4.1.1 Road network geometry

4.1.1.1 Identification

Name : Road network geometry
Identifier : ViaNova.Dbdd.ConceptualSchema.RoadNetworkGeometry

4.1.1.2 Description

[1] This schema describes the topology and geometry of generic road networks (e.g. motorway, carriageway or lane networks) in terms of a non-planar directed graph.

Note1 : Two-way roads may be modelled as two one-way roads.
Note2 : As roads may intersect at different levels, the graph is non-planar.

[2] The topological characteristics are modelled by means of edges and nodes. The geometric aspects by means of points and curves.

[3] The scheme is based on [prENV12160]. Notable differences are:

1. Underscore character are removed from the identifiers and capitals are used for readability (so the identifiers can be used in ASN.1 modules as well)
2. The point entity does not refer to an external Direct_Position, but uses WGS84 in stead
3. The entity Spatial_View is omitted
4. The entities Route and Segment are added.

4.1.1.3 Conceptual schema

```
SCHEMA ROADNETWORKGEOMETRY;
(*
This schema specifies the topology and geometry of an abstract road network in
terms of a directed non-planar graph.
*)

  TYPE interpolationMethod = ENUMERATION OF
    (SHORTEST_WAY,
     CIRCULAR_ARC,
     B_SPLINE,
     CLOTHOID);
  END_TYPE;

  TYPE direction = REAL;
    WHERE
      MinMaxDirection : {-PI < Self <= PI};
  END_TYPE;

  TYPE microdegrees = INTEGER;
    WHERE
      MinMax  : {-360000000 <= SELF <= 360000000};
  END_TYPE;

  TYPE decimetre = INTEGER;
  END_TYPE;

  ENTITY GeometricPrimitive
    ABSTRACT SUPERTYPE OF (ONEOF(Point, Curve));
  END_ENTITY;

  ENTITY Point
    SUBTYPE OF(GeometricPrimitive);
      latitude  : microdegrees;
      longitude : microdegrees;
      height    : OPTIONAL decimetre;
    WHERE
      MinMaxLatitude  : {-90000000 <= latitude <= 90000000};
      MinMaxLongitude : {-180000000 <= longitude <= 180000000};
  END_ENTITY;

  ENTITY Curve
    SUBTYPE OF(GeometricPrimitive);
      terrainPosition            : LIST [2:?] OF Point;
      interpolation              : interpolationMethod;
      tangentDirectionAtFirstEnd : OPTIONAL direction;
      tangentDirectionAtLastEnd  : OPTIONAL direction;
```

```
      curveLength               : decimetre;
    DERIVE
      endsAt                    : Point :=
TerrainPosition[hiIndex(TerrainPosition)];
      startsAt                  : Point := TerrainPosition[1];
  END_ENTITY;

  ENTITY TopologicalPrimitive
    ABSTRACT SUPERTYPE OF (ONEOF(Node, Edge, Route));
  END_ENTITY;

  ENTITY Node
    ABSTRACT SUPERTYPE
    SUBTYPE OF(TopologicalPrimitive);
      terrainpoint : Point;
  END_ENTITY;

  ENTITY TerminatingNode
    ABSTRACT SUPERTYPE
    SUBTYPE OF(Node);
    INVERSE
      isStartNodeOf : SET OF Edge FOR startsAt;
      isEndNodeOf   : SET OF Edge FOR endsAt;
    WHERE
      StartsEndsSomeEdge : SIZEOF(isStartNodeOF) + SIZEOF(isEndNodeOf) > 0;
  END_ENTITY;

  ENTITY Edge
    ABSTRACT SUPERTYPE
    SUBTYPE OF(TopologicalPrimitive);
      startsAt       : TerminatingNode;
      endsAt         : TerminatingNode;
      continuingEdge : OPTIONAL Edge;
      terrainCurve   : Curve;
    INVERSE
      supports       : SET OF IntermediateNode FOR isCoincidentWith;
      previousEdge   : SET [0:1] OF Edge FOR continuingEdge;
    WHERE
      Continuity                  : exists (continuingEdge) <= (continuingEdge in
endsAt.isStartNodeOf);
      CoincidentContinuationPoints : exists (continuingEdge) <= (
                                     terrainCurve.endsAt =
continuingEdge.terrainCurve.startsAt
                                     );
  END_ENTITY;

  ENTITY IntermediateNode
    ABSTRACT SUPERTYPE
