TECHNICAL SPECIFICATION

ISO/TS 20452

First edition
2007-06-15

Requirements and Logical Data Model for a Physical Storage Format (PSF) and an Application Program Interface (API) and Logical Data Organization for PSF used in Intelligent Transport Systems (ITS) Database Technology

Exigences et modèle de données logiques pour un format de stockage physique (PSF), une interface de programme d'application (API) et une organisation de données logiques pour un PSF utilisé dans la technologie de base de données des systèmes de transport intelligents (ITS)

Reference number
ISO/TS 20452:2007(E)

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

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

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

In other circumstances, particularly when there is an urgent market requirement for such documents, a technical committee may decide to publish other types of normative document:

— an ISO Publicly Available Specification (ISO/PAS) represents an agreement between technical experts in an ISO working group and is accepted for publication if it is approved by more than 50 % of the members of the parent committee casting a vote;

— an ISO Technical Specification (ISO/TS) represents an agreement between the members of a technical committee and is accepted for publication if it is approved by 2/3 of the members of the committee casting a vote.

An ISO/PAS or ISO/TS is reviewed after three years in order to decide whether it will be confirmed for a further three years, revised to become an International Standard, or withdrawn. If the ISO/PAS or ISO/TS is confirmed, it is reviewed again after a further three years, at which time it must either be transformed into an International Standard or be withdrawn.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO/TS 20452 was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*.

Introduction

ISO/NP 14826, *Physical Storage for TICS Database Technology*, was introduced into ISO/TC 204 with the objective of standardizing a physical storage format (PSF) for navigation map data and related information stored on physical media used by in-vehicle navigation systems. The intent was to facilitate an interoperable in-vehicle navigation market environment by developing a standard PSF that would enable navigation media offered by different providers to be used by any navigation system and navigation systems made by any developer to be able to read the same media.

There was widespread international participation in this effort. Many of the different companies within the different participating national delegations possessed their own respective formats[1)] that were commercially available. It was decided early on that since none of these existing formats would be adopted wholesale as the standard physical storage format, the functional requirements of these existing systems would be submitted and consolidated into a universal set and organized into the major categories of application functionality predominantly used by in-vehicle navigation systems.

This gathering of market-driven requirements was the first step of an agreed development process that would proceed according to a top-down development approach. A sequential work plan was defined which included a logical data model based on the requirements, followed by the development of a logical organization of the data types used in the model. This logical data organization (LDO) would be used as a basis for the definition of a physical data organization (PDO), which would be defined to a sufficient level of granularity to specify a single standard PSF.

It took several years to develop and gain consensus on the requirements, the logical data model, and the logical data organization. During the development there were several input documents submitted by various national delegations. At the beginning of the development of the PDO it was decided to use a Japanese PDO input document[2)] as a framework for the PDO discussion.

Shortly after the PDO discussion began, the project ISO/NP 14826 expired and there was not sufficient international support for resubmitting a new work item proposal to continue the work, nor was there consensus that the PDO work could be finished within an acceptable time frame. Consequently, a standard PSF as envisioned within the scope of the work item would not be realized.

However, the requirements, logical data model, and logical data organization documents developed in this process reflect international consensus and still provide value for the navigation system market and other emerging products and services which use navigation map data. Thus it was agreed to convert these documents into a Technical Specification which could be used for future developments.

This Technical Specification can help developers of applications that use map databases to realize efficiencies by providing guidelines on setting up an appropriate architecture for navigation systems. This provides a potential benefit to the developer's ability to develop systems in a shorter timeframe, thereby shortening time-to-market for products. Although this Technical Specification was originally developed for navigation system applications, it may also facilitate other market development activities by providing insight into solving common data modelling and organization issues in the fields of telematics and location-based services.

1) These formats are identified in the Bibliography of this Technical Specification.

2) Kiwi Format Specification version 1.2.2 (see Bibliography).

Requirements and Logical Data Model for a Physical Storage Format (PSF) and an Application Program Interface (API) and Logical Data Organization for PSF used in Intelligent Transport Systems (ITS) Database Technology

1 Scope

This Technical Specification describes the functional requirements and Logical Data Model for PSF and API and the Logical Data Organization for PSF that were completed under ISO/NP 14826. It does not specify a Physical Data Organization.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 14825, *Intelligent transport systems — Geographic Data Files (GDF) — Overall data specification*

3 Terms and definitions

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

3.1
Address Location
application category that deals with the task of expressing a real-world position in terms of the PSF data representation

NOTE Address Location is one of the six application categories supported by the PSF and the API.

3.2
address type
attribute of road section entity, specifying the type of house number ranges

EXAMPLE distinction between base address, county address, commercial address, etc., or no address.

3.3
application category
basic sub-function within the set of functionality for vehicle navigation and traveller information system applications

NOTE This Technical Specification identifies six application categories: Positioning, Route Planning, Route Guidance, Map Display, Address Location, Services and POI Information Access.

3.4
Application Program Interface
API
⟨ISO context⟩ specification interface and set of function calls between application software and data access libraries of vehicle navigation systems

3.5
base map
the whole of all transportation elements and all services, including their relationships to transportation elements

3.6
Branded Third Party Data
BTPD
information about services which is supplied by third party data providers (e.g. tourist or motoring organizations) who may impose proprietary restrictions on the use and presentation of the data

NOTE 1 Access to BTPD is subject to authorization and licensing.

NOTE 2 BTPD is a sub-set of Third Party Data (TPD).

3.7
cartographic feature
data model entity that represents geometrical information for display purposes, having non-explicit topology and 0-, 1- and 2-dimensional types

EXAMPLES Display Point, Polyline and Polygon.

3.8
cartographic text
data model entity that stores name text that is associated with all or part of a cartographic feature

NOTE It is language-dependent and can contain a suggested display location, orientation, language code, priority (or importance), suggested scale range, and bounding box.

3.9
condition
information related to link(s) which is composed of condition type, condition modifiers and condition scope

3.10
crossroad
data model entity that represents the single instance of the crossing of two named navigable features; it relates to the set of links and nodes which comprise the crossing, and to the crossing of the navigable features to a place

3.11
display point
0-dimensional type of cartographic feature

3.12
dummy point
non-required entity that represents a position along a link where the link crosses a parcel boundary and does not necessarily coincide with a shape point or node

3.13
geocoding
determination of a link or node based on address information describing and/or naming a location

3.14
intersection
GDF level 2 representation of a crossing which bounds a road or a ferry as a complex feature composed of one or more GDF level 1 junctions, road elements and enclosed traffic areas

3.15
junction
data model entity that represents a navigable feature which is either a named GDF junction or named GDF intersection, and that relates a named navigable feature to a set of links and nodes and a place

3.16
landmark
point, line or area feature that can be used to clarify the directions generated to describe a route

NOTE 1 It can be associated to a node or a link.

NOTE 2 A *landmark* cannot be in the *Services, Administrative Areas*, or *Public Transportation Feature* themes of the GDF; however a facility in which a s*ervice* is located can be a *landmark.*

3.17
layer
sub-set of map data resulting from a subdivision of data of the same coverage area based on contents (similar to ISO-GDF layer) and which is typically related to one or only a few of the application categories

EXAMPLE Route guidance data can be considered as one layer.

3.18
level
sub-set of map data resulting from classification of data of the same semantically contents based on the level of details/density, related to the concept of different map scales

NOTE *Level 0* is considered the lowest level (greatest detail); higher levels are numbered *level 1, level 2*, etc.

EXAMPLE Map display data can be organized into 6 levels representing different zoom scales.

3.19
link
directed topological connection between two nodes, composed of an ordered sequence of one or more *segments* and represented by an ordered sequence of zero or more *shape points*

3.20
Map Display
application category that deals with graphical information presentation

NOTE Map Display is one of the six application categories supported by the PSF and the API.

3.21
multilink
ordered aggregation of links which are at the same level, connected in sequence, share the same functional classification, form of way, direction of travel, and perhaps additional PSF-builder-specified characteristics, such that each link is contained in exactly one multilink

3.22
navigable feature name
data model entity that represents the name for the transportation element, including GDF road element, GDF ferry connection, GDF junction, GDF intersection

NOTE It is related to places, crossroads, junctions and road sections.

3.23
node
data model entity for a topological junction of two or more links or end bounding a link

NOTE A *link* stores the coordinate value of the corresponding GDF junction.

3.24
parcel
database partitioning unit, corresponding to a certain coverage area and associated with one *level* and containing data of one or more *layers*

NOTE 1 A *parcel* contains (at least) all *nodes* with positions enclosed by or located on the outline of its coverage area plus (parts of) all *links* attached to these *nodes.*

NOTE 2 It can be partitioned such that the amount of data of one *parcel* is nearly the same as that of another.

3.25
place
named area which can be used as part of address location

3.26
place class
attribute of place entity, classifying data into highest administrative or geographic division, administrative sub-division, postal, or colloquial (such as regions or neighbourhoods)

NOTE It is partially ordered as “place class A is below place class B” (does not imply strict or complete containment).

3.27
place level
level associated with places of place classification “administrative sub-division”

NOTE Higher/lower level situations are constituted by the occurrence of a parent/child place relationship between places.

3.28
place relationship
bivalent relationship between place entities, constituting the place tree, linking parent and child places (“place A is in place B”)

NOTE 1 It does not imply strict or complete containment.

NOTE 2 It is attributed as: address significant, official, postal, or useful for reverse geocoding.

3.29
Point of Interest
POI
destination and/or site of interest to travellers, usually non-commercial by nature

3.30
polygon
2-dimensional type of cartographic feature

3.31
polyline
1-dimensional type of cartographic feature

3.32
Positioning
application category that deals with the determination of vehicle location and map-matching

NOTE Positioning is one of the six application categories supported by the PSF and the API.

3.33
postal code
data model entity for a government-designated code used for specified regions for addressing

NOTE It is related to link, navigable feature name, place, and POI.

3.34
rectangle
unit of geographic space, defined by two parallels of min/max latitude and by two meridians of min/max longitude, that represents the coverage area of the map data enclosed by or located on the outline of the *rectangle*

3.35
regular parcel
parcel shaped like a rectangle

NOTE Regular parcels on the same generalization level are not intended to overlap.

3.36
reverse geocoding
determination of the address description of a link or node (i.e. determination of an upwards path across the place tree)

3.37
road
GDF level 2 feature composed of one, many or no road elements and joining two intersections, serving as the smallest independent unit of a road network at GDF level 2

3.38
Road Element Side
RES
basic component of the road section entity that represents the left or right side of a link, and corresponds to one or more unique combinations of a navigable feature and a house number range

3.39
road section
data model entity that represents the house number ranges of both sides of a street that carries a navigable feature name

NOTE It corresponds to a link (ID).

3.40
Route Guidance
application category that deals with the generation of graphical, textual, and/or audio instructions for following a planned route

NOTE Route Guidance is one of the six application categories supported by the PSF and the API.

3.41
Route Planning
application category that deals with the determination of routes between specified points

NOTE Route Planning is one of the six application categories supported by the PSF and the API.

3.42
segment
straight section of a link connecting either two successive shape points, or a shape point and a node, or two nodes in case the link does not contain shape points

3.43
service
data model entity for a commercial activity of interest to travellers as a destination and/or orientation that is associated with road element(s), by which it can be accessed, and place(s)

NOTE 1 *Service* is further described by attributes including (at least) name and type; it can be associated with other *services* by parent/child relationships (many to many).

NOTE 2 *Service* is used synonymously with *POI* within the logical data model.

3.44
service attribute
descriptive information of a service

3.45
Services and POI Information Access
application category that deals with the provision of POI information to the navigation application

NOTE Services and POI Information Access is one of the six application categories supported by the PSF and the API.

3.46
shape point
position along a link used to more accurately represent its geometric course, bounded by exactly two segments

3.47
signpost
data model entity for a directional sign that represents a logical relationship between signpost information and two associated links where the first link (mandatory) represents the road element along which the signpost is located, and the second link (optional) is the first road element which directs exclusively to the destination indicated on the signpost

NOTE The position of the signpost along the link and the link direction the signpost is facing are also stored.

