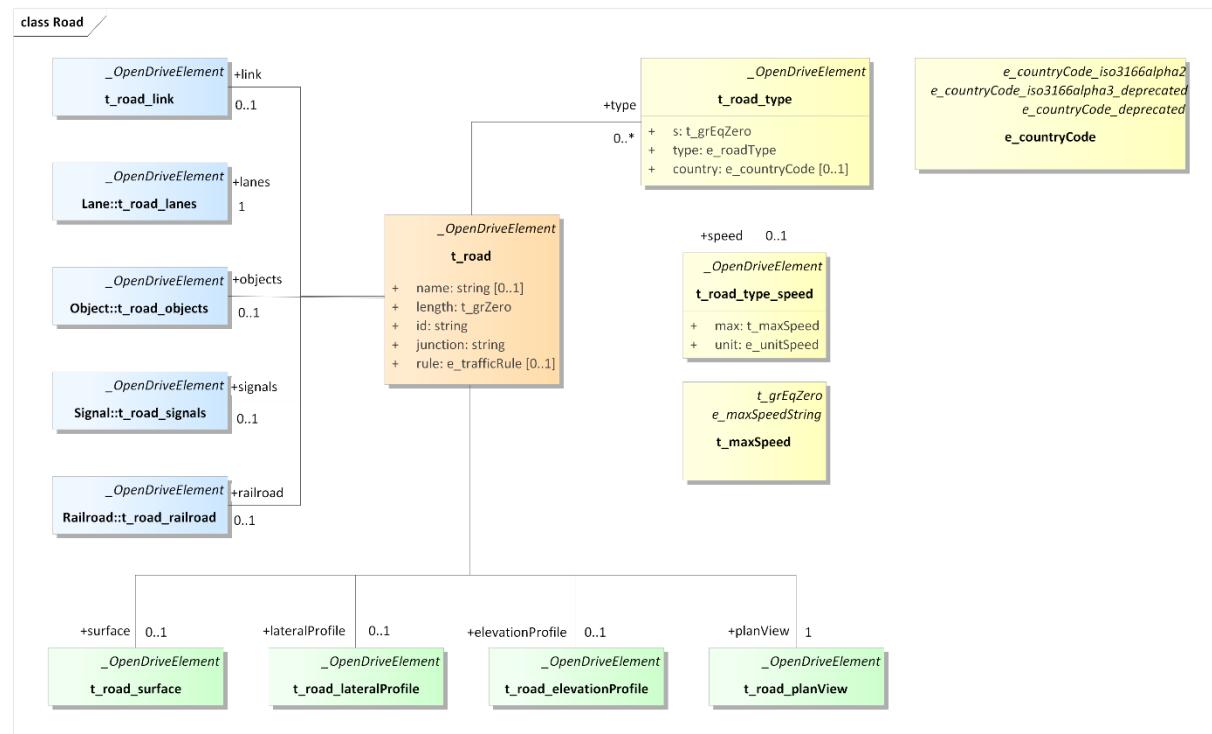


Figure 33: UML Model for Roads

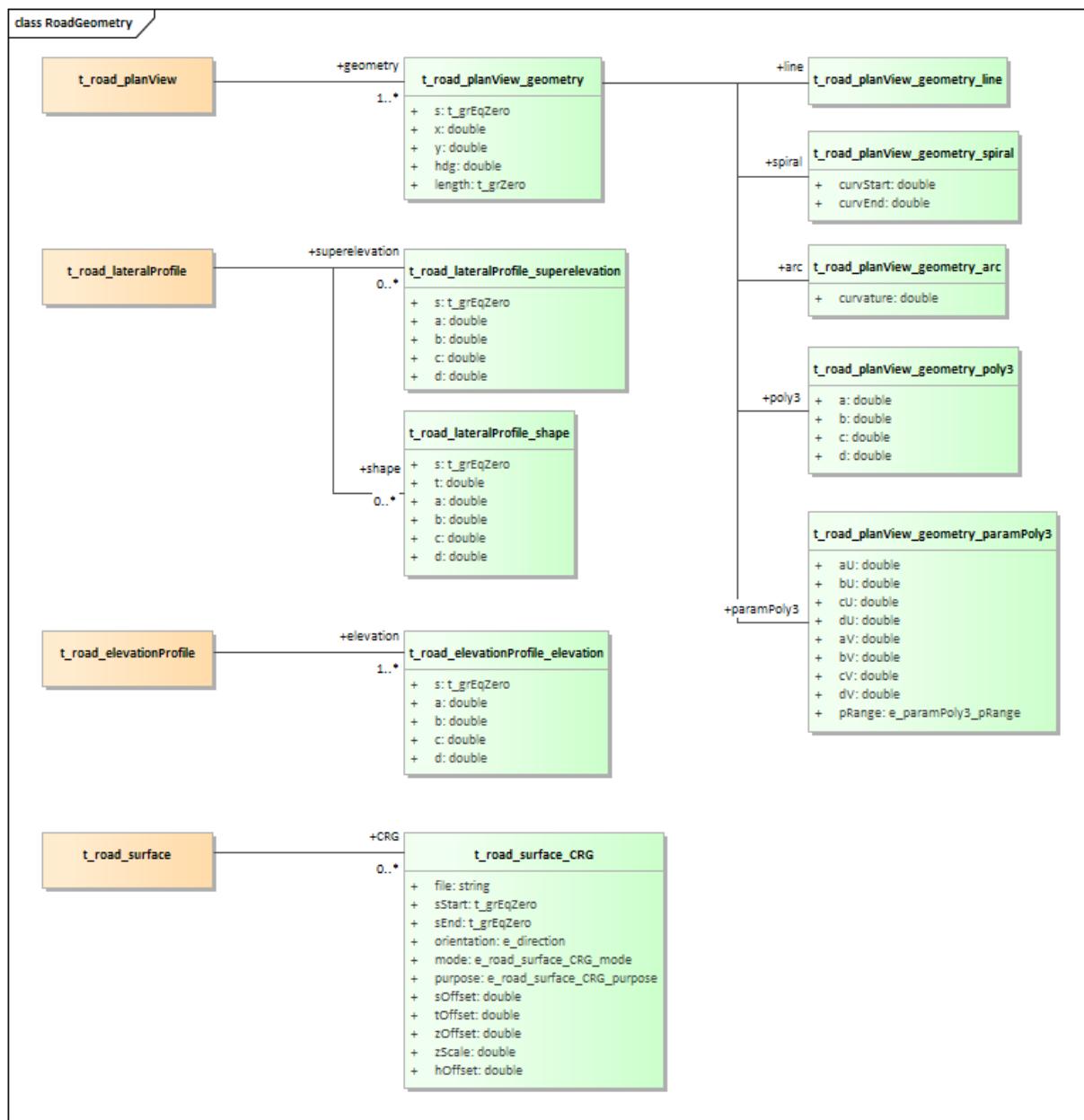


Figure 34: UML model for Road Geometry

Attributes

t_road

The road header element defines the basic parameters of an individual road.

attributes	name	type	unit	value	description
	name	string			Name of the road. May be chosen freely.
	length	t_grZero			Total length of the reference line in the xy-plane. Change in length due to elevation is not considered
	id	string		- ; [0; ∞[Unique ID within the database. If it represents an integer number, it should comply to uint32_t and stay within the given range.
	junction	string		-; -1	ID of the junction to which the road belongs as a connecting road (= -1 for none)
	rule	e_trafficRule		RHT ; LHT	Basic rule for using the road; RHT=right-hand traffic, LHT=left-hand traffic. When this attribute is missing, RHT is assumed.

Table 11: Attributes of the road element

8.1 Properties for road sections and cross section

Some properties of roads are described based on the cross section of the road. The road cross section is the orthogonal view of the road at a given point on the road reference line. An example of a property that refers to the road cross section is superelevation. Elements that are valid for a road cross section are valid for the whole width of the road at a given point on the reference line.

Other road properties are described based on the plan view of the road. This includes lanes and road elevation. For these properties, the term road section is used. Road sections describe parts of roads and their specific properties along the s-coordinate of the road reference line. Properties that are valid for a road section may not be valid for the whole width of the road, but for specific lanes only.

That means it is possible to create sections for different properties like road type or lane sections. Sections are created by an additional element within the <road> element, using new start s-coordinates. The length of a section is implicitly given by the difference between two given s-start positions. Sections shall be stored in ascending order of s-coordinates.

8.2 Road linkage

For applications to navigate through a road network, roads must be linked to each other. Roads may be connected to another road or a junction. Isolated roads are not connected to other roads or junctions.

Figure 35 shows scenarios of prohibited, allowed, and recommended road linkage. It is important that the lanes and reference lines of the roads to be linked have a direct linkage to its predecessor or successor. Overlaps or leaps should be avoided but are not prohibited if the reference lines are connected properly.

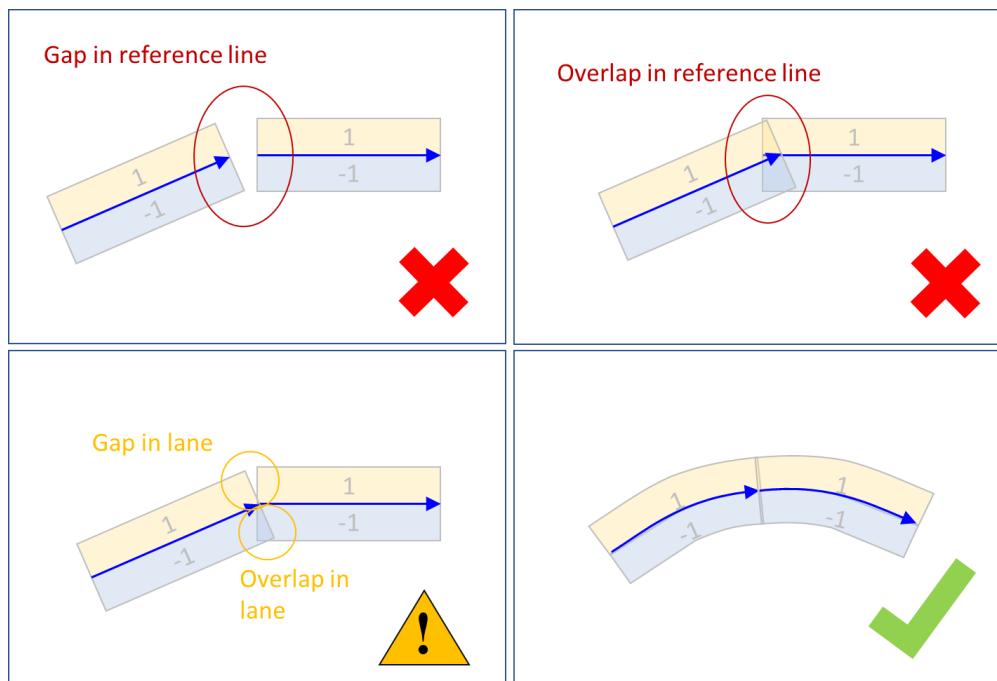


Figure 35: Allowed, prohibited, and recommended road linkage

Figure 36 shows the allowed scenarios for road linkage outside junctions, with two roads running in the same, opposite, or converging directions. Road linkage is not possible, if the two reference lines are not connected to each other.

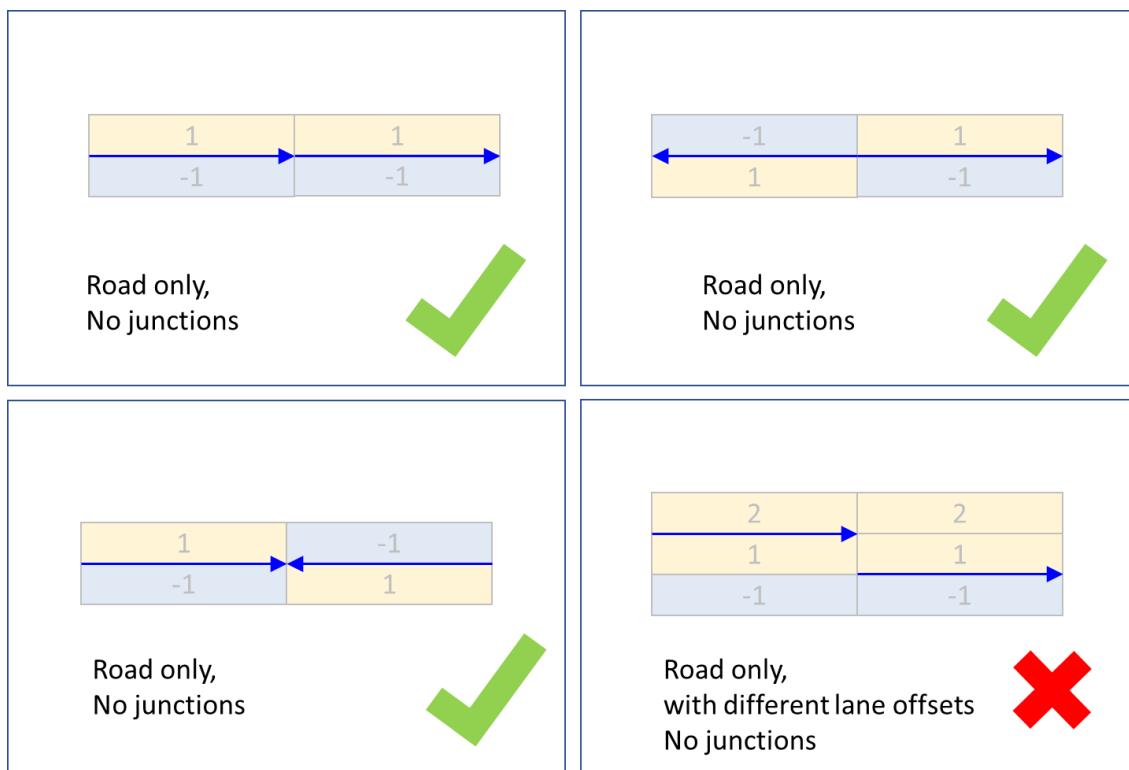


Figure 36: Allowed scenarios of road linkage

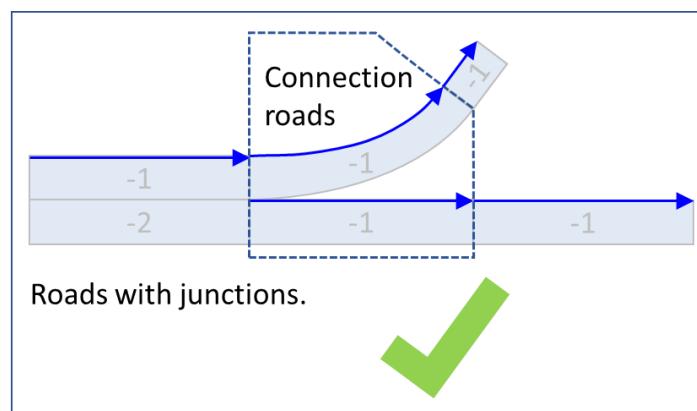


Figure 37: Allowed scenario of road linkage within a junction

In OpenDRIVE, road linkage is represented by the `<link>` element within the `<road>` element. The `<predecessor>` and `<successor>` elements are defined within the `<link>` element. For virtual and regular junctions, different attribute sets shall be used for the `<predecessor>` and `<successor>` elements.

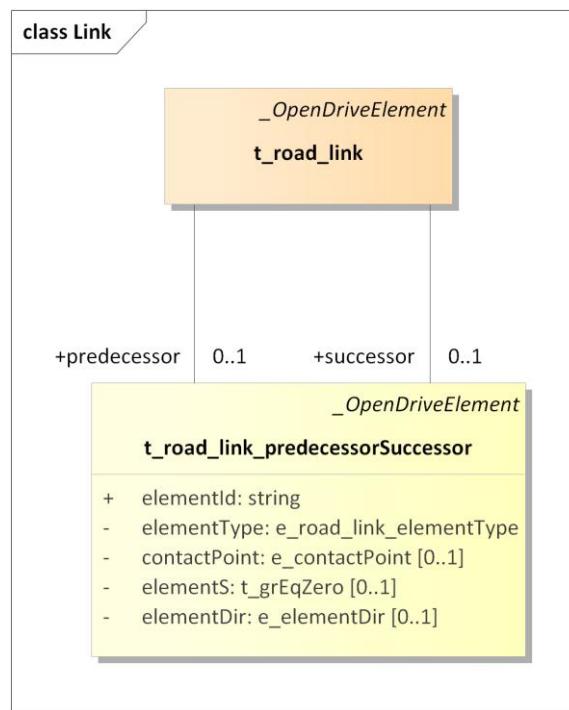


Figure 38: UML model for road linkage

Attributes

t_road_link

Follows the road header if the road is linked to a successor, a predecessor, or a neighbor. Isolated roads may omit this element.

t_road_link_predecessorSuccessor

For virtual and regular junctions, different attribute sets shall be used. @contactPoint shall be used for regular junctions; @elementS and @elementDir shall be used for virtual junctions.

attributes

	name	type	unit	value	description
	element Id	string			ID of the linked element
	element Type	e_road_link_element Type		road ; juncti on	Type of the linked element
	contact Point	e_contactPoint		start ; end	Contact point of link on the linked element

elementTypeS	t_grEqZero	m		Alternative to contactPoint for virtual junctions. Indicates a connection within the predecessor, meaning not at the start or end of the predecessor. Shall only be used for elementType "road"
elementDir	e_elementDir	+ ; -		To be provided when elementTypeS is used for the connection definition. Indicates the direction on the predecessor from which the road is entered.

Table 12: Attributes of the road link predecessorSuccessor element

Rules

The following rules apply to road linkage:

- Two roads shall only be linked directly, if the linkage is clear. If the relationship to successor or predecessor is ambiguous, junctions shall be used.
- A road may have another road or a junction as successor or predecessor. A road may also have no successor or predecessor.
- A road may serve as its own predecessor or successor.

Related topics

- Reference line
- Junctions
- Lane linkage

8.3 Road type

The road type defines the main purpose of a road and the associated traffic rules. Example road types are motorways and rural roads. The road type is valid for the entire road cross section.

The road type may be changed as often as needed within a `<road>` element. This may be done by defining different road types at given points along the reference line. One road type remains valid until another road type is defined.

In OpenDRIVE, the road type is represented by the `<type>` element within the `<road>` element. The road type itself is given in the `@type` attribute.

Attributes

t_road_type

A road type element is valid for the entire cross section of a road. It is valid until a new road type element is provided or until the road ends.

attributes	name	type	unit	value	description
	s	t_grEqZero	m	[0;∞[s-coordinate of start position
	type	e_roadType			Country code of the road, see ISO 3166-1, alpha-2 codes.[8]
	country	e_countryCode			Country code of the road, see ISO 3166-1, alpha-2 codes.[8]

Table 13: Attributes of the road type element

Rules

The following rules apply to road types:

- When the type of road changes, a new `<type>` element shall be created within the parent `<road>` element.
- Country code and state identifier may be added to the `<type>` element to specify which national traffic rules apply to this road type. The according data is stored in the application and not in OpenDRIVE.
- There shall only be ALPHA-2 country codes in use, no ALPHA-3 country codes, because only ALPHA-2 country codes support state identifiers.
- Single lanes may have another type than the road they belong to. Road type and lane type represent different properties and are both valid if specified.

Related topics

- Lane type
- Properties for road sections and cross section

8.3.1 Speed limits for road types

A speed limit may be defined for a road type. When the road type changes and a speed limit exists on that road section, a new speed element is required, because road types have no globally valid speed limits. The limit shall be defined for each road type element separately.

In OpenDRIVE, the speed limit is represented by the `<speed>` element within the `<type>` element.

Attributes

t_road_type_speed

Defines the default maximum speed allowed in conjunction with the specified road type.

attributes	name	type	unit	value	description
	max	t_max Speed		no limit ; undefined ; [0,∞[Maximum allowed speed. Given as string (only "no limit" / "undefined") or numerical value in the respective unit (see attribute unit). If the attribute unit is not specified, m/s is used as default.
	unit	e_unit Speed			Unit of the attribute max. For values, see chapter "units".

Table 14: Attributes of the road type speed element

Rules

The following rules apply to speed limits:

- A maximum speed may be defined as default value per road type element.
- Single lanes may have different speed limits than the road they belong to. They are defined as <laneSpeed>.
- Speed limits derived from signals shall always have preference.

Related topics

- Road type
- Properties for road sections and cross section
- Signals

8.4 Methods of Elevation

There are several ways to elevate a road or parts of a road:

- Road elevation specifies the elevation along the road reference line, that is in s-direction.
- The lateral profile, using superelevation and shape definition, specifies the elevation orthogonally to the reference line, that is in t-direction.

The types of road elevation are shown in figure 39

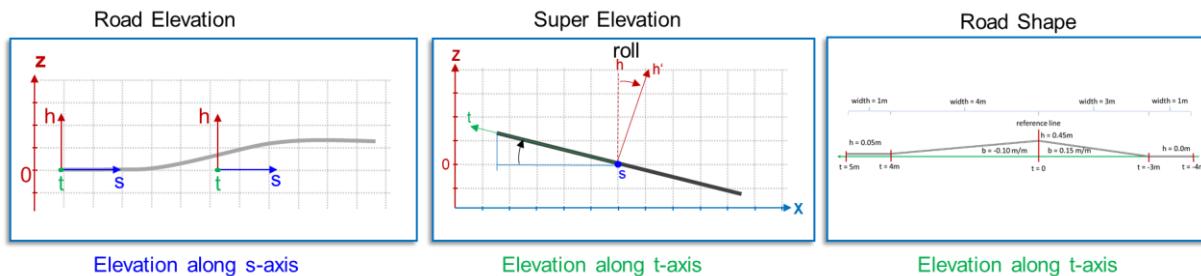


Figure 39 Types of road elevation

The s length does not change with the elevation.

8.4.1 Road elevation

A road may be elevated along its reference line. Road elevation is defined per road cross section at a given position on the reference line. Elevation is specified in meters. The default elevation of a road is zero. In case georeferencing is used, the definition of zero depends on it.

In OpenDRIVE, elevation is represented by the `<elevation>` element inside the `<elevationProfile>` element.

Attributes

`t_road_elevationProfile_elevation`

Defines an elevation element at a given position on the reference line. Elements shall be defined in ascending order along the reference line. The s length does not change with the elevation.

attributes					
name	type	unit	value	description	
s	t_grEqZero	m	[0; ∞ [s-coordinate of start position	
a	double	m] $-\infty$; ∞ [Polynom parameter a, elevation at @s (ds=0)	
b	double	1] $-\infty$; ∞ [Polynom parameter b	
c	double	1/m] $-\infty$; ∞ [Polynom parameter c	
d	double	1/m ²] $-\infty$; ∞ [Polynom parameter d	

Table 15: Attributes of the elevation element

Calculation

Road elevation is calculated with the following polynomial function of the third order:

$$\text{elev}(ds) = a + b*ds + c*ds^2 + d*ds^3$$

where

elev is the elevation (inertial z) at a given position
 a, b, c, d are the coefficients
 ds is the distance along the reference line between the start of a new elevation element and the given position.

ds restarts at zero for each element. The absolute position of an elevation value is calculated as follows:

$$s = s_{\text{start}} + ds$$

where

s is the absolute position in the reference line coordinate system
 s_{start} is the start position of the element in the reference line coordinate system

Rules

The following rules apply to road elevation:

- Roads shall be elevated along their reference line.
- Road elevation may be defined in combination with superelevation and road shape or standalone.
- Elevation elements shall be defined in ascending order. Because the elevation can go up and down, the elements shall be linked to the respective position on the reference line.
- The definition of road elevation remains valid until the next element of this type is defined.

Related topics

- Superelevation
- Shape definition

8.4.2 Superelevation

Superelevation is part of the lateral profile and describes the cross slope of the road. It may be used, for example, to incline the road to the inner side so that vehicles can drive through them more easily. For superelevated roads, the t-axis of a road is not parallel to the underlying terrain, as shown in figure 40. For this reason, a lateral profile is defined for the entire road cross section. Superelevation does not change the actual width of a lane, but it affects the projected width. The default value for superelevation is zero.

Mathematically, superelevation is defined as the roll angle of the road cross section around the reference line. That means, superelevation has positive values for roads falling to the right side and negative values for roads falling to the left side.

In the example in figure 40, the reference line is parallel to the y-axis, to simplify the given example.

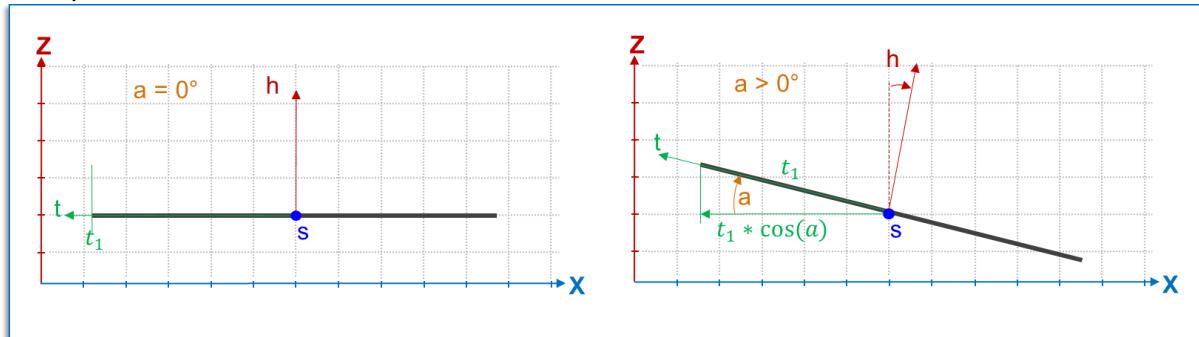


Figure 40: Superelevation

In OpenDRIVE, superelevation is represented by the `<superelevation>` element within the `<lateralProfile>` element.

Attributes

`t_road_lateralProfile_superelevation`

Defined as the road section's roll angle around the s-axis. Elements must be defined in ascending order along the reference line. The parameters of an element are valid until the next element starts or the road reference line ends. Per default, the superelevation of a road is zero.

attributes	name	Type	unit	value	Description
	s	t_grEqZero	m	[0;∞[s-coordinate of start position
	a	double	rad]∞;∞[Polynom parameter a, superelevation at @s (ds=0)
	b	double	rad/m]∞;∞[Polynom parameter b
	c	double	rad/m²]∞;∞[Polynom parameter c
	d	double	rad/m³]∞;∞[Polynom parameter d

Table 16: Attributes of the superelevation element

Calculation

Superelevation is calculated using the following polynomial function of the third order:

$$s_{\text{Elev}}(ds) = a + b*ds + c*ds^2 + d*ds^3$$

where

s_{Elev} is the superelevation at a given position
 a, b, c, d are the coefficients
 ds is the distance along the reference line between the start of a superelevation element and the given position.

ds restarts at zero for each element. The absolute position of a superelevation value is calculated as follows:

$$s = s_{start} + ds$$

where

s is the absolute position in the reference line coordinate system
 s_{start} is the start position of the element in the reference line coordinate system

Rules

The following rules apply to superelevation:

- When superelevation is defined, it shall apply to the entire road cross section.
- Single lanes of a road may be excluded from superelevation using the @level attribute.
- Road elevation may be defined in combination with superelevation.

Related topics

- Excluding lanes from road elevation
- Road elevation
- Shape definition

8.4.3 Shape definition

Some lateral road shapes are too complex to be described by superelevation alone. Shapes describe the elevation of a road's cross section at a given point on the reference line in a more detailed way. That means, there may be several shape definitions at one s-coordinate that have different t-values, thereby describing the curvy shape of the road.

If shapes are used without superelevation, the actual width of a lane might be changed due to its curvilinear shape. The projected width with respect to the planview is not affected.

If shapes are combined with superelevation as shown in Figure 42, the projected width with respect to the superelevated state does not change, but the projected width with respect to the planview is affected. Between shape profiles, at specific s-coordinates, the road shape is interpolated linearly. Combining shapes with non-linear lane offsets should be avoided.

The defined t-range must at least cover the maximum t-expansion of the entire <road> element, as shown in the figure 41.

In figure 41 is shown how to calculate the height information between two lateral profiles. In figure 41. The lateral profile at s_{R1} has five polynomial definition, while the lateral profile at

s_{R2} has three polynomial definitions. To calculate a point between two lateral profiles, interpolate linear between those two profiles, use the formulas shown in figure 41.

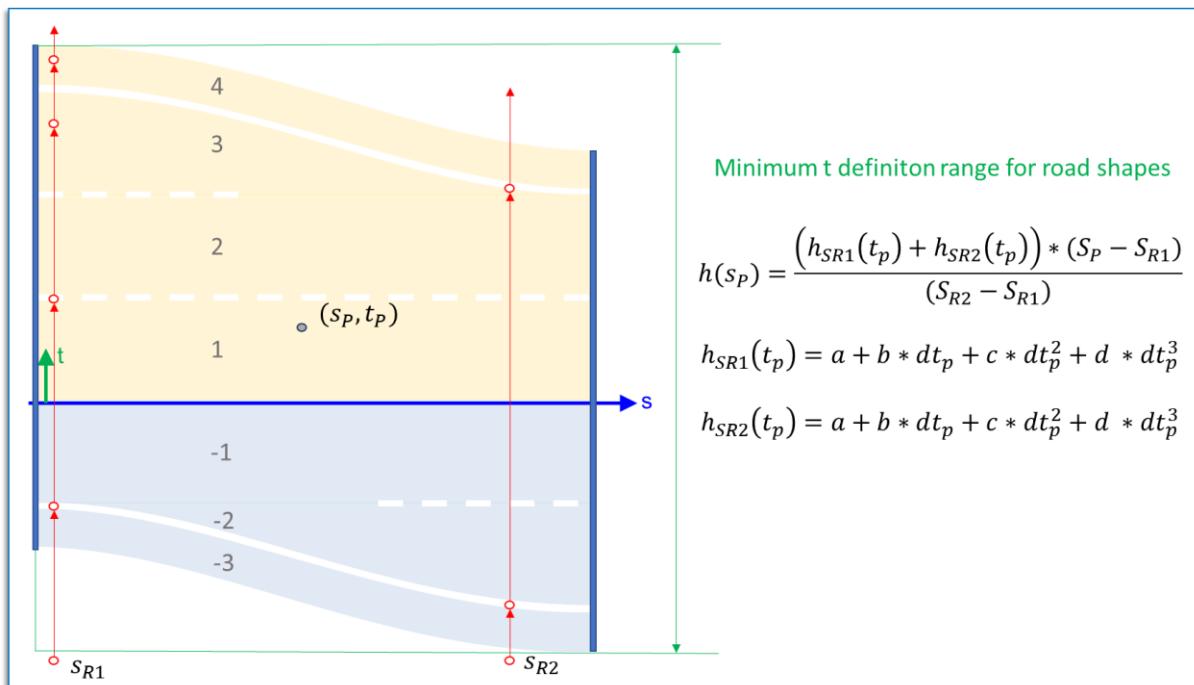


Figure 41: Minimum t-definition range for road shapes

Typical use cases are curved road surfaces on high-speed test tracks and crossfalls. The default value for shape is zero.

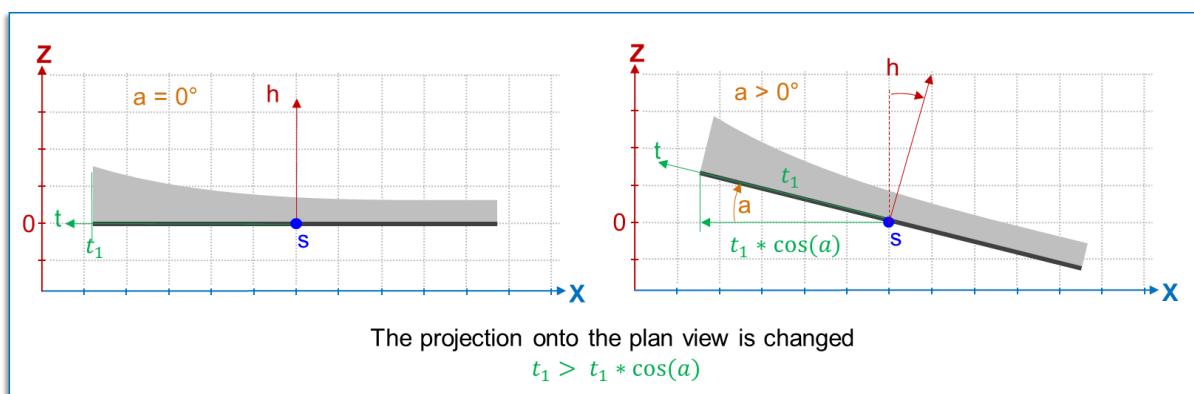


Figure 42: Shape definition (left) in combination with superelevation (right)

In OpenDRIVE, road shape is represented by the `<shape>` element within the `<lateralProfile>` element.

Attributes

`t_road_lateralProfile_shape`

Defined as the road section's surface relative to the reference plane. There may be several shape definitions at one s-position that have different t-values, thereby describing the curvy shape of the road.

attributes

name	type	unit	value	description
s	t_grEqZero	m	[0;∞[s-coordinate of start position
t	double	m]∞;∞[t-coordinate of start position
a	double	m]∞;∞[Polynom parameter a, relative height at @t (dt=0)
b	double	1]∞;∞[Polynom parameter b
c	double	1/m]∞;∞[Polynom parameter c
d	double	1/m ²]∞;∞[Polynom parameter d

Table 17: Attributes of the shape element

Calculation

The shape of the lateral profile is calculated with the following polynomial function:

$$h_{\text{Shape}}(ds) = a + b*ds + c*ds^2 + d*ds^3$$

where

- h_{Shape} is the height above the reference plane at a given position
 a, b, c, d are the coefficients
 ds is the distance perpendicular to the reference line between the start of a shape element and the given position

ds restarts at zero for each element. The absolute position of a shape value is calculated as follows:

$$t = t_{\text{start}} + ds$$

where

- t is the absolute position in the reference line coordinate system
 t_{start} is the start position of the element in the reference line coordinate system

Rules

The following rules apply to shapes:

- Shapes may be defined in combination with superelevation and road elevation.
- There should be no lane offset when using shapes.