    SUBTYPE OF(Node);
      isCoincidentWith : Edge;
      distance         : decimetre;
  END_ENTITY;

  ENTITY Route
    ABSTRACT SUPERTYPE
    SUBTYPE OF(TopologicalPrimitive);
      startsAt   : IntermediateNode;
      endsAt     : IntermediateNode;
      alongEdges : LIST [1:?] OF Edge;
    WHERE
      StartsAtFirstEdge : alongEdges[1] = startsAt.isCoincidentWith;
      EndsAtLastEdge    : alongEdges[SIZEOF(alongEdges)] =
endsAt.isCoincidentWith;
      continuity        : RouteIsContinuous (alongEdges);
  END_ENTITY;

  ENTITY Segment
    ABSTRACT SUPERTYPE
    SUBTYPE OF(Route);
    WHERE
      IsStraightOnRoute : RouteIsStraightOn (alongEdges);
  END_ENTITY;

  ENTITY UpstreamRoutes;
      routes : SET [1:?] OF Route;
    DERIVE
      endsAt : IntermediateNode := routes[1].endsAt;
  END_ENTITY;

(*
Checks whether or not a list of edges is continouos
*)
```

```
  FUNCTION RouteIsContinuous
  (route : LIST [1:?] OF Edge) : Boolean;
    LOCAL ok : BOOLEAN := True; END_LOCAL;
    REPEAT i := 1 TO SIZEOF(route)-1;
      IF route[i].EndsAT :<>: route[i+1].StartsAt THEN ok := FALSE; END_IF;
      IF not ok THEN ESCAPE; END_IF;
    END_REPEAT;
    Return (ok);
  END_FUNCTION;

  FUNCTION RouteIsStraightOn
  (route : LIST [1:?] OF Edge) : Boolean;
    LOCAL ok : BOOLEAN := True; END_LOCAL;
    REPEAT i := 1 TO SIZEOF(route)-1;
      IF route[i].EndsAT :<>: route[i].ContinuingEdge.StartsAt THEN ok := FALSE;
END_IF;
      IF not ok THEN ESCAPE; END_IF;
    END_REPEAT;
    Return (ok);
  END_FUNCTION;

  FUNCTION EdgesAreConnected
  (EdgeSet : AGGREGATE OF Edge) : BOOLEAN;
  LOCAL
    Found : SET[0:?] OF Edge;
    ToDo  : SET[0:?] OF Edge;
    Tnode : TerminatingNode;
    Cedge : Edge;
  END_LOCAL;

  Found := [EdgeSet[1]];
  ToDo  := [EdgeSet[1]];

  REPEAT UNTIL sizeof(ToDo) = 0;

    Tnode := ToDo[1].startsAt;
    REPEAT i:= 1 to hiIndex (Tnode.isStartNodeOf);
      CEdge := Tnode.isStartNodeOf[i];
      IF (Cedge in EdgeSet) and (not (CEdge in Found)) THEN
        Found := Found + CEdge;
        ToDO  := ToDo  + CEdge;
      END_IF;
    END_REPEAT;

    REPEAT i:= 1 to hiIndex (Tnode.isEndNodeOf);
      CEdge := Tnode.isEndNodeOf[i];
      IF (Cedge in EdgeSet) and (not (CEdge in Found)) THEN
        Found := Found + CEdge;
        ToDO  := ToDo  + CEdge;
      END_IF;
    END_REPEAT;

    Tnode := ToDo[1].endsAt;
    REPEAT i:= 1 to hiIndex (Tnode.isStartNodeOf);
      CEdge := Tnode.isStartNodeOf[i];
      IF (Cedge in EdgeSet) and (not (CEdge in Found)) THEN
        Found := Found + CEdge;
        ToDO  := ToDo  + CEdge;
      END_IF;
    END_REPEAT;

    REPEAT i:= 1 to hiIndex (Tnode.isEndNodeOf);
      CEdge := Tnode.isEndNodeOf[i];
      IF (Cedge in EdgeSet) and (not (CEdge in Found)) THEN
        Found := Found + CEdge;
        ToDO  := ToDo  + CEdge;
      END_IF;
    END_REPEAT;

    ToDo := ToDo - ToDo[1];
  END_REPEAT;

  if SIZEOF (EdgeSet - Found) > 0 then
    return (FALSE);
  else
     return (TRUE);
  end_if;

  END_FUNCTION;

  FUNCTION SameRouteEndings
    (routes: SET [1:?] OF ROUTE):BOOLEAN;
```

```
    REPEAT i:=2 TO hiIndex (routes);
      IF routes[i].endsAT <> routes[1].endsAT THEN
        return (False);
      END_IF;
    END_REPEAT;
    RETURN(True);
  END_FUNCTION;

  FUNCTION Distance2Point
    (c: Curve; d: INTEGER):Point;
    LOCAL
       p : Point;
    END_LOCAL;
    (* calculate the point co-ordinates for the point on the curve at distance d
from Curve.startsAt *)
    RETURN(p);
  END_FUNCTION;

END_SCHEMA;
```

Schema 4-1 Road network geometry schema

4.1.1.4 Entity descriptions

The various entities together with some auxiliary definitions are described below.