3.48
SuperLink
aggregation of linearly connected regular links present in the lowest level as a simplified representation of the road network in higher levels

3.49
symbol
data model entity that represents an icon associated with a cartographic feature

3.50
Third Party Data
TPD
additional descriptive and editorial information about services which is typically supplied by third party data providers (e.g. tourist or motoring organizations)

3.51
traffic location
data model entity that contains an external reference (e.g. VICS or RDS-TMC) and is linked to either place or transportation entities

3.52
transportation element
any feature from the *Roads and Ferries* feature theme of the GDF

4 Symbols and abbreviated terms

4.1 Abbreviations

AL	Address Location
API	Application Program Interface
BTPD	Branded Third Party Data (subset of Third Party Data – TPD)
DAL	Data Access Libraries
DBD	Detailed Background Data
DRD	Detailed Road Data
GDF	Geographic Data Files
ITRF	International Terrestrial Reference Frame
LDO	Logical Data Organization
LDM	Logical Data Model
LiQ	Location in Question
MD	Map Display
MI	Metadata Information
PDO	Physical Data Organization
POI	Point(s) of Interest
PSF	Physical Storage Format (the ISO entity defined by this Technical Specification)
RDS-TMC	Radio Data System-Traffic Message Channel
RES	Road Element Side
RP	Route Planning
TI	Traffic Information
TPD	Third Party Data
VICS	Vehicle Information and Communication System
WGS84	World Geodetic System 1984

4.2 Syntax notation used in data model diagrams

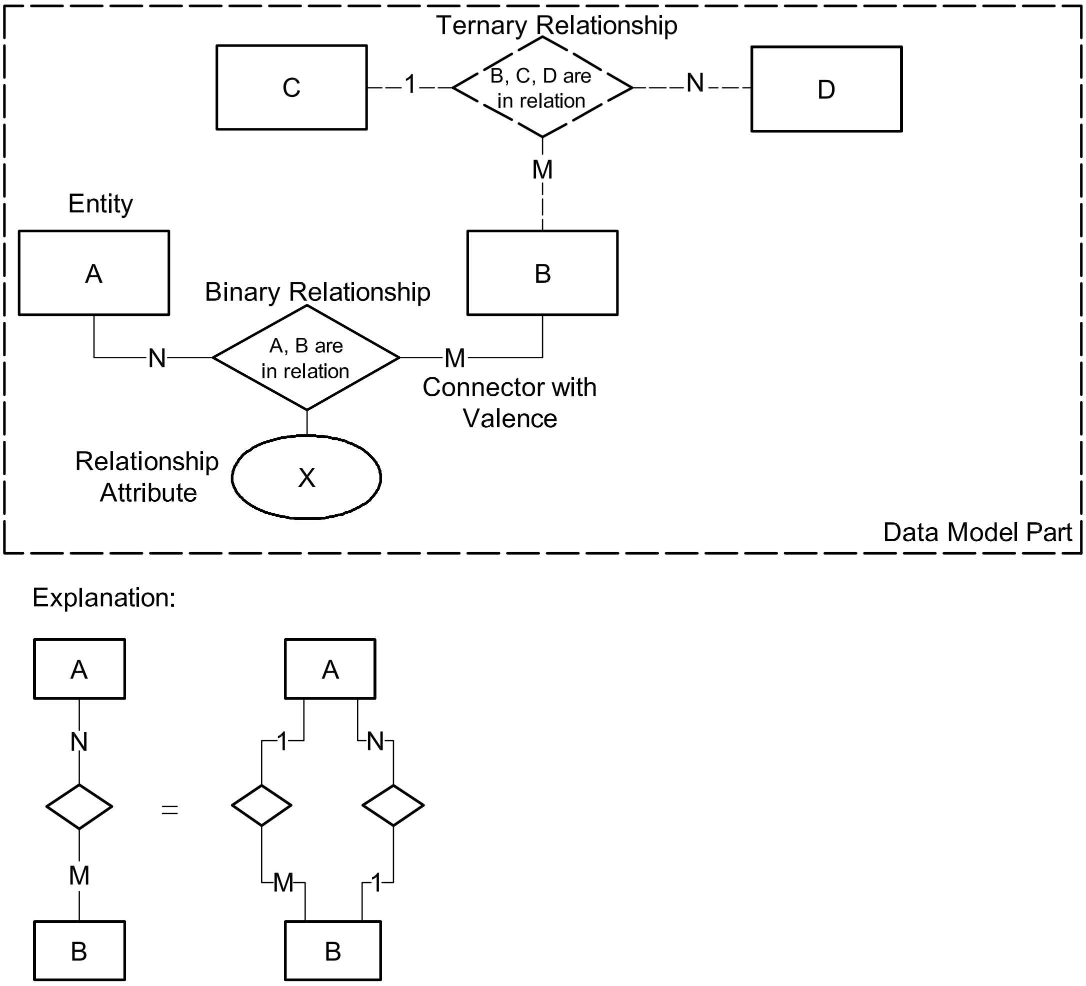


Figure 1 — Example for Data Model Notation

There are six components:

- Entity (any logical data entity);
- Binary Relationship (any relation between two entities);
- Ternary Relationship (any relation between three entities);
- Connector (notation component to connect entities with a relationship valence qualified by a number);
- Relationship Attribute (qualifies a relationship in more detail);
- Data Model Part (a conceptual sub-unit of the whole model).

In the example in Figure 1, there is an N-to-M binary relationship involving A and B, and qualified by relationship attribute X. Valences (the N and M in an N-to-M relationship, or the 1 and N in a one-to-N relationship) are interpreted as follows: An N-to-M binary relationship is always an abbreviation for two parallel relationships, as shown in Figure 1. This means that an A can correspond to multiple B's (hence the "M" on B), and in addition that a B can correspond to multiple A's (hence the "N" on A). When a variable like "N" or "M" is used, there is no implication that every A corresponds to the same number of B's or that every B corresponds to the same number of A's. Two different variables, N and M, are used to further imply that there is no correspondence between the number of B's per A and the number of A's per B. The existence of an N-to-M relationship in the Logical Data Model also does not imply that functionality will be supplied to follow the relationship in both directions. That is, it is not necessarily true that there need be both a function to find the B's corresponding to a given A and the A's corresponding to a given B. If there is a function to find the B's that correspond to a given A, then the 1-to-M relationship from A to B indicates that the function, given an A, might return multiple B's, and the N-to-1 relationship from A to B indicates that a given B might be in the lists returned for multiple distinct A's in separate calls.

Whether a relationship is characterised as 1-to-1, 1-to-N, or N-to-M, it is possible that a given individual entity may not participate in the relationship, or may participate in the relationship with fewer than the maximum possible number of corresponding entities. For example, there is a 1-to-N relationship between Links and Shape Points. This indicates that one Link can have multiple Shape Points, but that a given Shape Point can correspond to only one Link. However, a given Link might have no Shape Points, despite the multiplicity implied by "N". In order to keep the notation simple, we will consider this to be implicit, rather than writing expressions like "{0, 1}-to-{0, 1, …, N}".

The interpretation of ternary relationships and their valences is similar. A ternary relationship is a relationship among three entities. An example is shown as an M-to-1-to-N relationship among B, C, and D. This is interpreted as follows: There are correspondences between triples (b, c, d) of individual B's, C's, and D's. The valence M on B indicates that, if one selects a particular c0 from among the C's and a particular d0 from among the D's, there can be multiple B's that correspond to c0 and d0. There is no implication that different choices of c0 and d0 will correspond to the same number of B's. Similarly, the valence N on D indicates that, if one selects a particular b0 from among the B's and a particular c0 from among the C's, there can be multiple D's that correspond to b0 and c0. Again, there is no implication that different choices of b0 and c0 will correspond to the same number of D's. Different variables N and M are used for B and D in order to further imply that there is no correspondence between the number of B's corresponding to a fixed C and a fixed D and the number of D's corresponding to a fixed B and a fixed C. Finally, the valence 1 on C indicates that, if one selects a particular b0 from among the B's and a particular d0 from among the D's, there can be at most one C that corresponds to b0 and d0.

As in the case of binary relationships, the existence of a ternary relationship, regardless of its valences, does not imply that functionality will be supplied to follow the relationship in all three possible directions. Of the three possible functions in this example (given a B and a C, return the corresponding D's; given a B and a D, return the corresponding C; given a C and a D, return the corresponding B's), one, two, or all three functions may be defined. Suppose, for example, that only the first of those functions is defined. Then the correct interpretation of the valences is as follows: The "N" on D means that for a given B and a given C, the function might return multiple D's. The "M" on B means that, for a fixed c0 among the C's and a fixed d0 among the D's, there might be multiple B's such that the function, called with them and with c0, will return d0 among its list of D's. Finally, the "1" on C means that, for a fixed b0 among the B's and a fixed d0 among the D's, there is at most one C such that calling the function with b0 and with that C will return d0 among its list of D's.

Binary relationships shall be drawn in solid lines, while ternary relationships shall be drawn in dashed lines.

5 Application categories

5.1 Positioning

5.1.1 General

The Positioning function is used to determine location, for example latitude and longitude of a road network entity and for Map Matching. Map Matching is the method of determining where the navigation system has moved in the road network based on the navigation system's previous location and data about the navigation system's motion from external inputs.

5.1.2 Functional description

"Positioning" seeks a position and orientation of a navigation system relative to the transportation network with respect to the map data representing the real world. An application may dynamically determine the navigation system's current position while the navigation system is in motion. Map Matching can continue "in the background" even while other functions are being performed so the navigation system always "knows where it is". Map Matching algorithms are beyond the scope of this document.

For the purpose of positioning, the following functions shall be provided:

a) a single set of coordinates for an application-specified *Point Feature* in the *Roads and Ferries* theme;

b) the set of *Edges, Nodes* and/or *Intermediate Points* for an application-specified *Feature* or set of connected *Features* in the *Roads and Ferries* theme;

c) the set of topologically connected *Features* in the *Roads and Ferries* Theme connected to an application-specified *Feature* in the *Roads and Ferries* theme;

d) a single set of coordinates for an application-specified *Line Feature* in the *Roads and Ferries* theme and application-specified percentage of the distance along the *Feature*;

e) the set of *Features, Edges, Nodes* and/or *Intermediate Points* in the *Roads and Ferries* theme within an application-specified rectangle;

f) positioning related Attributes, Conditions and Relationships (i.e. Prohibited Maneuvers, Direction of Traffic Flow) for an application-specified Feature in the *Roads and Ferries* theme;

g) the entry and exit angles for the set of *Transportation Elements* connected to an application-specified *Intersection* or *Junction*;

Additionally,

h) the API shall allow a pre-fetch area of interest to be specified by a rectangle for retrieving Positioning data;

i) the specification shall support a single, world-wide, latitude/longitude-based coordinate reference system. The International Terrestrial Reference Frame (ITRF) is chosen because it is maintained by an international body. It is considered equivalent to WGS84 because the two systems currently have less than 1 m difference;

j) in datasets of Japan and Korea the Tokyo datum shall also be allowed because of the prevalent use of the Tokyo datum in existing navigation applications in Japan and Korea;

k) only one coordinate system can be used in a single piece of storage media;

l) identification of the coordinate system used in the PSF shall be by means of a meta-data identifier;

m) the API shall allow retrieval of the coordinate system identifier;

n) the API shall allow retrieval of coordinates in the coordinate system which is stored in the media.

5.1.3 Interaction of Positioning and other Application Categories

Positioning may provide position information to other applications which perform the following tasks:

a) when an application tracks progress along the route and provides manoeuver instructions at appropriate points to the end-user;

b) when an application determines whether the navigation system has left the planned route;

c) when an application calculates a route to the requested destination from the navigation system's current position;

d) when an application scrolls the displayed map;

e) when an application selects services by geographic proximity;

f) when an application is displaying the navigation system's position on a map;

g) when an application displays a map around a location relative to the navigation system's current position;

h) positioning may receive planned route information from the Route Planning application for use in Map Matching.

5.1.4 Requirements for Logical Data Model

The Logical Data Model is required to support at least the data identified in the Function Description. Other requirements for the Logical Data Model are defined below.

a) Only access to the lowest level of data is required.

b) Only access to the data represented in the *Roads and Ferries* theme is required.

c) Positioning data shall be organized into parcels.

d) In order to minimize the number of parcels accessed, any link crossing into a parcel, with or without a *node* or *intermediate point* in that parcel, shall be represented in that parcel.

e) In order to allow fast spatial access to parcels, parcels shall be accessed by their bounding rectangles. The shapes of parcels on the lowest level shall not overlap.

5.1.5 Metadata

Access to metadata is required in order to identify the geodetic system of the coordinate data in the database.

5.1.6 Extended Parcel Exposing Functions

For the requirements of this subclause, see 5.2.7.