Related topics

- Road elevation
- Superelevation
- Properties for road sections and cross section

8.5 Road surface

The description of the surface of a road is part of OpenCRG, not OpenDRIVE. It is possible to reference data created by OpenCRG in OpenDRIVE. Neither OpenDRIVE nor OpenCRG contain data regarding the visual representation of the road surface. With OpenCRG it is possible to model detailed road surface attributes, for example cobble stone or pot holes, as shown in figure 43.

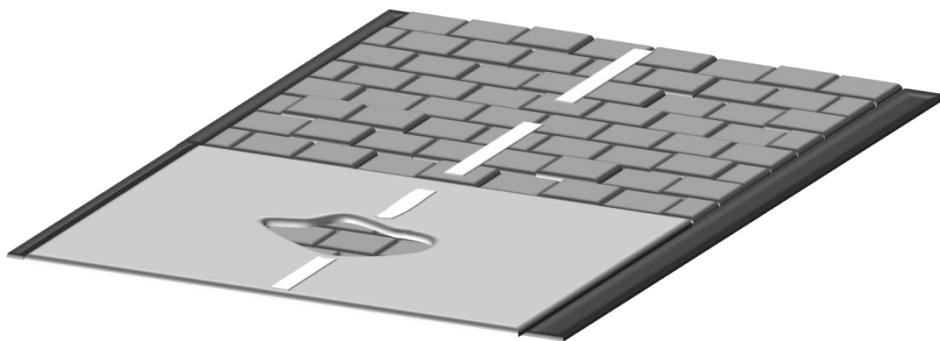


Figure 43: Road surface as defined in a CRG file

As the name indicates, CRG data is organized in a regular grid which is laid out along a reference line (comparable to an OpenDRIVE road's reference line). At each grid position, it contains the absolute elevation measured along a real road and some additional data which allows for the computation of the delta elevation relative to the reference line. The key to combining OpenDRIVE and CRG data is to define a correlation between the two reference lines and a rule for using the elevation data of both descriptions. CRG data may be offset from the OpenDRIVE road's reference line (see $@tOffset$) and it may be oriented in the same or opposite direction as the layout direction of the road (see orientation).

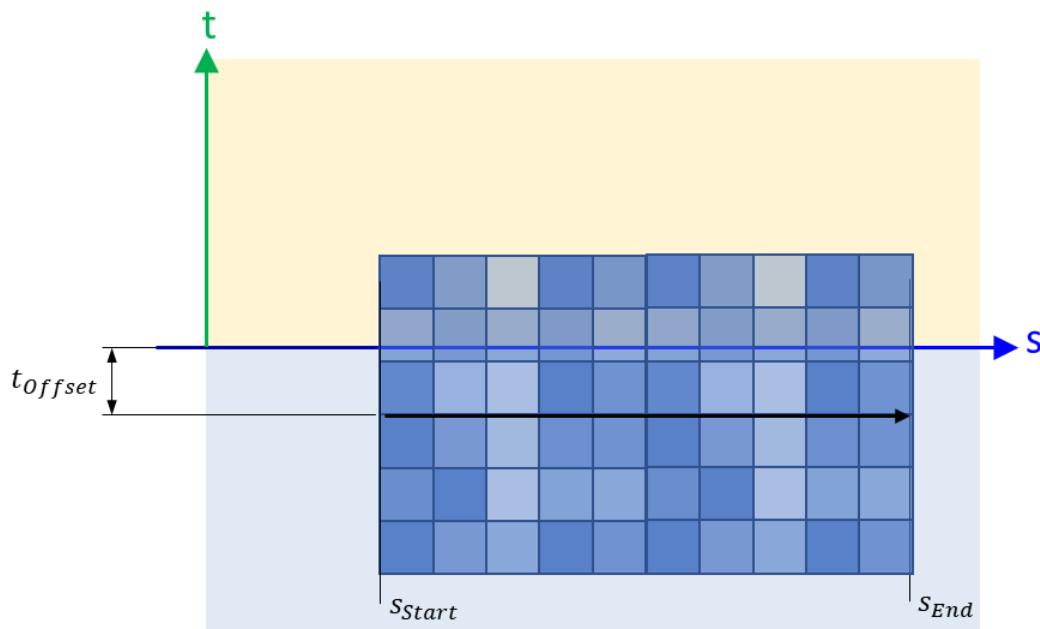


Figure 44: positioning of an OpenCRG file along the reference line

The CRG data may be applied to a given OpenDRIVE road in different modes:

@mode = attached:

The reference line of the CRG data set is replaced with the OpenDRIVE road's reference line, taking into account the @tOffset and the @sOffset parameters. The CRG local elevation values (calculated by evaluating the CRG grid and applying @zOffset and @zScale) will be added to the surface elevation data of the OpenDRIVE road (as derived from the combination of elevation, superelevation and crossfall). With this mode, the surface information relative to the original CRG data's reference line is transferred from an arbitrary CRG road to an OpenDRIVE road without having to make sure that the overall geometries of the road match. The original position, heading, curvature, elevation and superelevation of the CRG road are disregarded. The CRG grid is evaluated along the OpenDRIVE reference line instead of the CRG reference line.

$$\begin{pmatrix} u \\ v \end{pmatrix}_{CRG} = \begin{pmatrix} s - s_{Offset} \\ t - t_{Offset} \end{pmatrix}_{OpenDRIVE}$$

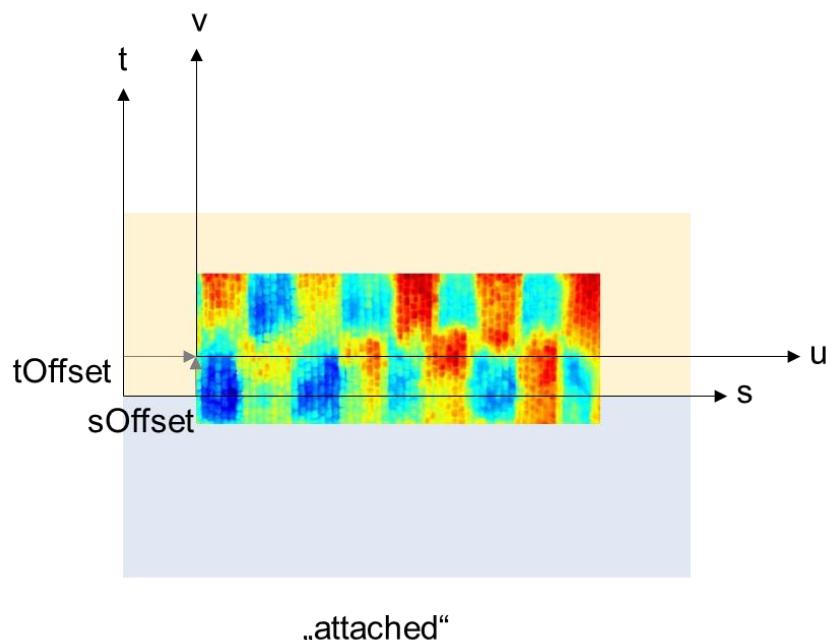


Figure 45: OpenCRG attachment mode, attached

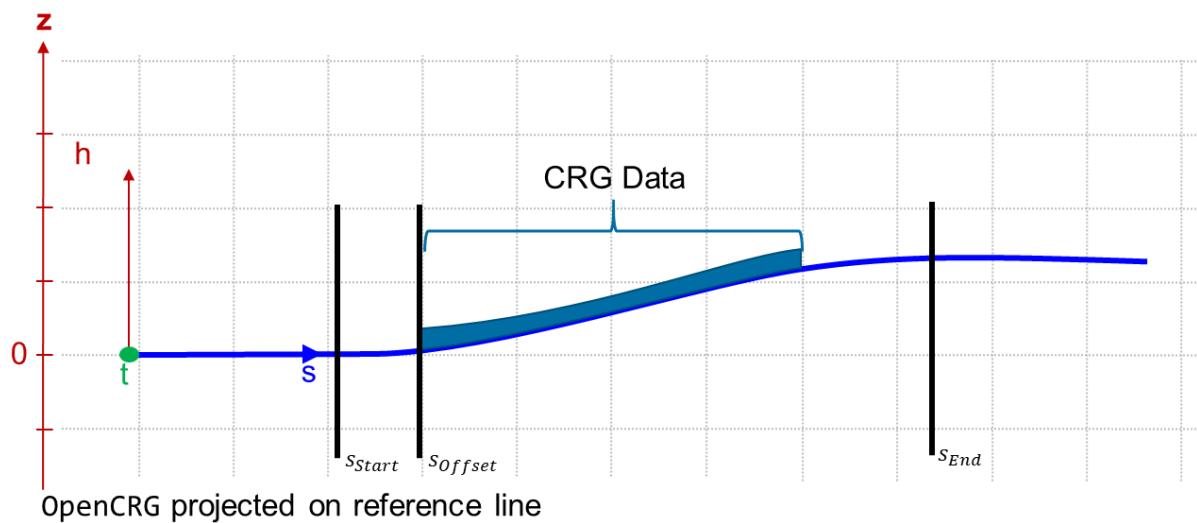


Figure 46: OpenCRG attached mode with elevation

@mode = attached0:

This mode is basically the same as the attached mode, with the only exception that only the CRG data's elevation value is considered (i.e. the OpenDRIVE elevation is set to zero). To avoid problems set @sStart and @sEnd exactly to the CRG data boundaries. Otherwise gaps with height zero can occur, as shown in the below figure.

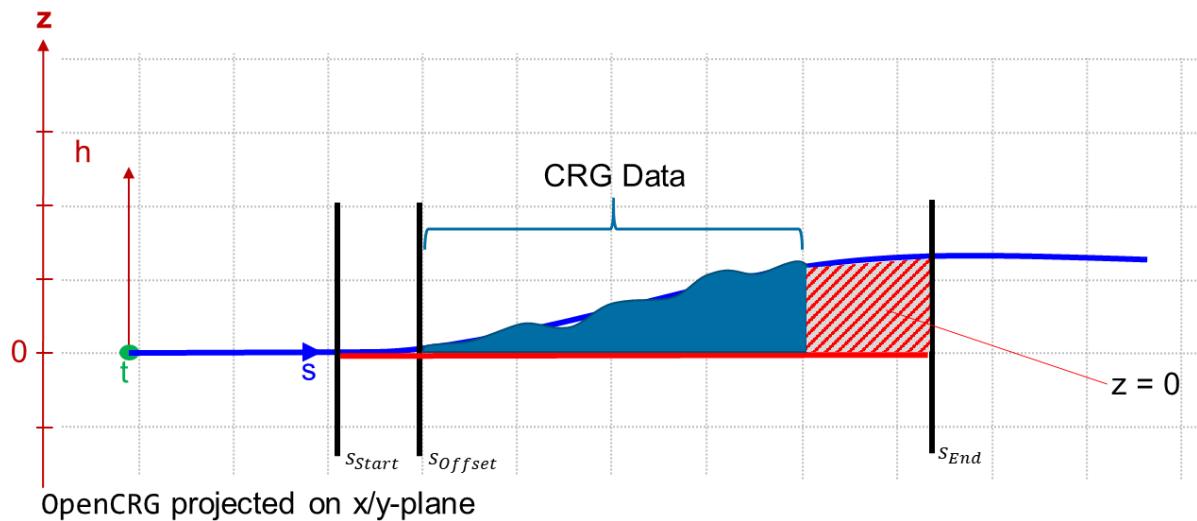


Figure 47: OpenCRG attached0 mode with elevated reference line

@mode = genuine:

The start point of the CRG data set's reference line is positioned relative to the point on the OpenDRIVE road's reference line at the position defined by @sStart, @sOffset and @tOffset. By providing offset values for the longitudinal (@sOffset) and lateral (@tOffset) displacement, the heading (@hOffset) and the elevation (@zOffset), the correlation between

the two description's reference lines is clear. In genuine mode, the CRG data will completely replace the OpenDRIVE elevation data, i.e. the absolute elevation of a given point of the road surface is directly computed from the CRG data (think of it as combining CRG and OpenDRIVE data with the OpenDRIVE elevation, superelevation and crossfall all being zero). When using this method, it must of course be made sure that the geometry of the CRG data matches – within certain tolerance – the geometry of the underlying OpenDRIVE road.

$$\begin{pmatrix} u \\ v \end{pmatrix}_{CRG} = \begin{pmatrix} \cos(p_{offset}) & \sin(p_{offset}) \\ -\sin(p_{offset}) & \cos(p_{offset}) \end{pmatrix} \begin{pmatrix} s - s_{offset} \\ t - t_{offset} \end{pmatrix}_{OpenDRIVE}$$

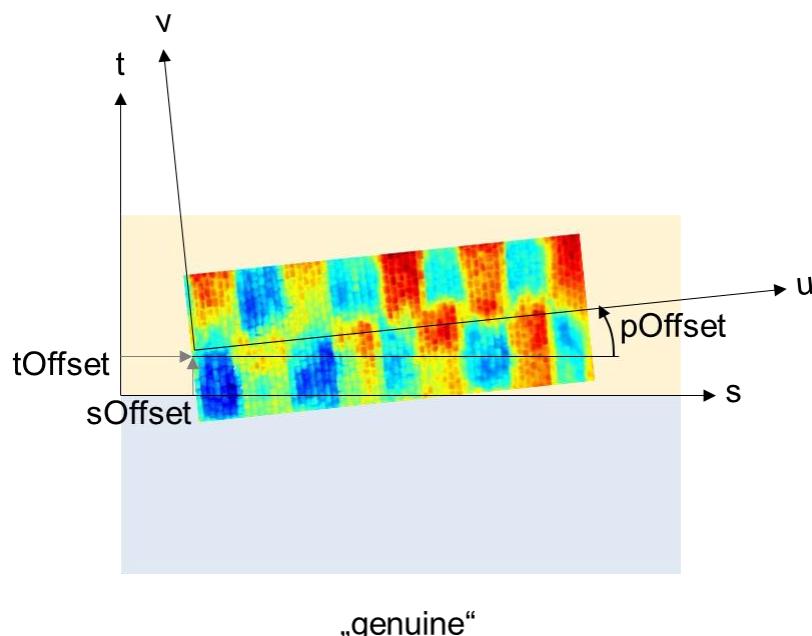


Figure 48: OpenCRG attachment mode, genuine

@mode = global:

The CRG data set is just referenced from a given track or a junction record but neither translatory nor rotatory transformation is applied. All data in the CRG file remains in its native co-ordinate system. Elevation data is interpreted as inertial data, i.e. AS IS.

Since CRG data may only cover parts of a road's surface, it must be made sure that outside the valid CRG area, the elevation information derived from OpenDRIVE data can still be used.

In OpenDRIVE, the road surface is represented by the `<surface>` element within the `<road>` element. Data described in OpenCRG is represented by the `<CRG>` element within the `<surface>` element.

NOTE: if more than one CRG entry is given for the same physical property (attribute purpose) at a given location, then the last entry in the sequence of occurrence in the OpenDRIVE file shall be the relevant one. All others will be ignored.

@orientation = same, opposite

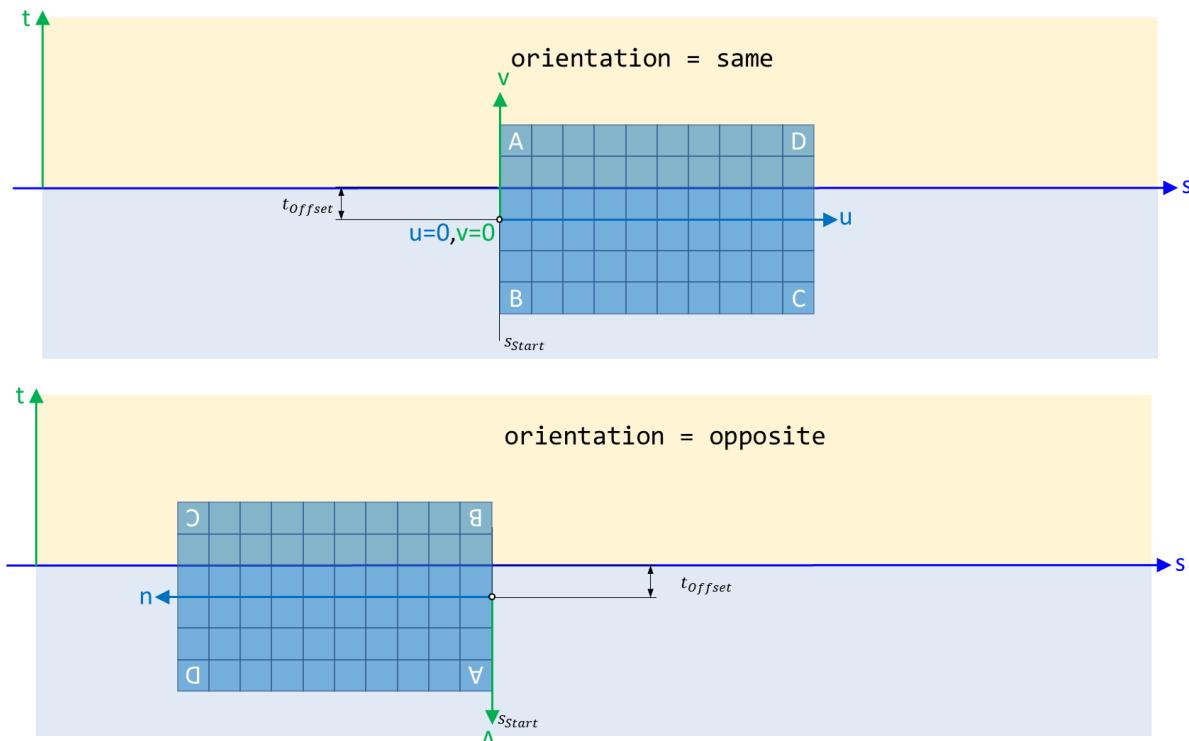


Figure 49: OpenCRG orientation

$$\begin{pmatrix} u \\ v \end{pmatrix}_{CRG} = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} s - s_{offset} \\ t - t_{offset} \end{pmatrix}_{OpenDRIVE}$$

The orientation attribute rotates the CRG file at the origin of the u/v coordinate system of the OpenCRG. The value “same” has a rotation angle of 0° and the value “opposite” has a value of 180°. The t-offset is not affected by the orientation attribute.

Attributes

t_road_surface_CRG

attributes					
name	type	unit	value	description	
file	string			Name of the file containing the CRG data	
sStart	t_grEq Zero	m	[0;∞[Start of the application of CRG data (s-coordinate)	
sEnd	t_grEq Zero	m	[0;∞[End of the application of CRG (s-coordinate)	
orientation	e_direction		same ; opposite	Orientation of the CRG data set relative to the parent <road> element	

mode	e_road_surface_CRG_mode		attached ; attached0 ; genuine ; global	Attachment mode for the surface data, see specification.
purpose	e_road_surface_CRG_purpose		elevation ; friction	Physical purpose of the data contained in the CRG file; if the attribute is missing, data will be interpreted as elevation data.
sOffset	double	m]-∞;∞[s-offset between CRG center line and reference line of the road (default = 0.0)
tOffset	double	m]-∞;∞[t-offset between CRG center line and reference line of the road (default = 0.0)
zOffset	double	m]-∞;∞[z-offset between CRG center line and reference line of the road (default = 0.0)
zScale	double]-∞;∞[z-scale factor for the surface description (default = 1.0)
hOffset	double	rad]-∞;∞[Heading offset between CRG center line and reference line of the road (required for mode genuine only, default = 0.0)

Table 18: Attributes of the road surface CRG element

XML example

```
<surface>
  <CRG      file="fancyData.crg"      sStart="0.0"      sEnd="100.0"
    orientation="same" mode="attached" tOffset="0.0">
  </CRG>
</surface>
```

Related topics

- Road

8.6 Use cases for roads

The following sections contain some sample use cases for modelling roads in OpenDRIVE.

8.6.1 Modelling a road shape with a linear crossfall

Many roads have a crossfall, for example, to provide a drainage gradient so that water runs off the surface into the gutter.

Figure 50 shows a sample definition of a two-lane road with a crossfall.

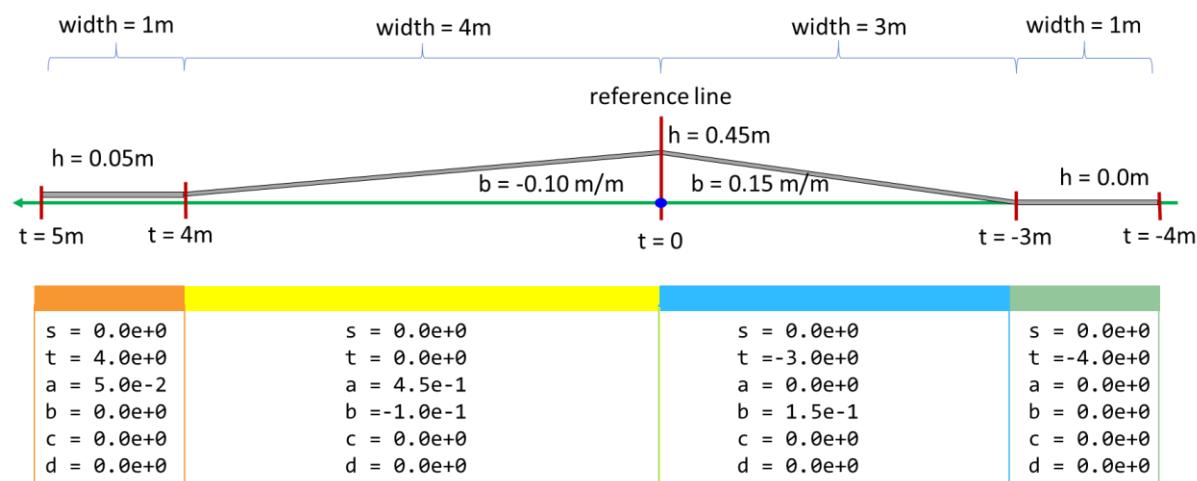


Figure 50: Example crossfall modelled with road shape

The linear crossfall has the following properties:

- The width of the road start at $t=-4$. Because the values are 0, nothing changes in the height up to $t=-3$.
- From $t=-3$ to $t=0$, there is a linear rise from 0.15 meter per meter. That means, at $t=0$ (middle of the road), the road has reached a height of 0.45m.
- From $t=0.45m$, the road has a linear fall by 0.1 meter per meter. That means, when reaching $t=4$, the road has a height of 0.05m ($0.45m - 0.40m$ is 0.05m; road loses 0.1 m per meter over a distance of 4m. When starting at 0.45m, the endpoint is at 0.05m).

XML structures used

To describe the crossfall, `<shape>` elements within `<lateralProfile>` are used.
See use case file UC-RoadShape.xodr

Using OpenDRIVE structures

To model road shapes, shape elements shall start at the right side of the road, that is in positive t-direction. That means, the elements start with negative t-values.

XML example

```
<lateralProfile>
  <shape
    s="0.000000000000000e+00"
    t="-4.00000000000000e+00"
    a="0.000000000000000e+00"
    b="0.000000000000000e+00"
    c="0.000000000000000e+00"
    d="0.000000000000000e+00"/>
  <shape
    s="0.000000000000000e+00"
    t="-3.00000000000000e+00"
    a="0.000000000000000e+00"
    b="1.49999999999999e-01"
    c="0.000000000000000e+00"
    d="0.000000000000000e+00"/>
  <shape
    s="0.000000000000000e+00"
    t="0.000000000000000e+00"
    a="4.50000000000001e-01"
    b="-1.00000000000001e-01"
    c="0.000000000000000e+00"
    d="0.000000000000000e+00"/>
  <shape
    s="0.000000000000000e+00"
    t="4.00000000000000e+00"
    a="5.00000000000003e-02"
    b="0.00000000000000e+00"
    c="0.00000000000000e+00"
    d="0.00000000000000e+00"/>
</lateralProfile>
```

9 Lanes

In OpenDRIVE, all roads contain *lanes*. Each road shall have at least one lane with a width larger than 0. The number of lanes per road is not limited.

For the definition and description of lanes in OpenDRIVE, the *center lane* is required. The center lane has no width and serves as reference for lane numbering. The center lane itself has the lane number 0. The numbering of the other lanes starts at the center lane: Lane numbers descend to the right, meaning negative t-direction, and ascend to the left, meaning positive t-direction.

Figure 51 illustrates the center lane for a road with multiple traffic lanes and different driving directions. In this case, the center lane separates the driving directions, depending on left- and right-hand traffic, specified in Road type. Because no lane offset is used, the center lane is identical to the road reference line.

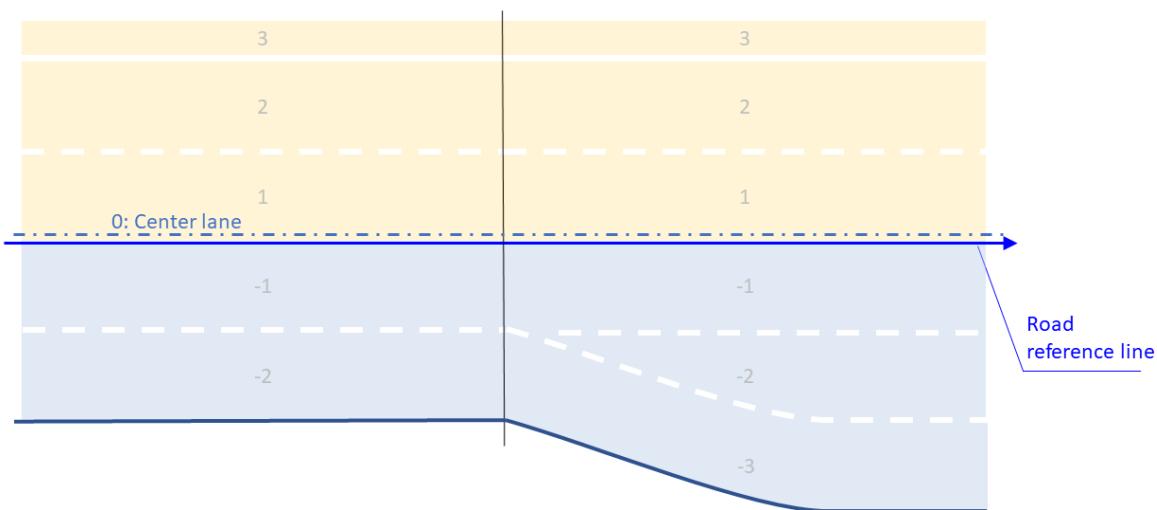


Figure 51: Center lane for road with lanes of different driving directions

Figure 52 illustrates the center lane for a road with lanes that have the same driving direction, meaning a one-way road.

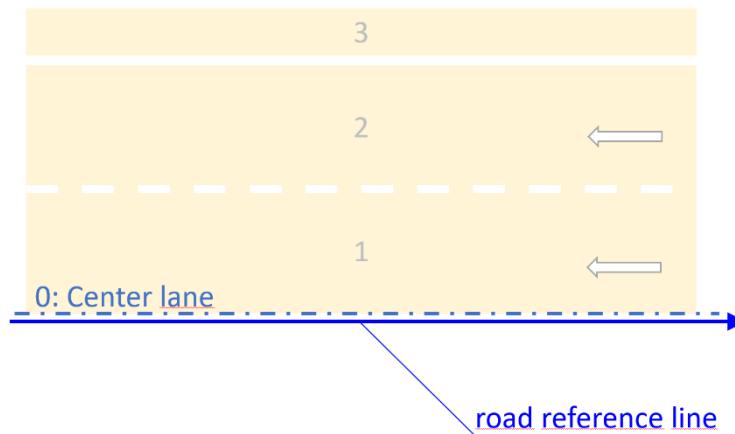


Figure 52: Center lane for road with lanes of identical driving direction

In OpenDRIVE, lanes are represented by the `<lanes>` element within a `<road>` element.

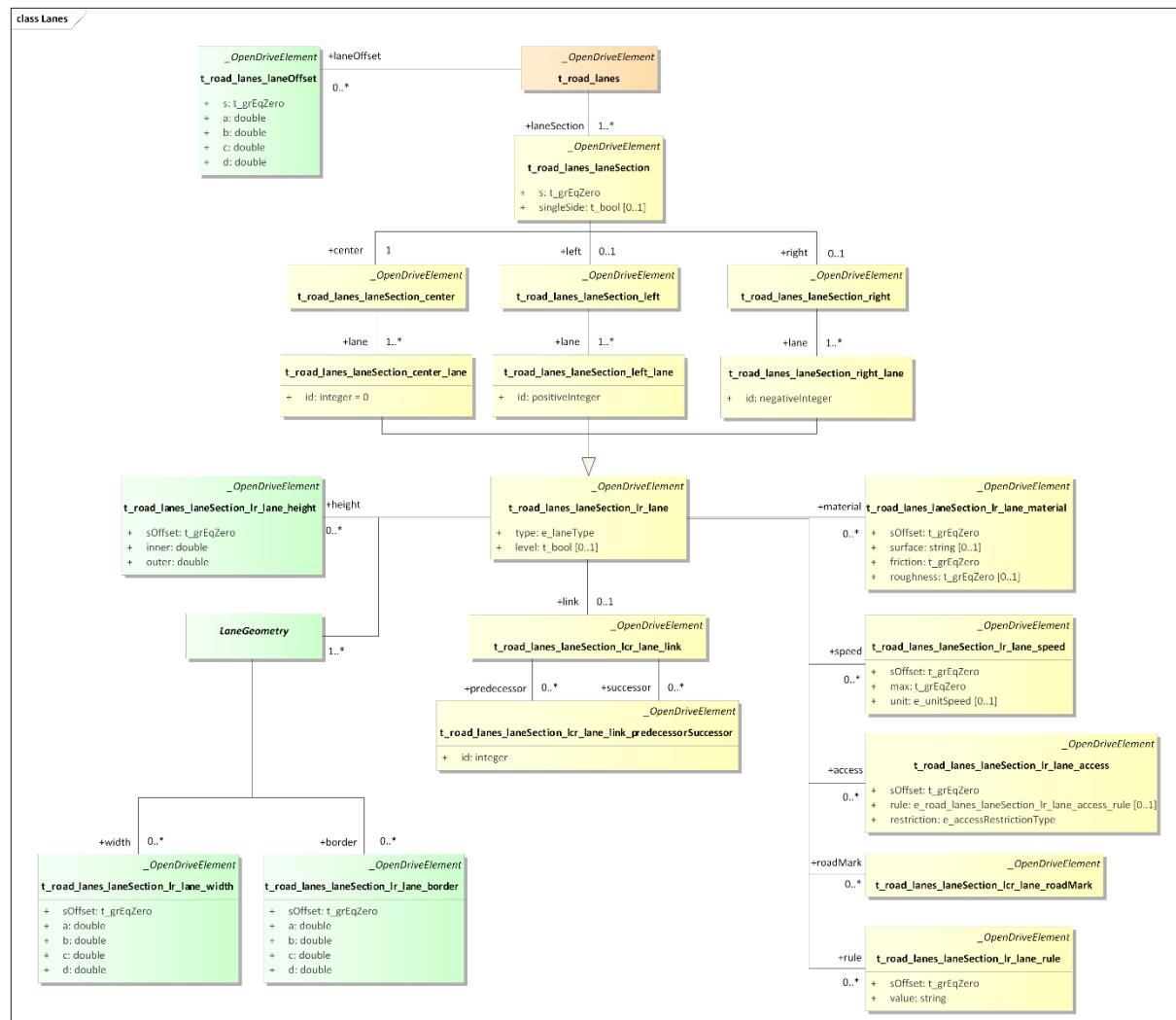


Figure 53: UML model for lanes

Rules

The following rules apply for the use of lanes:

- Each road shall have a center lane and one lane with a width larger than 0.
- Roads may have as many lanes as needed.
- The center lane shall have no width, meaning that the `<width>` element shall not be used for the center lane.
- The center lane shall have the lane number 0.
- Lane numbering shall start with 1 next to the center lane, descend in negative t-direction and ascend in positive t-direction.
- Lane numbering shall be consecutive without any gaps.
- Lane numbering shall be unique per lane section.
- There may be bidirectional lanes. This is specified using the `@type` attribute of the `<lane>` element.

XML example

```
<lanes>
    <laneSection s="0.000000000000000e+00">
        <left>
            <lane id="2" type="border" level="false">
                <link>
                </link>
                <width sOffset="0.0" a="1.0" b="0.0" c="0.0" d="0.0"/>
            </lane>
            <lane id="1" type="driving" level="false">
                <link>
                </link>
                <width sOffset="0.0" a="4.0" b="0.0" c="0.0" d="0.0"/>
            </lane>
        </left>
        <center>
            <lane id="0" type="none" level="false">
                <link>
                </link>
            </lane>
        </center>
        <right>
            <lane id="-1" type="driving" level="false">
                <link>
                </link>
                <width sOffset="0.0" a="4.0" b="0.0" c="0.0" d="0.0"/>
            </lane>
            <lane id="-2" type="border" level="false">
                <link>
                </link>
                <width sOffset="0.0" a="1.0" b="0.0" c="0.0" d="0.0"/>
            </lane>
        </right>
    </laneSection>
</lanes>
```

Related topics

- Lane grouping within lane sections

9.1 Lane grouping within lane sections

For easier navigation through an OpenDRIVE road description, the lanes within a lane section are grouped into left, center, and right lanes. Within these *groups*, the lanes are described by `<lane>` elements.

Because lane numbers descend in negative t-direction and ascend in positive t-direction, applications can derive the direction of a lane from the lane number given in the ID attribute, unless `@type` is bi-directional.