Carrigeway Intermediate Node0, CwIntermediateNode0
A node on an edge in the carriageway network which may or may not coincide with a terminating node.

Curve
Bounded continuous 1-dimensional geometric primitive. [CEN prENV12160]

Edge
1-dimensional topological primitive being an oriented connection between two terminating nodes, which may be the same. [CEN prENV12160]
The continuing edge, if exists, is a next edge which can be reached without tuning right or left.
The terrain curve represents the centreline of the edge.

Geometric primitive
Description, partial or total, of the spatial aspects of an object by means of co-ordinates and mathematical functions. [CEN prENV12160]

Geometry
Metric spatial aspects of Geographic Information [CEN prENV12160].

Intermediate node
A node coincident with an edge without terminating it. [CEN prENV12160]
The distance is the distance along the terrainCurve of the edge from its startsAt point.

Node
0-dimensional topological primitive. [CEN prENV12160]
Within this context each node is a connected node, i.e. a node related to one or more edges.

Point
A 0-dimensional geometric primitive. [CEN prENV12160]
The geodetic position of a node is described in WGS geodetic co-ordinates [WGS].

RdCoordinates
The cartesian coordinates of a point with respect to the Dutch Stereographic Grid (New Numbering), called 'Rijksdriehoek coördinaten' in dutch.

Route
A route through a network. A route may encompass one or more edges and starts and ends with an intermediate node on the first and last edge respectively.

Segment
A route for which each non-first edge is the continuing edge of its predecessor.

Terminating node
A node terminating an edge. [CEN prENV12160]

Topological primitive
Description, partial or total, of the topological aspects of an object. [CEN prENV12160]

Topology
Non-metric discrete spatial aspects of Geographic Information. [CEN prENV12160]

Upstream Routes
A set of routes all ending in the same intermediate node.

Wgs Geodetic Coordinates
Geodetic coordinates of a point with respect to the WGS84 coordinate system (see [WGS84]) The attribute latitude represents the geodetic latitude and the attribute longitude the geodetic longitude (see CEN287011]). The height is ellipsoidal height.

4.1.1.5 Type descriptions

The various types together with some auxiliary definitions are described below.

Decimetre
One tenth of a metre.

Note1 : The decimetre is also used in GDF as a unit measurement. So GDF information can be obtained without loss of accuracy.

Note: As 40.000 km = 40 million metres = 400 million decimetres < 2.000 million, any distance on earth can be represented by a four bytes integer if measured in decimetres. At the same time the decimetre is more then accurate for all known traffic applications.

Direction
An angle measured in radials defining the direction of a straight curve.

Ellipsoidal height (h)
The distance of a point from the geodetic ellipsoid measured along the perpendicular to the ellipsoid at that point, with the heigth of points outside the ellipsoid being treated as positive [CEN287011], called geodetic height (h) in [WGS84].

Microdegrees
One millions of a degree.

Note1 : The unit microdegrees is also used by GDF and ILOCS uses 10 microdegrees as its unit of measurement. So information on angles can be obtained form both sources without loss of accuracy.

Note2: As 360 million is less then 2.000 million, any value between -360 and 360 degrees can be represented by a four byte integer if measured in microdegrees.

Interpolation method
Defines the different allowed interpolation methods

4.1.1.6 Integrity constraints

[1] A route shall be continuous, each end of a road edge within a route coincides with the start of its successor on the route. This is formally described by the function RouteIsContinuous.

[2] A segment shall be a route on which the successor for each road edge on the route shall be the continuing edge of its predecessor. This is formally described by the function RouteIsStraightOn

[3] The carriageway edges of a motorway edge or node shall form a connected graph. This is formally described by the function EdgesAreConnected

[4] A set of upstream routes shall all end at the same point. This is formally described by the function SameRouteEndings

4.1.1.7 Other security aspects

There are no additional security requirements

4.1.1.8 Performance aspects

There are no additional performance requirement

4.1.1.9 Implementation aspects

There are no additional implementation requirements.