5.2 Route Planning

5.2.1 General

The Route Planning function is used to determine routes from one user-specified location to another.

5.2.2 Functional description

Navigation applications may calculate routes based on attributes of the transportation network. Applications may allow end-users to specify criteria for the route such as "shortest distance", "no highways", etc. As a basic operation, a user indicates a departure position, which could be the navigation system's current position, and selects a destination (place to go) and possibly one or more waypoints. A suitable route is then calculated. Routing is not limited to automobile transportation only. This function supports routing via any mode represented in the database. This may include rail and water ferries, taxis, and routes only accessible by bicycle or foot. Other forms of public transportation may be considered in the future.

The route calculation algorithms are outside the scope of this functional description.

To improve data access speed, the Logical Data Organization groups transportation features into levels. The higher levels contain only the more significant features (e.g. highways and main roads). These may be aggregated. Correspondences between features at different levels shall be made available to the application. The functions specified in the requirements below allow selection by level.

The Route Planning application provides the following methods of accessing data that can be used for routing:

a) via the set of topologically connected *Links* for an application-specified *Link* at an application-specified level;

b) via routing-related Attributes for an application-specified Transportation Element or set of connected Transportation Elements, such as: Node Coordinates (of the bounding Nodes of a Link), Measured Length, Functional Road Class, Number Of Lanes, Average Speed, Divided Road Element, Form Of Way, as well as access characteristics, Conditions, and other Relationships;

c) via navigation attributes for *Roads* and *Intersections*;

d) via corresponding *Link* for an application-specified *Link* at an application-specified different level;

e) via a set of topologically connected *GDF Roads* for an application specified *GDF Road* at an application specified level at certain levels to be determined;

f) via a set of GDF Road Elements and GDF Junctions, which comprise a GDF Road or GDF Intersection;

g) via the GDF Road or GDF Intersection for an application-specified GDF Road Element or GDF Junction;

h) via the corresponding entity representing a GDF *Junction or Intersection* for an application-specified entity representing a GDF *Junction or Intersection* at an application-specified different level;

i) via effective time or date periods for turn, travel, or other Conditions;

j) via location references which are stored in the database for an application-specified set of *Transportation Elements*;

k) via a set of *Transportation Elements* for an application-specified location reference, which is stored in the database;

l) via the entry and exit angles for the set of *Links* connected to an application-specified *Intersection* or *Junction;*

m) via historic and forecast traffic conditions, incidents, and events information for a specified *Transportation Element* or set of *Transportation Elements*;

n) via a DAL capable of providing transparent access to static and dynamic traffic information. It shall not preclude or require the integration of dynamic traffic information from external systems;

o) via an API allowing a pre-fetch area of interest specified by feature ID or rectangle for retrieving Route Planning data at an application-specified level.

5.2.3 Interaction of Route Planning and other application categories

The Route Planning application can interact with other application categories as follows.

a) The Route Planning application accepts other information from the Positioning application when calculating a route to the requested destination from the navigation system's current position.

b) The Route Planning application provides information about the planned route to the Positioning application when determining whether the navigation system has left the planned route.

c) The Route Planning application provides information about the planned route to the Route Guidance application for generating driving instructions.

d) The Route Planning application provides information about the planned route to the Services and POI Information Access application for geographic selection of services with proximity to the planned route.

e) The Route Planning application accepts input from the Services and POI Information Access and Address Location application when determining end-points or way-points for a route.

f) The Route Planning application provides information about the planned route to the Map Display Application when indicating the course of the planned route on the graphical map display.

5.2.4 Requirements for Logical Data Model

The logical data model is required to support at least the data identified in the function description. Other requirements for the logical data model are defined below.

a) Only access to the data represented in the *Roads and Ferries* theme is required. Enclosed traffic areas shall be represented by links and nodes.

b) The shape of a parcel on a given level shall be contained in the shape of exactly one parcel at a higher level. The shapes of parcels on the same level shall not overlap.

c) For route planning data, references to parcels on the same level and on the level(s) above and below are required.

d) In order to have optimally filled parcels, parcels may have different coverage sizes.

e) For route planning data, no intermediate points are required for the representation of links. A representation of turn angles, link length and the link cost are required.

f) There is no requirement to create an additional node where a link crosses a parcel boundary.

g) For route planning data, links crossing a parcel boundary should be stored as a whole in those parcels where they are connected to other links in the same parcel.

h) In order to have fast access to parcels, parcels shall be accessed by their bounding rectangles.

i) A separate computation is required to find nodes or links in the network data corresponding to origin, intermediate and destination points. The manner in which the nodes or links are found is outside the scope of this Technical Specification.

The logical data model entities used in Route Planning are described below. Entities of these types are stored in parcels and uniquely identified with database IDs.

5.2.5 Logical Data Model Entities

The logical data model entities used in Route Planning are described below in alphabetical order.

— **Complex Intersection**

The Complex Intersection entity is used to store information about a generalized road crossing involving Links and Nodes associated with intersection-internal elements such as dual-carriageway sections, slip lanes, and roundabouts. This information includes navigation attributes. The Complex Intersection corresponds to the concept of GDF Intersections.

— **Complex Road**

The Complex Road entity is used to store information about a (generalized) topological connection between Complex Intersections, including navigation attributes. It corresponds to the concept of GDF Roads.

— **Condition**

The Condition entity is used to store information about restrictions or additional attributes associated with a maneuver. This information includes a condition type which defines the type of information stored in the record as well as modifier fields that provide additional information about the condition. Conditions can specify information for a single link or for a series of connected links referred to as a maneuver. For example, a toll cost condition is defined in terms of a single link, whereas, a prohibited driving maneuver condition is defined in terms of two or more connected links.

The Condition entity also contains scope information to specify the lanes of the link that are affected by the condition and the types of vehicles that are affected by the condition. The Condition entity may also include a *Validity Period* which defines the time during which the condition is in effect.

— **Link**

The Link entity is used to store information about *Road Elements* and other Transportation Elements. A Transportation Element is any *Road Element, Road, Ferry* or *Ferry Connection*. This information includes the identifiers for the end nodes of the link as well as a collection of navigation attributes. The navigation attributes provide all of the information necessary for an application to determine a weighting factor for the link during route calculation.

There are two types of link entities: Regular Links and SuperLinks.

- Regular Links are the fundamental units of the navigable road network. They contain two (and only two) nodes, one at each end;
- at route calculation levels above level 0, some Links may be aggregated into SuperLinks. The purpose of the SuperLinks is to provide a simplified representation of the road network which can help speed up route calculation. A SuperLink represents a set of linearly connected links with sufficiently similar characteristics.

— **Node**

The Node entity constitutes connectivity within the transportation network. It represents the topological junction between two or more links, or the end-bound of a link. The Node entity stores the coordinate value of the corresponding junction.

5.2.6 Metadata

The API will allow the retrieval of the number of Route Planning functional levels for a dataset, and provide information as to which roads are contained in each level.

5.2.7 Extended Parcel Exposing Functions

The Extended Parcel Exposing Functions shall be as follows.

a) For a given level, rectangle, and entity type (e.g. link), retrieve a list of parcel IDs representing parcels that contain elements of the given type and are lying completely or partially inside the rectangle.

b) For a given parcel ID, retrieve the parcel information, that is: level; bounding box; data size; availability in cache; list of overlapping parcels at the next lower level; and list of overlapping or abutting parcels at the same level.

c) For a given parcel ID and entity type, retrieve all the entities of that type that lie within the parcel.

d) For a given parcel ID, notify the DAL that the application is no longer using the parcel.

e) For a given parcel ID and given ID of a boundary element (e.g. a boundary node or boundary link), retrieve a list of overlapping or abutting parcels at the same level to which the boundary element connects.

f) For an entity's ID, retrieve the parcel ID of the parcel in which the element is stored.

g) If it is decided that some entities with the same geometry can be contained in more than one parcel at the same level, then there shall be a function that, for a given entity's ID (e.g. a node's ID or a link's ID), retrieves a list of IDs describing the same geometry in another parcel at the same level.

5.3 Route Guidance

5.3.1 General

The Route Guidance function is used to generate instructions for following a route.

5.3.2 Functional description

The Route Guidance function generates step-by-step instructions for following a route. These instructions may include compass heading, distance, road names, sign text, Landmarks, and still or motion images. These instructions may also include maneuver details such as turn angle, merges, and road name changes. Route Guidance may be given using text, voice or graphics.

Route Guidance provides the following methods of accessing data used for guidance of a route:

a) via guidance-relevant features and relationships related to an application-specified Transportation Element, or set of Transportation Elements, such as: intersecting *Road Elements*, *Signpost Information*, Conditions and Landmarks along the Transportation Element;

b) via guidance attributes for an application-specified Transportation Element, or set of Transportation Elements, such as: *Road Names, Length*, *Direction Of Traffic Flow,* and *Form Of Way*;

c) via indicating whether an application-specified *Junction* is a part or all of an *Intersection*;

d) via indicating whether an application-specified link is a regular link, a superlink or part of a superlink;

e) via connected Transportation Elements for an application-specified Junction;

f) via connected Transportation Elements for an application-specified *Intersection* which are not part of that Intersection, such as the set of *Road Elements* connected to a Roundabout;

g) via component Transportation Elements of an application-specified Intersection, such as the set of *Road Elements* and *Junctions* which make up a roundabout;

h) via data about the transition from an application-specified Link to an application-specified series of connected Links, such as: the existence of a tollbooth or gate;

i) via node and Intermediate Point positions for *Line Features* to support displaying manoeuvre "arrows" for route guidance;

j) via cartographic data for the *Line Features* comprising an intersection for the derivation of intersection schematics;

k) via entry and exit angles for the set of *Transportation Elements* connected to an application-specified *Intersection* or *Junction*;

l) via phonetic strings in an application-specified language for pronunciation of any named entity in the database;

m) via phonetic strings in an application-specified language for commonly used guidance words;

n) via digitized pronunciation data in an application-specified language for commonly used guidance words;

o) via API allowing a prefetch area of interest specified by a rectangle for retrieving Route Guidance data;

p) via image data for optional picture guidance.

5.3.3 Route Guidance and Positioning Interaction

The Route Guidance application may receive input from other applications to perform the following functions:

a) provide guidance while the navigation system is in motion, based on a calculated route and map matching;

b) track progress along the route and provide manoeuver instructions at appropriate points to the end-user.

5.3.4 Map Display and Route Guidance Interaction

The application may highlight the point on a displayed map of a particular routing manoeuvre (from a calculated route).

5.3.5 Requirements for Logical Data Model

The Logical Data Model is required to support at least the data identified in the function description. Other requirements/recommendations for the Logical Data Model are given below.

a) Route Guidance data shall be organized into Parcels.

b) In order to allow fast access to Parcels, Parcels shall be accessed by their bounding rectangles.

c) In order to have optimally filled Parcels, Parcels may have different coverage area sizes.

The Logical Data Model entities used in Route Guidance are described below.

5.3.6 Data Model Entities

The logical data model entities for Route Guidance are described below in alphabetical order.

— **Complex Intersection** (see 5.2.5)

— **Condition** (see 5.2.5)

— **Landmark**

The Landmark entity is used to associate a Link or Node with any *Point, Line,* or *Area* features that can be used to clarify the directions that are generated to describe a route. The Landmark entity contains the feature ID of the *Point, Line* or *Area Feature* that is located along a Link or at the Node and information about the location of the feature in relation to the Link or Node. Landmarks are not features from the *Services, Administrative Areas* or *Public Transportation* Feature themes. However, a facility in which a service is located may be a landmark.

— **Link** (see 5.2.5)

— **Node** (see 5.2.5)

— **Signpost**

The Signpost entity provides a logical relationship between two Links and the actual signpost information associated with the two links. The first link (mandatory component) represents the Road Element along which the signpost is located. The position of the signpost along the link and the Link direction the signpost is facing is also stored. The second link (optional component) is associated with the destination specified on the Signpost. It is the first Road Element which directs exclusively to the destination indicated on the signpost (e.g. a city). The Signpost entity contains the Signpost Content attribute.

— **Signpost Content** (an attribute)

This attribute provides a description of the content of signposts, e.g. geographic names, road numbers, directional arrows, pictograms (such as the airport symbol), etc. The attribute corresponds to a specific combination of two links: the link related to the signpost location and the link to which the signpost information directs.

A physical signpost may contain multiple instances of the signpost content attribute for different directions and different languages.