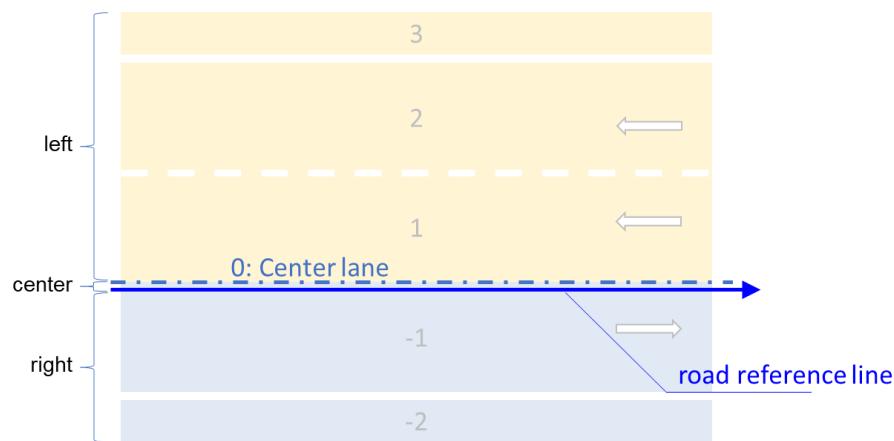


Figure 54: Lane grouping with left, center, right

In OpenDRIVE, *lane groups* are represented by `<center>`, `<right>` and `<left>` elements within the `<laneSection>` element. The ID attribute is defined within the `<lane>` element that is nested under a `<center>`, `<right>`, or `<left>` element.

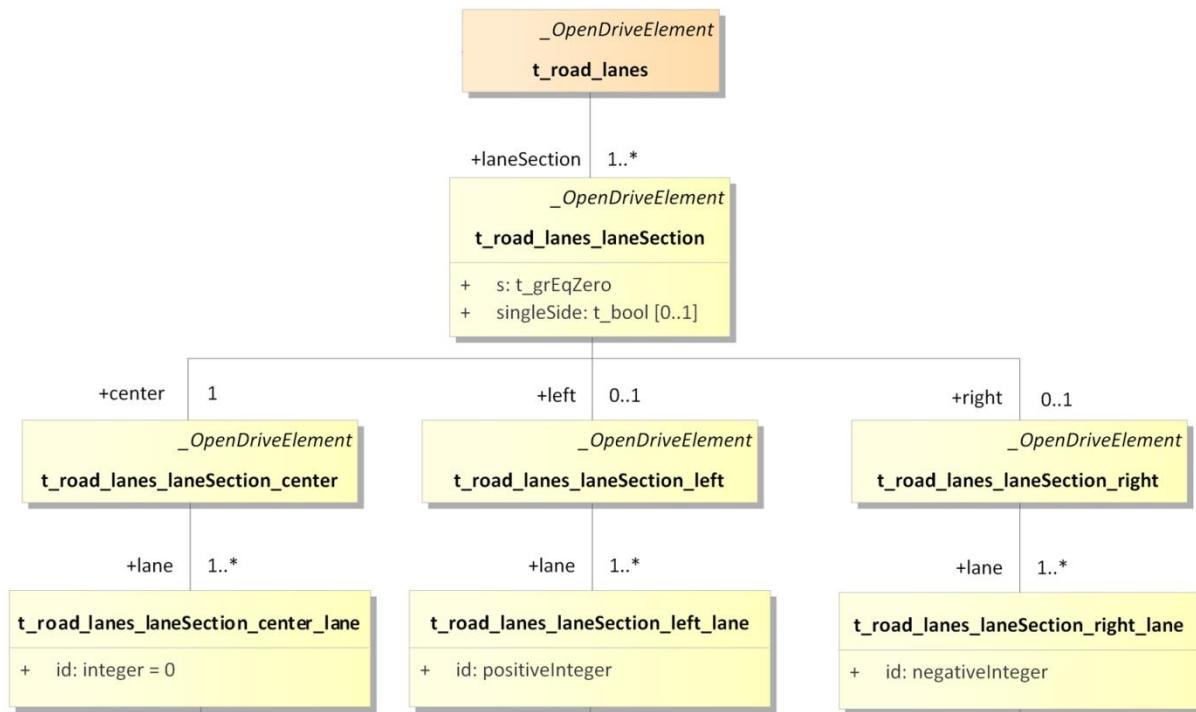


Figure 55: UML model for `t_road_lanes`

XML Example

See XML example from Lanes

Rules

The following rules apply for lane grouping:

- Lanes with positive ID run on the left side of the center lane, while lanes with negative ID run on the right side of the center lane.
- Each lane section shall contain at least one `<right>` or `<left>` element.
- One `<center>` element shall be defined for each s-coordinate.
- Each lane section may contain one `<center>` element.
- For better orientation, lanes should be listed from left to right, that is with descending ID.

Related topics

- Lane sections
- Lanes

9.2 Lane sections

Lanes may be split into multiple lane sections. Each lane section contains a fixed number of lanes. Every time the number of lanes changes, a new lane section is required, as shown in figure 56. Lane sections are defined in ascending order along the road reference line.

Figure 56 shows a road section that is divided into different lane sections. A new lane section is required when the number of lanes changes.

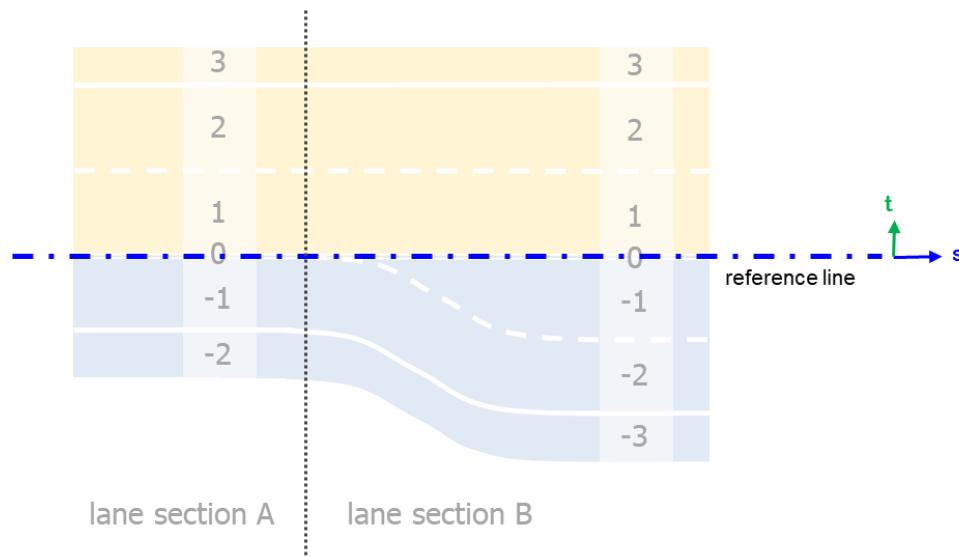


Figure 56: A road section with lane sections

To simplify the use of lane sections for complex roads, they may be defined for one side of the road only using the `@singleSide` attribute. This principle is shown in figure 57.

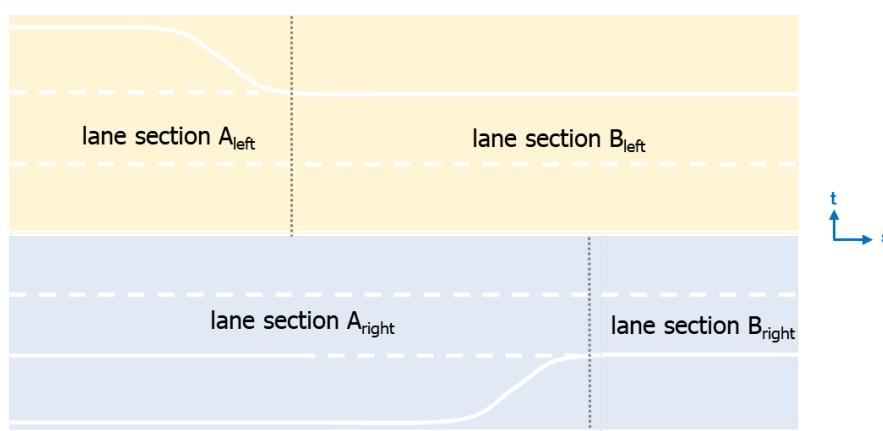


Figure 57: Lane sections defined separately for both sides of the road

In OpenDRIVE, lane sections are represented by `<laneSection>` elements within a `<lanes>` element.

Attributes

t_road_lanes_laneSection

Lanes may be split into multiple lane sections. Each *lane section* contains a fixed number of lanes. Every time the number of lanes changes, a new lane section is required.

attributes	name	type	unit	value	description
	s	t_grEqZero	m	[0;∞[s-coordinate of start position
	singleSide	t_bool		true ; false	Lane section element is valid for one side only (left, center, or right), depending on the child elements.

Table 19: Attributes of the laneSection element

t_road_lanes_laneSection_center

For easier navigation through an OpenDRIVE road description, the lanes within a lane section are grouped into left, center, and right lanes. Each lane section shall contain one <center> element and at least one <right> or <left> element.

t_road_lanes_laneSection_center_lane

Lane elements are included in left/center/right elements. Lane elements should represent the lanes from left to right, that is, with descending ID.

attributes	name	type	unit	value	description
	id	integer]∞;∞[ID of the lane

Table 20: Attributes of the laneSection center lane element

t_road_lanes_laneSection_left

For easier navigation through an OpenDRIVE road description, the lanes within a lane section are grouped into left, center, and right lanes. Each lane section shall contain one <center> element and at least one <right> or <left> element.

t_road_lanes_laneSection_left_lane

Lane elements are included in left/center/right elements. Lane elements should represent the lanes from left to right, that is, with descending ID.

attributes

name	type	unit	value	description
id	positive integer]-∞;∞[ID of the lane

Table 21: Attributes of the laneSection left lane element

t_road_lanes_laneSection_right

For easier navigation through an OpenDRIVE road description, the lanes within a lane section are grouped into left, center, and right lanes. Each lane section shall contain one <center> element and at least one <right> or <left> element.

t_road_lanes_laneSection_right_lane

Lane elements are included in left/center/right elements. Lane elements should represent the lanes from left to right, that is, with descending ID.

attributes

name	type	unit	value	description
id	negative integer]-∞;∞[ID of the lane

Table 22: Attributes of the laneSection right lane element

Rules

The following rules apply for lane sections:

- Each road shall have at least one lane section.
- Lane sections shall be defined in ascending order.
- There shall always be exactly one center lane at each s-position.
- Using lanes with a width of 0 for long distances should be avoided.
- A new lane section shall be defined each time the number of lanes change.
- A lane section shall remain valid until a new lane section is defined.
- The properties of lanes inside a lane section may be changed as often as needed.
- Lane sections may be defined for one side of the road only using the @singleSide attribute.

Related topics

- Lanes
- Lane grouping within lane sections
- Road reference line

9.3 Lane offset

A lane offset may be used to shift the center lane away from the road reference line. This makes it easier to model local lateral shifts of lanes on roads, for example for left turn lanes.

A combination of lane offset and shape definition can lead to inconsistencies depending on the interpolation used for lane offset. Because linear interpolation is used for the road shape along the reference line, linear interpolation should also be used for the offset definition to enable consistent combined use of both definitions.

Figure 58 shows the offset of the center lane away from the road reference line.

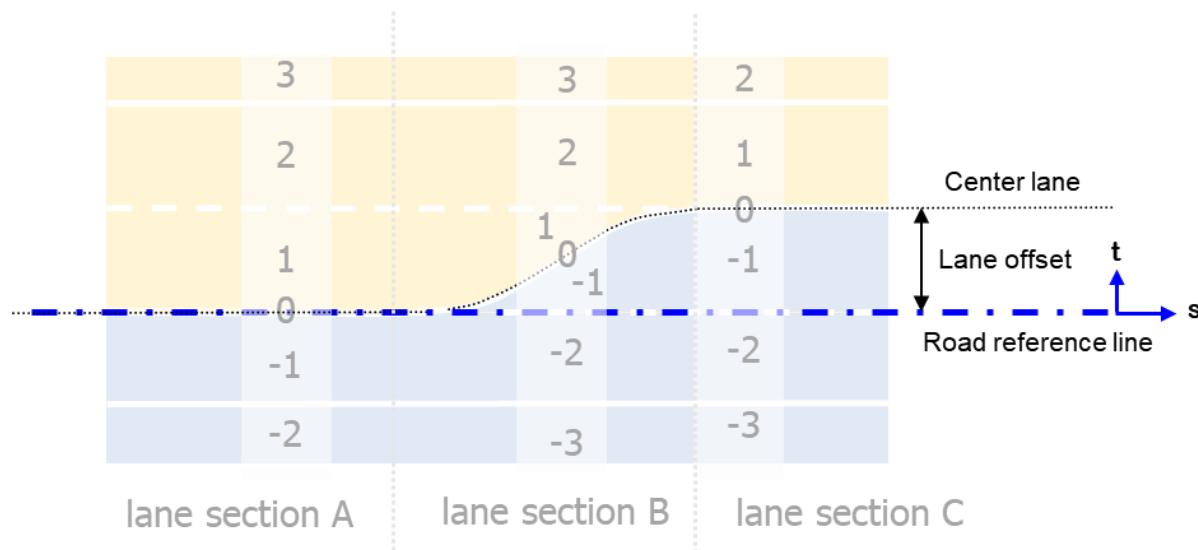


Figure 58: Lane offset

In OpenDRIVE, a lane offset is represented by a `<laneOffset>` element within a `<lanes>` element.

Attributes

`t_road_lanes_laneOffset`

A *lane offset* may be used to shift the center lane away from the road reference line.

attributes				
name	type	unit	value	description
<code>s</code>	<code>t_grEqZero</code>	m	$[0; \infty[$	s-coordinate of start position
<code>a</code>	double	m	$]-\infty; \infty[$	Polynom parameter a, offset at $@s$ ($ds=0$)
<code>b</code>	double	1	$]-\infty; \infty[$	Polynom parameter b
<code>c</code>	double	$1/m$	$]-\infty; \infty[$	Polynom parameter c
<code>d</code>	double	$1/m^2$	$]-\infty; \infty[$	Polynom parameter d

Table 23: Attributes of the `laneOffset` element



Calculation

The offset at a given point is calculated with the following polynomial function of the third order:

$$\text{offset } (\text{ds}) = a + b*\text{ds} + c*\text{ds}^2 + d*\text{ds}^3$$

where

<code>offset</code>	is the lateral offset at a given position
<code>a, b, c, d</code>	are the coefficients
<code>ds</code>	is the distance along the road reference line between the start of a lane offset element and the given position
<code>new</code>	

`ds` restarts at zero for each element. The absolute position of an offset value is calculated as follows:

$$s = s_{\text{start}} + ds$$

where

<code>s</code>	is the absolute position in the reference line coordinate system
<code>s_{start}</code>	is the start position of the offset element in the reference line system
coordinate	

A new lane offset element is required each time the polynomial function changes.

XML example

```
<lanes>
  <laneOffset s="25.0" a="0.0" b="0.0" c="3.9e-03" d="-5.2e-05"/>
  <laneOffset s="75.0" a="3.25" b="0.0" c="0.0" d="0.0"/>
  ...
</lanes>
```

This example can be found in Ex_Simple-LaneOffset.xodr.

Rules

The following rules apply for lane offsets:

- Lane offsets shall not be used together with road shapes.
- A new lane offset shall start when the underlying polynomial function changes.
- There shall be no offset if border definitions are present.

Related topics

- Lane borders
- Shape definition
- Lanes

9.4 Lane linkage

To enable lane navigation, linkage information for lanes may be stored in OpenDRIVE. Linkage is described by means of predecessor and successor information for each lane. Both lanes and junctions may serve as predecessor or successor of lanes. Lanes may be linked to lanes on the same or another road.

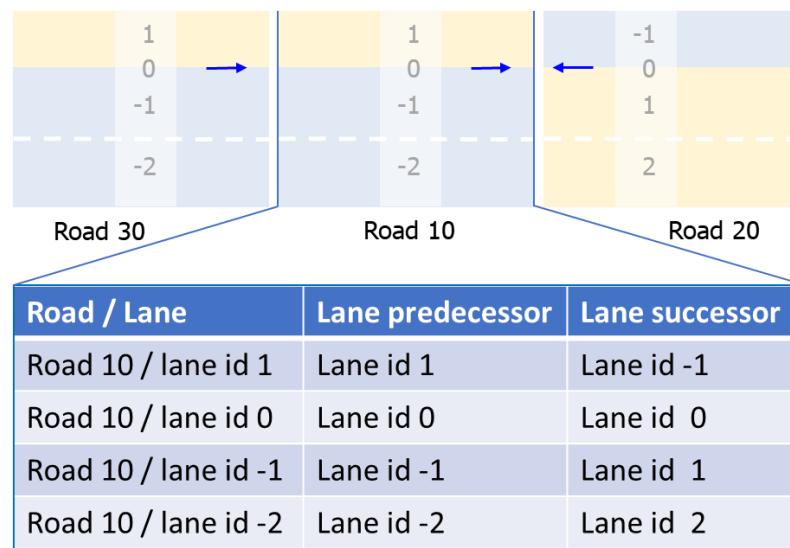


Figure 59: Lane links for road with id 10

In OpenDRIVE, lane linkage is represented by the `<link>` element within the `<lane>` element. The `<predecessor>` and `<successor>` elements are defined within the `<link>` element.

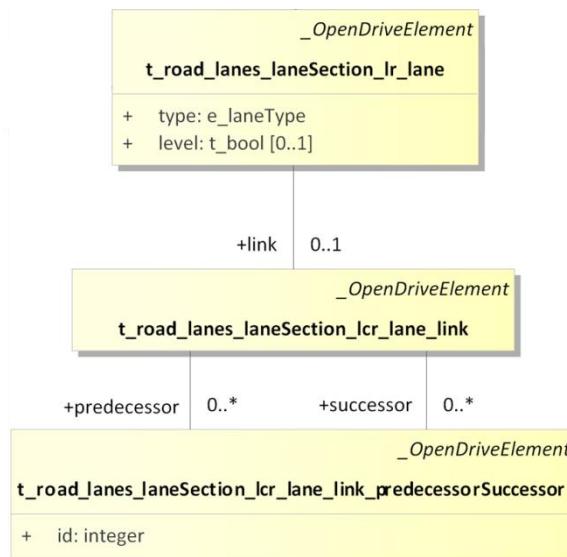


Figure 60: UML Model t_road_lanes_laneSection_lcr_lane_link

Attributes

`t_road_lanes_laneSection_lcr_lane_link`

For links between lanes with an identical reference line, the lane predecessor and successor information provide the IDs of lanes on the preceding or following lane section. This element may only be omitted, if lanes end at a junction or have no physical link.

t_road_lanes_laneSection_lcr_lane_link_predecessorSuccessor

attributes

name	type	unit	value	description
id	integer]-∞;∞[ID of the preceding / succeeding linked lane

Table 24: Attributes of the predecessor successor elements

Rules

The following rules apply to lane linkage:

- A lane may have another lane as successor or predecessor.
- Two lanes shall only be linked if their linkage is clear. If the relationship to successor or predecessor is ambiguous, junctions shall be used.
- The <link> element shall be omitted if the lane ends in a junction or has no link.

Related topics

- Road linkage
- Junctions
- Lanes

9.5 Lane properties

Lane properties describe the use and shape of lanes. Lane properties are defined per lane section but may change within that section. If a property is not specifically defined for a lane section, applications can apply default properties.

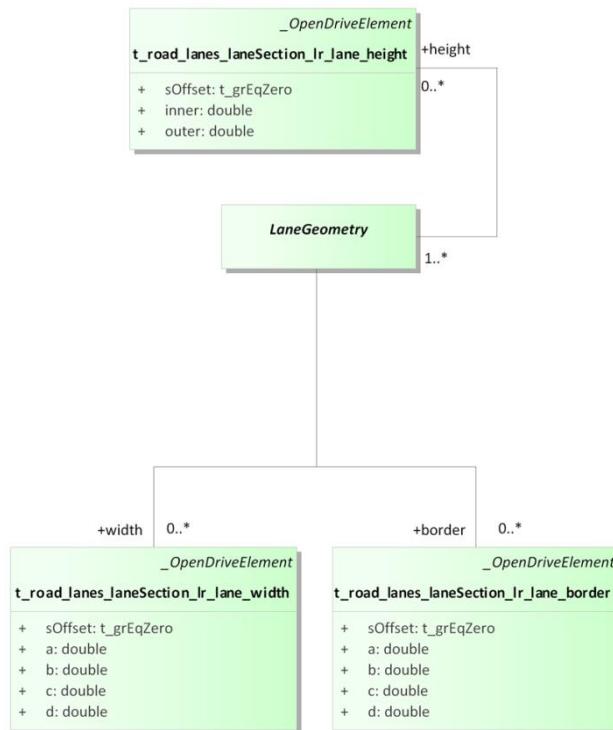


Figure 61: UML model lane properties

Examples of lane properties are lane width, lane border, and speed limits.

Rules

The following rules apply for lane properties:

- Lane properties shall be defined relative to the start of the corresponding lane section.
- A specific lane property shall remain valid until another lane property of that type is defined or the lane section ends.
- Lane properties of identical types shall be defined in ascending order.

9.5.1 Lane width

The width of a lane is defined along the t-coordinate. The width of a lane may change within a lane section.

Lane width and lane border elements are mutually exclusive within the same lane group. If both width and lane border elements are present for a lane section in the OpenDRIVE file, the application must use the information from the <width> elements.

In OpenDRIVE, lane width is described by the <width> element within the <lane> element.

Attributes

t_road_lanes_laneSection_lr_lane_width

The width of a lane shall be defined at least once per lane section. The center lane shall have no width, meaning that the `<width>` element shall not be used for the center lane. The width of a lane shall remain valid until a new width element is defined or the lane section ends.

attributes

name	type	unit	value	description
sOffset	t_grEqZero	m	[0; ∞ [s-coordinate of start position of the <code><width></code> element, relative to the position of the preceding <code><laneSection></code> element
a	double	m] $-\infty$; ∞ [Polynom parameter a, width at @s (ds=0)
b	double	1] $-\infty$; ∞ [Polynom parameter b
c	double	1/m] $-\infty$; ∞ [Polynom parameter c
d	double	1/m ²] $-\infty$; ∞ [Polynom parameter d

Table 25: Attributes of the lane width element

Calculation

The width at a given point is calculated with the following polynomial function of the third order:

$$\text{Width (ds)} = a + b*ds + c*ds^2 + d*ds^3$$

where

- `width` is the width at a given position
`a, b, c, d` are the coefficients
`ds` is the distance along the road reference line between the start of a lane width element and the given position
`new`

`ds` restarts at zero for each element. The absolute position of a width value is calculated as follows

$$s = s_{\text{section}} + \text{offset}_{\text{start}} + ds$$

where

- `s` is the absolute position inside the reference line coordinate system
`ssection` is the start position of the preceding lane section element in the track coordinate system
`offsetstart` is the offset of the element relative to the preceding lane section

Figure 62: Change of lane width per lane section shows the change in lane width in positive s-direction, starting at different offset positions.

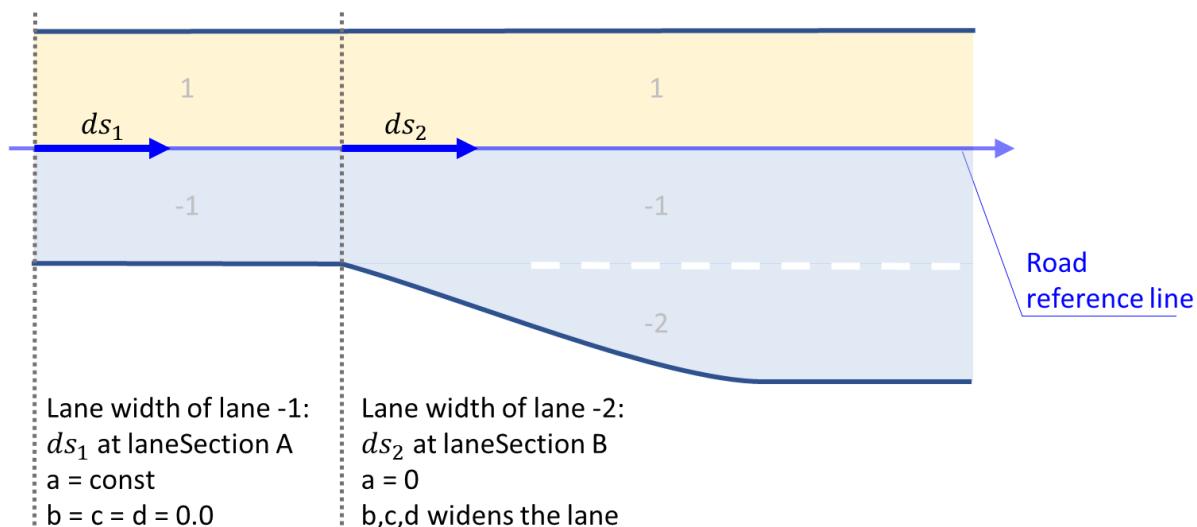


Figure 62: Change of lane width per lane section

XML example

See the sample file Ex_Lane-Width.xodr

Rules

The following rules apply to lane width:

- The width of a lane shall be defined at least once per lane section.
- The width of the lane shall be defined for the full length of the lane section. This means that there must be a `<width>` element for $s=0$.
- The center lane shall have no width, meaning that the `<width>` element shall not be used for the center lane.
- The width of a lane shall remain valid until a new width element is defined or the lane section ends.
- A new width element shall be defined when the variables of the polynomial function change.
- Several width elements per lane section shall be defined in ascending order.
- Width elements shall not be used together with border elements in the same lane group.

Related topics

- Lane sections
- Lane grouping within lane sections
- Superelevation
- Shape definition

9.5.2 Lane borders

Lane borders are another method to describe the width of lanes. Instead of defining the width directly, lane borders describe the outer limits of a lane, independent of the parameters of their inner borders. In this case, inner lanes are defined as lanes which have the same sign for their ID as the lane currently defined, but with a smaller absolute value for their ID.

Especially when road data is derived from automatic measurements, this type of definition is easier than specifying the lane width because it avoids creating many lane sections.

Lane width and lane border elements are mutually exclusive within the same lane group. If both width and lane border elements are present for a lane section in the OpenDRIVE file, the application shall use the information from the <width> elements.

In OpenDRIVE, lane borders are represented by the <border> element within the <lane> element.

Attributes

t_road_lanes_laneSection_lr_lane_border

attributes

name	type	unit	value	description
sOffset	t_grEqZero	m	[0; ∞ [s-coordinate of start position of the <border> element , relative to the position of the preceding <laneSection> element
a	double	m]- ∞ ; ∞ [Polynom parameter a, border position at @s (ds=0)
b	double	1]- ∞ ; ∞ [Polynom parameter b
c	double	1/m]- ∞ ; ∞ [Polynom parameter c
d	double	1/m ²]- ∞ ; ∞ [Polynom parameter d

Table 26: Attributes of the lane border element

Calculation

The border position at a given point is calculated with the following polynomial function of the third order:

$$t_{\text{border}}(\text{ds}) = a + b * \text{ds} + c * \text{ds}^2 + d * \text{ds}^3$$

where

t_{border}

a, b, c, d

ds

is the t-position of the border at a given ds-position

are the coefficients

is the distance along the road reference line between the start of the element and the given position

ds restarts at zero for each element. The absolute position of a border offset value is calculated by

$$s = s_{\text{section}} + \text{offset}_{\text{start}} + ds$$

where

s	is the absolute position in the reference line coordinate system
s_{section}	is the start position of the preceding lane section element in the track coordinate system
$\text{offset}_{\text{start}}$	is the offset of the element relative to the preceding lane section element

Figure 63 illustrates this convention for a lane with varying border shape over a given range:

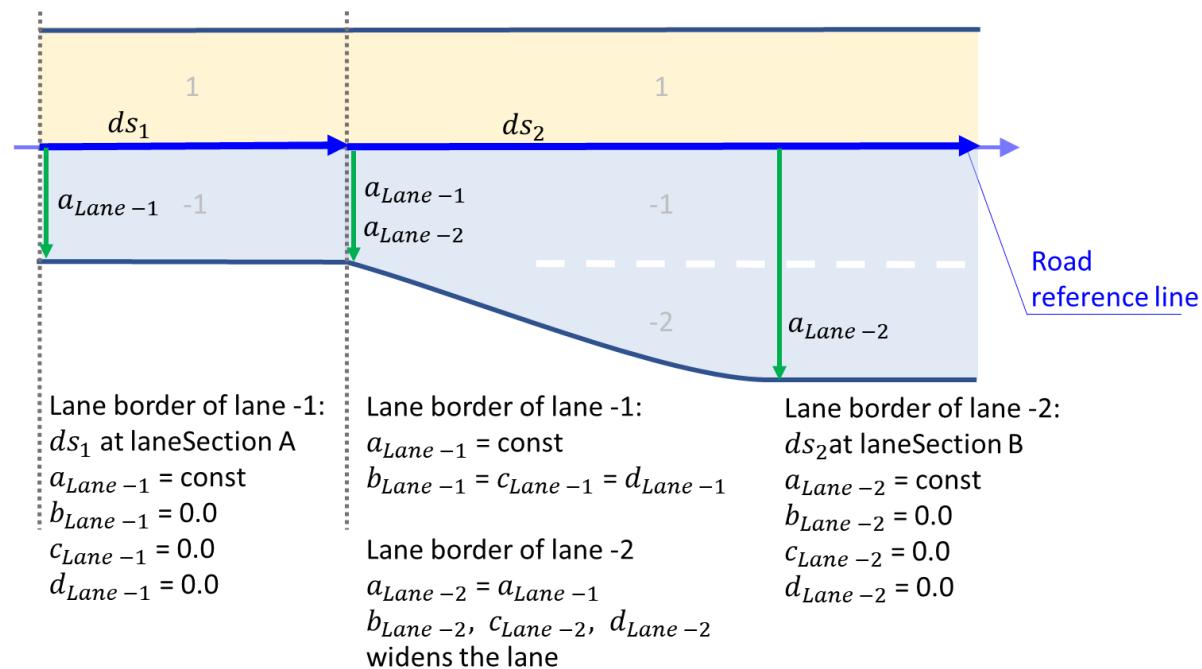


Figure 63: Lane with varying border shape

XML Example

See the sample file Ex_Lane-Border.xodr

Rules

The following rules apply to lane borders:

- Width elements shall not be used together with border elements in the same lane group.
- Border elements shall not exist together with lane offset.
- A new border element shall be defined when the variables of the polynomial function change.

Related topics

- Lane sections
- Lane grouping within lane sections

9.5.3 Lane type

The lane type is defined per lane. A lane type defines the main purpose of a lane and its corresponding traffic rules.

The available lane types are:

- **shoulder**: Describes a soft border at the edge of the road.
- **border**: Describes a hard border at the edge of the road. It has the same height as the drivable lane.
- **driving**: Describes a “normal” drivable road that is not one of the other types.
- **stop**: Hard shoulder on motorways for emergency stops
- **none**: Describes the space on the outermost edge of the road and does not have actual content. Its only purpose is for applications to register that OpenDRIVE is still present in case the (human) driver leaves the road.
- **restricted**: Describes a lane on which cars should not drive. The lane has the same height as drivable lanes. Typically, the lane is separated with lines and often contains dotted lines as well.
- **parking**: Describes a lane with parking spaces.
- **median**: Describes a lane that sits between driving lanes that lead in opposite directions. It is typically used to separate traffic in towns on large roads.
- **biking**: Describes a lane that is reserved for cyclists.
- **sidewalk**: Describes a lane on which pedestrians can walk.
- **curb**: Describes curb stones. Curb stones have a different height than the adjacent drivable lanes.
- **exit**: Describes a lane that is used for sections that are parallel to the main road. It is mainly used for deceleration lanes.
- **entry**: Describes a lane type that is used for sections that are parallel to the main road. It is mainly used for acceleration lanes.
- **onramp**: A ramp leading to a motorway from rural or urban roads.
- **offRamp**: A ramp leading away from a motorway and onto rural urban roads.
- **connectingRamp**: A ramp that connects two motorways, for example, motorway junctions.

For the use of the lane types, see figure 64 to figure 68.