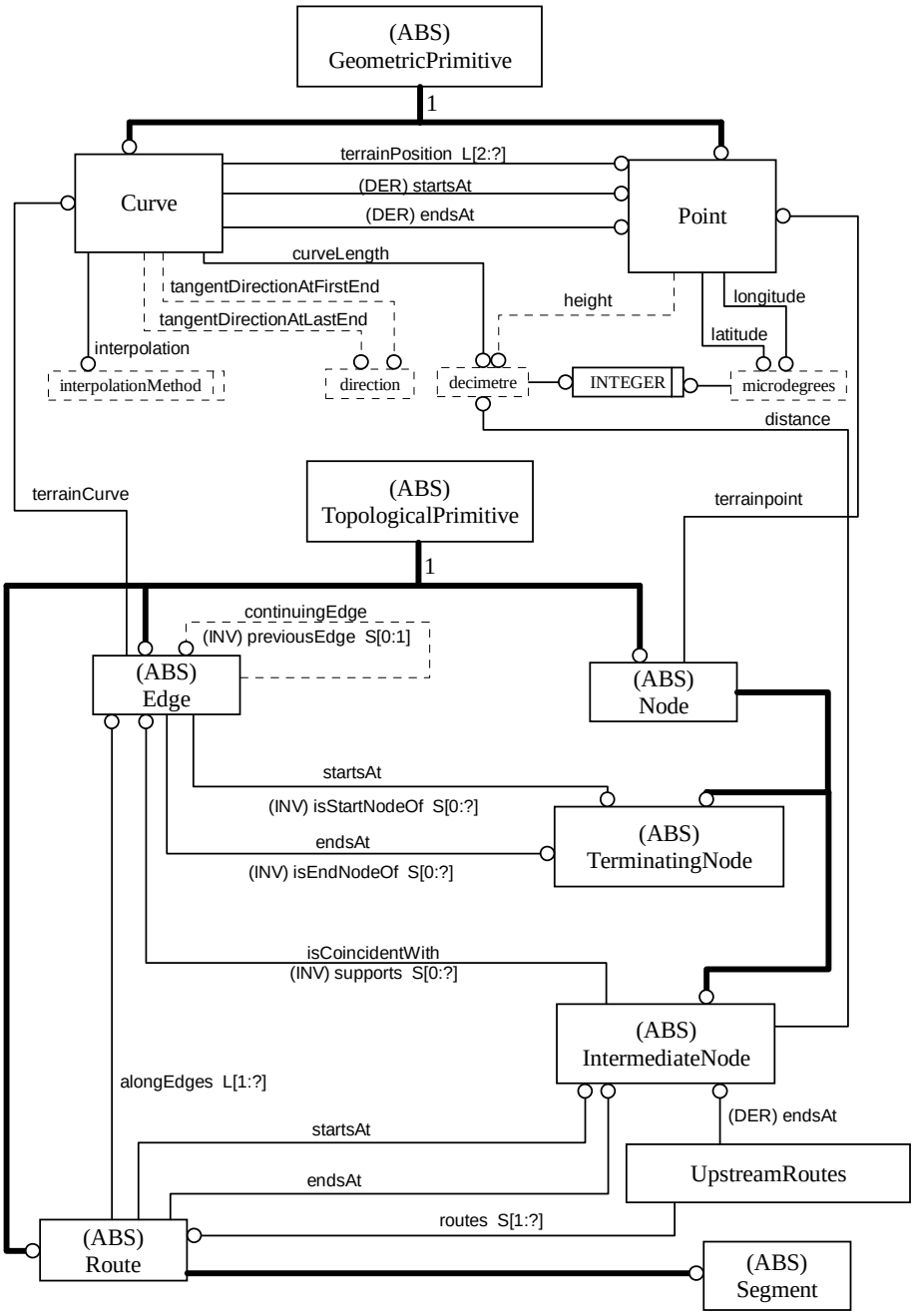
4.1.1.10 Abbreviations

CEN	Comité Européen de Normalisation (European Committee for Standardization)
GDF	Geographic Data File
ILOCS	Intersection LOCationS
prENV	European Prestandard
WGS	World Geodetic System

4.1.1.11 References

See section 2 for a complete reference description.

[prENV12160]	Geographic Information - Data Description - Spatial schema November 1996
[CEN287011]	prENV 287011:1995 Geographic Information - Referencing - Position CEN/TC287/WG4
[GDF]	…
[ILOCS]	…
[WGS84]	MIL-STD-2401 US Department of Defence World Geodetic System (WGS) 11 January 1994


(ABS)
GeometricPrimitive
1
Curve
terrainPosition L[2:?]
(DER) startsAt
(DER) endsAt
Point
curveLength
tangentDirectionAtFirstEnd
tangentDirectionAtLastEnd
height
longitude
latitude
interpolation
interpolationMethod
direction
decimetre
INTEGER
microdegrees
distance
terrainCurve
(ABS)
TopologicalPrimitive
terrainpoint
1
continuingEdge
(INV) previousEdge S[0:1]
(ABS)
Edge
(ABS)
Node
startsAt
(INV) isStartNodeOf S[0:?]
endsAt
(INV) isEndNodeOf S[0:?]
(ABS)
TerminatingNode
isCoincidentWith
(INV) supports S[0:?]
(ABS)
IntermediateNode
alongEdges L[1:?]
(DER) endsAt
startsAt
endsAt
UpstreamRoutes
routes S[1:?]
(ABS)
Route
(ABS)
Segment