5.3.7 Metadata

The API shall provide access to the following Metadata information:

a) which character set is used to represent the data;

b) which phoneme set is used for the phonetic data;

c) which binary format is used for digitized sounds;

d) which languages are “supported” in the database;

e) on which side of the road traffic travels in a particular region;

f) which file format(s) is used for still and/or motion image data.

5.3.8 Extended Parcel Exposing Functions

For the requirements of this subclause, refer to 5.2.7.

5.4 Map Display

5.4.1 General

The Map Display function is used to display a map of a specified geographic area. An application may display maps to the end-user. The application may also accept end-user input that references the map display (such as from a point and click device).

5.4.2 Functional description

An application may display *Points Features, Lines Features, Areas Features*, Cartographic Text and Symbols for a specified geographic area. This may include roads, physical features, administrative boundaries, and names for all of these. Text and symbols can be positioned on a display to annotate this map.

The Map Display function provides cartographic data that can be used to display a map of any application-specified arbitrarily-oriented rectangle in the database. The data consist of the following database entities to support a variety of map drawing styles: Cartographic Features, Cartographic Text and Symbols.

The application may allow the map to be zoomed in or out. The application may display different levels of detail on a map display based on the zoom level. The application may allow the map to be rotated and scrolled. When scrolling, if detailed data are not available, the application may automatically zoom the map out to a level where data are available. The application may allow the end-user to access additional information by selecting objects on the display. The application may display multiple windows. Generating map images and managing displays are beyond the scope of this function.

To facilitate data access speed, this application groups cartographic data into *level*s. The higher *level*s contain only the more significant cartographic features. The set of cartographic data is also selectable by *level*.

Map Display provides the following methods of accessing data:

a) via the Cartographic Features, Cartographic Text and Symbols for an application-specified rectangle, *level* and *Feature* type;

b) via the coordinates for application-specified Cartographic Features;

c) via attributes for Cartographic Features, such as: feature type, name, and functional classification;

d) via the complete or partial Cartographic Features associated with application-specified Transportation Elements;

e) via the area (size) of an application-specified *Area Feature*;

f) via the ability to retrieve additional information for *Point, Line* and *Area Features* which are associated with Cartographic Features which may have been selected from the displayed map;

g) via the Cartographic Text associated with a Cartographic Feature;

h) via the Symbol associated with a Cartographic Feature;

i) via returning the Cartographic Features and Cartographic Text in “draw-order”. For example, if water is drawn before bridges, water features should be returned before bridge features.

j) When no map data are available for an area requested by a function, the function may distinguish between the case of “off the map” and the case of “no data at this location at this *level* ”.

k) The API shall allow a pre-fetch area of interest to be specified by a rectangle and application specified *level* for retrieving Map Display data.

5.4.3 Interaction between Map Display and other Application Categories

There are interactions in both directions between Map Display and other application categories.

5.4.3.1 Other inputs used for Map Display

An application may receive inputs for Map Display and other Application Categories to perform the following:

a) display a map at the current location of the navigation system;

b) display a marker on the map indicating the navigation system’s current location;

c) scroll the displayed map as the navigation system moves, maintaining the position of the marker indicating the navigation system’s current location;

d) display a map at any location selected by the end-user. The end-user may specify the location as an address, an intersection, a service, or by cursor position on the display;

e) provide latitude and longitude, street address, and other information for a point indicated by the cursor on the display;

f) highlight a route on the display;

g) highlight the point on a displayed map of a particular routing manoeuver.

5.4.3.2 Services provided by Map Display for other Application Categories

This function provides the following methods of accessing data that can be used for other application categories:

a) via cartographic data for selected line features so that other application categories (e.g. Route Guidance) may highlight a route and/or individual line features of a route. The cartographic data include: display class, speed limit, number of lanes, line feature attributes, and intermediate points;

b) via information for *Features* selected on a displayed map to other application categories (e.g. to select a destination for Route Planning).

5.4.4 Requirements for Logical Data Model

The logical data model is required to support at least the data identified in the function description. Other requirements for the logical data model are given below.

a) Several *levels* of data are required for cartographic data, corresponding to different map scale ranges. At the higher *level*s, the drawing detail for line features and area features is generalized.

b) Access is required for data from all GDF Feature Themes, as well as attributes and conditions.

c) Map display data shall be organized into Parcels.

d) In order to allow easy identification of Parcels, Parcels shall be rectangular.

e) Links crossing Parcel boundaries shall be cut at the Parcel boundary.

f) In order to minimize the number of Parcels accessed, any link crossing into a parcel, with or without a *node* or *intermediate point* in that Parcel, shall be represented in that parcel.

5.4.5 Logical Data Model Entities

The following are the requirements for logical data model entities for the Map Display application category.

— **Cartographic Feature**

The Cartographic Feature entity consists of three types – the Display Point, Polyline, and Polygon entities.

- The **Display Point** type is used to represent Services, Signposts and other point features such as toll booths. Depending on the level of generalization, a Display Point may also be used to represent an *Area Feature*.

- The **Polyline** type is used to represent *Line Features* such as *Road Elements* and *Railway Elements*. A cartographic polyline does not necessarily correspond to a single *Road Element* or *Line Feature*. Depending on the level of generalization, a Polyline may also be used to represent an *Area Feature*. For Map Display data, topological connectivity is not relevant. One cartographic Polyline can correspond to many *Line Features*, as shown in Figure 2 below.

- The **Polygon** type is used to represent *Area Features* such as parks and lakes. To aid in polygon filling, these positions are returned in fixed order for the outer boundary of the polygon and any enclaves.

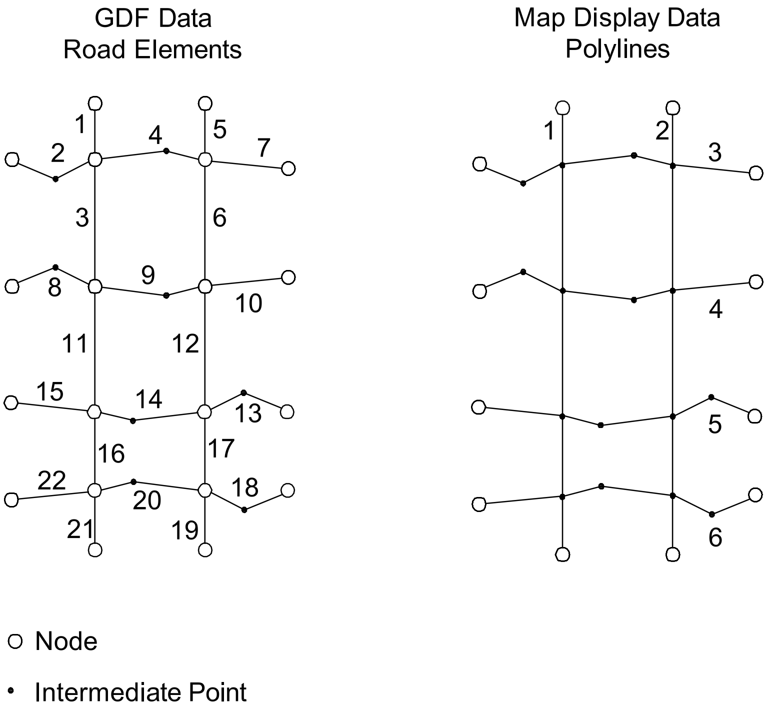
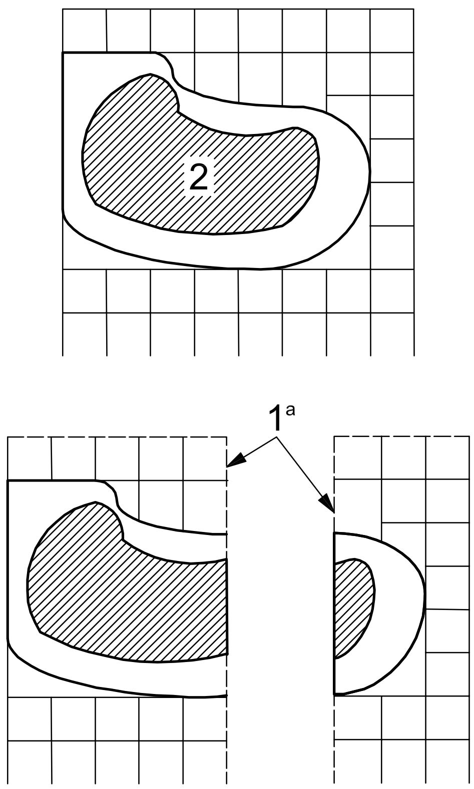


Figure 2 — Example of polyline representation

An *Area Feature* may be represented as multiple Polygons. A *Line Feature* may be represented as multiple Polylines. If a feature is split, new points are created and, for polygons, new boundary lines are added to close it (see cartographic feature polygon example in Figure 3).



Key

1 boundary

2 lake

[a] Lake is split and closed.

Figure 3 — Representation of a split area feature

Each of these virtual entities contains attributes that are specific to that particular type of entity. These attributes provide the application with the information it needs to determine the entity's display characteristics (e.g., positions, line width, color, fill pattern, icon, etc.).

— **Cartographic Text**

The Cartographic Text entity is used to store the name text that is associated with all or part of a cartographic feature. In addition to the text, this entity may contain a suggested location, orientation, language code, priority (or importance), suggested scale ranges, and bounding box that can be used to position the text on the displayed map. Cartographic Text entities are language dependent and different Cartographic Text entities can be associtated to the same Cartographic Feature for different languages.

There is a many-to-many relationship between Cartographic Text entities and Cartographic Features.

— **Symbol**

A symbol is a graphic associated with a cartographic feature.

5.4.6 Metadata

Metadata fulfil the following functions:

a) provide the number of cartographic *level*s defined in a given dataset;

b) provide scale factor range which is appropriate for each cartographic *level*;

c) provide information about the contents of each *level*.

5.4.7 Extended Parcel Exposing Functions

For the requirements of this subclause, refer to 5.2.7.

5.5 Address Location

5.5.1 General

This function is used to access data that are used to determine positions, both on the earth and in the map data representation of the earth.

5.5.2 Functional description

Address Location is the determination of a location based on information describing or naming the location. An application may determine locations based on various types of information. For example, this information can be an address or cross-streets. There are also two basic methods of address location:

— *Geocoding*: determining a link or a node, or a polygon or representative point by its address description.

— *Reverse geocoding*: determining an address description of a link or node or representative point or area.

End users or applications may not know the complete specification of a location. For example, they may not know the complete address and administrative area, or they may not know whether a street is a "street" or an "avenue". They may need to search the database based on the information they do know, and examine a set of locations matching their criteria.

Address Location may support different entry orders by means of appropriate data structures. Typically, a hierarchichal top-down entry order may be used. However, permutations thereof shall be supported (e.g. street name first).

Address Location may support extensions to the search criteria when no match is found. The user may demand an expanded search area (e.g. areas close to the specified place(s) or spelling tolerance for similarly pronounced/written names).

5.5.3 General requirements

5.5.3.1 Country/Language dependencies

For any search that queries names (street/place/intersection/junction), return the language code and the name-type attributes (official or alternative name) for the name.

Language codes may be specified when retrieving names.

5.5.3.2 Official abbreviations

The PSF and the API shall support access to full names as well as official abbreviations of names.

5.5.4 Functional requirements

5.5.4.1 Place searches

This function provides the following methods for accessing places.

a) For a specified place and specified child place classes: return all children meeting the specifications with their place classes and relationships. This search method may be further restricted by spatially qualifying a specified rectangle. This search method may also be restricted by qualifying a specified partial or complete representation of a name with an exact or phonetic match.

b) For a specified place and specified parent place classes: return all parents meeting the specifications with their place classes and relationships. This search method may also be restricted by qualifying a specified partial or complete representation of a name with an exact or phonetic match.

c) For a specified place class and a partially spelled place name and, optionally, a specified parent place: return the set of unique next successor characters for all places which match that partially spelled place name of that class within that parent place.

d) For a specified place: return a latitude/longitude rectangle which bounds that place.

e) For a specified place and navigable feature name: return all the child places with that navigable feature name.

f) For a specified place: retrieve an external location reference.

g) For a specified external location reference: retrieve a place.

h) For a specified class of places in the address tree (e.g. municipalities, settlements, postal places): return all places in the specified class. The Address Location model shall allow for this function without the necessity to join separate place tables in the tree. Those classes of places which can be returned as a single class are defined by metadata.

i) For a specified place: return the attributes of that place.

j) For a specified rectangle and specified place class: return all places of the specified class within the rectangle.

k) For a specified place: return a polygonal cartographic feature representing the place.

l) For a specified place: return a representative point in the interior of the place.