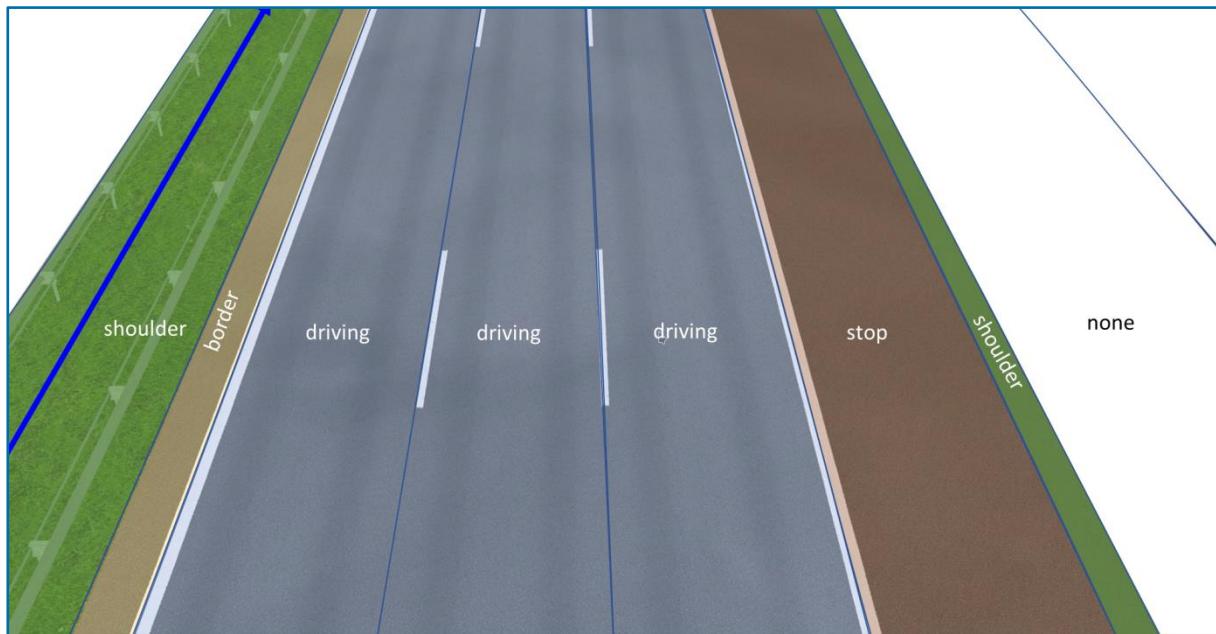


Figure 64: Lane types for a motorway

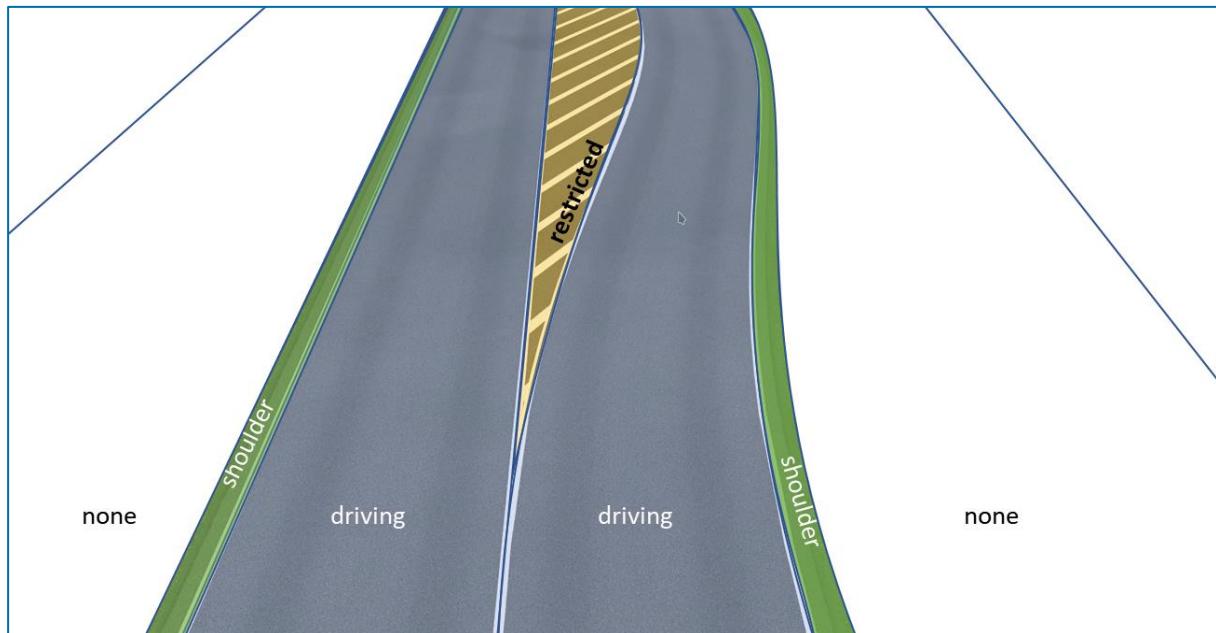


Figure 65: Lane types for a rural road

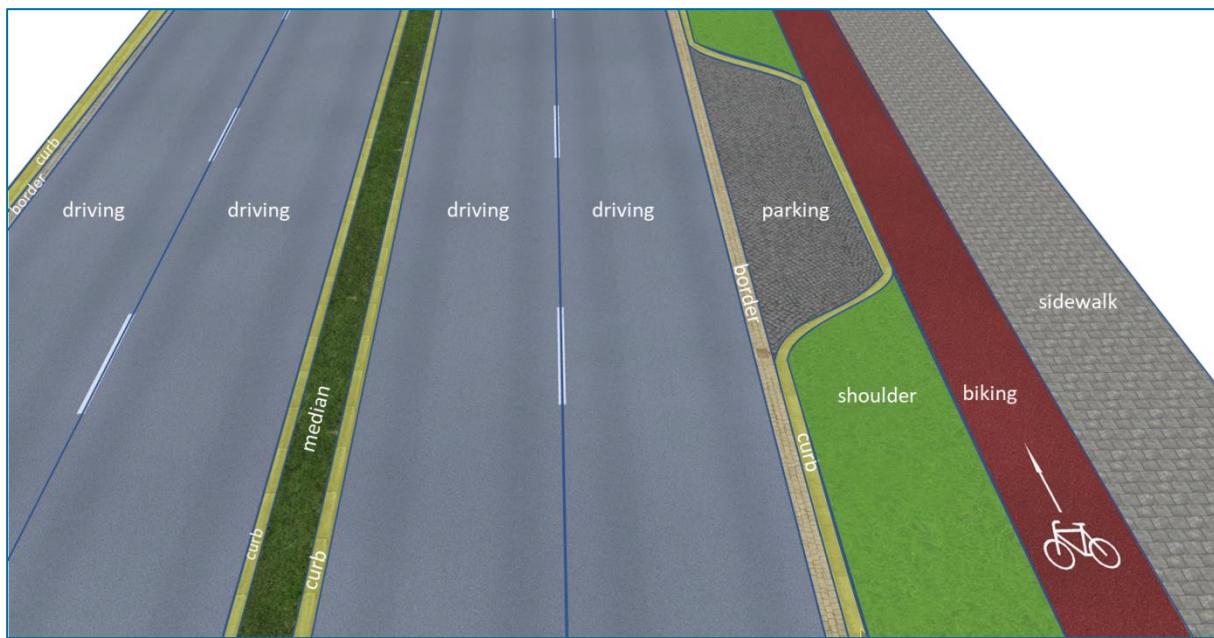


Figure 66: Lane types for an urban road



Figure 67: Lane types for motorway exit and entry



Figure 68: Lane types for motorway connecting to another motorway

In OpenDRIVE, lane types are represented by the attribute @type element within the <lane> element.

Attributes

t_road_lanes_laneSection_lr_lane

Lane elements are included in left/center/right elements. Lane elements should represent the lanes from left to right, that is, with descending ID.

attributes					
	name	type	unit	value	description
	type	e_laneType			Type of the lane. For values see UML model.
	level	t_bool		true ; false	"true" = keep lane on level, that is, do not apply superelevation; "false" = apply superelevation to this lane (default, also used if attribute level is missing)

Table 27: Attributes of the lane type

Rules

The following rules apply to lane types:

- The lane type may be changed as often as needed by using a new lane section.

Related topics

- Lanes

9.5.4 Lane material

OpenDRIVE provides an element to store information on the material of lanes (apart from OpenCRG), meaning their surface, friction properties, and roughness. If no material is defined, applications can apply default values.

In OpenDRIVE, lane material is represented by the `<material>` element within the `<lane>` element.

Attributes

`t_road_lanes_laneSection_lr_lane_material`

Stores information about the material of lanes. Each element is valid until a new element is defined. If multiple elements are defined, they must be listed in ascending order.

attributes	name	type	unit	value	description
	sOffset	t_grEqZero	m	[0;∞[s-coordinate of start position, relative to the position of the preceding <code><laneSection></code> element
	surface	string			Surface material code, depending on application
	friction	t_grEqZero		[0;∞[Friction coefficient
	roughness	t_grEqZero		[0;∞[Roughness, for example, for sound and motion systems

Table 28: Attributes of the lane material element

Rules

The following rules apply to lane material:

- The center lane shall have no material elements.
- The material elements of a lane shall remain valid until another material element starts or the lane section ends.
- If there are multiple material elements per lane section, the elements shall be defined in ascending order regarding s-position.

Related topics

- Lanes

- Lane sections
- Lane grouping within lane sections

9.5.5 Lane speed limit

The maximum speed allowed on a lane may be defined. Lane speed limits overrides road speed limits.

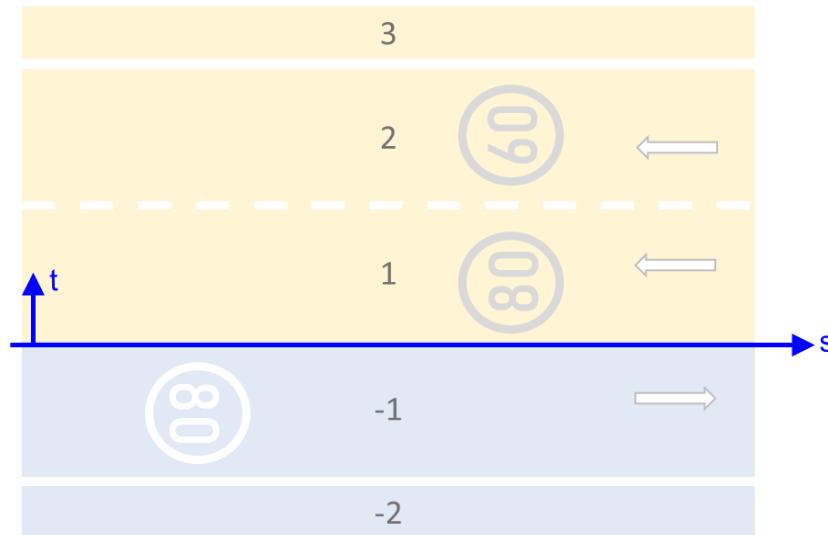


Figure 69: Lane-specific speed limits

In OpenDRIVE, lane speed is represented by the `<speed>` element within the `<lane>` element.

Attributes

`t_road_lanes_laneSection_lr_lane_speed`

Defines the maximum allowed speed on a given lane. Each element is valid in direction of the increasing s-coordinate until a new element is defined.

attributes					
	name	type	unit	value	description
	sOffset	t_grEqZero	m	[0; ∞ [s-coordinate of start position, relative to the position of the preceding <code><laneSection></code> element
	max	t_grEqZero	m/s	[0; ∞ [Maximum allowed speed
	unit	e_unitSpeed			Unit of the attribute max. For values, see UML Model

Table 29: Attributes of the speed element



XML example

```
<lane id="-1" type="driving" level="false">
  <link>
    <successor id="-3"/>
  </link>
  <width sOffset="0.0" a="2.0" b="0.0" c="0.0" d="0.0"/>
  <speed sOffset="0.0" max="80.0" unit="km/h"/>
  <height sOffset="0.0" inner="0.12" outer="0.12"/>
</lane>
```

Rules

The following rules apply to lane speed limits:

- The center lane shall have no speed limit.
- The speed limit of a lane shall remain valid until another speed limit is defined or the lane section ends.
- If there are multiple lane speed limit elements per lane section, the elements shall be defined in ascending order.
- Speed limits derived from signals shall always have preference.

Related topics

- Lanes
- Signals
- Speed limits for road types

9.5.6 Lane access

Lanes can be restricted to specific road users, such as trucks or busses. Such restrictions may be defined in OpenDRIVE in addition to restrictions described by signals.

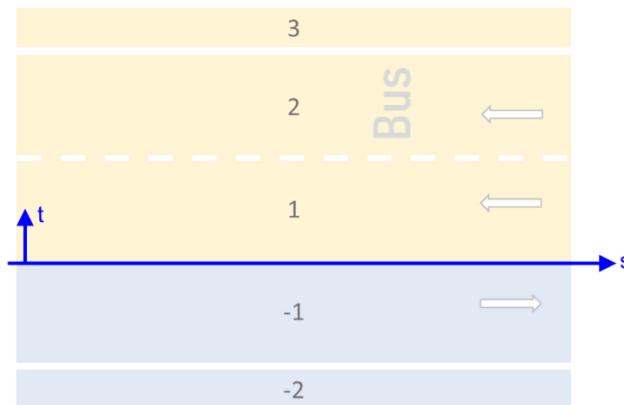


Figure 70: Lane access, bus lane

OpenDRIVE provides the `<access>` element within the `<lane>` element to describe lane access rules.

Attributes

`t_road_lanes_laneSection_lr_lane_access`

Defines access restrictions for certain types of road users.

Each element is valid in direction of the increasing `s` co-ordinate until a new element is defined. If multiple elements are defined, they must be listed in ascending order.

attributes					
	name	type	unit	value	description
	<code>sOffset</code>	<code>t_grEqZero</code>	m	<code>[0;∞]</code>	s-coordinate of start position, relative to the position of the preceding <code><laneSection></code> element
	<code>rule</code>	<code>e_road_lanes_laneSection_lr_lane_access_rule</code>			specifies whether the participant given in the attribute <code>@restriction</code> is allowed or denied access to the given lane
	<code>restriction</code>	<code>e_accessRestrictionType</code>			Identifier of the participant to whom the restriction applies. For values, see UML Model

Table 30: Attributes of the access element

XML example:

```
<lane id="2" type="driving" level="false">
  <link>
    <successor id="2"/>
  </link>
  <width sOffset="0.0" a="2.0" b="0.0" c="0.0" d="0.0"/>
  <access sOffset="0.0" rule="allow" restriction="bus"/>
</lane>
```

Rules

The following rules apply to lane access rules:

- The center lane shall have no access rules.
- The access rules of a lane shall remain valid until another access rule is defined or the lane section ends.
- If there are multiple access rule elements per lane section, the elements shall be defined in ascending order.
- Lane access elements may start at identical offset positions.
- If no `<access>` element is present within a lane element, then there are no restrictions.
- If `deny` value is present in the `<rule>` element, all other vehicles are still allowed.
- If `allow` value is present in the `<rule>` element, all other vehicles are banned.
- At a given s-position, either only `deny` or only `allow` values shall be given, not mixed.
- For a new s-position, all restrictions must be defined again, even if only a subset changes.
- The restriction `deny=none` is used to revert all previous restrictions.

Related topics

[Lanes](#)

- Lane sections

9.5.7 Lane height

Lane height shall be defined along the h-coordinate. Lane height may be used to elevate a lane independent from the road elevation. Lane height is used to implement small-scale elevation such as raising pedestrian walkways, as shown in Figure 71. Lane height is specified as offset from the road (including elevation, superelevation, shape) in z direction.

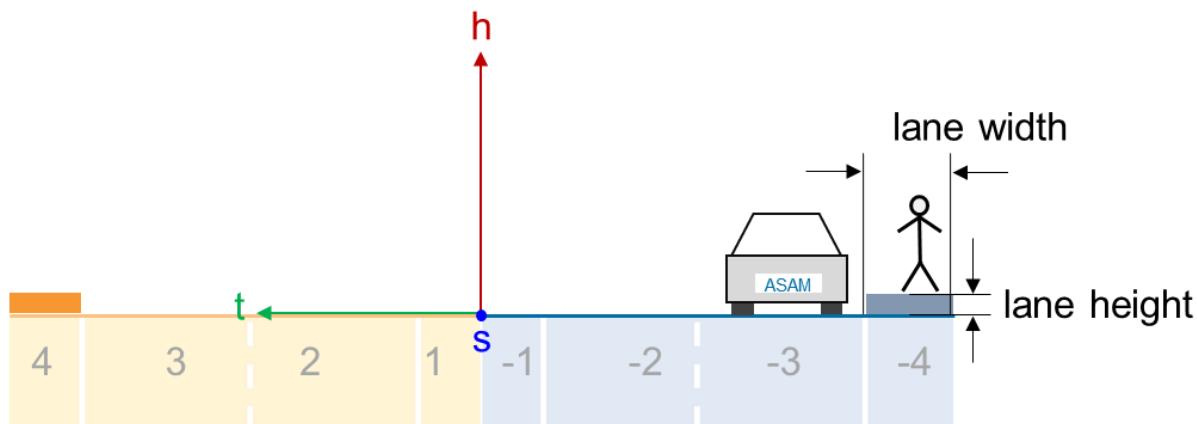


Figure 71: Lane height

In OpenDRIVE, lane height is represented by the `<height>` element within the `<lane>` element.

Attributes

`t_road_lanes_laneSection_lr_lane_height`

attributes

name	type	unit	value	description
<code>sOffset</code>	<code>t_grEqZero</code>	m	<code>[0;∞]</code>	s-coordinate of start position, relative to the position of the preceding <code><laneSection></code> element
<code>inner</code>	double	m	<code>]∞;∞[</code>	Inner offset from road level
<code>outer</code>	double	m	<code>]∞;∞[</code>	Outer offset from road level

Table 31: Attributes of the lane height element

XML example

```
<lane id="-2" type="sidewalk" level="false">
  <link>
    <successor id="-3"/>
  </link>
  <width sOffset="0.0" a="2.0" b="0.0" c="0.0" d="0.0"/>
  <height sOffset="0.0" inner="0.12" outer="0.12"/>
</lane>
```

Rules

The following rules apply to lane height:

- To modify the lane height, for example for curbstones, the element <height> shall be used.
- The center lane shall not be elevated by lane height.
- Lane height shall not be used to define road elevation or superelevation.
- Lane height shall be used for small scale elevation only.

Related topics

- Road elevation
- Lanes
- Lane grouping within lane sections

9.5.8 Excluding lanes from road superelevation

Single lanes may be excluded from superelevation to cover use cases like roads with curbstones, borders, or sidewalks without superelevation.

Figure 72 shows the use of the attribute @level, which excludes the outermost lanes of a road from superelevation.

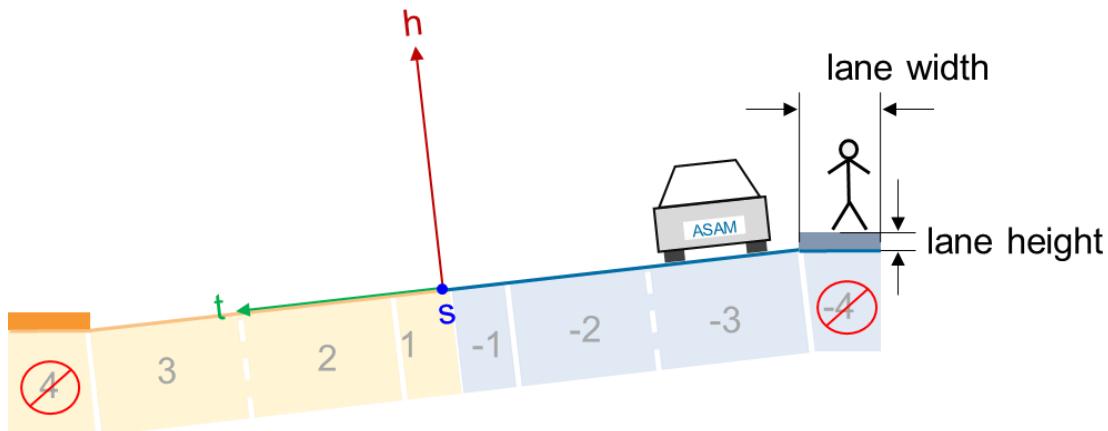


Figure 72: Lanes excluded from road elevation

OpenDRIVE provides the attribute @level for excluding lanes from road superelevation. When the attribute is set to TRUE for a lane, then this lane is excluded from both superelevation and road shape definition of the road. The elevation of the lane stays on the same height as the inner connecting lane.

There may be multiple outer lanes with level=TRUE, for example, for a bike lane followed by a sidewalk.

Rules

The following rules apply for excluding lanes from road elevation:

- If a lane has @level = TRUE, then at least on one side there shall be only lanes with @level = TRUE until the edge of the road is reached.
- There may be multiple outer lanes with @level = TRUE.

Related topics

- Superelevation
- Lane type

9.6 Road markings

Lanes on roads can have different lane markings, for example lines of different colors and styles. OpenDRIVE provides the `<roadMark>` element for road markings. The road mark information defines the style of the line at the lane's outer border. For left lanes, this is the left border, for right lanes the right one. The style of the center line that separates left and right lanes is determined by the road mark element for the center lane.

For each lane within a road cross section, multiple road mark elements may be defined. Several attributes may be used to describe the properties of the lane markings, for example `@type`, `@weight`, and `@width`.

There are two ways to specify the type of road marking:

- The `@type` attribute within the `<roadMark>` element allows to enter keywords that are stored in the application. They are used to describe simplified road marking types like solid, broken, or grass.
- The `<type>` element contains further `<line>` elements that allows to describe the road marking in a more detailed way.

In OpenDRIVE, road markings are represented by `<roadMark>` elements within `<lane>` elements.

Attributes

`t_road_lanes_laneSection_lcr_lane_roadMark_type`

Each type definition shall contain one or more line definitions with additional information about the lines that the road mark is composed of.

attributes	name	type	unit	value	description
	name	string			Name of the road mark type. May be chosen freely.
	width	t_grEq Zero	m	[0; ∞]	Accumulated width of the road mark. In case of several <code><line></code> elements this <code>@width</code> is the sum of all <code>@width</code> of <code><line></code> elements and spaces in between, necessary to form the road mark. This attribute supersedes the definition in the <code><roadMark></code> element.

Table 32: Attributes of the `roadMark` element

`t_road_lanes_laneSection_lcr_lane_roadMark_type_line`

A road mark may consist of one or more elements. Multiple elements are usually positioned side-by-side. A line definition is valid for a given length of the lane and will be repeated automatically.

attributes	name	type	unit	value	description
	length	t_grZero	m	[0;∞[Length of the visible part
	space	t_grEqZero	m	[0;∞[Length of the gap between the visible parts
	tOffset	double	m]~;∞[Lateral offset from the lane border. If <sway> element is present, the lateral offset follows the sway.
	sOffset	t_grEqZero	m	[0;∞[Initial longitudinal offset of the line definition from the start of the road mark definition
	rule	e_roadMarkRule		no passing ; caution ; none	Rule that must be observed when passing the line from inside, for example, from the lane with the lower absolute ID to the lane with the higher absolute ID. For values see UML Model.
	width	t_grZero	m	[0;∞[Line width
	color	e_roadMarkColor			Line color. If given, this attribute supersedes the definition in the <roadMark> element. For values see UML Model.

Table 33: Attributes of the roadMark line element

Rules

The following rules apply to road markings:

- <roadMark> elements shall only be used to describe the outer lane marking.
- The centerline of the lane marking shall be positioned on the lane's outer border line in such a way that the outer half of the lane marking is physically placed on the next lane.

Related topics

- Road marking types and lines
- Explicit road marking types and lines
- Offsets in road marking

9.6.1 Road marking types and lines

Detailed information about road marking types and lines may be defined in `<type>` elements within the `<roadMark>` element. Each `<type>` definition contains one or more `<line>` definitions with additional information about the lines of the road marking.

Road marking information in the `<type>` element is more specific than the information given in the `@type` attribute within the `<roadMark>` element.

The outline of the road marking is described by the attributes `@length` and `@space`:

- `@length` represents the visible part of the line.
- `@space` describes the non-visible part.

The position of the road marking in relation to the reference line may be described by defining the lateral offset. A line definition is valid for a given length of the lane and will be repeated automatically. The optional `@rule` attribute for lines defines the traffic rule for passing the lane from the inside.

In OpenDRIVE, road marking types and lines are represented by `<type>` elements within `<roadmark>` elements. The line definitions are contained in `<line>` elements within the `<type>` element.

Attributes

`t_road_lanes_laneSection_lcr_lane_roadMark`

Defines the style of the line at the outer border of a lane. The style of the center line that separates left and right lanes is determined by the road mark element for the center lane.

attributes	name	type	unit	value	description
	sOffset	t_grEqZero	m	[0; ∞ [s-coordinate of start position of the <code><roadMark></code> element, relative to the position of the preceding <code><laneSection></code> element
	type	e_roadMarkType			Type of the road mark. For values see UML model.
	weight	e_roadMarkWeight			Weight of the road mark. This attribute is optional if detailed definition is given below. For values see UML model.
	color	e_roadMarkColor			Color of the road mark. For values see UML model.
	material	string			Material of the road mark. Identifiers to be defined by the user, use "standard" as default value.
	width	t_grEqZero	m	[0; ∞ [Width of the road mark. This attribute is optional if detailed definition is given by <code><line></code> element.

laneChange	e_road_lanes_laneSection_lcr_lane_roadMark_laneChange		increase ; decrease ; both ; none	Allows a lane change in the indicated direction, taking into account that lanes are numbered in ascending order from right to left. If the attribute is missing, "both" is used as default. For values see UML model.
height	double	m]-∞;∞[Height of road mark above the road, i.e. thickness of the road mark.

Table 34: Attributes of the roadMark type element

Related topics

- Road markings
- Explicit road marking types and lines
- Offsets in road marking

9.6.2 Explicit road marking types and lines

Irregular road markings that cannot be described by repetitive line patterns may be described by individual road marking elements. These explicit definitions also contain `<line>` elements for the line definition, however, these lines will not be repeated automatically as in repetitive road marking types.

In OpenDRIVE, irregular road marking types and lines are represented by `<explicit>` elements within `<roadmark>` elements. The line definitions are contained in `<line>` elements within the `<explicit>` element.

The `<explicit>` element should specifically be used for measurement data.

Attributes

`t_road_lanes_laneSection_lcr_lane_roadMark_explicit`

`t_road_lanes_laneSection_lcr_lane_roadMark_explicit_line`

Specifies a single line in an explicit road mark definition.

attributes

name	type	unit	value	description
length	t_grZero	m	[0;∞[Length of the visible line
tOffset	double	m]-∞;∞[Lateral offset from the lane border. If <code><sway></code> element is present, the lateral offset follows the sway.
sOffset	t_grEqZero	m	[0;∞[Offset of start position of the <code><line></code> element, relative to the <code>@sOffset</code> given in the <code><roadMark></code> element

rule	e_roadMarkRule		no passing; caution : none	Rule that must be observed when passing the line from inside, that is, from the lane with the lower absolute ID to the lane with the higher absolute ID. For values see UML Model.
width	t_grZero	m	[0;∞[Line width. This attribute supersedes the definition in the <roadMark> element.

Table 35: Attributes of the roadMark explicit line element

t_road_objects_object_laneValidity

May replace the default validity with explicit validity information for an object. Multiple validity elements may be defined per object.

attributes	name	type	unit	value	description
	fromLane	integer]-∞;∞[Minimum ID of the lanes for which the object is valid
	toLane	integer]-∞;∞[Maximum ID of the lanes for which the object is valid

Table 36: Attributes of the object laneValidity element

Related topics

- Road markings
- Road marking types and lines
- Offsets in road marking

9.6.3 Offsets in road markings

To describe lane markings that are not straight, but have sideway curves, <sway> elements may be used. A <sway> element relocates the lateral reference position for the following (explicit) type definition and thus defines an offset. The sway offset is relative to the nominal reference position of the lane marking, meaning the lane border.

The main use case for sways are roads leading through construction sites. The driving lanes are between the yellow lines. The white lines are swayed and only markings.

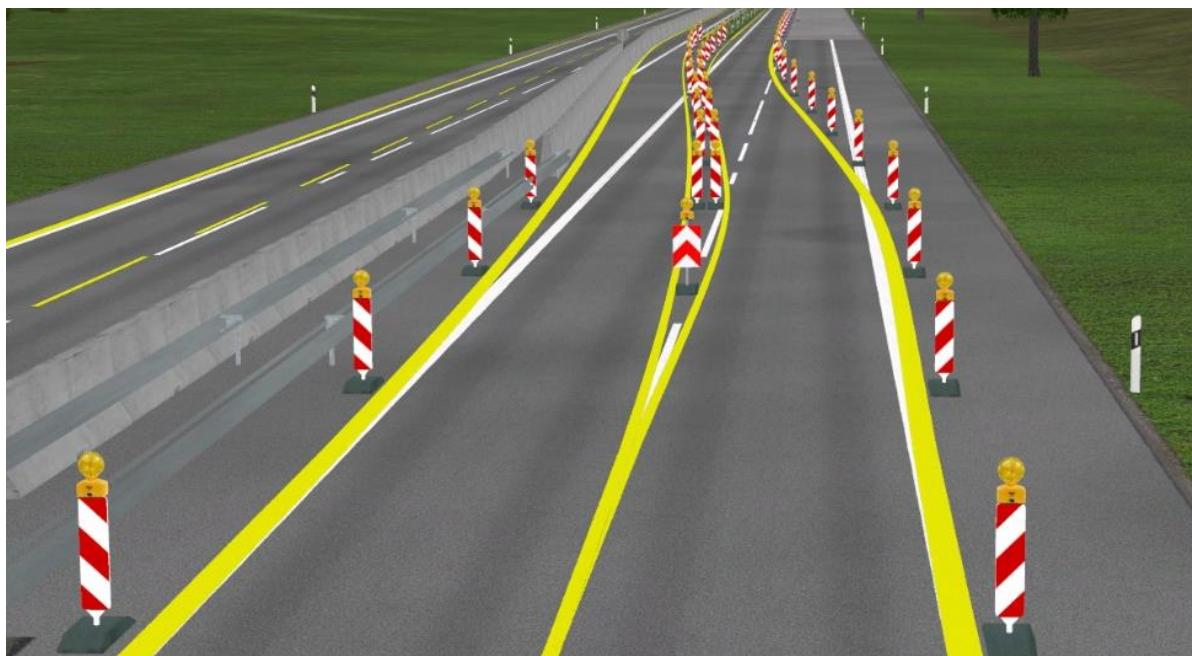


Figure 73: Road marking with sway and offset

Offsets from the lateral reference position are defined by `<sway>` elements within the `<roadMark>` element.

Attributes

`t_road_lanes_laneSection_lcr_lane_roadMark_sway`

Relocates the lateral reference position for the following (explicit) type definition and thus defines an offset. The sway offset is relative to the nominal reference position of the lane marking, meaning the lane border.

attributes	name	type	unit	value	description
	ds	<code>t_grEqZero</code>	m	$[0; \infty[$	s-coordinate of start position of the <code><sway></code> element, relative to the <code>@sOffset</code> given in the <code><roadMark></code> element
	a	double	m	$]^{-\infty}; \infty[$	Polynom parameter a, sway value at <code>@s</code> (<code>ds=0</code>)
	b	double	1	$]^{-\infty}; \infty[$	Polynom parameter b
	c	double	$1/m$	$]^{-\infty}; \infty[$	Polynom parameter c
	d	double	$1/m^2$	$]^{-\infty}; \infty[$	Polynom parameter d

Table 37: Attributes of the sway element

Calculation

For the definition of sways, the lateral reference position at a given point is calculated with the following polynomial function of the third order:

$$tOffset (ds) = a + b*ds + c*ds^2 + d*ds^3$$

where

- tOffset** is the lateral offset of the lateral reference position from the lane border at a given **ds** position
- a, b, c, d** are the coefficients
- ds** is the distance along the reference line between the start of the element and the given position.

ds starts at zero for each element and is relative to the **sOffset** value given in the **<roadMark>** element.

Related topics

- Road markings
- Road marking types and lines
- Explicit road marking types and lines

9.7 Specific lane rules

It is possible to define special rules for certain lanes that are not specifically defined in the OpenDRIVE standard and will be stored in the used application.

In OpenDRIVE, lane rule is represented by the **<rule>** element within the **<lane>** element.

Attributes

t_road_lanes_laneSection_lr_lane_rule

Used to add rules that are not covered by any of the other lane attributes that are described in this specification.

attributes				
name	type	unit	value	description
sOffset	t_grEqZero	m	[0; ∞ [s-coordinate of start position, relative to the position of the preceding <laneSection> element
value	string			Free text; currently recommended values are "no stopping at any time" "disabled parking" "car pool"

Table 38: Attributes of the lane rule element



Rules

The following rules apply to lane rule:

- Applications may have specific lane rules that are only valid in the respective application, but not in OpenDRIVE.

Related topics

- Lanes

10 Junctions

Junctions are areas where three or more roads meet. There are two different kinds of roads with relation to junctions, as shown in Figure 74 Incoming roads: These roads contain lanes that lead into a junction.

- Connecting roads: These roads represent the paths through a junction.

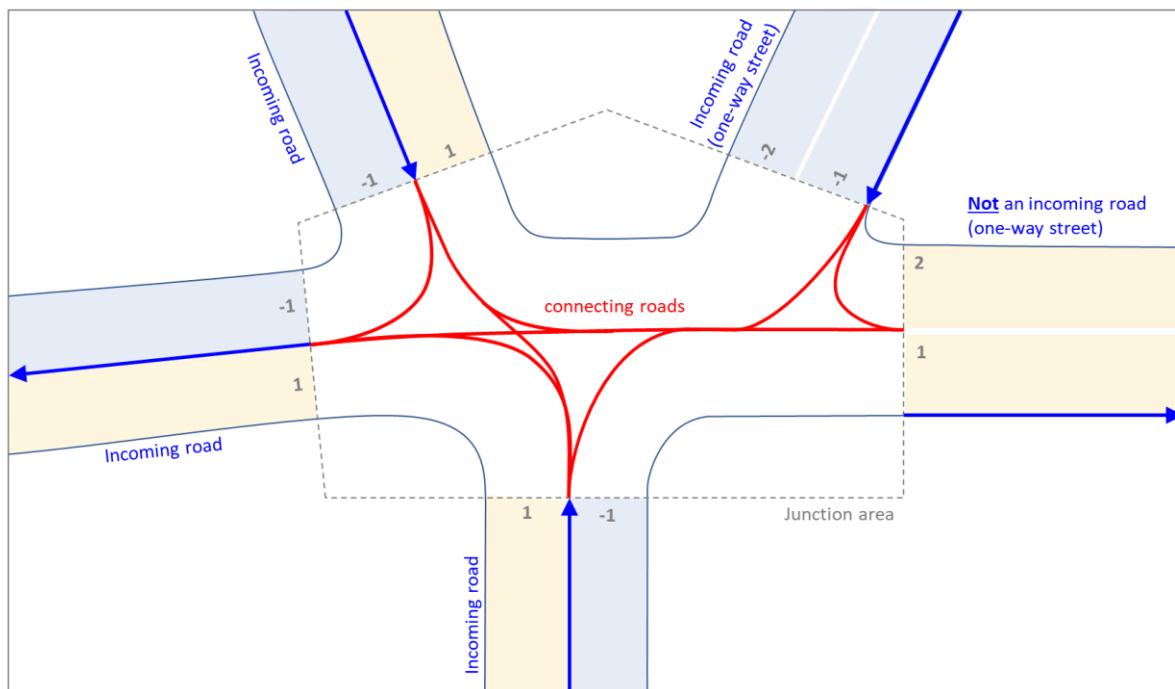


Figure 74: Types of roads in a junction (right-hand traffic)

In OpenDRIVE, junctions are represented by `<junction>` elements. Connecting roads are represented by `<connection>` elements within a `<junction>` element.