Support for methods k) and l) above is optional by place class and by geographic region on the part of the data supplier. However, the API and the PSF shall support methods k) and l), and a DAL shall support the data if it is present.

5.5.4.2 Street name searches

For a specified list of places, return a list of street names for all road elements in the union of all the places. For every street name in the list, provide the relevant place information that is necessary to identify the street name uniquely within the specified list of places. The following restrictions can be put on this search:

a) as above, qualified spatially by a specified rectangle;

b) as above, qualified by a specified partial or complete phonetic representation of the street names;

c) as above, qualified by a specified partial or full spelling match to the beginning of the street name or to appropriate individual words within the street name. For example, “John F. Kennedy Street” can also be found when searching by “Kennedy”;

d) for a specified list of places and a specified street name, return the list of house number ranges for road elements having the specified street name within the union of specified places;

e) for a specified list of places and a specified street name, return the list of road elements having the specified street name within the union of specified places;

f) for a specified list of places and a specified street name, return the list of crossroads involving road elements which intersect the specified street name. The application can specify whether the crossroads are returned alphabetically by intersecting street name or sequentially. Sequential order means that the crossroads are returned in the order in which they are encountered in driving along the road, as far as possible, without specifying the order of various discontinuous lengths of road with the specified name;

g) for a partially spelled street name and, optionally, a specified place, return the set of unique immediate successor characters for all streets which match that partially spelled street name in the specified place;

h) for a specified rectangle, return all street names within the specified rectangle;

i) for a specified set of places and given a fully or partially spelled street name, return the street names, in alphabetical order, of road elements in the union of all the places starting with the street names that match the given partially spelled street name;

j) for a specified set of places and given a fully or partially spelled street name, return the street names, in alphabetical order, of road elements in the union of all the places preceding and including the street names that match the given partially spelled street name.

5.5.4.3 Intersection and junction name searches

For a specified list of places, return a list of intersections or junctions for all road elements in any of these places. The following restrictions can be put on this search:

a) as above, qualified spatially by a specified rectangle;

b) as above, qualified by a specified partial or complete phonetic representation of the intersection or junction names;

c) as above, qualified by the street name of a participating road element (e.g., Exit 50 of Interstate 4);

d) as above, qualified by a specified partial or full spelling match to the beginning of the intersection or junction name or to appropriate individual words within the intersection or junction name. For example, “John D. Rockefeller Plaza” can also be found when searching by “Rockefeller”;

e) for a partially spelled intersection name or junction name and, optionally, a specified place, return the set of unique immediate successor characters for all intersection names or junction names which match that partially spelled name in that place;

f) for a specified junction name or street name, return street names and junction names containing the name in a combined view, without the need for the application to join a street-name list and a junction-name list.

5.5.4.4 Postal code searches

This function provides the following methods for accessing postal codes.

a) For a specified postal code, return the list of road elements that have this postal code.

b) For a specified postal code, return the list of navigable feature names that contain this postal code.

c) For a specified postal code, return the list of places that contain this postal code.

d) For a specified place, return the list of postal codes that are valid for this place.

e) For a specified navigable feature name, return the list of postal codes that are valid for this navigable feature name.

f) For a specified road element, return the list of postal codes that are valid for this road element.

5.5.4.5 Specific point references

This function provides the following methods for the location of specific points:

a) via the line feature which matches a specified street name and house number in a specified place. If the specified street name, house number information, and place name combination is not unique, all matching locations are returned;

b) via the line features and point features for the intersection of road elements with a specified pair of street names in a specified place. If the pair of street names and places does not specify a unique point (e.g., the streets cross in more than one place), the data for all matching intersections are returned;

c) via the line features and point features for a specified named intersection or junction in a specified place. If the specified name and place are not unique, all matching intersections or junctions are returned;

d) via the set of line features for a specified place. For example, a block (banchi) in Japan;

e) via the set of line features and point features for a specified bounding rectangle.

5.5.4.6 Partly overlapping features at different levels

The PSF and the API shall support representation and access of the address location features (place features and street features), that are partly located in multiple places at a next higher level of the address hierarchy.

The PSF and API shall allow identification of the part of the above-mentioned address location feature that belongs to the place at the next higher level in the address hierachy, for those cases where it is address significant.

The following two cases are distinguished:

a) one of the above-mentioned Address Location features partly covering multiple places;

EXAMPLE El Camino Real, San Francisco Bay Area.

b) multiple Address Location features, all having the same name, each located in a different place.

The PSF shall support the described cases, although the availability of the data contents is optional. The determination of the rules for what represents a single Address Location feature is left to the data supplier.

The API shall offer access to the information and it has to be able to handle the optional availability of the information.

5.5.4.7 Disambiguation

5.5.4.7.1 Place names

Applications shall be able to identify places with ambiguous names. A place is considered to have an ambiguous name if any other child of its parents has the same name. An indication of the existence of an ambiguity shall be returned when the name of a place is ambiguous.

5.5.4.7.2 Street names

To the extent that street names exist in places, applications shall be able to identify all unique street names within a place, i.e. Road Elements shall be unambiguously associated with the place(s) (e.g. cities in the U.S.) within which they are contained. Typically, both sides of a Road Element are associated with the same place. However, in order to also allow for other cases where Road Elements form the border of places, Road Element Sides (RES) shall be introduced as the lowest level children in the model which carries the street name(s).

5.5.4.7.3 Crossroads

Applications shall be able to identify ambiguous crossroad situations where a street is intersected more than once by streets of the same name that may or may not be the same physical street. An indication of the existence of an ambiguity shall be returned when there are multiple crossings with the same pairs of names.

5.5.4.8 Direct access

When a user is stepping through successive lists of places, it is not difficult to determine which place class to use. When a user types input in “free form”, it may be difficult to determine which place classes are associated and have been used. In particular, the relative order of place classes in a user input string generally varies from country to country. Metadata are required that describe a set of place classes in the relative order used in that country. The metadata also indicate whether the place classes are required or optional. The GDF and the PSF shall enable the representation of these metadata. These metadata shall be carried to the application through the API.

5.5.5 Interactions between Address Location and other Application Categories

Address Location may receive input from other applications to perform the following tasks:

— *geocoding* of a service via its associated address provided by the Service and POI Information Access application;

— reverse *geocoding* of a Road Element Side (RES) closest to a position (coordinates) which is provided by “cursor input” through the Map Display application.

Address Location may provide the following method of accessing data that can be used for other application categories:

— provide place (see place search above) to the Service and POI Information Access application for the selection of a service.

5.5.6 Requirements for the Logical Data Model

5.5.6.1 General description

The logical data model is required to support at least the data identified in the functional description. Other requirements are defined below.

5.5.6.2 Logical Data Model entities

5.5.6.2.1 General description

The Logical Data Model entities for Address Location are described below.

The entities listed immediately below have been described in the clauses noted:

— **Link** (see Route Planning);

— **Node** (see Route Planning).

In addition to the entities listed above, the following entities are used for Address Location:

5.5.6.2.2 Road Section

5.5.6.2.2.1 General description

The Road Section entity provides a logical relationship between a link, a navigable feature name (representing a name associated with the link), a place, and a house number range for each Road Element Side (RES). Note that streets are not represented as completed wholes (e.g. a set of continuous links which share the same name), but through a series of Road Section entities which correspond to the same name.

A link may be associated with multiple Road Sections due to the following.

— A link may have multiple names (each corresponding to one or both RES).

— A link may have multiple house number ranges (each relating to different types of addresses).

The Road Section entity allows a house number range, or ranges, to be paired with any or all of the names associated with a link. In this way, each combination of a name and a house number range corresponds uniquely to an RES.

In addition to the link ID, the navigable feature name ID and the house number ranges for both RES, the Road Section entity contains the type of address as an attribute. See Figure 4 below, providing an example for a situation in the USA.

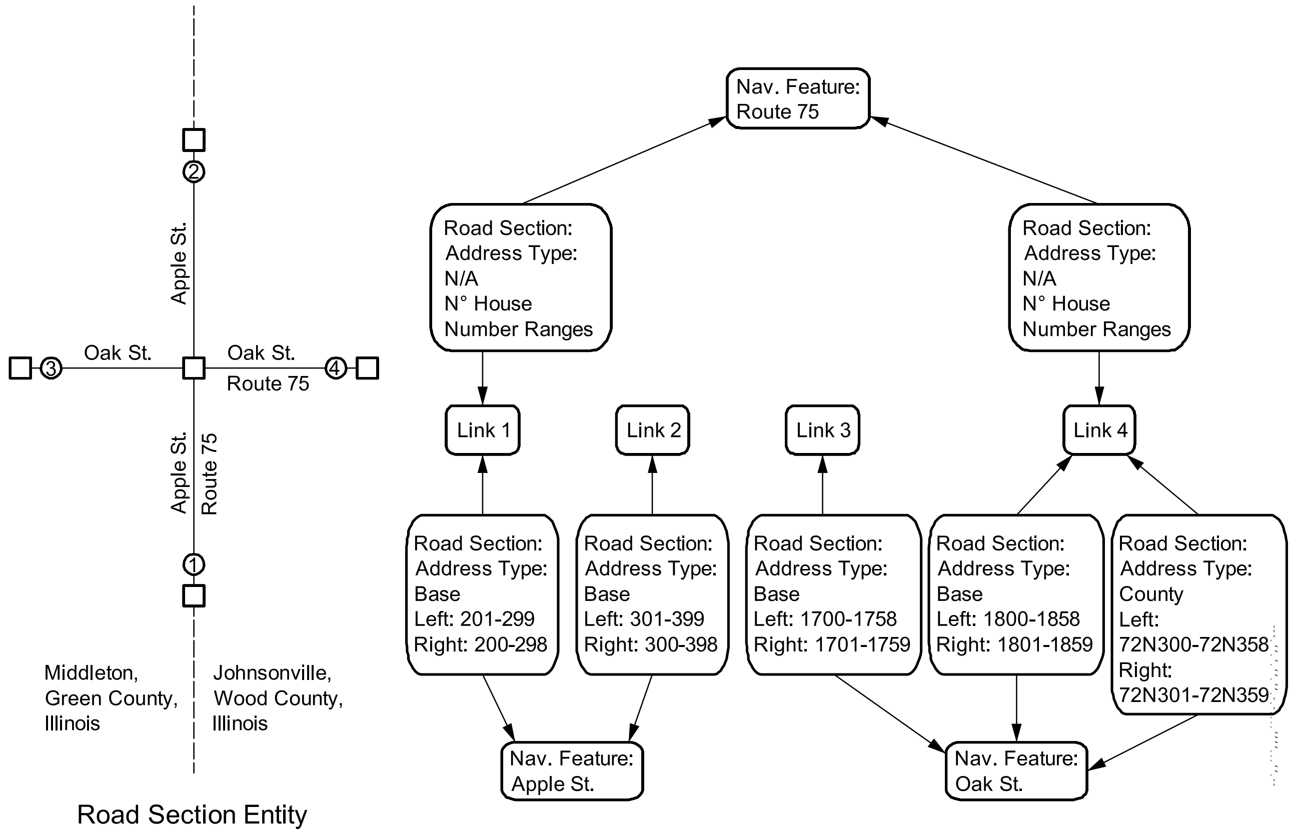


Figure 4 — USA Example of Road Sections

5.5.6.2.2.2 Address Type (an attribute)

This attribute provides a description of the type of house number information (e.g. base, county, commercial, etc.) that is associated with the Road Section and makes it unique.

5.5.6.2.3 Crossroad

The Crossroad entity is used to store information about a single instance of an intersection of two named navigable features. This information is primarily used for specifying the source or destination for a route. The Crossroad entity contains the navigable feature name IDs of the primary name and the intersecting navigable feature name. In addition, the Crossroad entity contains the location of the intersection in terms of a node ID. The Crossroad entity relates to the set of links and nodes which comprise the crossing. It also relates the crossing of the navigable features to a place.

5.5.6.2.4 Junction

The Junction entity provides a logical relationship between a node, a navigable feature name, a set of associated links, and a place. This entity is used to represent a named (GDF) junction or named (GDF) intersection, which occurs when a feature name is associated with a single intersection node or a node which generalizes a complex intersection. It is primarily used for specifying the source or destination for a route.