Outgoing roads are not specifically defined as element or attribute in OpenDRIVE. Incoming roads serve as outgoing roads. These roads are implicitly defined as outgoing by the connecting roads that lead into them.

class Junction

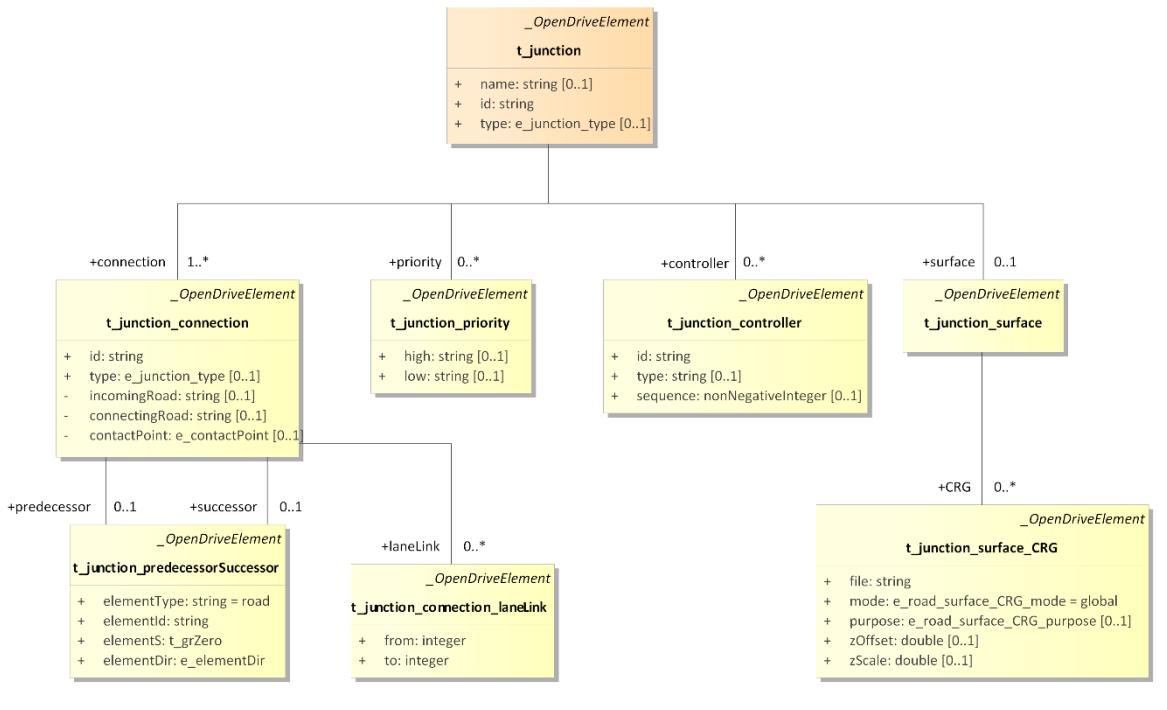


Figure 75: UML model for junctions

Attributes:

`t_junction`

Contains information about all possible connections between roads meeting at a physical junction.

attributes

name	type	unit	value	description
name	string			Name of the junction. May be chosen freely.
id	tstring			Unique ID within database
type	e_junction_type	default; virtual		Type of the junction; regular junctions are of type "default". The attribute is mandatory for virtual junctions

Table 39: Attributes of the junction element

Rules

The following rules apply to junctions:

- Junctions shall only be used when roads cannot be linked directly. They clarify ambiguities for the linking. Ambiguities are caused when a road has two or more possible predecessor or successor roads.
- Unlike roads, junctions do not have a predecessor or successor.
- A junction may have an own name to distinguish it from other junctions.
- Junctions should not be used when only two roads met.

Related topics

- Incoming roads
- Connecting roads
- Road linkage

10.1 Incoming roads

Incoming roads contain lanes that lead into a junction. Because outgoing roads are not specifically defined in OpenDRIVE, incoming roads may also serve as outgoing roads, see Figure 74.

To specify a road as incoming road, its ID is referenced in the `<connection>` element using the `@incomingRoad` attribute.

Attributes

t_junction_connection

Provides information about a single connection within a junction.

attributes

name	type	unit	value	description
id	string			Unique ID within the junction
type	e_junction_type		default ; virtual	Type of the connection, regular connections are type “default” mandatory attribute for virtual connections
incomingRoad	string			ID of the incoming road
connectingRoad	string			ID of the connecting road
contactPoint	e_contactPoint		start ; end	Contact point on the connecting road. For values, see UML Model

Table 40: Attributes of the junction connection element

t_junction_connection_laneLink

Provides information about the lanes that are linked between an incoming road and a connecting road. It is strongly recommended to provide this element. It is deprecated to omit the `<laneLink>` element.

attributes

name	type	unit	value	description
from	integer			ID of the incoming lane
to	integer			ID of the connection lane

Table 41: Attributes of the lane linkage of a connection element

XML example

```
<junction name="myJunction" id="555" >
  <connection id="0"
    incomingRoad="1"
    connectingRoad="2"
    contactPoint="start">
    <laneLink from="-2" to="-1"/>
  </connection>
```

Rules

The following rules apply to incoming roads:

- If the incoming road has lanes leading out of a junction, it also serves as the road leading out off a junction..
- Connecting roads shall not be incoming roads.

Related topics

- Connecting roads
- Junctions
- Road linkage

10.2 Connecting roads

Connecting roads link the roads that meet in a junction. They describe the paths that a vehicle can travel across a junction. Connecting roads are modelled in the same way as standard roads.

The paths described by a connecting road base on its lanes. The connecting road specifies the connections between the lanes of an incoming road and the lanes of an outgoing road of the same junction. If the lanes of an incoming and outgoing road are not linked, this means that there is no traversable path between these lanes.

Figure 74 and Figure 76 show the connecting roads inside the junction area that connect the incoming and outgoing roads.

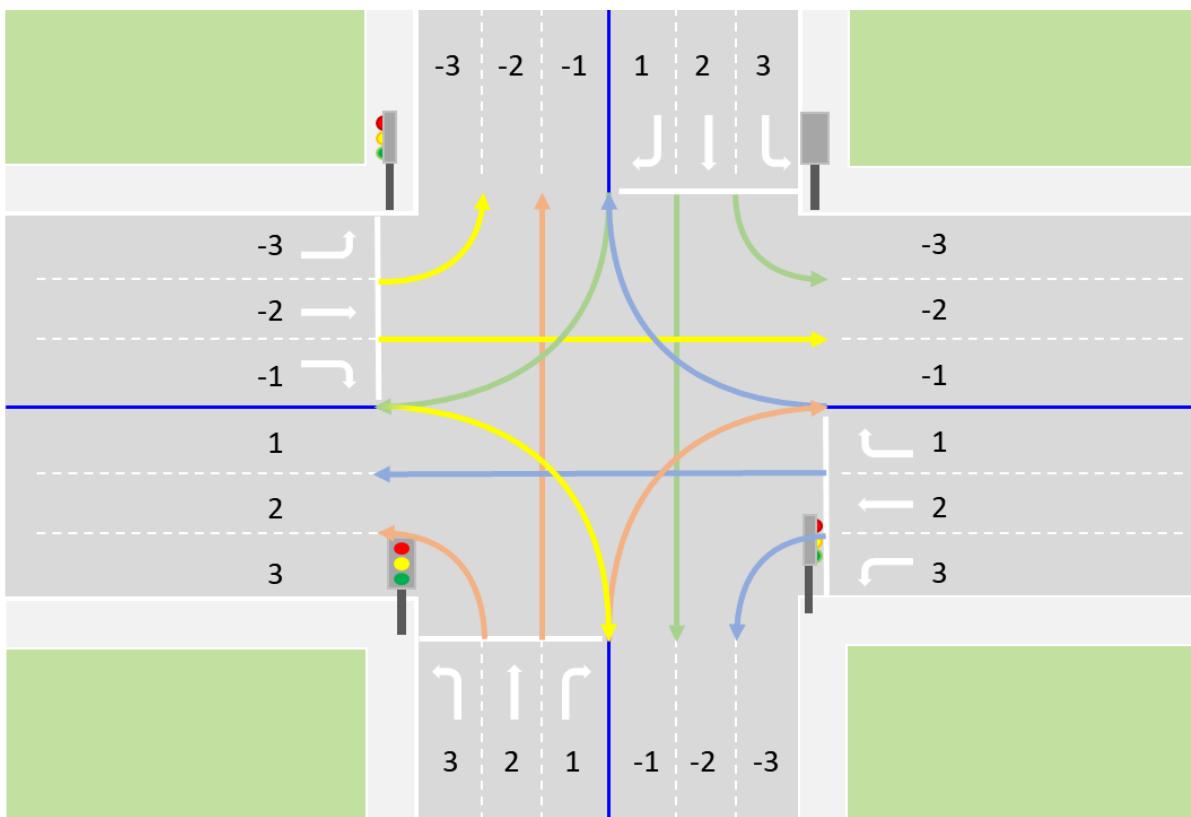


Figure 76: Connecting roads (left hand traffic)

In OpenDRIVE, connecting roads are represented by `<connection>` elements within a `<junction>` element. The lane connections of a connecting road are described by `<laneLink>` elements within the `<connection>` element. The attributes `@from` and `@to` specify the lanes that are linked.

XML example

See example / use case files:

- Ex_LHT-Complex-X-Junction.xodr (left hand traffic)
- UC_Simple-X-Junction.xodr (right hand traffic)

Rules

The following rules apply to connecting roads:

- Each connecting road shall be represented by exactly one `<connection>` element. A connecting road may contain as many lanes as required.
- An incoming road with multiple lanes may be connected to the lanes of the road leading out off the junction in different ways:
 - By multiple connecting roads, each with one `<laneLink>` element for the connection between two specific lanes. Lane changes within this junction are not possible.
 - By one connecting road with multiple `<laneLink>` elements for the connections between the lanes. Lane changes within this junction are possible.
- The linked lanes shall fit smoothly as described for roads.
- Connecting roads may be reserved for specific vehicles, for example busses.

Related topics

- Incoming roads
- Junctions
- Road linkage
- Lane linkage
- Priorities of connecting roads within a junction
- Lane access

10.2.1 Priorities of connecting roads within a junction

If an incoming road is linked to a road leading out of a junction with multiple connecting roads to represent several possible lane connections, then one of these connections may be prioritized. Assigning a priority is only required if the application is unable to derive priorities from signals before or inside a junction or from the lanes leading into a junction.

<priority> elements are used to assign priorities to connecting roads. The possible attributes are @high and @low.

Attributes

t_junction_priority

If an incoming road is linked to an outgoing road with multiple connection roads to represent several possible lane connections, then one of these connections may be prioritized. At least one attribute must be given.

attributes	name	type	unit	value	description
	high	string			ID of the prioritized connecting road
	low	string			ID of the connecting road with lower priority

Table 42: Attributes of the priority element

Rules

The following rules apply for priorities of connecting roads within a junction:

- Priority elements shall only be used if there are no signals.

Related topics

- Connecting roads
- Junctions

10.2.2 Direction of connecting roads

Connecting roads inside a junction may have different directions. For ease of use, the reference line of the connecting roads should be placed in driving direction if the driving direction is unique.

The attribute @contactPoint inside the <connection> element is used to specify the direction of a connecting road.

Rules

The following rules apply for the direction of connecting roads:

- The value “start” shall be used to indicate that the connecting road runs along the linkage indicated in the `<laneLink>` element.
- The value “end” shall be used to indicate that the connecting road runs along the opposite direction of the linkage indicated in the `<laneLink>` element.

Related topics

- Connecting roads
- Road type

10.3 Road surface within junctions

Roads may be elevated within a junction. This is accomplished by describing the road surface in correspondence with the descriptions used in OpenCRG. Using the road surface enables the description of complex elevations within a junction, including overlapping roads. All existing descriptions of road elevation within a junction are superseded by the `<surface>` element.

Because OpenCRG is not part of OpenDRIVE, the `<surface>` element is not described in this specification..

Attributes

t_junction_surface

Used to describe the road elevation profile within a junction. When a `<junction>` element contains a `<surface>` element, the `<surface>` element supersedes all elevation data for connecting roads.

t_junction_surface_CRG

Data described in OpenCRG are represented by the `<CRG>` element within the `<surface>` element.

attributes				
name	type	unit	value	description
file	string			Name of the file containing the CRG data
mode	e_road _surfa ce_CRG_mo de	global		Attachment mode for the surface data.
purpose	e_road _surfa ce_CRG_pur pose	elevat ion ; friction		Physical purpose of the data contained in the CRG file; if the attribute is missing, data will be interpreted as elevation data.

zOffset	double	m]-∞;∞[z offset between CRG center line and inertial xy-plane (default = 0.0)
zScale	double]-∞;∞[z scale factor for the surface description (default = 1.0)

Table 43: Attributes of a CRG element within a surface element

e_road_surface_CRG_mode

attributes	name	type	unit	value	description
	attachecd	string			
	attachecd0	string			
	genuine	string			
	global	string			

Table 44: Attributes of the road surface CRG mode element

e_road_surface_CRG_purpose

attributes	name	type	unit	value	description
	elevation	string			Defines if the linked CRG file contains elevation values
	friction	string			Defines if the linked CRG file contains friction values

Table 45: Attributes of the road surface CRG purpose

Related Topics

- Surface (CRG) for roads

10.4 Virtual junctions

Virtual junctions are used to describe connections within a road that do not require the main road to be split up like in normal junctions. By this, virtual junctions can be placed more easily. Their only use case are driveways that lead to parking lots or residential estates.

Figure 77 show how virtual junctions are applied. The transparent blue roads (road 2,4,5) are part of a virtual junction.

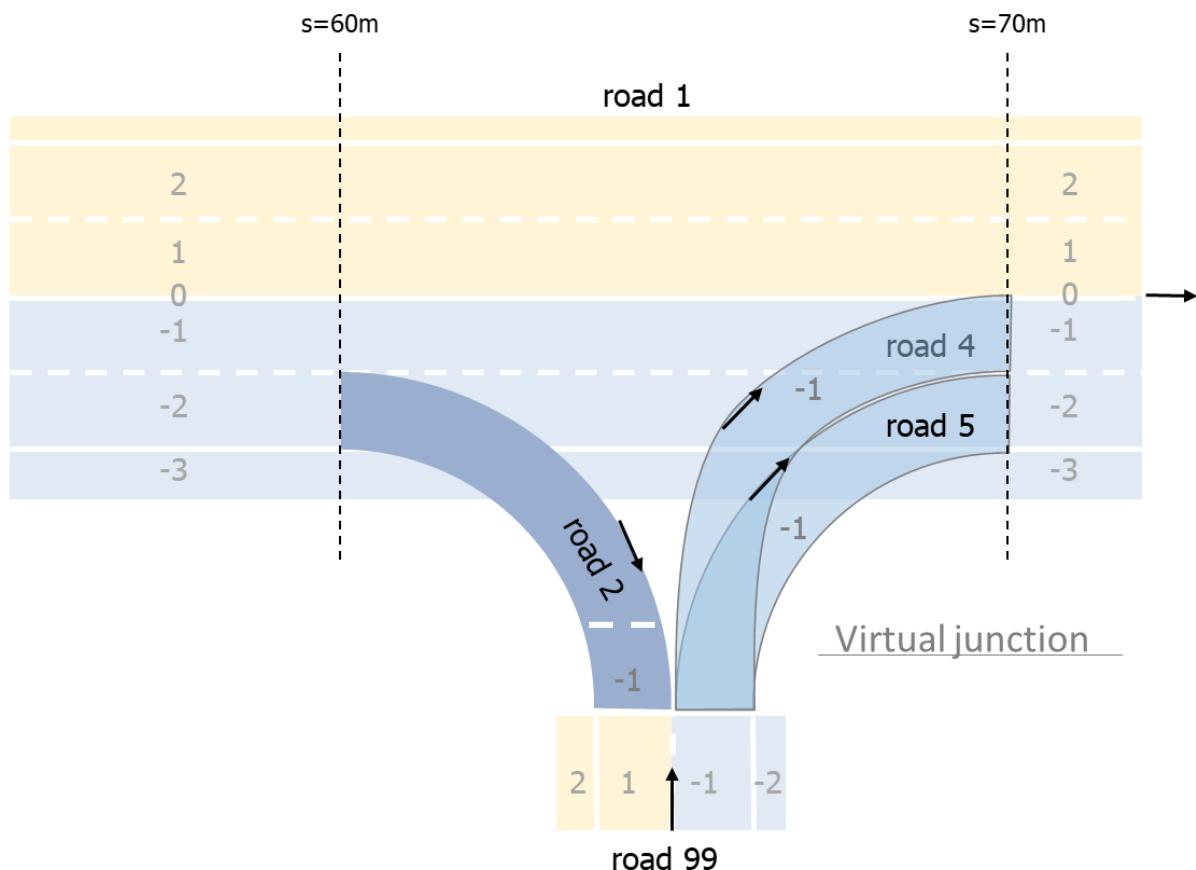


Figure 77: Example of a Virtual junction showing a parking lot entry and exit.

It is not mandatory for road 99 to be defined; It is sufficient to have the connecting roads only.

Virtual junctions are modelled by <junction> elements with the attribute @type.

Attributes

t_junction_predecessorSuccessor

Provides detailed information about the predecessor / successor road of a virtual connection. Currently, only the @elementType "road" is allowed.

attributes	name	type	unit	value	description
	elementType	string		road	Type of the linked element Currently only "road" is allowed.
	elementId	string			ID of the linked element
	elementS	t_grZero		[0;∞[s-coordinate where the connection meets the preceding / succeeding road.
	elementDir	e_elementDir		+ ; -	Direction, relative to the s-direction, of the connection on the preceding / succeeding road

Table 46: Attributes of junction types

XML example

```
<junction name="myJunction" type="virtual" id="555" >
  <connection id="0"
    incomingRoad="1"
    connectingRoad="2"
    contactPoint="start">
    <laneLink from="-2" to="-1"/>
  </connection>
  <connection id="1"
    incomingRoad="-1"
    connectingRoad="4"
    contactPoint="start">
    <laneLink from="-1" to="-1"/>
  </connection>
  <connection id="2"
    incomingRoad="-1"
    connectingRoad="5"
    contactPoint="start">
    <laneLink from="-1" to="-1"/>
  </connection>
</junction>
```

Rules

The following rules apply for virtual junctions:

- The main incoming road within a virtual junction does not need to end before the junction area.
- Virtual junctions shall not replace regular junctions and crossings that connect multiple regular roads.
- Virtual junctions shall be used for branches off the main road only. The main road always has priority.
- Virtual junctions shall not have controllers and therefore no traffic lights.
- If no incoming road is defined the attribute @incomingRoad has the value -1.

Related topics

- Controller
- Virtual connections
- Junctions
- Incoming roads
- Connecting roads

10.4.1 Virtual connections

Virtual connections indicate possible connections between two roads or one or more lanes of two roads. Because the indicated connections are only virtual, no real path is defined. That means that the course of the reference line is not changed.

Virtual connections describe topological connections between roads and lanes. They do not need to be geometrically correct. This feature has been implemented as convenience function to make OpenDRIVE more beginner-friendly.

For a picture of virtual connections, see Figure 78.

Virtual connections are modelled by @type within the <junction> element.

The predecessor and successor roads of a virtual connection are described by <predecessor> and <successor> elements within the <connection> element.

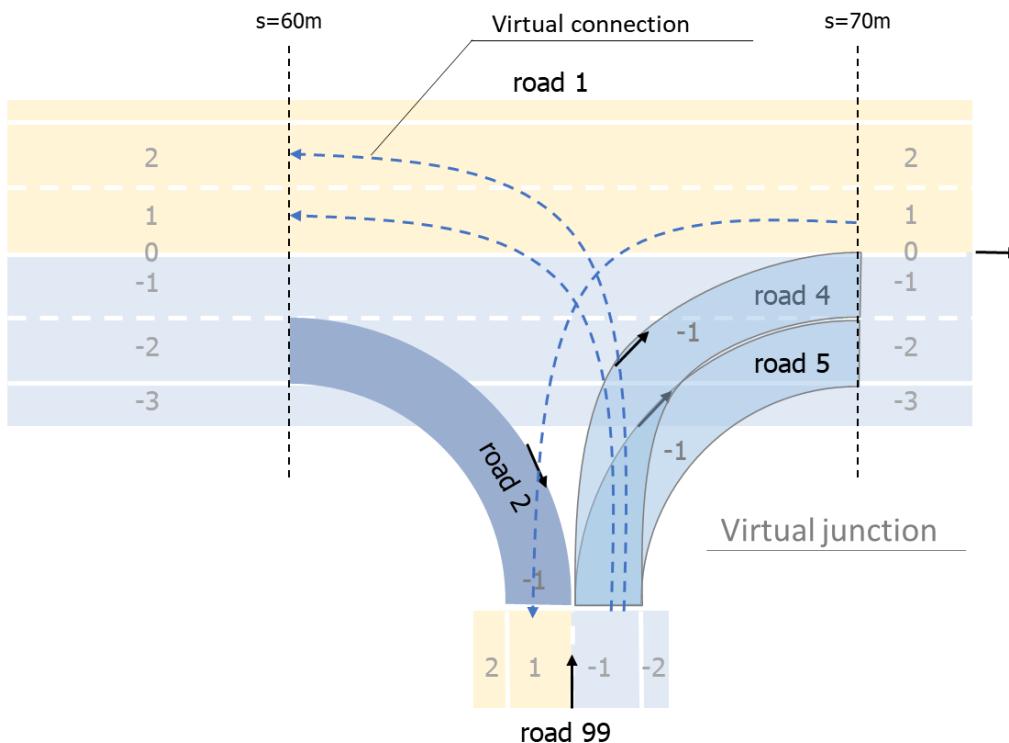


Figure 78: Virtual junction with virtual connections

Attributes

See Attributes 10 Junctions

XML example

```
<junction name="myJunction" type="virtual" id="555" >
    <connection id="0" incomingRoad="1" connectingRoad="2" contactPoint="start">
        <laneLink from="-2" to="-1"/>
    </connection>
    <connection id="1" incomingRoad="99" connectingRoad="4" contactPoint="start">
        <laneLink from="-1" to="-1"/>
    </connection>
    <connection id="2" incomingRoad="99" connectingRoad="5" contactPoint="start">
        <laneLink from="-1" to="-2"/>
    </connection>
    <connection id="3" type="virtual">
        <predecessor elementType="road" elementId="99" contactPoint="end"/>
        <successor elementType="road" elementId="1" elements="60.0" elementDir="-"/>
        <laneLink from="-1" to="1"/>
    </connection>
    <connection id="4" type="virtual">
        <predecessor elementType="road" elementId="99" contactPoint="end"/>
        <successor elementType="road" elementId="1" elements="60.0" elementDir="-"/>
        <laneLink from="-1" to="2"/>
    </connection>
    <connection id="5" type="virtual">
        <predecessor elementType="road" elementId="1" elements="70.0" elementDir="-"/>
        <successor elementType="road" elementId="99" >
        <laneLink from="1" to="1"/>
    </connection>
</junction>
```

```
</connection>
</junction>
```

Rules

The following rules apply for virtual connections:

- Virtual connections shall not replace regular geometrical elements described by road linkage and lane linkage.
- Virtual connections shall only be defined in virtual junctions

Related topics

- Virtual junctions
- Road linkage
- Lane linkage

10.5 Junction groups

Two or more junctions may be grouped in junction groups to indicate that these junctions belong to the same roundabout.

Figure 79 shows how the junctions 1,2,3 are aggregated in the junction group A.

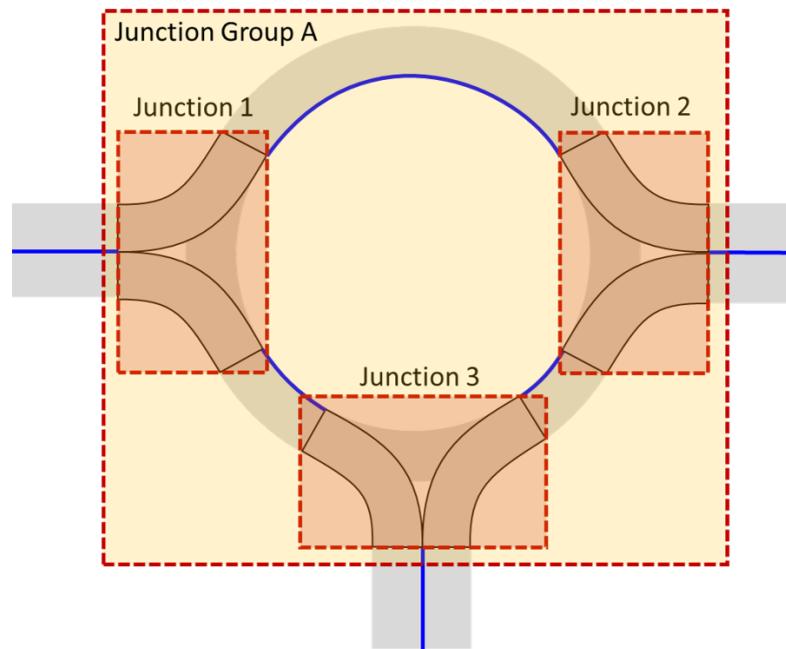


Figure 79: Junction group with three junctions

Junction groups are described by `<junctionGroup>` elements. The junctions that belong to the junction group are specified by `<junctionReference>` elements.

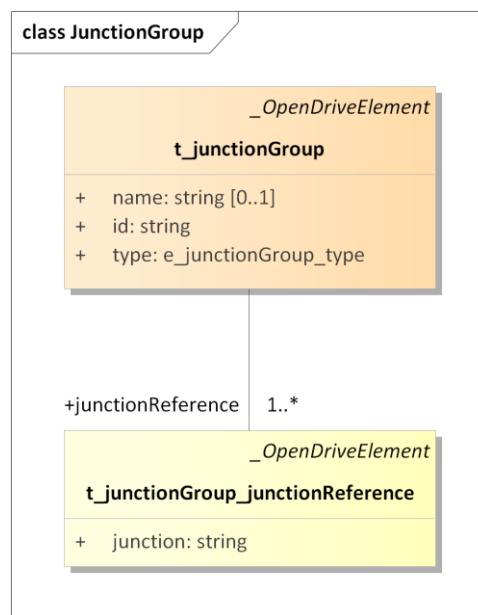


Figure 80: UML model for junction group

Attributes

t_junctionGroup

Two or more junctions may be grouped in junction groups to indicate that these junctions belong to the same roundabout.

The <junctionGroup> element is split into a header element and a series of member elements.

attributes

name	type	unit	value	description
name	string			Name of the junction group. May be chosen freely.
id	string			Unique ID within database
type	e_junctionGroup_type		roundabout; unknown	Type of junction group For values see UML Model

Table 47: Attributes of the junctionGroup element

t_junctionGroup_junctionReference

References to existing junction elements.

attributes

name	type	unit	value	description
junction	string			ID of the junction

Table 48: Attributes of the junction reference element

XML example

UC_2Lane-Roundabout-3Arms.xodr

Related topics

- Junctions
- Controllers for junctions

10.5.1 Controllers for junctions

Controllers may be used to manage signals within a junction, as shown in Figure 81. The use of controllers is described in the signal chapter. The function of the element within the <junction> element is to list the existing controllers and prioritize them in relation to other controllers inside the junction.

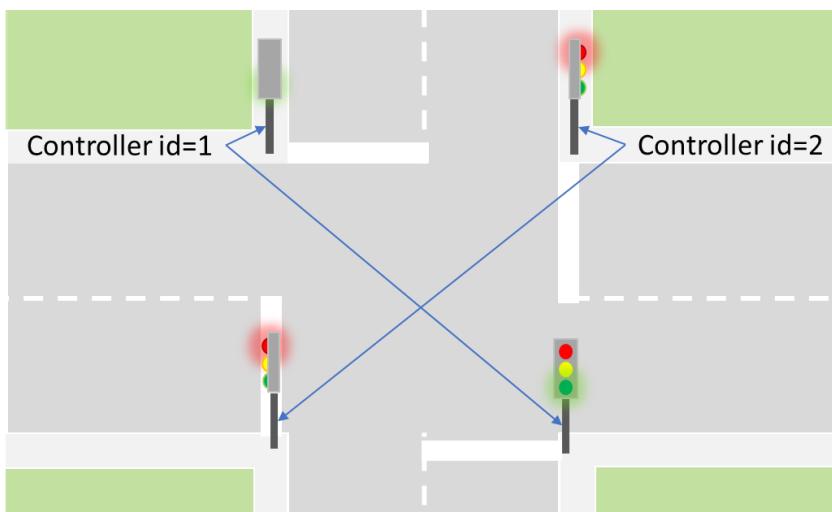


Figure 81: X-Junction with four traffic lights and two controllers

Controllers for junctions are described by `<controller>` elements within the `<junction>` element.

The `@type` of control depends on the application and is not specified in OpenDRIVE.

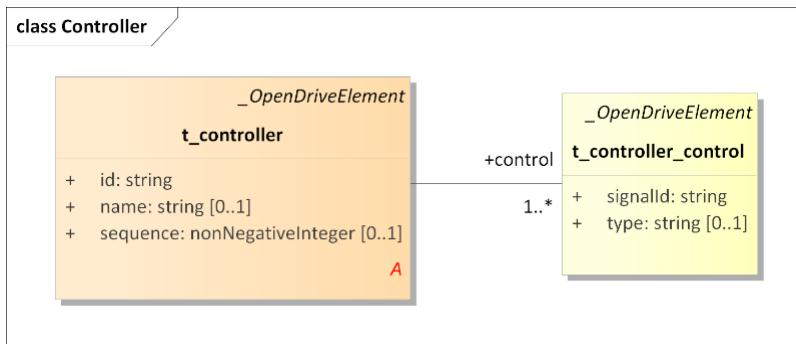


Figure 82: UML model for controller

Attributes

t_junction_controller

Lists the controllers that are used for the management of a junction.

attributes	name	type	unit	value	description
	id	string			ID of the controller
	type	string			Type of control for this junction. Free text, depending on the application.
	sequence	nonNegativeInteger		[0;∞[Sequence number (priority) of this controller with respect to other controllers in the same junction

Table 49: Attributes of the junction controller element

XML example

UC_Simple-X-Junction-TrafficLights.xodr

Related topics

- Controllers
- Junction groups
- Junctions

11 Objects

Objects are items that influence a road by expanding, delimiting, and supplementing its course. The most common examples are parking spaces, crosswalks, and traffic barriers.

There are two ways to describe the bounding box of objects, as shown in Figure 83

- For an angular object: definition of the width, length and height
- For a circular object: definition of the radius and height

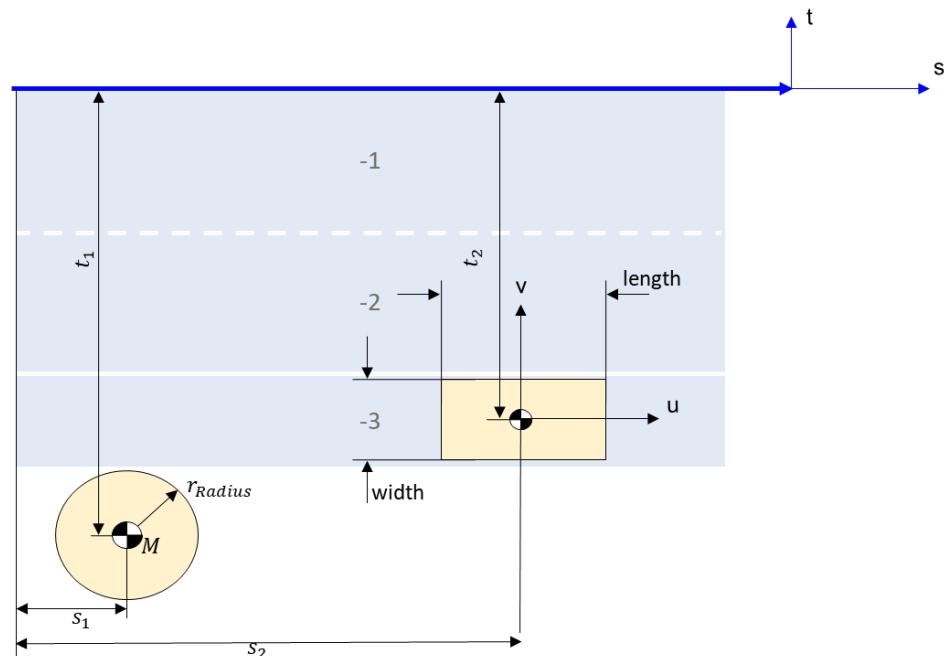


Figure 83: A circular and angular object

Complex objects may be further described using `<outline>` elements. If an `<outline>` is defined, it supersedes the bounding box.

Objects in OpenDRIVE do not change their position. They may be declared dynamic or static:

- Dynamic objects are static but have one or more moveable part. Examples are fans in tunnels or windmills.
- Stationary objects are completely static without any movable part. Examples are buildings or trees.

In OpenDRIVE, objects are represented by the `<objects>` element within the `<object>` element. Objects are defined per `<road>` element.