5.5.6.2.5 Navigable Feature Name

The navigable feature name entity is used to store information about a transportation element (e.g. GDF road element, GDF ferry element, GDF intersection, and GDF junction). This information includes a unique identifier for the feature and the name of the feature broken into the following component parts: directional prefix, street-type prefix, street-name body, trailing spaces, street-type suffix, and directional suffix, where applicable. This entity relates to Crossroads, Junctions, and Road Sections.

5.5.6.2.6 Place

5.5.6.2.6.1 General description

Places represent named areas which can be used during address location. Examples of places are: USA, Hollywood (colloquial, fuzzy), New York City, Borough of Queens, Neighborhood of Flushing, and Chicagoland.

The Place entity is used to store information about a named place. Named places can be categorized as administrative areas or districts. Administrative areas are governmental entities such as Country, State and City in the United States, or Country, Bundesland, Kreis, and Municipality in Germany. Districts are named places that are not administrative areas, such as neighborhoods or informally named regions (e.g., the San Francisco Bay Area).

Places are treated as separate entities in the logical data model rather than being treated as a link attribute because a link can be associated with a variable number of places. Also, the Place entity allows Services to be directly associated with a place rather than indirectly associated through a link. This aids in the ability to query services by place.

5.5.6.2.6.2 Place class (an attribute)

Places may be classified. Places are classified as highest administrative or geographic division (e.g. country, country group, or continent), administrative subdivision, Postal, or Colloquial (such as regions or neighborhoods). As described in the diagram, it is difficult to assign a “place level” to the administrative subdivision, because they are inconsistently applied. For example, a single place may be considered a 2nd place level subdivision in one area, and a similar subdivision may be a 3rd place level subdivision in another area. A place or place level needs an additional textual description defined in the metadata in order to describe, for example, the type of place level.

Place classes are partially ordered. That is, if place class A is below place class B, then places of classification A are contained in places of classification B. Note that this does not necessarily mean complete containment. If places of classification A are below places of classification B and places of classification B are below places of classification C, then places of classification A are below places of classification C. If places of classification A are below places of classification B, then places of classification B are not below places of classification A. A place of classification A cannot be below another place of classification A. It may be that places of classification A are not below places of classification B and places of classification B are not below places of classification A.

For three objects A, B, C:

— if A is subordinate to B, then B is not subordinate to A;

— if A is subordinate to B and B is subordinate to C, then A is subordinate to C;

— A is not subordinate to A;

— it may be that A is not subordinate to B and B is not subordinate to A.

5.5.6.2.6.3 Place relationships

These places are related to each other. Relationships can be defined as Place A is in Place B. This does not imply strict or complete containment. For example, New York City is in Queens County. Other parts of New York City can be in other counties.

These relationships are further attributed to describe their applicability in the address location application category. The following relationships are at least required:

— Address-significant, that is, they are relationships commonly used by people to describe successive stages of refinement in specifying a location. (For example, when selecting the next subordinate levels for a place selected from a menu.)

— Type (Official, postal, or other).

— Useful for reverse geocoding (for a particular user-category).

Road Elements (and other elements from the Transportation Network) may be related to one or more places. These relationships are also attributed as described above.

The combinations of these attributes and relationships determine the behavior of the address location function.

5.5.6.2.6.4 USA example for administrative structures and places

Based on the example in Figures 5 and 6, the following is a partial list of the features, attributes and relationships which could be included in a database. The examples shown pertain to the United States. Please note that the words in parentheses are not part of the definition, but only included for clarity.

- **Features**
  - Flushing is a Place
  - Queens (borough) is a Place
  - New York City is a Place
  - New York (state) is a Place
  - Queens County is a Place
- **Attributes**
  - Flushing is of Place Type Postal
  - Queens (borough) is of Place Type Administrative Subdivision
  - New York City is of Place Type Administrative Subdivision
  - New York (state) is of Place Type Administrative Subdivision
  - Queens (County) is of Place Type Administrative Subdivision
- **Relationship** [with attributes in square brackets]
  - New York City is in Queens
  - New York is in New York (state) [Address-Significant]
  - Queens (borough) is in Queens (county)
  - Flushing is in New York City [Address Significant, Useful for Reverse Geocoding]
  - Flushing is in Queens (borough) [Address-Significant]
  - Road Element is in Flushing [Address Significant, Useful for Reverse Geocoding]

- Road Element is in Queens (borough)
- Road Element is in New York City [Address Significant]

Also:

- **Features**
  - Rosemont is a Place
  - Des Plaines is a Place
- **Attributes**
  - Rosemont is of Place Type Administrative Subdivision
  - Des Plaines is of Place Type Administrative Subdivision
- **Relationship** [with attributes in square brackets]
  - RE 1 is in Des Plaines [Officially, Address Significant]
  - RE 2 is in Rosemont [Officially, Address Significant]
  - RE 2 is in Des Plaines [Postal]
  - RE 3 is in Rosemont [Officially]

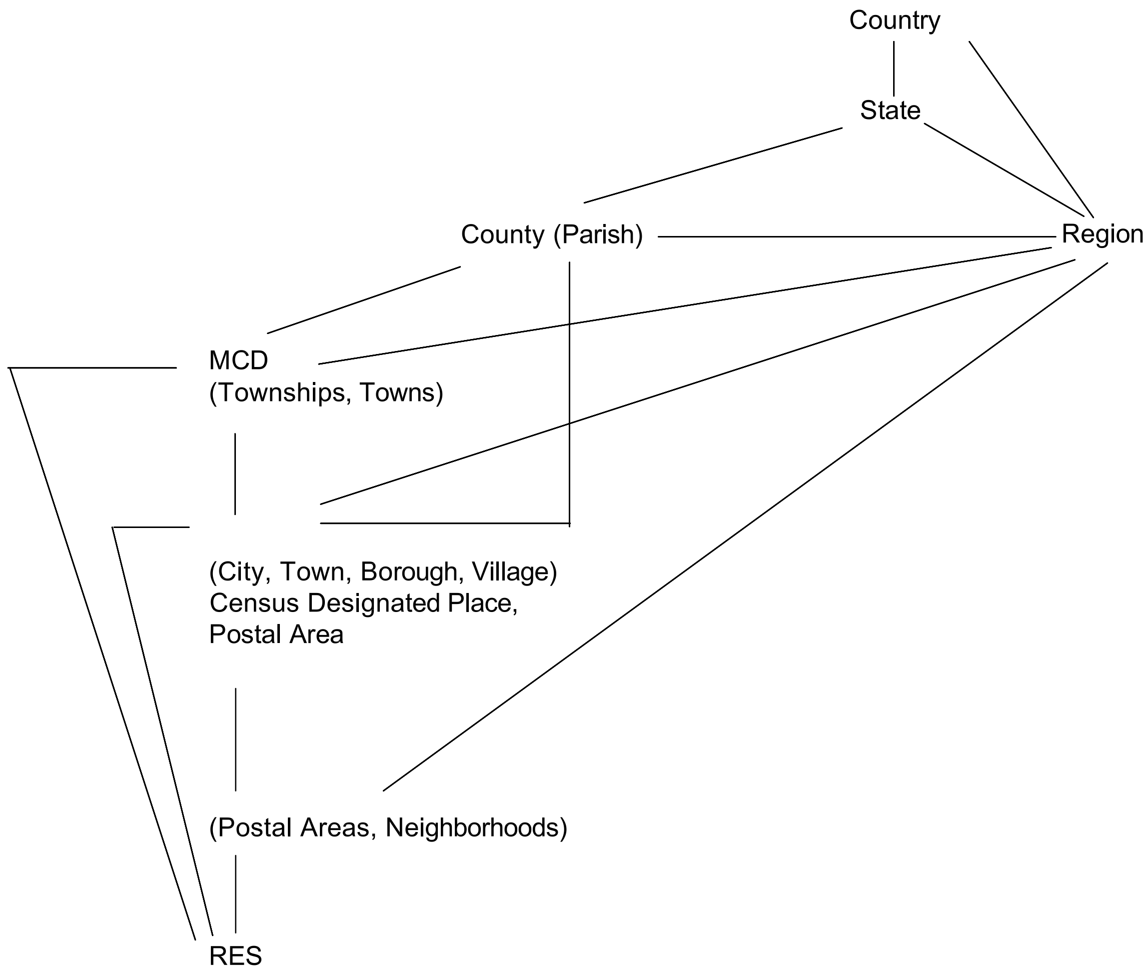


Figure 5 — USA example of administrative structure

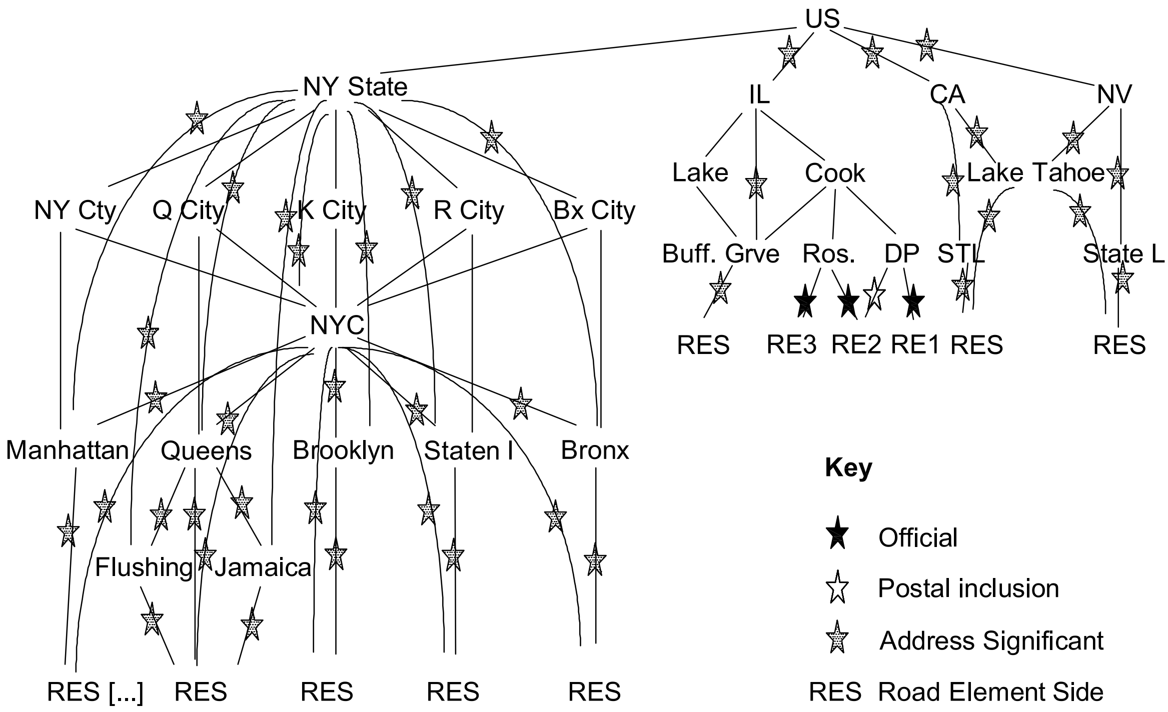


Figure 6 — USA example of administrative places

5.5.7 Metadata

Metadata can consist of the following:

a) a set of place classes in the relative order used, on a country basis, for enabling “free form” entry of places (direct access). The metadata shall indicate whether the place classes are required or optional. The GDF and the PSF shall enable the representation of these metadata. These metadata shall be carried to the application through the API;

b) textual description of places or place levels describing, for example, the type of place level;

c) sorting order of characters per country and language-dependent name-match rules. The criteria and definition for matching names (e.g. place names, street names, or junction names) is country/language-dependent. For example, in Germany, the characters “ü” and “ö” shall also be accessible by users as Specification “u” and “o” characters and, additionally, as “ue” and “oe” characters. This means that “München” matches “Munchen” and “Muenchen”;

d) language-dependent rules for official abbreviations of words;

e) sets of place classes that can be viewed as a single class of places by a user.

5.5.8 Extended Parcel exposing functions

For the requirements of this subclause, refer to 5.2.7.

5.6 Service and POI Information Access

5.6.1 General

The Service and Points-of-Interest (POI) Information Access function provides access to data which are commonly used as origins or destinations for a route and which contain information useful to travelers. Services are single point or area locations that are typically known by name rather than address. Services include traveler-related commercial services such as hotels, restaurants, and gas stations. Services also include locations or points of interest to travelers, such as national parks, monuments, and tourist attractions. Services can be categorized by type (e.g. airport, city centre, hotel) and may carry a variety of other attribute information (e.g. rating, cuisine type, credit cards accepted).