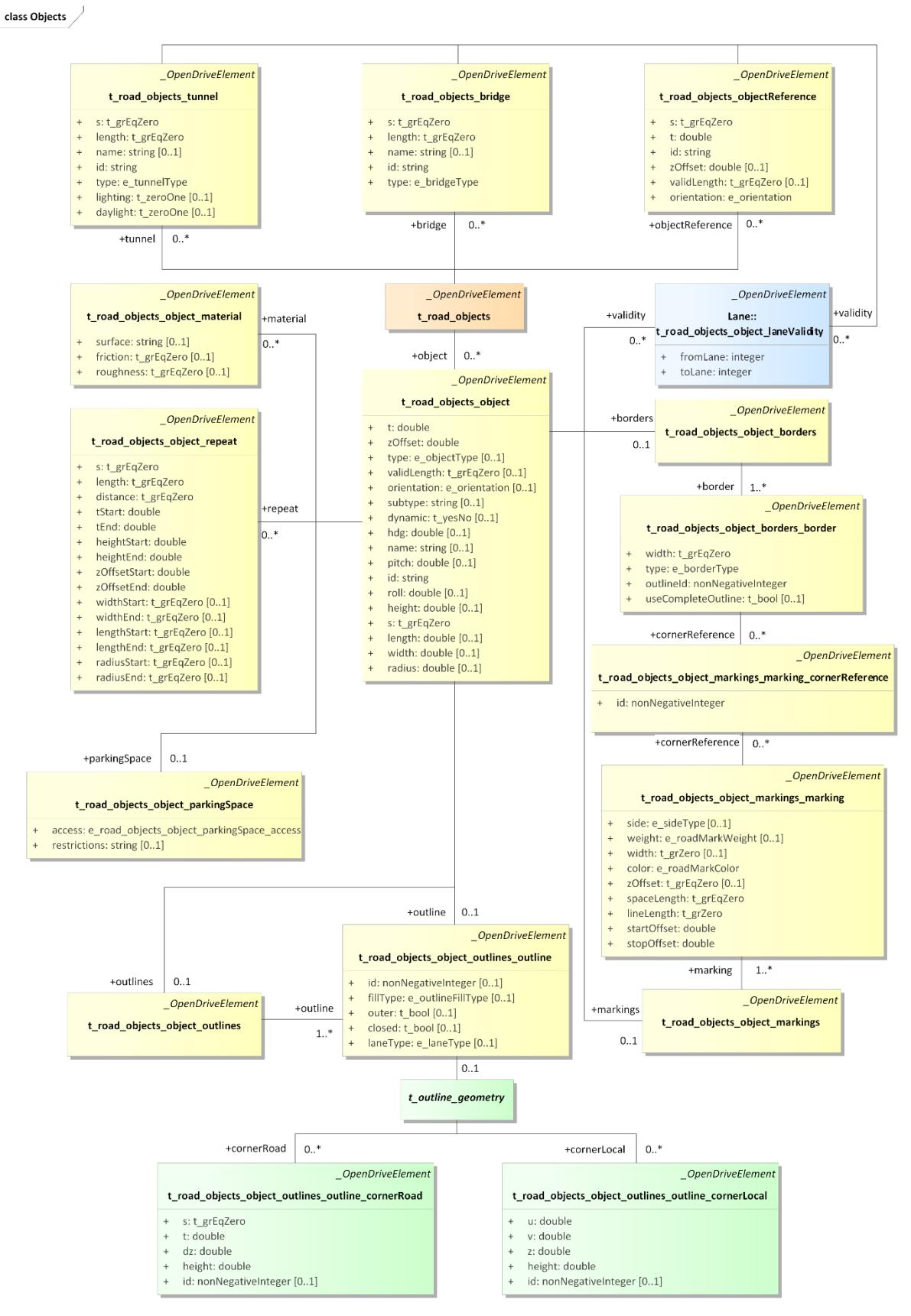


Figure 84: UML modell for objects

Attributes

t_road_objects

Container for all objects along a road.

t_road_objects_object

Describes common 3D objects that have a reference to a given road. *Objects* are items that influence a road by expanding, delimiting, and supplementing its course. The most common examples are parking spaces, crosswalks, and traffic barriers.

There are two ways to describe the bounding box of objects.

- For an angular object: definition of the width, length and height.
- For a circular object: definition of the radius and height.

attributes

name	type	unit	value	description
t	double	m]-∞;∞[t-coordinate of object's origin
zOffset	double	m]-∞;∞[z-offset of object's origin relative to the elevation of the reference line
type	e_objectType			Type of object. For values, see UML. For a parking space, the <parkingSpace> element may be used additionally.
validLength	t_grEqZero	m	[0;∞[Validity of object along s-axis (0.0 for point object)
orientation	e_orientation		+ ; - ; none	"+" = valid in positive s-direction "-" = valid in negative s-direction "none" = valid in both directions (does not affect the heading)
subtype	string			Variant of a type
dynamic	t_yesNo		yes ; no	Indicates whether the object is dynamic or static, default value is "no" (static). Dynamic object cannot change its position.
hdg	double	rad]-∞;∞[Heading angle of the object relative to road direction
name	string			Name of the object. May be chosen freely.
pitch	double	rad]-∞;∞[Pitch angle relative to the x/y-plane
id	string			Unique ID within database
roll	double	rad]-∞;∞[Roll angle relative to the x/y-plane
height	double	m]-∞;∞[Height of the object's bounding box. @height is defined in the local coordinate system u/v along the z-axis
s	t_grEqZero	m	[0;∞[s-coordinate of object's origin

length	double	m	[0;∞[Length of the object's bounding box, alternative to @radius. @length is defined in the local coordinate system u/v along the v-axis
width	double	m	[0;∞[Width of the angular object's bounding box, alternative to @radius. @width is defined in the local coordinate system u/v along the u-axis
radius	double	m	[0;∞[Radius of the circular object's bounding box, alternative to @length and @width. @radius is defined in the local coordinate system u/v"

Table 50: Attributes of the road objects element

XML example

```
<objects>
  <object
    type="building"
    name="ExampleBuilding"
    id="1"
    s="80.0"
    t="17.0"
    zOffset="0.0"
    orientation="none"
    length="12.15"
    width="22.415"
    height="11.84"
    hdg="1.44"
    pitch="0.0"
    roll="0.00">
  </object>
</objects>
```

Rules

The following rules apply to objects:

- The type of an object shall be given by the @type attribute.
- An object may either be dynamic or static.
- Objects derived from OpenSCENARIO shall not be mixed with objects described in OpenDRIVE.
- The direction for which objects are valid shall be specified.
- The origin position of the object shall be described with s- and t-coordinates.
- Objects may be of circular or angular shape. The possibilities are mutually exclusive. The shape is defined by the used attributes.

Related topics

- Repeating objects
- Roads
- Object outline

11.1 Repeating objects

To avoid lengthy XML code, objects of the same type may be repeated. The attributes of the repeated object may be changed. This element is mainly used to describe railings, railing posts, street lamp.

Figure 85 shows one larger angular object that repeats another object.

Figure 86 and Figure 87 show several smaller repeated objects, angular (Figure 86) and circular (Figure 87).

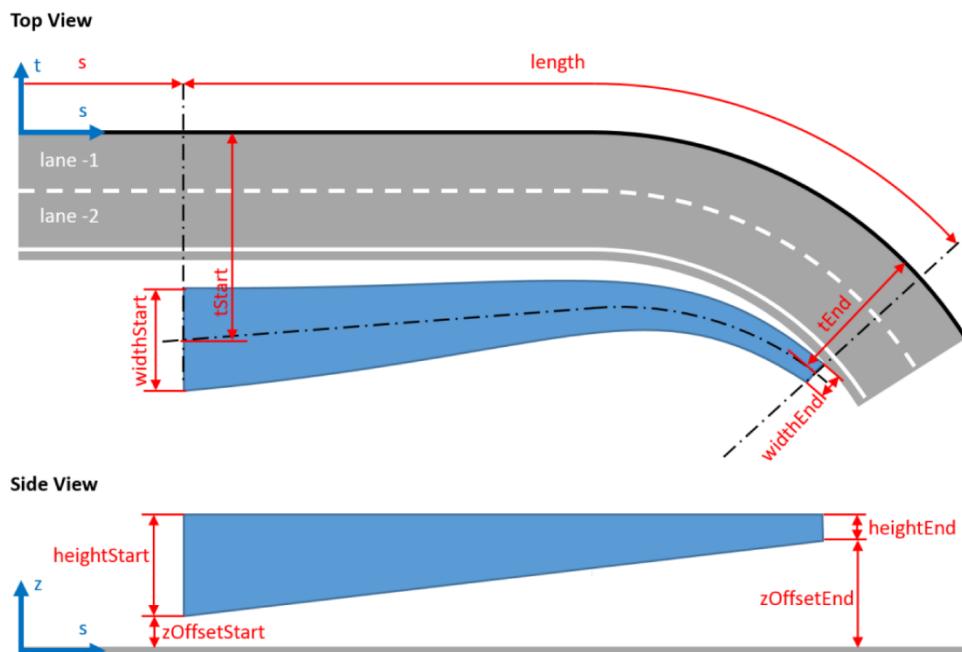


Figure 85: One larger angular repeated object

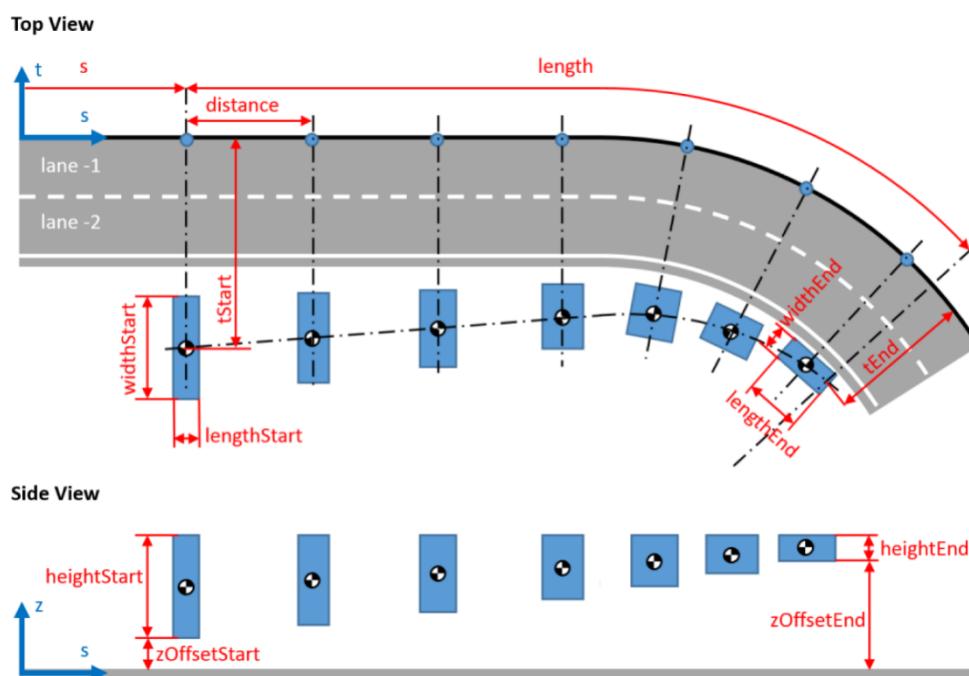


Figure 86: Several smaller angular repeated objects

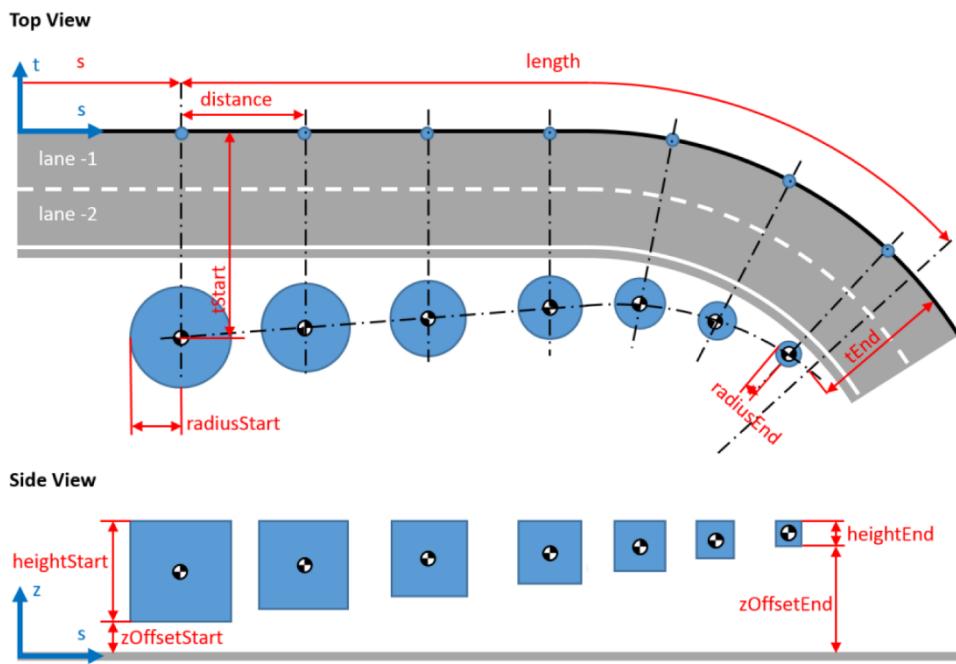


Figure 87: Several smaller circular repeated objects

In OpenDRIVE, repeating objects are represented by the `<repeat>` element within the `<object>` element

Attributes

`t_road_objects_object_repeat`

To avoid lengthy XML code, objects of the same type may be repeated. Attributes of the repeated object shall overrule the attributes from the original object. If attributes are omitted in the repeated objects, the attributes from the original object apply.

attributes					
	name	type	unit	value	description
	s	t_grEqZero	m	[0;∞[s-coordinate of start position, overrides the corresponding argument in the original <code><object></code> record
	length	t_grEqZero		[0;∞[Length of the repeat area, along the reference line in s-direction.
	distance	t_grEqZero	m	[0;∞[Distance between two instances of the object; If this value is zero, then the object is treated like a continuous feature, for example, a guard rail, a wall, etc.
	tStart	double	m]⁻∞;∞[Lateral offset of objects reference point at @s
	tEnd	double	m]⁻∞;∞[Lateral offset of object's reference point at @s + @length

heightStart	double	m]-∞;∞[Height of the object at @s
heightEnd	double	m]-∞;∞[Height of the object at @s + @length
zOffsetStart	double	m]-∞;∞[z-offset of the object at @s, relative to the elevation of the reference line
zOffsetEnd	double	m]-∞;∞[z-offset of the object at @s + @length, relative to the elevation of the reference line
widthStart	t_grEqZero	m	[0;∞[Width of the object at @s
widthEnd	t_grEqZero	m	[0;∞[Width of the object at @s + @length
lengthStart	t_grEqZero			
lengthEnd	t_grEqZero			
radiusStart	t_grEqZero			
radiusEnd	t_grEqZero			

Table 51: Attributes of the repeat element for objects

XML example

```
<objects>
  <object
    type="streetLamp"
    name="ExampleStreetLamp"
    id="2"
    s="15.00"
    t="5.0"
    zOffset="0.0"
    orientation="none"
    length="0.14"
    width="1.28"
    height="7.35"
    hdg="0.0"
    pitch="0.00"
    roll="0.0000">
    <repeat
      s="15.0"
      length="180.0"
      distance="60.00"
      tStart="5.0"
      tEnd="5.0"
      widthStart="1.28"
      widthEnd="1.28"
      heightStart="7.35"
      heightEnd="7.35"
      zOffsetStart="0.0"
```

```
    zOffsetEnd="0.0"/>
  </object>
</objects>
```

Rules

The following rules apply to repeating objects:

- Parameters of the repeated object may differ from the original object.
- Parameters of the repeated object shall overrule the parameters from the original object.

Related topics

- Objects
- Junction groups

11.2 Object outline

Objects may have an outline that is too complex to be described by parameters for angular and circular objects alone. Therefore, the outline of polygonal objects or non-rectangular objects may be described in a more detailed way.

An outline defines a series of corner points, including the height of the object relative to the road reference line. The inner area of the described outline may be filled with a filling type, such as grass, concrete, asphalt, or pavement.

The definition of the outline of objects is mainly used for traffic islands, irregularly shaped parking spaces and special road marking.

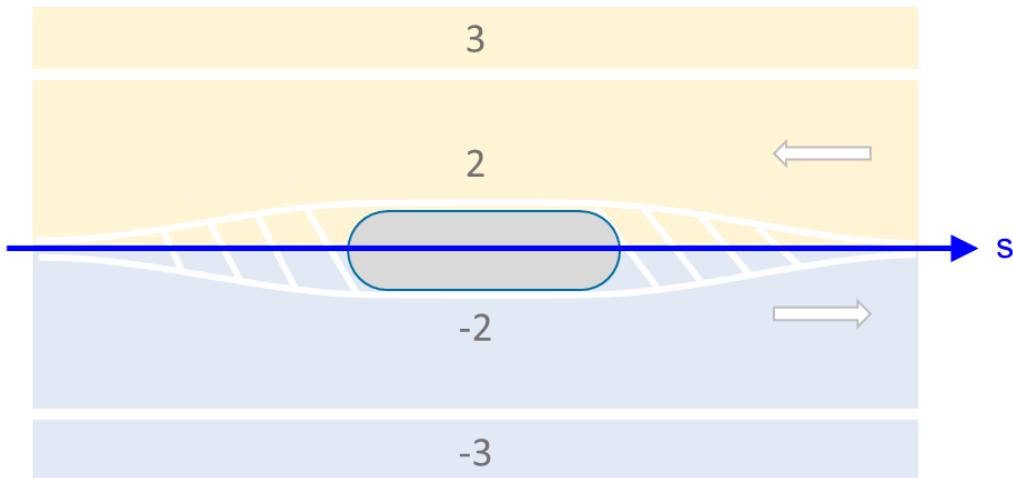


Figure 88: Traffic island as object

In OpenDRIVE, the outline of objects is represented by the `<outlines>` element within the `<object>` element.

The `<outlines>` element serves as a wrapper for the `<outline>` element, which itself contains further elements to describe, for example, corner roads, bridges and borders.

Attributes

t_road_objects_object_outlines_outline

Defines a series of corner points, including the height of the object relative to the road reference line. For areas, the points should be listed in counter-clockwise order.

An `<outline>` element shall be followed by one or more `<cornerRoad>` element or by one or more `<cornerLocal>` element.

OpenDRIVE 1.4 outline definitions (without `<outlines>` parent element) shall still be supported, but deprecated.

attributes	name	type	unit	value	description
	id	nonNegativeInteger		[0;∞[ID of the outline. Must be unique within one object.
	fillType	e_outlineFillType			Type used to fill the area inside the outline. For values see UML Model.
	outer	t_bool		true ; false	Defines if outline is an outer outline of the object.
	closed	t_bool		true ; false	If true, the outline describes an area, not a linear feature.
	laneType	e_laneType			Describes the lane type of the outline. For values see UML Model.

Table 52: Attributes of the outline element

XML example:

Ex_TrafficIsland-CornerRoad.xodr

Rules

The following rules apply to the outline of objects:

- An `<outline>` element shall be followed by one or more `<cornerRoad>` elements or by one or more `<cornerLocal>` element.
- The `<outline>` element may represent an area or a line feature.
- The inner area of the described outline may be filled with a filling type.
- An outline may be specified as an objects outer or inner outline. It may be specified if the described outline is located at the outer border of the object.
- It may be specified like which lane type the object is treated by the application.
- All points of the `<outline>` element must be located inside the bounding box.

Related topics

- Corner roads
- Corner local
- Objects

11.2.1 Corner roads

Corner roads are mandatory elements inside an `<outline>` element. They are used to describe non-linear forms of objects. They are mutually exclusive with `<cornerLocal>` elements. Corner roads describe the outline of objects relative to the road reference line with their s- and t-coordinates.

The shape of an object may be described by the height of the object at a corner and the difference in height relative to the reference line.

Figure 89 shows a non-linear object with several corner points that is described by the s- and t-coordinates along the reference line. Corner road helps to position objects along a road, for example concrete barriers.

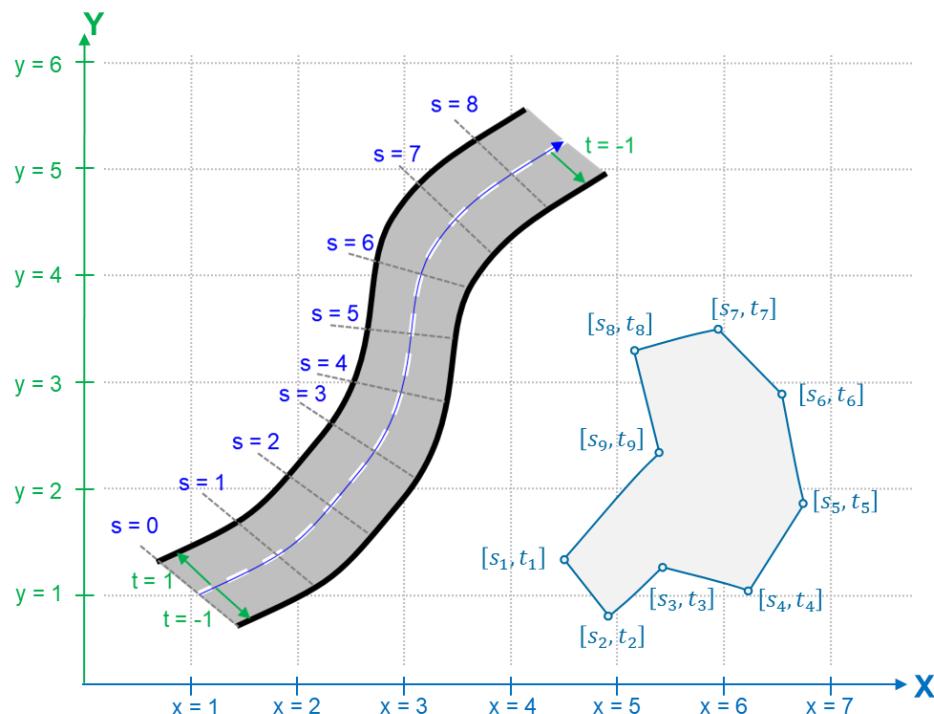


Figure 89: An object described by corner road coordinates

In OpenDRIVE, corner roads are represented by the `<cornerRoad>` element within the `<outline>` element.

Attributes

t_road_objects_object_outlines_outline_cornerRoad

Defines a corner point on the object's outline in road coordinates.

attributes

name	type	unit	value	description
s	t_grEqZero	m]-∞;∞[s-coordinate of the corner
t	double	m]-∞;∞[t-coordinate of the corner
dz	double	m]-∞;∞[dz of the corner relative to road reference line
height	double	m]-∞;∞[Height of the object at this corner, along the z-axis
id	nonNegativeInteger		[0;∞[ID of the outline point. Shall be unique within one outline.

Table 53: Attributes of the cornerRoad element

XML example

See example file Ex_TrafficIsland-cornerRoad.xodr

Rules

The following rules apply to corner roads:

- There shall be at least two <cornerRoad> elements inside an <outline> element.
- There shall be no <cornerLocal> element next to a <cornerRoad> element inside the same <outline> element.

Related topics

- Corner local
- Object outline
- Superelevation

11.2.2 Corner local

Corner local are mandatory elements inside an <outline> element. They are used to describe non-linear forms of objects. They are mutually exclusive with <cornerRoad>. Corner local describes the outline of objects within a local u/v-coordinate system.

Figure 90 shows a non-linear object with several corner points that is described within a local coordinate system. Corner local helps to position objects beyond a road, relative to a single point, for example buildings or traffic islands.

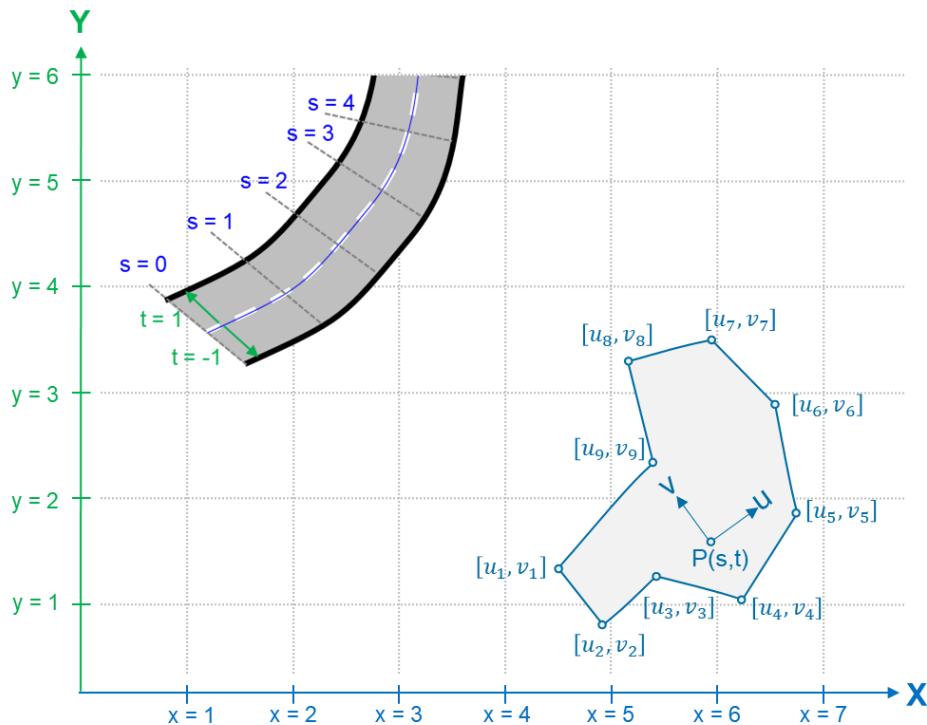


Figure 90: An object described by corner local coordinates

In OpenDRIVE, corner locals are represented by the `<cornerLocal>` element within the `<outline>` element.

Attributes

`t_road_objects_object_outlines_outline_cornerLocal`

Used to describe non-linear forms of objects. Defines a corner point on the object outline relative to the object pivot point in local u/v-coordinates. The pivot point and the orientation of the object are given by the s/t/heading arguments of the `<object>` element.

attributes					
	name	type	unit	value	description
	u	double	m	$]^{-\infty}; \infty[$	Local u-coordinate of the corner
	v	double	m	$]^{-\infty}; \infty[$	Local v-coordinate of the corner
	z	double	m	$]^{-\infty}; \infty[$	Local z-coordinate of the corner
	height	double	m	$]^{-\infty}; \infty[$	Height of the object at this corner, along the z-axis
	id	nonNegativeInteger		$[0; \infty[$	ID of the outline point. Shall be unique within one outline.

Table 54: Attributes of the `cornerLocal` element

XML example

See use case file UC_2Lane-Roundabout-3Arms.xodr

Rules

The following rules apply to corner local:

- There shall be at least two `<cornerLocal>` elements inside an `<outline>` element.
- There shall be no mixture of `<cornerRoad>` and `<cornerLocal>` elements inside the same `<outline>` element.

Related topics

- Corner road
- Object outline

11.3 Object material

Objects placed on a road, such as patches, may consist of a different material than the surrounding road. Therefore, the material of the object may be defined separately. In OpenDRIVE, it is possible to describe the surface, roughness and friction. The values depend on the application and are not defined in OpenDRIVE.

In OpenDRIVE, object material is represented by the `<material>` element within the `<object>` element

Attributes

t_road_objects_object_material

Describes the material properties of objects, for example, patches that are part of the road surface but deviate from the standard road material. Supersedes the material specified in the `<road material>` element and is valid only within the outline of the parent road object.

attributes	name	type	unit	value	description
	surface	string			Surface material code, depending on application
	friction	t_grEqZero		[0; ∞ [Friction value, depending on application
	roughness	t_grEqZero		[0; ∞ [Roughness, for example, for sound and motion systems, depending on application

Table 55: Attributes of the material element within an object

Rules

The following rules apply to object material:

- The material of objects may differ from the surrounding road.

Related topics

- Objects

11.4 Lane validity for objects

Per default, objects are valid for all lanes of a road. Lane validity enables rendering of objects for specific lanes only.

In OpenDRIVE, lane validity is represented by the `<validity>` element within the `<object>` element.

Rules

The following rules apply to lane validity for objects:

- An object may be valid for specified lanes.
- An object may be valid for one lane only.

Related topics

- Objects
- Object reference

11.5 Access rules to parking spaces

Objects of type parking space are defined as all other object types using the `@type=parkingSpace` within the `<object>` element. The outline of the parking space is described by `<cornerRoad>` or `<cornerLocal>` elements, as shown in Figure 91.

The access to a specified parking space may be restricted to a certain group, for example handicapped persons or residents, or a certain group of vehicles, for example busses. Further restrictions depend on the application and are not part of OpenDRIVE.

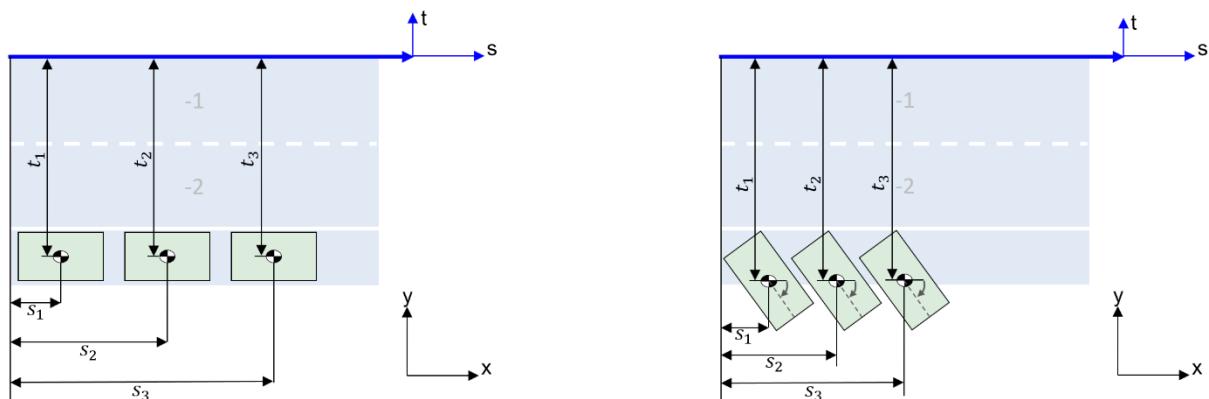


Figure 91: Parking spaces rectangular (left figure) and rhomboid (right figure)

In OpenDRIVE, access rules for parking spaces are represented by the `<parkingSpace>` element within the `<object>` element.

Attributes

t_road_objects_object_parkingSpace

Details for a parking space may be added to the object element.

attributes	name	type	unit	value	description
	access	e_road_object_s_object_parkingSpace_access		all ; car ; women ; handicapped ; bus ; truck ; electric ; residents	Access definitions for the parking space. Parking spaces tagged with "women" and "handicapped" are vehicles of type car. For values see UML Model
	restrictions	string			Free text, depending on application

Table 56: Attributes of the parkingSpace element

XML example

See example files:

- Ex_Parkingspace_Rectangular.xodr
- Ex_Parkingspace_rhomboind.xodr

Rules

The following rules apply to access rules to parking spaces:

- The access to a specified parking space may be limited to a specified group of persons or vehicles.
- Further access restrictions may be defined, but are not part of OpenDRIVE.

Related topics

- Objects
- Corner road
- Corner local

11.6 Object marking

Marking describes the road marks of any objects like crosswalks, stopping-lines, and parking spaces. Marking is defined either in accordance to the bounding box of the element or by referencing outline points of the object.

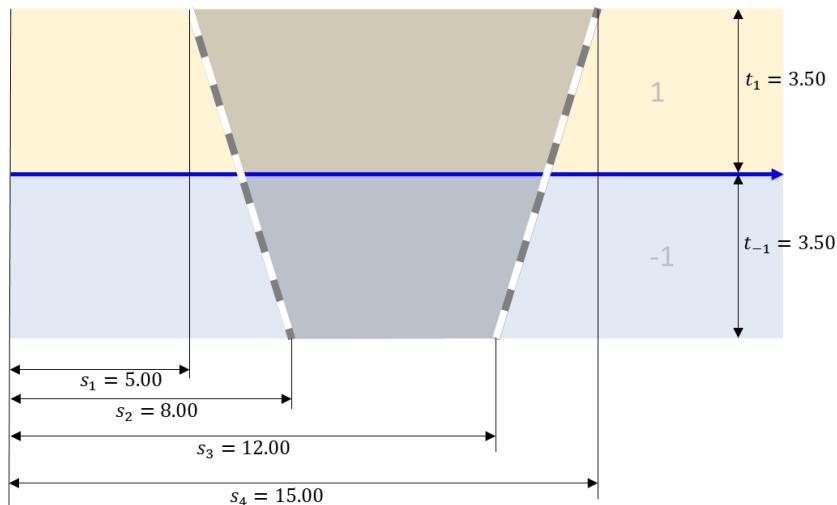


Figure 92: Crosswalk in OpenDRIVE

In OpenDRIVE, the marking of objects is represented by the `<markings>` element within the `<object>` element.

The `<markings>` element serves as a wrapper for the `<marking>` element, which contains further information about the marking.

The marking may be defined for a straight line from one outline point to another by referencing the ID of the respective outline points. For this purpose, the `<cornerReference>` element inside the `<marking>` element is used.

Attributes

t_road_objects_object_markings

Describes the appearance of the parking space with multiple marking elements.

t_road_objects_object_markings_marking

Specifies a marking that is either attached to one side of the object bounding box or referencing outline points.

attributes	name	type	unit	value	description
	side	e_side Type			Side of the bounding box described in <object> element in the local coordinate system u/v. For values see UML model.
	weight	e_road MarkW eight		stand ard ; bold	Optical "weight" of the marking. For values see UML model.
	width	t_grZer o	m	[0; ∞ [Width of the marking.
	color	e_road MarkC olor			Color of the marking. For values see UML model.
	zOffset	t_grEq Zero	m	[0; ∞ [Height of road mark above the road, i.e. thickness of the road mark
	spaceLe ngth	t_grEq Zero	m	[0; ∞ [Length of the gap between the visible parts
	lineLen gth	t_grZero	m	[0; ∞ [Length of the visible part
	startOf fset	double	m]~; ∞ [Lateral offset in u-direction from start of bounding box side where the first marking starts
	stopOff set	double	m]~; ∞ [Lateral offset in u-direction from end of bounding box side where the marking ends

Table 57: Attributes of the marking element

t_road_objects_object_markings_marking_cornerReference

Specifies a point by referencing an existing outline point.

attributes	name	type	unit	value	description
	id	nonNe gativel nteger		[0; ∞ [Index of outline point

Table 58: Attributes of the cornerReference element

XML example

```
<objects>
  <object type="crosswalk" id="10" s="10.0" t="0.0" zOffset="0.0"
    orientation="none" length="10.0" width="7.0" hdg="0.0" pitch="0.0"
    roll="0.0">
    <outlines>
      <outline id="0">
        <cornerRoad s="5.0" t="3.5" dz="0.0" height="4.0" id="0"/>
        <cornerRoad s="8.0" t="-3.5" dz="0.0" height="4.0" id="1"/>
        <cornerRoad s="12.0" t="-3.5" dz="0.0" height="4.0" id="2"/>
        <cornerRoad s="15.0" t="3.5" dz="0.0" height="4.0" id="3"/>
      </outline>
    </outlines>
    <markings>
      <marking width="0.1" color="white" zOffset="0.005" spaceLength
        ="0.05" lineLength ="0.2" startOffset="0.0" stopOffset="0.0">
        <cornerReference id="0"/>
        <cornerReference id="1"/>
      </marking>
      <marking width="0.1" color="white" zOffset="0.005" spaceLength
        ="0.05" lineLength ="0.2" startOffset="0.0" stopOffset="0.0">
        <cornerReference id="2"/>
        <cornerReference id="3"/>
      </marking>
    </markings>
  </object>
</objects>
```

Rules

The following rules apply to object marking:

- The marking of an object shall either completely or partially be defined on its outline
- The color of the marking shall be defined.
- If no outline is used, the attribute @side is mandatory.
- If an outline is used at least two <cornerReference> are mandatory

Related topics

- Object outline
- Objects
- Object borders

11.7 Object borders

Objects may have a border, that is a frame of a defined width. Different border types are available, currently concrete and curb, for example for traffic islands.

In OpenDRIVE, object borders are represented by the `<borders>` element within the `<object>` element. The `<borders>` element serves as a wrapper for the `<border>` element, which itself contains further attributes to describe the borders.

Attributes

`t_road_objects_object_borders`

Objects may have a border, that is, a frame of a defined width. Different border types are available.

`t_road_objects_object_borders_border`

Specifies a border along certain outline points.

attributes	name	type	unit	value	description
	width	t_grEqZero	m	[0; ∞ [Border width
	type	e_borderType			Appearance of border. For values see UML Model.
	outlineId	nonNegativeInteger		[0; ∞ [ID of the outline to use
	useCompleteOutline	t_bool		true; false	Use all outline points for border. "true" is used as default.

Table 59: Attributes of the border element

Rules

The following rules apply to object borders:

- If complete outline is true, `<cornerReference>` shall not be defined
- If complete outline is false, at least two `<cornerReference>` are mandatory

Related topics

- Object outline
- Objects
- Object marking

11.8 Object reference

It is possible to link an object with one or more roads, signals or other objects. These links represent a logical connection between the two elements.

An object reference is used, for example, if a pedestrian crossing crosses several roads. In this case, the pedestrian crossing is defined for one road only, and then referenced by the other roads that it crosses.

The lane validity element may be used to indicate for which lane the object reference is valid.

In OpenDRIVE, object reference is represented by the `<objectReference>` element within the `<objects>` element.

Attributes

`t_road_objects_objectReference`

It is possible to link an object with one or more roads, signals or other objects using a `<objectReference>` element. The referenced objects require a unique ID.

The object reference element consists of a main element and an optional lane validity element.

attributes					
	name	type	unit	value	description
	s	t_grEqZero	m	[0; ∞ [s-coordinate
	t	double	m] $-\infty$; ∞ [t-coordinate
	id	string			Unique ID of the referred object within the database
	zOffset	double	m] $-\infty$; ∞ [z offset relative to the elevation of the reference line
	validLength	t_grEqZero	m	[0; ∞ [Validity of the object along s-axis (0.0 for point object)
	orientation	e_orientation		+ ; - ; none	"+" = valid in positive s-direction "-" = valid in negative s-direction "none" = valid in both directions

Table 60: Attributes of the `objectReference` element

Related topics

- Lane validity
- Objects

11.9 Tunnels

Tunnels are modeled as objects in OpenDRIVE. By definition, tunnels are valid for the complete cross section of a road. Different properties of tunnels may be described: length, whether the tunnel represents an underpass and is open to daylight, and the light conditions.

Figure 93 shows a tunnel that is valid for the whole cross section of the road.

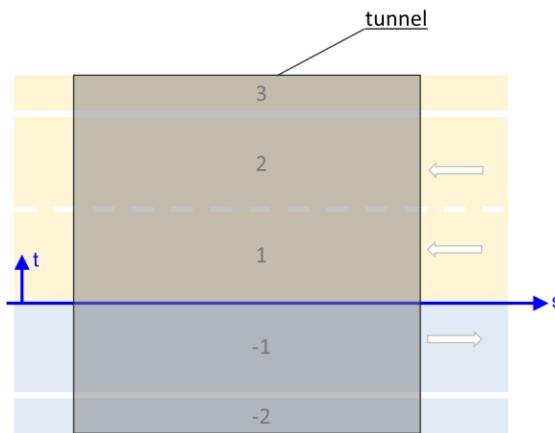


Figure 93: A Tunnel

In OpenDRIVE, tunnels are represented by the `<tunnel>` element within the `<objects>` element.

Attributes

`t_road_objects_tunnel`

Tunnels are modeled as objects in OpenDRIVE. Tunnels apply to the entire cross section of the road within the given range unless a lane validity element with further restrictions is provided as child element.

attributes				
name	type	unit	value	description
s	t_grEqZero	m	[0;∞[s-coordinate
length	t_grEqZero	m	[0;∞[Length of the tunnel (in s-direction)
name	string			Name of the tunnel. May be chosen freely.
id	string			Unique ID within database
type	e_tunnElType			Type of tunnel. For values see UML Model.
lighting	t_zeroOne		[0;1]	Degree of artificial tunnel lighting. Depends on the application.
daylight	t_zeroOne		[0;1]	Degree of daylight intruding the tunnel. Depends on the application.

Table 61: Attributes of the tunnel element

XML example

```
<objects>
  <tunnel
    s="50.0"
    length="100.0"
    name="ExampleTunnel"
    id="1"
    type="standard"
    lighting="0.2"
    daylight="0.9" />
</objects>
```

Rules

The following rules apply to tunnels:

- Tunnels may be restricted to certain lanes, using the `<laneValidity>` element.
- The `@type` of the tunnel shall be specified.

Related topics

- Bridges
- Road cross section
- Lane validity

11.10 Bridges

Bridges are modeled as objects in OpenDRIVE. By definition, bridges are valid for the complete cross section of a road. Different properties of bridges may be described: length and type, such as concrete, steel, wood or brick.

Figure 94 shows a bridge that is valid for the whole cross section of the road

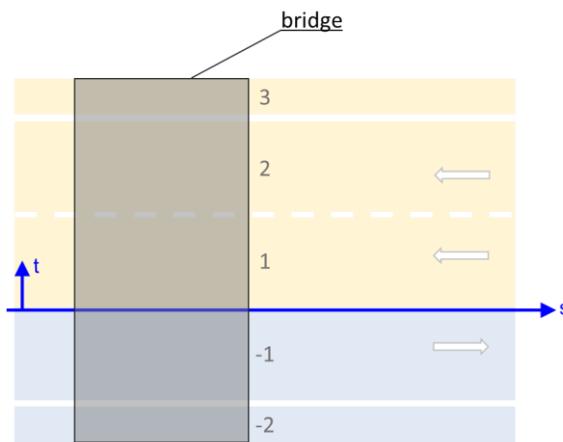


Figure 94: A bridge

In OpenDRIVE, bridges are represented by the `<bridge>` element within the `<objects>` element.

Attributes

t_road_objects_bridge

Bridges are modeled as objects in OpenDRIVE. Bridges are valid for the whole cross section of a road unless a lane validity record with further restrictions is provided as child element.

attributes	name	type	unit	value	description
	s	t_grEqZero	m	[0;∞[s-coordinate
	length	t_grEqZero	m	[0;∞[Length of the bridge (in s-direction)
	name	string			Name of the bridge. May be chosen freely.
	id	string			Unique ID within database
	type	e_bridgeType		concrete; steel; brick; wood	Type of bridge. For values see UML Model.

Table 62: Attributes of the bridge element

XML example

```
<objects>
  <brIDGE
    s="50.0 "
    length="100.0"
    name="ExampleBridge"
    id="1"
    type="concrete" />
</objects>
```

Rules

The following rules apply to bridges:

- Bridges may be restricted to certain lanes, using the `<laneValidity>` element.
- The `@type` of the bridge shall be specified.

Related topics

- Tunnels
- Road cross section
- Lane validity

12 Signals

Signals are traffic signs, traffic lights, and specific road marking for the control and regulation of road traffic, as shown in Figure 95.

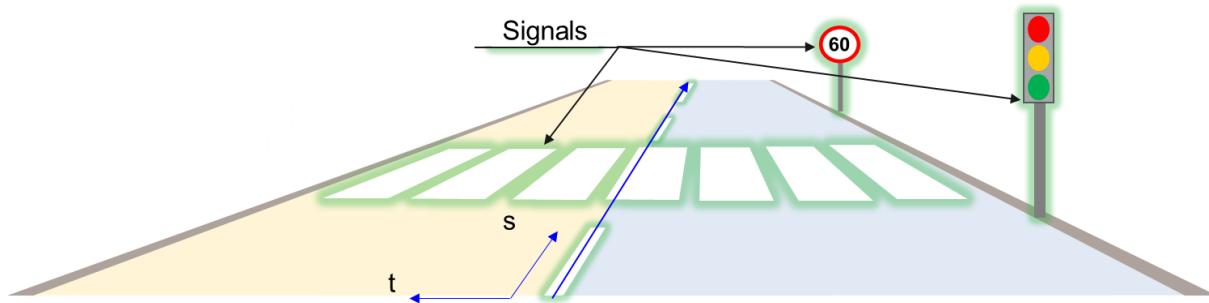


Figure 95: Signals in OpenDRIVE

Signals have different functions and properties:

- They are used to control traffic behavior, for example, with speed limits and turn restrictions, and to alert road traffic about dangerous situations.
- They can be static or dynamic. Static signals, like stop signs, do not change their meaning. Dynamic signals, like traffic lights, may change their meaning during the simulation. Their state may be defined in OpenSCENARIO.

Signals shall be placed in relation to a specific road. The position of the signal is described relative to the road reference line, using the s- and t- coordinates. Signals shall be positioned in such a way that it is clear to which road or lane they belong and where their validity starts. Ambiguity about their interpretation shall be avoided.

Traffic rules are different for each country. The country of the signal is specified in the attribute @country. When placing signals in OpenDRIVE, country-specific legislation and traffic rules should be considered. Legislative changes are indicated by the year when the rules come into force.

Height and width of a signal are not required but are recommended for proper representation of the signal.

Road marks, that are not binding to traffic, are not defined as signals, but only as objects.

A signal with the attributes @type and @subtype is only unique in combination with @country and @countryRevision.

```
<signals>
  <signal s="3981.4158159146"
    t="-14.0503"
    id="5000162"
    name="Vorschriftzeichen"
    dynamic="no"
    orientation="+"
    zOffset="3.8835"
    country="Germany"
    type="274"
    subtype="100"
    countryRevision = "2017"
    value="100"
    unit="km/h"
    width="0.77"
    height="0.77"
    hOffset="0.0" />
</signals>
```

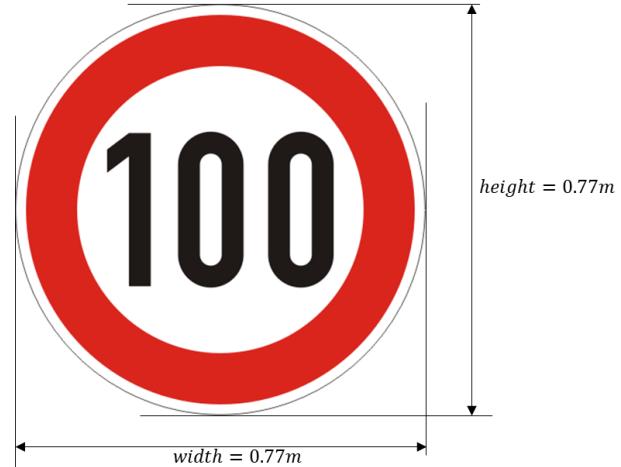


Figure 96: Width and height for signals

In OpenDRIVE, signals are represented by the `<signals>` element within the `<road>` element.

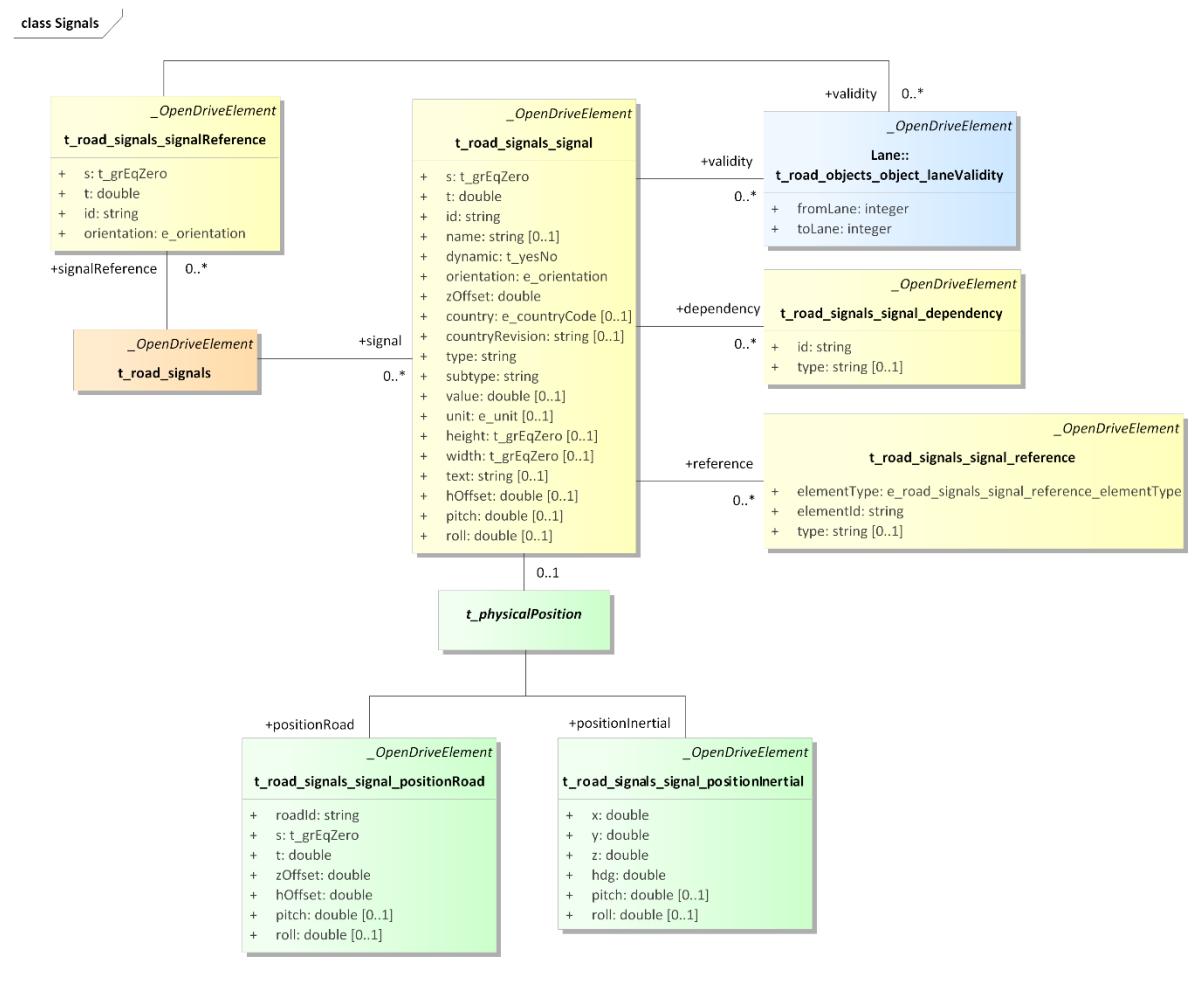


Figure 97: UML model for signals

Attributes

t_road_signals

The `<signals>` element is the container for all signals along a road.

t_road_signals_signal

Used to provide information about signals along a road. Consists of a main element and an optional lane validity element. The tag for a signal is `<signal>`.

attributes

name	type	unit	value	description
s	t_grEqZero	m	[0;∞[s-coordinate
t	double	m]−∞;∞[t-coordinate
id	string			Unique ID of the signal within the OpenDRIVE file

name	string			Name of the signal. May be chosen freely.
dynamic	t_yesNo		yes ; no	Indicates whether the signal is dynamic or static. Example: traffic light is dynamic
orientation	e_orientation		+ ; - ; none	"+" = valid in positive s- direction "-" = valid in negative s- direction "none" = valid in both directions
zOffset	double	m]-∞;∞[z offset from the road to bottom edge of the signal. This represents the vertical clearance of the object. Relative to the reference line.
country	e_countryCode			Country code of the road, see ISO 3166-1, alpha-2 codes.
country Revision	string			
type	string		-1 ; none	Type identifier according to country code or "-1" / "none". See extra document.
subtype	string			Subtype identifier according to country code or "-1" / "none"
value	double			Value of the signal, if value is given, unit is mandatory
unit	e_unit			Unit of @value
height	t_grEq Zero	m	[0;∞[Height of the signal, measured from bottom edge of the signal.
width	t_grEq Zero	m	[0;∞[Width of the signal
text	string			Additional text associated with the signal, for example, text on city limit "City\nBadAibling"
hOffset	double	rad]-∞;∞[Heading offset of the signal (relative to @orientation, if orientation is equal to "+" or "-"). Heading offset of the signal (relative to reference line, if orientation is equal to "none")
pitch	double	rad]-∞;∞[Pitch angle of the signal, relative to the inertial system (xy-plane)
roll	double	rad]-∞;∞[Roll angle of the signal after applying pitch, relative to the inertial system (x'y'-plane)

Table 63: Attributes of the signal element

XML example

```
<signals>
  <signal>
    s="3981.4158159146"
    t="-14.0503"
    id="5000162"
    name="Vorschriftzeichen"
    dynamic="no"
    orientation="+"
    zOffset="3.8835"
    country="DE"
    countryRevision="2017"
    type="274"
    subtype="100"
    value="100"
    unit="km/h"
    height="0.77"
    width="0.77"
    hOffset="0.0"
  </signal/>
</signals>
```

Rules

The following rules apply to signals:

- Signals shall have a specific type and subtype.
- If present, signals shall be used in priority to other traffic rules.
- A country code shall be added to refer to country-specific rules using the @country attribute.
- The year the traffic rules come into force may be specified in the @countryRevision attribute.
- Signals may be valid for one direction or both directions.
- Signals may be dynamic or static.

Related topics

- Roads
- Signal positioning
- Controllers

12.1 Lane validity for signals

By default, signals are valid for all lanes of a road. Lane validity offers the possibility to restrict the validity of a signal to specific lanes only.

In Figure 98, signals in the shape of a road mark specify the speed limit of different lanes.

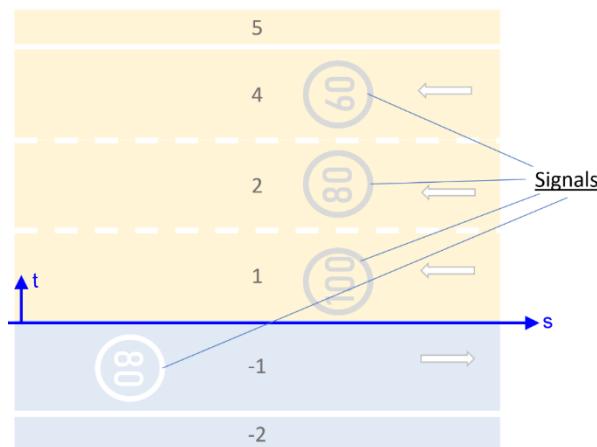


Figure 98: Lanes with signals in the shape of road marks

In OpenDRIVE, lane validity is represented by the `<validity>` element within the `<signal>` element.

Rules

The following rules apply to lane validity for signals:

- A signal may be valid for one or more lanes.

Related topics

- Signals
- Signal dependency
- Signal positioning

12.2 Signal dependency

Signal dependency means that one signal controls the output of another signal. For example, warning lights can be automatically turned on, when a traffic light goes red.

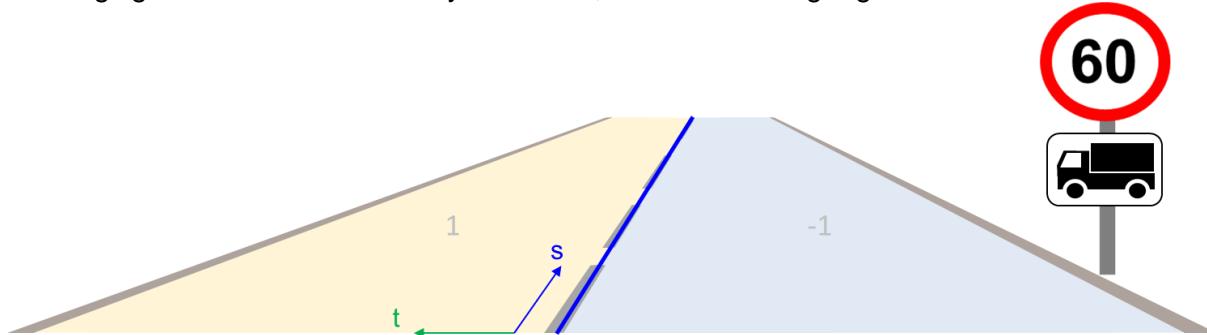


Figure 99: Lane and type specific speed limit

Rules regarding the type of dependency are defined in the application and are not stored in OpenDRIVE.

In OpenDRIVE, signal dependency is represented by the `<dependency>` element within the `<signal>` element.

Attributes

t_road_signals_signal_dependency

Signal dependency means that one signal controls the output of another signal. A signal may have multiple dependency elements.

attributes					
	name	type	unit	value	description
id	string				ID of the controlled signal
type	string				Type of the dependency, Free text, depending on application

Table 64: Attributes of the dependency element

XML example:

```
<signals>
  <signal
    s="50.0"
    t="-4.0" id="1"
    name="SpeedLimit60"
    dynamic="no" orientation="+"
    zOffset="1.90"
    type="274"
    country="DE"
    countryRevision="2013"
    subtype="56"
    value="60.0"
    unit="km/h"
```

```
    hOffset="0.0"
    pitch="0.0"
    roll="0.0"
    height="0.61"
    width="0.61">
    <dependency id="2" />
</signal>
<signal
    s="50.0"
    t="-4.0"
    id="2"
    name="LorriesOnly"
    dynamic="no"
    orientation="+"
    zOffset="1.56"
    type="1048"
    country="DE"
    countryRevision="2013"
    subtype="12"
    hOffset="0.0"
    pitch="0.0"
    roll="0.0"
    height="0.33"
    width="0.60">
    </signal>
</signals>
```

Rules

The following rules apply to lane signal dependency:

- A signal may have multiple dependencies.
- The type of dependency is not specifically defined in OpenDRIVE and may be set in the application

Related topics

- Links between signals and objects
- Controllers
- Signals

12.3 Links between signals and objects

OpenDRIVE offers the possibility to link a signal to an object or another signal.

References to signals and objects are defined within the `<signal>` element and not the superordinate `<signals>` element. It is valid for the specific signal only.

The type of linkage depends on the application and is not defined within OpenDRIVE.

Another example is shown in Figure 98. A speed limit is described as road mark object but referenced as a signal.

In OpenDRIVE, signal reference is represented by the `<reference>` element within the `<signal>` element.

Attributes

`t_road_signals_signal_reference`

Provides a means to link a signal to a series of other elements (for example, objects and signals).

attributes	name	type	unit	value	description
	element Type	e_road _signal s_sign al_refe rence_ eleme ntType		object ; signal	Type of the linked element, For values see UML Model
	element Id	string			Unique ID of the linked element
	type	string			Type of the linkage Free text, depending on application

Table 65: Attributes of the reference element

XML example

Ex_Crosswalk.xodr

Rules

The following rules apply to lane signal reference:

- A signal may be linked to an object or another signal.

Related topics

- Signals
- Objects

12.4 Signal positioning

A signal should be placed next to the road for which it is valid, enabling the application to identify the signals validity. This is called the logical position of a signal. The s-position of the signal describes the position on the road where the signal takes effect.

In certain situations, physical and logical position of a signal are different, as shown in Figure 100. OpenDRIVE offers two possibilities to describe the physical deviation of a signal. The possibilities are mutually exclusive. The positioning of the signal has no influence on its content.

- A signal may be positioned at another physical position that is described with a reference line coordinate system.

The A signal whose physical position deviates from its logical position is represented by the `<positionRoad>` element within the `<signal>` element. That means, the ID of the specified road is referenced, together with the s- and t-coordinates of the road.

Examples are different positions of stop signs and stop lines.

- A signal may be positioned at another physical position that is described with an inertial coordinate system. A signal whose physical position deviates from its logical position and is positioned using inertial coordinates is represented by the `<positionInertial>` element within the `<signal>` element.

Inertial coordinates are used, for example, if the signal is not placed next to a road, but on the other side of the street or hanging over a junction.

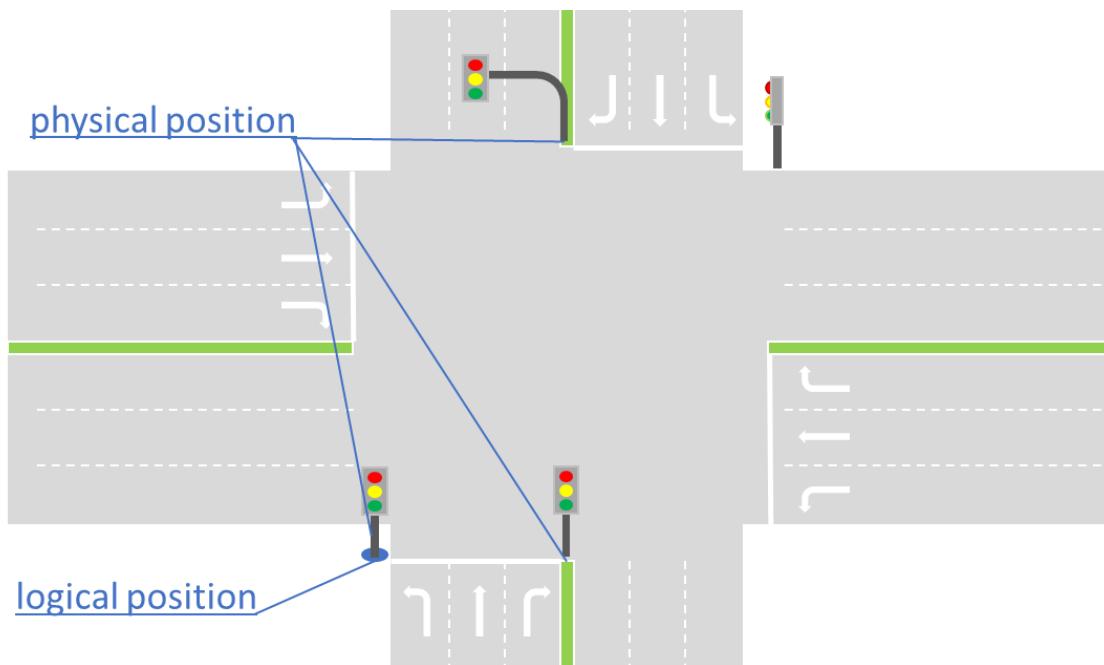


Figure 100: Junction with signals at physical and logical positions

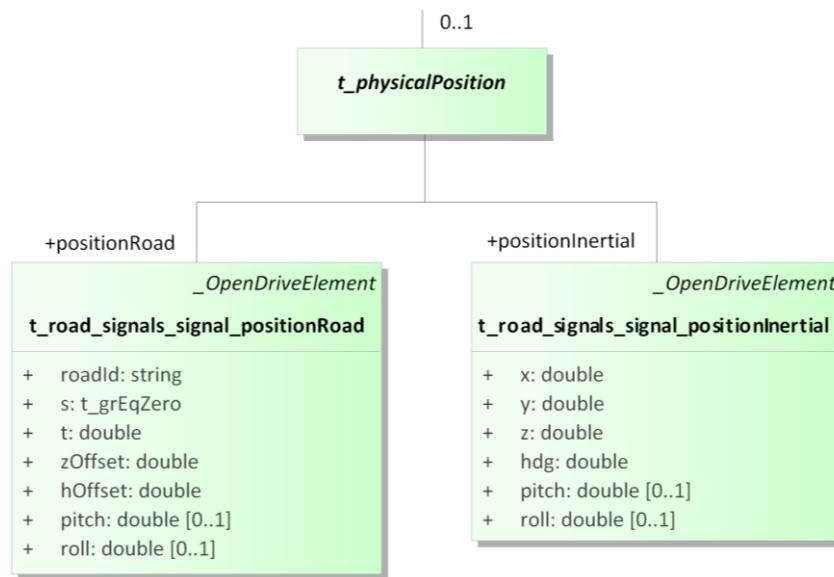


Figure 101: UML Model Physical Position

Attributes

t_road_signals_signal_positionInertial

Describes the reference point of the physical position in inertial coordinates in cases where it deviates from the logical position. Defines the inertial position.

attributes	name	type	unit	value	description
	x	double	m]-∞;∞[x-coordinate
	y	double	m]-∞;∞[y-coordinate
	z	double	m]-∞;∞[z-coordinate
	hdg	double	rad]-∞;∞[Heading of the signal, relative to the inertial system
	pitch	double	rad]-∞;∞[Pitch angle of the signal after applying heading, relative to the inertial system (x'y'-plane)
	roll	double	rad]-∞;∞[Roll angle of the signal after applying heading and pitch, relative to the inertial system (x''y''-plane)

Table 66: Attributes of the positionInertial element

t_road_signals_signal_positionRoad

Describes the reference point of the physical position road coordinates in cases where it deviates from the logical position. Defines the position on the road.

attributes

name	type	unit	value	description
roadId	string			Unique ID of the referenced road
s	t_grEq Zero	m	[0;∞[s-coordinate
t	double	m]∞;∞[t-coordinate
zOffset	double	m]∞;∞[z offset from road level to bottom edge of the signal
hOffset	double	rad]∞;∞[Heading offset of the signal (relative to @orientation)
pitch	double	rad]∞;∞[Pitch angle of the signal after applying hOffset, relative to the inertial system (x'y'-plane)
roll	double	rad]∞;∞[Roll angle of the signal after applying hOffset and pitch, relative to the inertial system (x''y''-plane)

Table 67: Attributes of the positionRoad element

XML Example

See use case file UC_LHT-Complex-TrafficLights.xodr

Rules

The following rules apply to signal positioning:

- Signals should be placed next to the road for which they are valid.
- The physical position of signals may deviate from their logical position.

Related topics

- Signals
- Signals and land validity

12.5 Reuse of signal information

OpenDRIVE offers the possibility to copy signal information between signals by referencing the signal content. This convenience option avoids inconsistencies when positioning multiple signals of the same type and content. For the signal with copied content, it shall be specified for which direction of the road the signal applies. Signals may be valid for multiple roads at once, typically in a junction. Signals references may be used to attach the physical signal logical to several roads (e.g. traffic lights).

Corresponding to the `<signal>` element, a signal with copied content may be supplemented with lane validity. This enables to include or exclude certain lanes from the validity range of the signal.

An example of a signal with copied content is a speed limit signal positioned at an incoming road to a junction. The signal is valid for every connection road inside the junction. Signals on the connection road may reference and copy the signal content. This ensures that all signals follow the same speed limit.

In OpenDRIVE, a reference to another signal for reuse of signal information is represented by the `<signalReference>` element within the `<signal>` element.

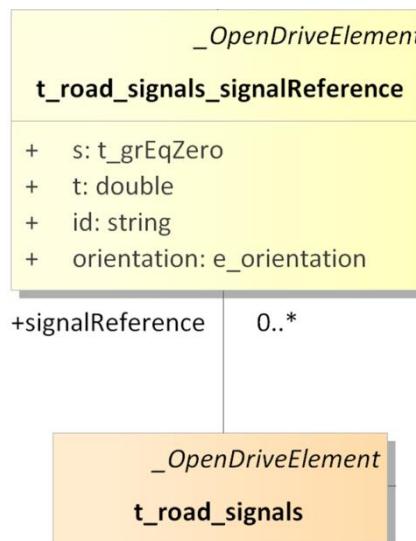


Figure 102: UML Model Signal Reference

Attributes:

t_road_signals_signalReference

Refers to the same, that is, identical signal from multiple roads.

The referenced signals require a unique ID. The `<signalReference>` element consists of a main element and an optional lane validity element.

attributes

name	type	unit	value	description
s	t_grEqZero	m	[0; ∞ [s-coordinate
t	double	m] $-\infty$; ∞ [t-coordinate
id	string			Unique ID of the referenced signal within the database
orientation	e_orientation		+ ; - ; none	"+" = valid in positive s-direction "-" = valid in negative s-direction "none" = valid in both directions

Table 68: Attributes of the signalReference element

Rules

The following rules apply for the purpose of reusing signal information:

- A lane validity element may be added for every `<signalReference>` element.
- Signal reference shall be used for signals only.
- For the signal that reuses the content of another signal, the direction for which it is valid shall be specified.