Typically, third party organizations, such as tourist or motoring organizations, can offer a rich content of traveler information which may be of interest to the user. This type of service information is called Third Party Data (TPD). The amount of service information supplied by Third Parties may vary and may consist of comprehensive Service data, including locational aspects and a linkage to the road network. Some TPD may originate from a party which has imposed proprietary restrictions on the use of the data. This is a subset referred to as Branded Third Party Data (BTPD) which imposes additional requirements.

5.6.2 Functional description

An application may provide Service data to the end-user. Also, an application may allow the use of Services in Address Location, Route Planning, and Map Display. An application may provide information about Services, including Third Party Data (TPD). The Services may be selectable by types, geographic areas (e.g. within a rectangle or within distance of a point), places (e.g. Administrative Areas, Districts, Postal Areas), Service attributes, or whether the Service is associated with TPD. Services may be associated with Road Elements or other components of the Transportation Network based on their location. This provides a location on a road element which gives access to a Service.

Additionally, searches for Services may be qualified by an application-specified partial or full spelling match to the beginning of the Service type, attribute, name, or to any individual word within the type, attribute or name.

Services may be associated with each other. A primary Service is called a parent. A parent Service may have many secondary Services called child Services. A child Service may also have many parent Services. One example of how this relationship is used is in the definition of an Airport Service that has multiple *Parking Lots*. In this case, the Airport is designated as the parent Service and the *Parking Lots* are designated as children of that parent.

A Service may be associated with multiple places. For example, the Dallas/Fort Worth Airport is physically located in Arlington and Grapevine. It is also logically associated with Dallas and Fort Worth.

Service and POI Information Access shall support different entry orders by means of appropriate data structures. Typically, a hierarchical top-down entry order may be used. However, permutations thereof shall also be supported, e.g. POI brand name first.

Service and POI Information Access shall support extensions to the search criteria when no match is found. The user may demand an expanded search area, i.e. areas close to the specified place(s) or spelling tolerance for similarly pronounced/written names.

Service data may be accessed by the following methods:

a) via Service attribute data for an application-specified Service (to the extent they exist in the database) for example: name, address, phone number, chain, facility type, and days and times the Service is open;

b) via the coordinates of an application-specified Service;

c) via the related Road Elements and position along the Road Elements for the entry to an application-specified Service;

d) via the related Services of an application-specified Road Element;

e) via the set of Services within an application-specified set of places;

f) via the set of Services within an application-specified set of rectangles;

g) via the set of Services where an application-specified partial or full spelling matches the beginning of the Service name;

h) via the set of Services which meet any conjunction (logical AND) of the following criteria:

   — an application-specified string which matches the full, partial, or phonetic spelling of the Service name. The application specifies whether a partial, phonetic or complete match is required,

   — any of a set of application-specified values of an application-specified Service attribute that can be used as a selection criterion as specified by metadata. More than one criterion of this type may be ANDed together,

   — any of a set of application-specified Service types,

   — the union of an application-specified set of places,

   — the union of an application-specified set of rectangles;

— via the set of Services that are within an application-specified distance of an application-specified place;

— via the selection of Services by relationships to other Services (parent-child relationships);

— via the set of Service types for all Services which exist in an application-specified place;

— for an application-specified place level, chosen from a set of place levels specified in metadata, via the set of places for which an application-specified Service type exists;

- via the set of attribute data values which exist for an application-specified attribute, for Services which meet application-specified criteria, based on availability specified in metadata;
- via a Service given an application-specified external location reference;
- for a given place, service type, and partially spelled service name, via retrieval of the next successor characters;

Additionally, Service data shall

- support the application in presenting the Service data in the same style as provided by the Service data vendor. This includes providing all attributes and attribute values as provided by the Service data vendor;
- support the representation and retrieval of icons;
- support the representation and retrieval of images, sound tracks, hyper-links, and other multi-media;
- for a specified postal code, return the list of Services that have this postal code;
- for a specified Service, return the list of postal codes that are valid for this Service.

5.6.3 Country/Language dependencies

For any search that queries names, the Service shall return the language code and the name-type attributes (official or alternative name) for the name.

Language codes may be specified when retrieving names.

5.6.4 Third Party Data (TPD)

5.6.4.1 General description

Third Party Data is considered a special type of service information. All the functional requirements for Services apply to the special TPD type of services.

Additionally, if there are multiple TPD entities (from different vendors) for a particular real-world service, it should be possible for the multiple TPD entities to be related each other. This functionality is dependent upon this information being available in the data.

An application may provide TPD to the end users. This function provides the following additional methods for accessing TPD. For TPD not related to a base map, no requirements are specified.

a) via a set of TPD from an application-specified vendor which meet application-specified criteria as described above;

b) via supporting the use of TPD from multiple vendors simultaneously in the same database;

c) via accessing TPD from different vendors. The TPD from different vendors may have different sets of attributes, may have different sets of attribute values, and may cover different geographic areas;

d) for a given item of TPD, via identifying the vendor of that TPD;

e) via related transportation entities for application-specified TPD;

f) via related TPD from an application-specified vendor for application-specified transportation entities;

g) the integration between TPD and the base map requires a relationship between the TPD and the transportation entities. This can be done either by a direct relationship (e.g. by an ID) or by an indirect relationship (e.g. by positions);

h) via identifying vendors for which TPD is available.

i) for a specified service entity (including a TPD entity), via the set of service entities (including TPD entities) which represent the same real-world object;

j) via allowing for the identification of mutliple TPD entities which represent the same real-world entity.

5.6.4.2 Branded Third Party Data

BTPD shall exhibit the following characteristics.

a) The API shall allow for the access to BTPD from multiple vendors on a single medium subject to authorization or licensing.

b) There shall be a function call that returns the data formatted to support a BTPD vendor's requirements for symbology, presentation, etc. (e.g., an HTML-formatted "page").

c) Vendor-specific presentation shall be totally transparent to the application. Additionally, vendor-specific presentation shall be transparent to the DAL in order to allow the BTPD vendor to control the presentation without having to be involved in DAL development. This also allows DAL developers to implement a DAL that is independent of various BTPD vendor presentations. Specifications of how vendor-specific information should appear shall be stored on the media. Typically, this vendor-specific presentation information is written by the supplier of the TPD.

d) An application can query the BTPD on a select set of attributes specified by the BTPD vendor. The set of attributes shall be transparent to the DAL and the application, in order to allow DAL and application developers to develop systems which are independent of various BTPD suppliers. The semantic description and format of the attributes which can be queried shall be stored on the media.

Conceptually, the BTPD consists of three components (see Figure 7):

— fielded BTPD records for the set of attributes to be queried on;

— some form of metadata for each service type which describes the field contents, format, and range of values for the attributes which can be queried;

— presentation data at least including the attributes not used for queries and formatting information.

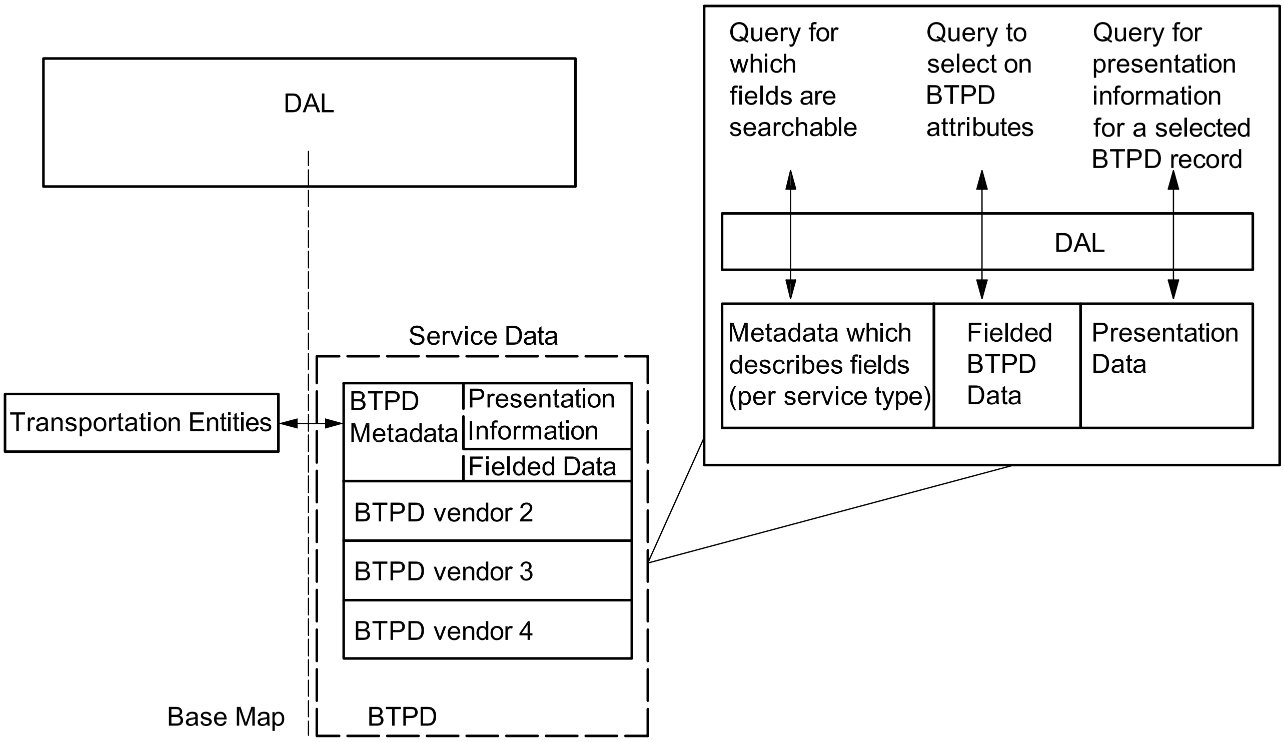


Figure 7 — Structure of BTPD

5.6.5 Requirements for the Logical Data Model

5.6.5.1 General description

The Logical Data Model used for this application category is required to support at least the data identified in the functional description. Other requirements are defined below.

5.6.5.2 Logical Data Model entity

The Logical Data Model entity for Services and POI Information Access is described below.

— **Service**

The Service entity includes both service features and Points of Interest (POIs) used as destinations and/or for orientation. A Service feature is a commercial activity of interest to travelers. A POI is a destination and/or site of interest to travelers which is usually non-commercial by nature. Both types are used synonymously within the Logical Data Model. A service may be associated with other services by parent/child relationships (many-to-many) and may be associated with places. A service entity may have service attributes.

5.6.6 Metadata

The API shall provide, and the PSF shall supply, the following metadata information:

a) the list of Service attributes or combination of Service attributes which can be used as selection criteria for a given Service Type;

b) the list of fields in the TPD for a specific vendor and Service type;

c) the data type and size for a given field;

d) the list of values for enumerated data types and the ranges for numeric data types for a given field;

e) sorting order of characters per country and language-dependent name-match rules. The criteria and definition for matching service names, types and attributes is country/language dependent. For example, in Germany, the characters "ü" and "ö" shall also be accessible by users as specification "u" and "o" characters and, additionally, as "ue" and "oe" characters. This means that "München" matches "Munchen" and "Muenchen";

f) language-dependent rules for official abbreviations of words;

g) indices and field descriptions for Services;

h) the set of service types that are available in the database.

5.6.7 Extended Parcel Exposing Functions

For the requirements of this subclause, refer to 5.2.7.

6 Logical Data Model

6.1 Overall model

The overall model as shown in Figure 8 presents the relations between the following data model parts:

— **Map entities**

They consist of the following parts:

- — Transportation entities.

  The data set is used for Positioning, Route Planning and Route Guidance for different levels of generalization and aggregation.

- — Cartographic entities.

  The data set is used to store map data for display at different map scales.

— **Address Location entities**

The data set is used to relate named features to elements of the Transportation entities in order to specify a destination.

— **Service/POI entities**

The data set contains information about services, POIs, as well as Third Party Data.

— **Dynamic Traffic Information entities**

The data set is used to relate traffic information (e.g. RDS-TMC, VICS) to locations of the road network.

Each data model part is presented as a separate box. Only those major entities of a data model part are shown which have a relation to other data model parts. However, the overall model does not cover relationships that reflect all relations between entities within the boxes nor relations between different levels of generalization and aggregation. Such relationships are described in more detail in the individual models of the data model parts.

The relationship of "Service/POI entities" and "Transportation entities" respectively with "Cartographic entities" indicates that information from the first two data model parts can be displayed in a map.