Related topics

- Lane validity
- Signals
- Signal positioning

12.6 Controllers

Controllers provide identical states for one or more dynamic signals. Controllers serve as wrappers for the behaviour of a group of signals. Controllers are used for dynamic speed control on motorways, and to control traffic light switching phases.

Different from signal dependency, controllers are high-level elements that do not depend on other signals.

Additional content for controllers, such as traffic light phases, is stored outside the OpenDRIVE file.

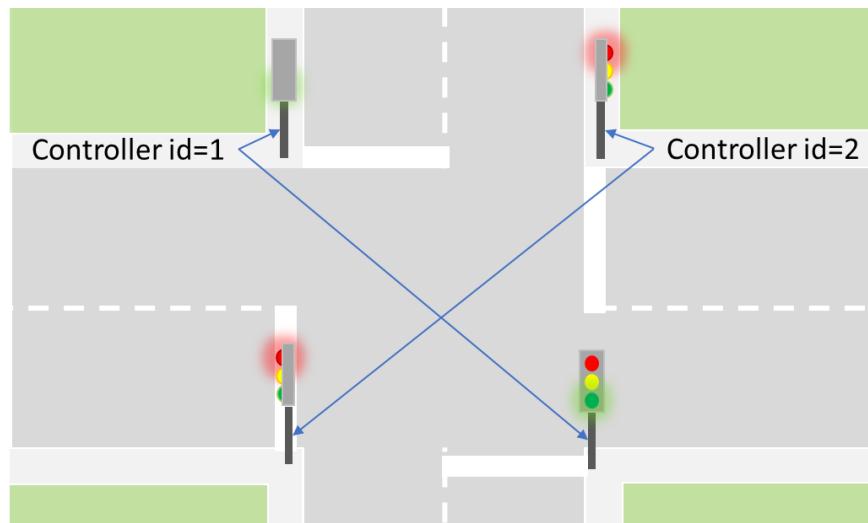


Figure 103: Junction with four traffic lights, controlled by two controllers

In OpenDRIVE, controllers are represented by the `<controller>` element within the `<OpenDRIVE>` element.

The ID of the referenced signal is stored in the `<control>` element within the `<controller>` element.

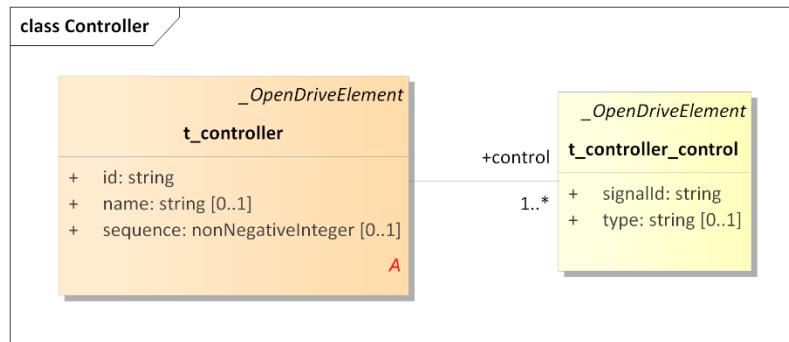


Figure 104: UML Model Controller

Attributes

t_controller

Controllers provides identical states for one or more dynamic signals. Controllers serve as wrappers for the behaviour of a group of signals. Controllers are used for dynamic speed control on motorways, and to control traffic light switching phases.

attributes				
name	type	unit	value	description
id	string			Unique ID within database
name	string			Name of the controller. May be chosen freely.
sequence	nonNegativeInteger		[0;∞[Sequence number (priority) of this controller with respect to other controllers of same logical level

Table 69: Attributes of the controller element of a signal

t_controller_control

Provides information about a single signal controlled by the corresponding controller.

attributes				
name	type	unit	value	description
signalId	string			ID of the controlled signal
type	string			Type of control. Free Text, depends on the application.

Table 70: Attributes of the control element

XML example

See use case file UC_Simple-X-Junction-TrafficLights.xodr

Rules

The following rules apply to controllers:

- Controllers shall be valid for one or more signals.

Related topics

- Signal dependency
- Controllers for junctions
- Signals

13 Railroads

In addition to roads, OpenDRIVE offers the possibility to model rail-based transport systems, that is, trams and streetcars. OpenDRIVE cannot be used for complex railway networks and railway signals. OpenDRIVE describes rail networks only where roads and rails meet.

In OpenDRIVE, railroads are represented by the `<railroad>` element within the `<road>` element.

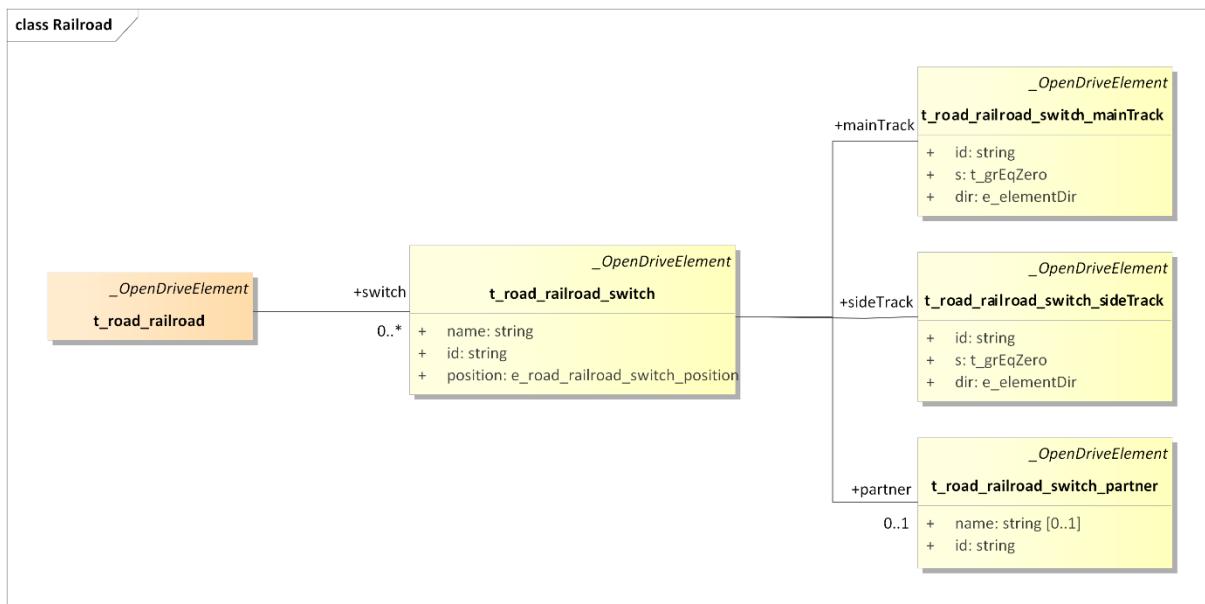


Figure 105: UML Model Railroad

Rules

The following rules apply to railroads:

- Each railway track requires one road.

Related topics

- Railroad tracks
- Switches
- Stations

13.1 Railroad tracks

In OpenDRIVE, rails are always described in connection with the roads on which one pair of rails run. It is not possible to define rails outside roads. Regardless of a tram or sharing the same space with none railway traffic or separate of none railway traffic, it always needs a separate road.

Rails are defined per lane using the @type attribute.

Because rail-based traffic differs from road traffic, the following recommendations apply to the modeling of railroads:

Figure 106 shows the difference between the use of the reference line for roads and railroads.

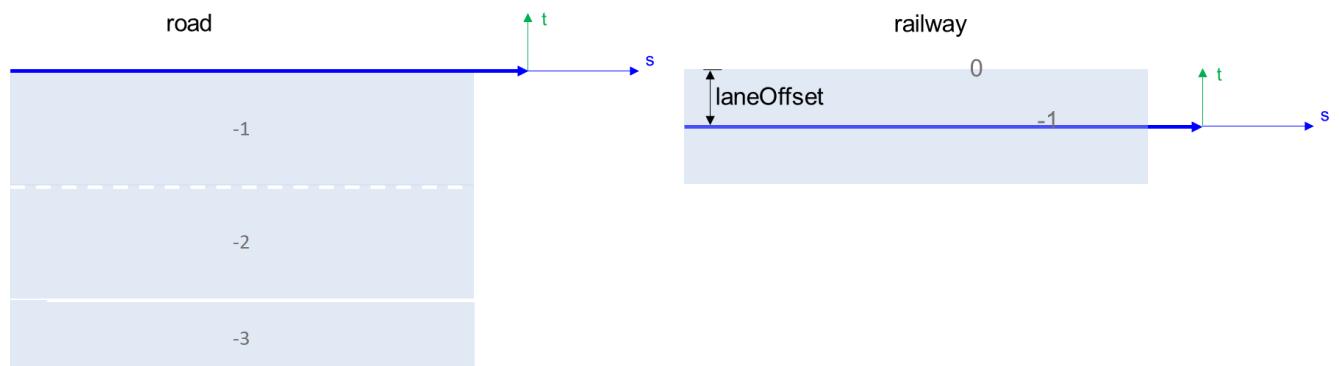


Figure 106: Reference lines for roads and railroads

In OpenDRIVE, *railroad tracks* are represented by the @type attribute within the <lane> element. The values for railroad tracks are tram and rail.

Rules

The following rules apply to railroads:

- The reference line must be in the center of the pair of rails.
- There is only one tram xor one rail lane per road.
- The Width of the lane must be at least the width rail-bound vehicles

Related topics

- Railroads
- Switches

13.2 Switches

Rail-bound vehicles use switches to change their tracks. In contrast to junctions, a switch can guide the vehicles into two directions only.

There are two different types of switches:

- Dynamic switches split the railroad track in two tracks leading in two directions. Dynamic switches can be changed during the simulation.
- Static switches split the railroad track in two tracks leading in two directions. Static switches cannot be changed during the simulation.

Switches may be placed at an arbitrary position on a main track.

Figure 107 shows two partnered switches (12 and 32). A side track (2) connects two main tracks (1 and 3).

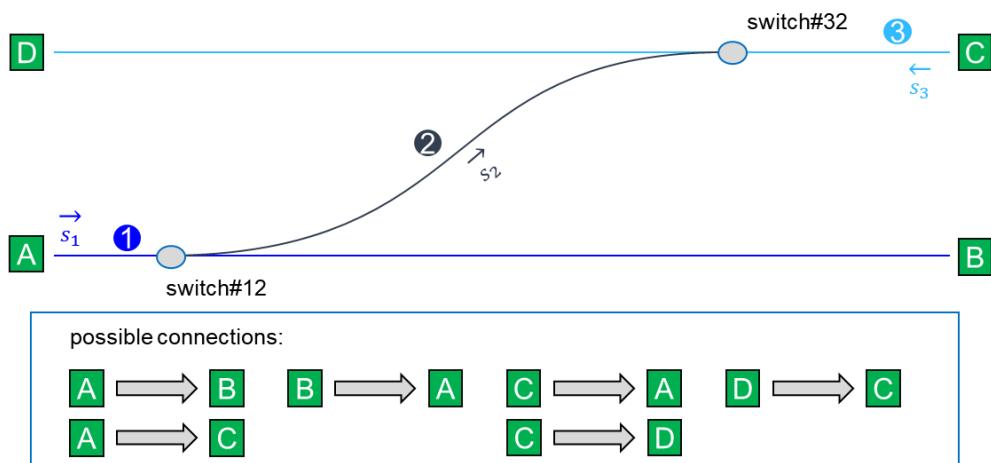


Figure 107: Railroad switches

In OpenDRIVE, switches are represented by the `<switch>` element within the `<railroad>` element.

Attributes

t_road_railroad_switch

attributes					
	name	type	unit	value	description
	name	string			Unique name of the switch
	id	string			Unique ID of the switch; preferably an integer number, see uint32_t
	position	e_road_railroad_switch_position			Either a switch can be operated (dynamic) or it is in a static position. For values see UML Model

Table 71: Attributes of the switch element

XML example

```

<railroad>
  <switch name="ExampleSwitch12" id="12" position="dynamic">
    <mainTrack id="1" s="1.000000000000000e+01" dir="+"/>
    <sideTrack id="2" s="0.000000000000000e+00" dir="+"/>
    <partner name="ExampleSwitch32" id="32"/>
  </switch>
</railroad>

```

Rules

The following rules apply to switches:

- A switch may be either dynamic or static.

Related topics

- Railroad track
- Main track
- Side track
- Partner switch

13.2.1 Main track

A *main track* represents the main course for rail bound traffic. A main track has the same properties as a side track. The two track types have been implemented as a convenience function to simplify the modeling of tracks entering and coming out of switches.

For a picture, see Figure 107

In OpenDRIVE, main tracks are represented by the `<mainTrack>` element within the `<switch>` element.

Attributes

t_road_railroad_switch_mainTrack

attributes					
	name	type	unit	value	description
	<code>id</code>	string			Unique ID of the main track, that is, the <code><road></code> element. Must be consistent with parent containing this <code><railroad></code> element.
	<code>s</code>	<code>t_grEqZero</code>	m	<code>[0;∞[</code>	s-coordinate of the switch, that is, the point where main track and side track meet
	<code>dir</code>	<code>e_elementDir</code>		<code>+ ; -</code>	direction, relative to the s-direction, on the main track for entering the side track via the switch

Table 72: Attributes of the `mainTrack` element

Rules

The following rules apply to main tracks:

- Main tracks shall not be used to connect two switches.

Related topics

- Side track
- Partner switches

13.2.2 Side track

A side track connects two switches that are placed on main tracks. A side track has the same properties as a main track. The two track types have been implemented as convenience function to simplify the modeling of tracks entering and coming out of switches.

For a picture, see Figure 107

In OpenDRIVE, side tracks are represented by the `<sideTrack>` element within the `<switch>` element.

Attributes

t_road_railroad_switch_sideTrack

attributes					
	name	type	unit	value	description
	id	string			Unique ID of the side track, that is, the <code><road></code> element
	s	t_grEq Zero	m	[0; ∞ [s-coordinate of the switch on the side track
	dir	e_elementDir		+ ; -	direction, relative to the s-direction, on the side track for after entering it via the switch

Table 73: Attributes of the `sideTrack` element

Rules

The following rules apply to side tracks:

- Side tracks shall be used to link two switches only.

Related topics

- Main track
- Partner switches

13.2.3 Partner switches

For convenience reasons, two switches may be declared partner switches. This describes a connection between two switches that are linked by a side track. These two switches need to be set consistently.

For a picture, see Figure 107. Here switches #12 and #32 are partner switches

In OpenDRIVE, partner switches are represented by the `<partner>` element within the `<switch>` element.

Attributes

t_road_railroad_switch_partner

Indicates the switch that leads out of a side track after it has been entered.

attributes

name	type	unit	value	description
name	string			Unique name of the partner switch
id	string			Unique ID of the partner switch

Table 74: Attributes of the partner element

Rules

The following rules apply to partner switches:

- Partner switches shall be used to indicate that a side track links two switches.
- Single switches do not have partner switches.

Related topics

- Main track
- Side track

13.3 Stations

Rail-bound vehicles like trams need stations for people to get on and off. Each station shall have at least one platform, which may be further divided into segments. The platforms determine the physical extent of a station.

The `<station>` element may also be used for bus stations.

Figure 108 shows two scenarios for stations:

- In the first scenario, one platform is referenced by the roads 1 and 3, running in different driving directions. The platform consists of one segment only.
- In the second scenario, platform 1 is referenced by road 3 only. Platform 2 is referenced by road 1 and 2. Platform 2 is split into two segments.

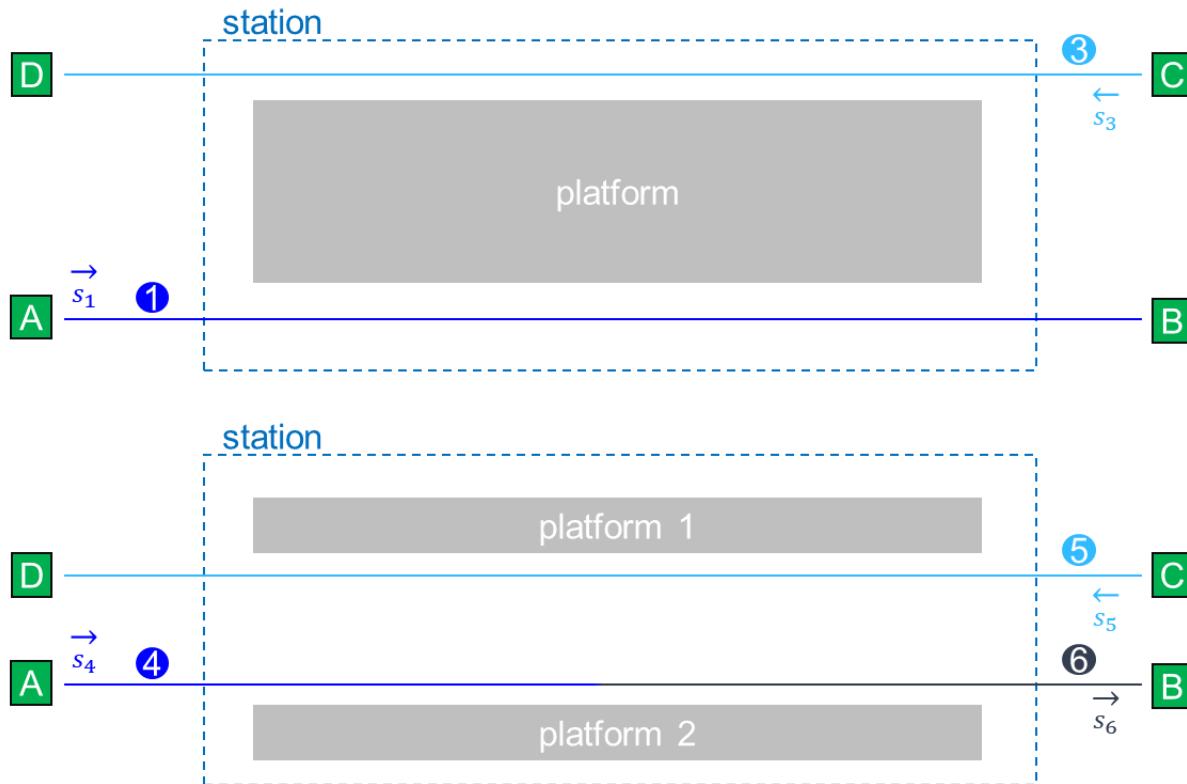


Figure 108: Railroad stations

In OpenDRIVE, stations are represented by the `<station>` element within the `<OpenDRIVE>` element.

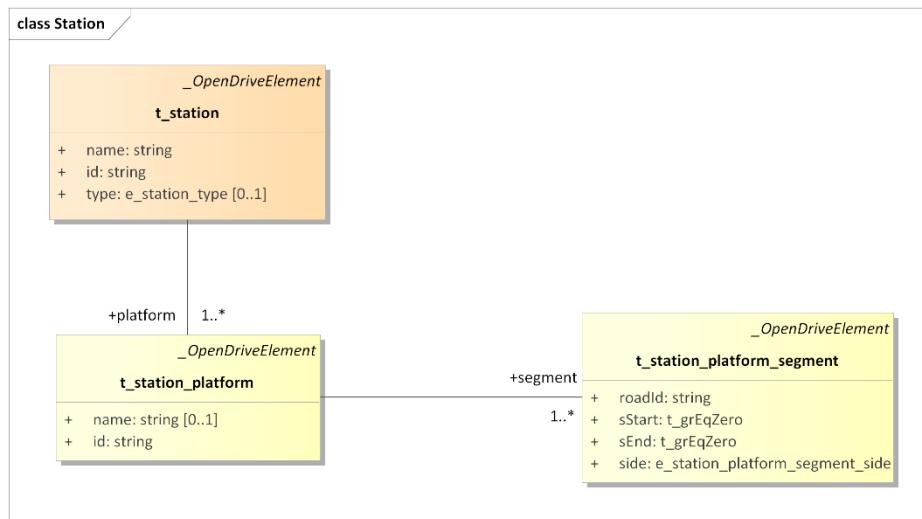


Figure 109: UML Model Station

Attributes

t_station

Defines stations for tram and railroad applications and for automotive environments. May refer to multiple tracks and is therefore defined on the same level as junctions.

attributes

name	type	unit	value	description
name	string			Unique name of the station
id	string			Unique ID within database
type	e_stati on_typ e		small ; medium ; large	Type of station. Free text, depending on the application. e.g.: small, medium, large

Table 75: Attributes of the station element

XML example

See sample file Ex_Railway-station.xodr

Rules

The following rules apply to railroads:

- A `<station>` element shall be followed by at least one `<platform>` element.
- The type of the station may be further specialized by the `@type` attribute. The values are stored in the used application.

Related topics

- Platforms
- Segments
- Railroads

13.3.1 Platforms

A station shall contain at least one platform. A platform shall be referenced by one or more railroad track.

See picture in Figure 108

In OpenDRIVE, railroads are represented by the `<platform>` element within the `<station>` element.

Attributes

`t_station_platform`

Each station element must contain at least one platform element. Each platform element must contain at least one reference to a valid track segment.

attributes	name	type	unit	value	description
	name	string			Name of the platform. May be chosen freely.
	id	string			Unique ID within database

Table 76: Attributes of the platform element

Rules

The following rules apply to platforms:

- There shall be at least one platform per station.
- A platform shall contain at least one segment.

Related topics

- Stations
- Segments

13.3.2 Segments

Platforms may be further divided into *segments*. This is useful if a bi-directional railroad track runs along the same platform. A platform shall contain at least one segment.

In OpenDRIVE, railroads are represented by the `<segment>` element within the `<platform>` element.

Attributes

t_station_platform_segment

Each platform element is valid on one or more track segments. The <segment> element must be specified.

attributes	name	type	unit	value	description
	roadId	string			Unique ID of the <road> element (track) that accompanies the platform
	sStart	t_grEq Zero	m	[0; ∞ [Minimum s-coordinate on <road> element that has an adjacent platform
	sEnd	t_grEq Zero	m	[0; ∞ [Maximum s-coordinate on <road> element that has an adjacent platform
	side	e_station_platform_segment_side		left ; right	Side of track on which the platform is situated when going from sStart to sEnd. For values see UML Model

Table 77: Attributes of the segment element

Rules

The following rules apply to segments:

- There shall be at least one segment per platform.

Related topics

- Stations
- Platforms

14 Bibliography

- [1] Spatial Reference: EPSG:4326:WGS 84; <http://spatialreference.org/ref/epsg/4326/>
- [2] ISO: ISO 8855:2011 - Road vehicles -- Vehicle dynamics and road-holding ability -- Vocabulary; www.iso.org/standard/51180.html
- [3] ASAM e.V.: ASAM CRG - Open Curved Regular Grid; www.asam.net
- [4] W3C: Extensible Markup Language (XML) 1.0 (Fifth Edition); www.w3.org/TR/2008/REC-xml-20081126
- [5] Spatial Reference: EPSG:32632:WGS 84 / UTM Zone 32N; <http://spatialreference.org/ref/epsg/32632/>
- [6] Wikipedia: Euler spiral; http://en.wikipedia.org/wiki/Euler_spiral
- [7] ISO: ISO 8601 - Date and time format; www.iso.org/iso-8601-date-and-time-format.html
- [8] ISO: ISO 3166-1:2013: Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes; www.iso.org/standard/63545.html
- [9] ASAM e.V.: Style Guide for ASAM OpenDRIVE Databases; www.asam.net
- [10] Booch, G., Rumbaugh, J., Jacobson, I. (1997): Unified Modeling Language User Guide. Addison-Wesley.
- [11] Robert G. Burger and R. Kent Dybvig: "Printing floating-point numbers quickly and accurately." In proceedings of ACM SIGPLAN 1996 conference on Programming Language Design and Implementation, Philadelphia, PA, USA, May 1996, pp. 108-116
- [12] Ulf Adams: "Ryū: fast float-to-string conversion." ACM SIGPLAN Notices, Vol. 53, No. 4, April 2018, pp. 270-282
- [13] William D. Clinger: "How to Read Floating Point Numbers Accurately." ACM SIGPLAN Notices, Vol. 25, No. 6, June 1990, pp. 92-101

Figure Directory

Figure 1: UML notation (see ISO TS 19103, Geographic information - Conceptual schema language)	11
Figure 2: UML attribute notation	12
Figure 3: Relation between OpenDRIVE, OpenCRG, and OpenSCENARIO	13
Figure 4: UML model OpenDRIVE Core	19
Figure 5: available coordinate systems in OpenDRIVE.....	21
Figure 6: Coordinate systems in OpenDRIVE interacting with another	21
Figure 7: Inertial coordinate system with defined rotations	22
Figure 8: combined heading, pitch, roll on a cube.....	23
Figure 9: Reference line coordinate system.....	23
Figure 10: Reference line system with defined rotations	24
Figure 11: Heading in reference line.....	24
Figure 12: Roll in reference line	25
Figure 13: Elevation of reference line	25
Figure 14: Local system with defined rotations	26
Figure 15: Local coordinate systems with heading, pitch, roll	26
Figure 16: Local coordinate system with respect to reference line system.....	27
Figure 17: Summary of coordinate system in OpenDRIVE	27
Figure 18: Offset for the geoReference	29
Figure 19: Geometry elements in OpenDRIVE	30
Figure 20: The individual parts of a road	31
Figure 21: UML model Road Geometry including the Reference Line elements	32
Figure 22: A straight line.....	34
Figure 23: Road geometry described by a spiral.....	35
Figure 24: Road geometry described by an arc	36
Figure 25: Creating a reference line from geometry elements	37
Figure 26: A cubic polynom	38
Figure 27: A curve which can't be represented by a cubic polynom w.r.t x parameter	39
Figure 28: Transformation from a u/v- to a x/y coordinate system with $a=b=0$	40
Figure 29: Transformation from a u/v- to a x/y coordinate system with $a!=0$ and $b!=0$	40
Figure 30: A parametric cubic polynom for interpolation of u coordinate	43
Figure 31: A parametric cubic polynom for interpolation of v coordinate	43
Figure 32: A parametric cubic polynom	44
Figure 33: UML Model for Roads.....	46
Figure 34: UML model for Road Geometry	47
Figure 35: Allowed, prohibited, and recommended road linkage.....	49
Figure 36: Allowed scenarios of road linkage	50
Figure 37: Allowed scenario of road linkage within a junction	50
Figure 38: UML model for road linkage.....	51
Figure 39 Types of road elevation	55
Figure 40: Superelevation	57
Figure 41: Minimum t-definiton range for road shapes.....	59
Figure 42: Shape definition (left) in combination with superelevation (right).....	59
Figure 43: Road surface as defined in a CRG file	61
Figure 44: positioning of an OpenCRG file along the reference line	61
Figure 45: OpenCRG attachement mode, attached	62
Figure 46: OpenCRG attached mode with elevation	63
Figure 47: OpenCRG attached0 mode with elevated reference line	63
Figure 48: OpenCRG attachement mode, genuine	64
Figure 49: OpenCRG orientation	65
Figure 50: Example crossfall modelled with road shape	67

Figure 51: Center lane for road with lanes of different driving directions	69
Figure 52: Center lane for road with lanes of identical driving direction	70
Figure 53: UML model for lanes	70
Figure 54: Lane grouping with left, center, right	72
Figure 55: UML model for t_road_lanes	73
Figure 56: A road section with lane sections	74
Figure 57: Lane sections defined separately for both sides of the road	74
Figure 58: Lane offset	77
Figure 59: Lane links for road with id 10	79
Figure 60: UML Model t_road_lanes_laneSection_lcr_lane_link	79
Figure 61: UML model lane properties	81
Figure 62: Change of lane width per lane section	83
Figure 63: Lane with varying border shape	85
Figure 64: Lane types for a motorway	87
Figure 65: Lane types for a rural road	87
Figure 66: Lane types for an urban road	88
Figure 67: Lane types for motorway exit and entry	88
Figure 68: Lane types for motorway connecting to another motorway	89
Figure 69: Lane-specific speed limits	91
Figure 70: Lane access, bus lane	93
Figure 71: Lane height	95
Figure 72: Lanes excluded from road elevation	97
Figure 73: Road marking with sway and offset	103
Figure 74: Types of roads in a junction (right-hand traffic)	106
Figure 75: UML model for junctions	107
Figure 76: Connecting roads (left hand traffic)	110
Figure 77: Example of a Virtual junction showing a parking lot entry and exit	114
Figure 78: Virtual junction with virtual connections	117
Figure 79: Junction group with three junctions	119
Figure 80: UML model for junction group	119
Figure 81: X-Junction with four traffic lights and two controllers	121
Figure 82: UML model for controller	121
Figure 83: A circular and angular object	123
Figure 84: UML model for objects	124
Figure 85: One larger angular repeated object	127
Figure 86: Several smaller angular repeated objects	127
Figure 87: Several smaller circular repeated objects	128
Figure 88: Traffic island as object	130
Figure 89: An object described by corner road coordinates	132
Figure 90: An object described by corner local coordinates	134
Figure 91: Parking spaces rectangular (left figure) and rhomboid (right figure)	136
Figure 92: Crosswalk in OpenDRIVE	138
Figure 93: A Tunnel	143
Figure 94: A bridge	144
Figure 95: Signals in OpenDRIVE	146
Figure 96: Width and height for signals	147
Figure 97: UML model for signals	148
Figure 98: Lanes with signals in the shape of road marks	151
Figure 99: Lane and type specific speed limit	152
Figure 100: Junction with signals at physical and logical positions	155
Figure 101: UML Model Physical Position	156
Figure 102: UML Model Signal Reference	158
Figure 103: Junction with four traffic lights, controlled by two controllers	160
Figure 104: UML Model Controller	160



Figure 105: UML Model Railroad	162
Figure 106: Reference lines for roads and railroads	163
Figure 107: Railroad switches	164
Figure 108: Railroad stations.....	168
Figure 109: UML Model Station	169

Table Directory

Table 1: Units	9
Table 2: Rules for using modal verbs	10
Table 3: Typographical conventions	11
Table 4: Attributes of the OpenDRIVE element	16
Table 5: Attributes of the header element.....	17
Table 6: Attributes of the planView element.....	33
Table 7: Attributes of the spiral element	35
Table 8: Attributes of the arc element.....	36
Table 9: Attributes of the poly3 element	40
Table 10: Attributes of the paramPoly3 element	44
Table 11: Attributes of the road element.....	48
Table 12: Attributes of the road link predecessorSuccessor element.....	52
Table 13: Attributes of the road type element	53
Table 14: Attributes of the road type speed element.....	54
Table 15: Attributes of the elevation element.....	55
Table 16: Attributes of the superelevation element.....	57
Table 17: Attributes of the shape element	60
Table 18: Attributes of the road surface CRG element	66
Table 19: Attributes of the laneSection element	75
Table 20: Attributes of the laneSection center lane element	75
Table 21: Attributes of the laneSection left lane element	76
Table 22: Attributes of the laneSection right lane element	76
Table 23: Attributes of the laneOffset element.....	77
Table 24: Attributes of the predecessor successor elements	80
Table 25: Attributes of the lane width element	82
Table 26: Attributes of the lane border element	84
Table 27: Attributes of the lane type	89
Table 28: Attributes of the lane material element.....	90
Table 29: Attributes of the speed element	91
Table 30: Attributes of the access element	93
Table 31: Attributes of the lane height element.....	95
Table 32: Attributes of the roadMark element	98
Table 33: Attributes of the roadMark line element.....	99
Table 34: Attributes of the roadMark type element	101
Table 35: Attributes of the roadMark explicit line element.....	102
Table 36: Attributes of the object laneValidity element	102
Table 37: Attributes of the sway element.....	103
Table 38: Attributes of the lane rule element	104
Table 39: Attributes of the junction element.....	107
Table 40: Attributes of the junction connection element.....	108
Table 41: Attributes of the lane linkage of a connection element	108
Table 42: Attributes of the priority element	111
Table 43: Attributes of a CRG element within a surface element	113
Table 44: Attributes of the road surface CRG mode element.....	113
Table 45: Attributes of the road surface CRG purpose	113
Table 46: Attributes of junction types.....	115
Table 47: Attributes of the junctionGroup element	120
Table 48: Attributes of the junction reference element	120
Table 49: Attributes of the junction controller element	121
Table 50: Attributes of the road objects element.....	126
Table 51: Attributes of the repeat element for objects.....	129
Table 52: Attributes of the outline element.....	131

Table 53: Attributes of the cornerRoad element	133
Table 54: Attributes of the cornerLocal element	134
Table 55: Attributes of the material element within an object	135
Table 56: Attributes of the parkingSpace element	137
Table 57: Attributes of the marking element	139
Table 58: Attributes of the cornerReference element.....	139
Table 59: Attributes of the border element.....	141
Table 60: Attributes of the objectReference element	142
Table 61: Attributes of the tunnel element	143
Table 62: Attributes of the bridge element	145
Table 63: Attributes of the signal element.....	149
Table 64: Attributes of the dependency element.....	152
Table 65: Attributes of the reference element	154
Table 66: Attributes of the positionInertial element	156
Table 67: Attributes of the positionRoad element	157
Table 68: Attributes of the signalReference element	159
Table 69: Attributes of the controller element of a signal	161
Table 70: Attributes of the control element	161
Table 71: Attributes of the switch element	164
Table 72: Attributes of the mainTrack element	165
Table 73: Attributes of the sideTrack element.....	166
Table 74: Attributes of the partner element.....	167
Table 75: Attributes of the station element.....	169
Table 76: Attributes of the platform element	170
Table 77: Attributes of the segment element	171



Association for Standardization of
Automation and Measuring Systems

E-mail: support@asam.net

Web: www.asam.net

© by ASAM e.V., 2020