The data model part "Transportation entities" contains an entity "Link". Relationships exist between a *Link* and entities from other data model parts (see Figure 8):

— *Navigable Feature Name* entity representing addresses from the "Address Location Model";

— *Postal Code* entity from the "Address Location Model";

— *Place* entity from the "Address Location" or "Service/POI entities" *Service* entity from the "Service/POI entities";

— *Traffic Location* entity from the "Dynamic Traffic Information entities".

Based upon a specified address, specified service or specified traffic location, the corresponding *map location* shall be described in terms of those *Link* entities which represent the parts of the road network (i.e. a sub-set of links) that provide access to the *location in question* (LiQ).

NOTE *Location-in-Question (LiQ)* is used as a collective term for all locational information required to describe the map location of above-mentioned entities in terms of the corresponding link(s). For any of these entities, this includes a position relative to the link(s).

LiQ requirements are as follows:

— **to relate the LiQ to a Link:**

  Because an address of service destination directly represents the LiQ, the Link itself does not completely specify a destination. Therefore, an indication is required to state at which position along the particular link the destination is found. This is presented in Figure 8 by means of the relations "*located at position along*" and "*references to position along*".

  In case there is no explicit relation known from the LiQ to a *Link* during the creation of the Navigation database, a coordinate pair for the LiQ shall be available and is used at run-time to resolve the corresponding link(s) together with the position along the link(s).

  The "Address Location entities" and the "Service/POI entities" do not have an explicit relation with each other. They are two independent (but similar) models for describing destinations. For services, relations can be "created" at run-time by using, for instance, the coordinates of *Services*. (This requires a relation between coordinates and an LiQ within a given model, possibly by means of a spatial index.)

— **to relate a Link to the LiQ:**

  A *Link* is related to the LiQ, and can be realised by means of a coordinate along the *Link* or by means of a mechanism defined by the Logical Data Organization. (The latter option is especially useful for the relation between the "Transportation entities" and the "Address Location entities".)

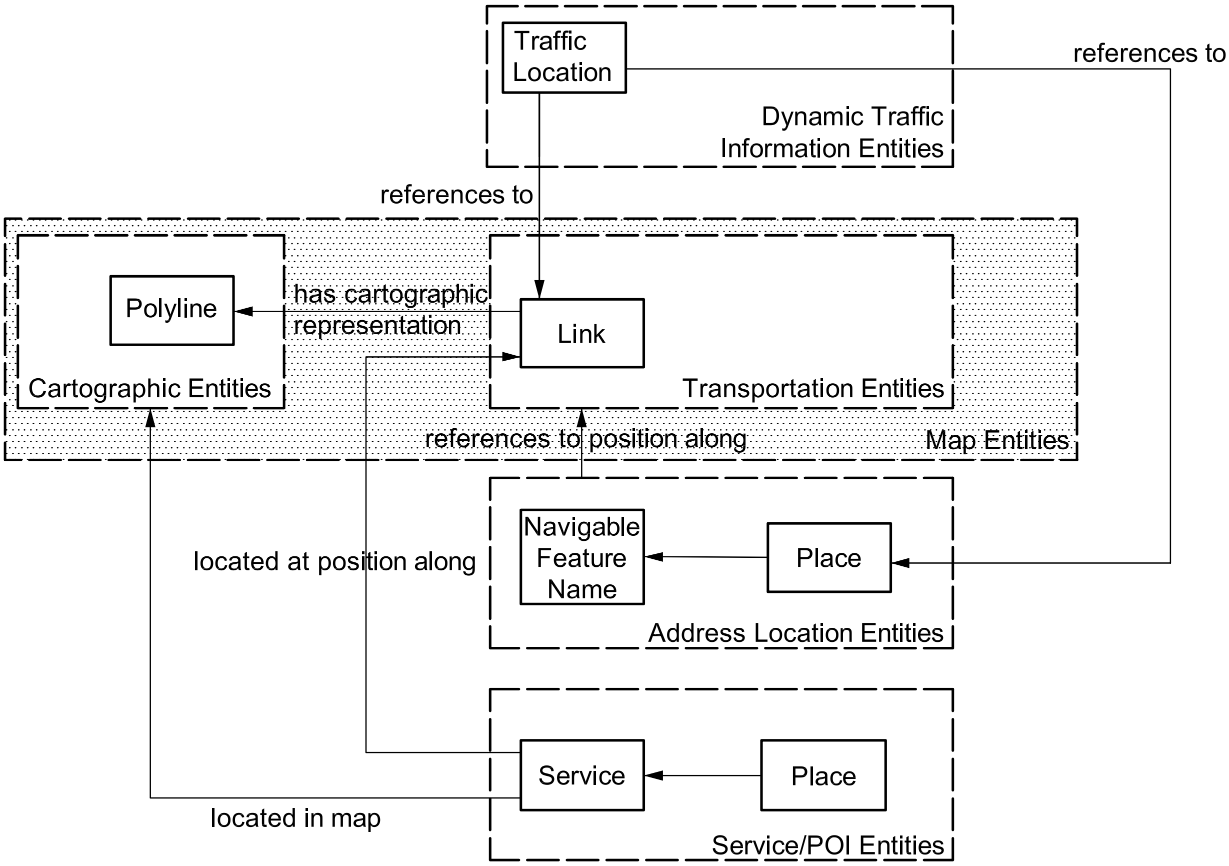


Figure 8 — Overall model

6.2 Transportation Entities

In general, there are a number of data levels, represented by an ER diagram, and there are relations between the levels. This is indicated in Figure 9.

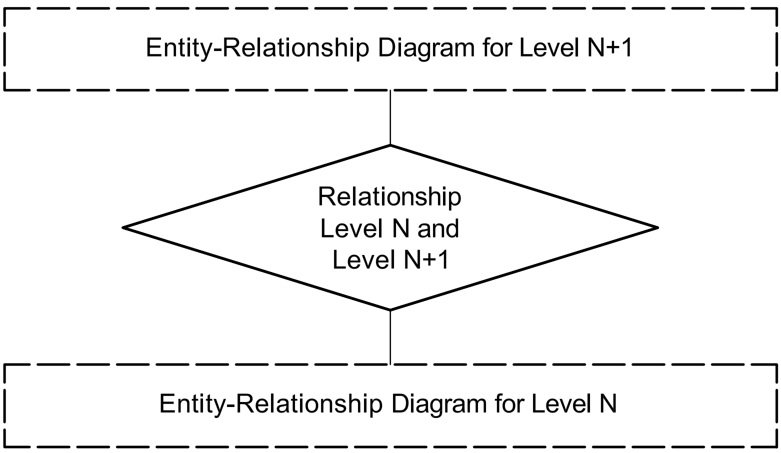
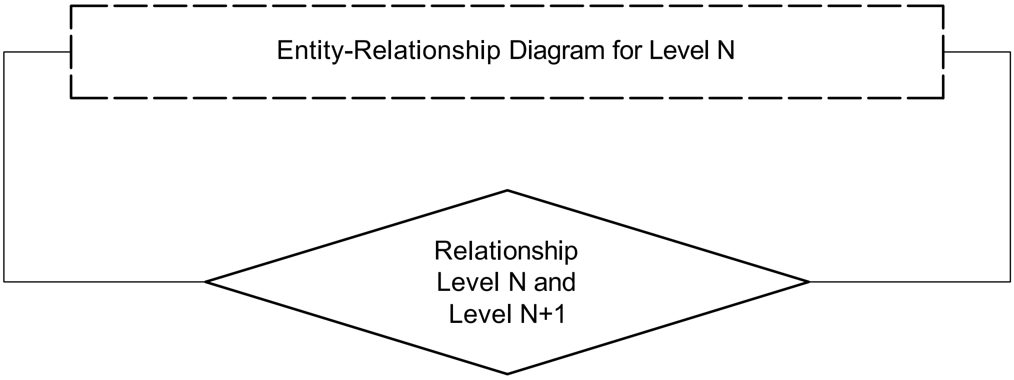


Figure 9 — Relationship across Levels

Considering that the ER diagrams for level N and level N+1 will be the same, any level can be represented by one and the same ER diagram (see Figure 11), with the exception of the attributes: the lowest level can contain more attributes than the higher levels.

Therefore, conceptually the ER diagrams of the transportation entities can be presented as shown in Figure 10:



NOTE The ER diagram for level N is presented inside a bounding box, while the relations "Relationship Level N and Level N+1" are presented outside the bounding box.

Figure 10 — Convention for relationships across levels

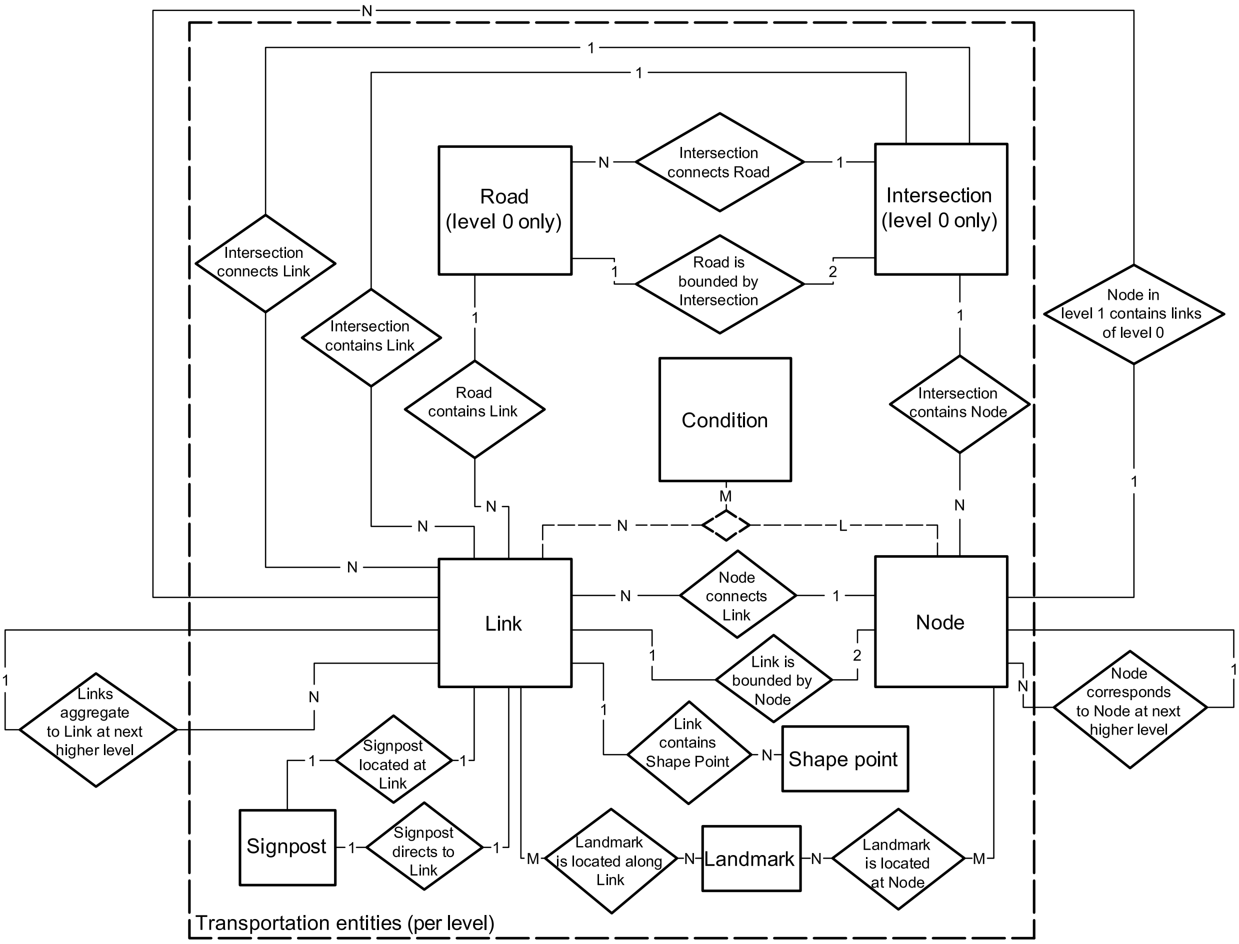


Figure 11 — Logical Data Model of Transportation Entities

At level 0, links may be aggregated "widthwise" (links with opposing directions of travel on the same multicarrigeway may be aggregated) into roads. Links and nodes may also be aggregated into intersections. Roads and intersections exist only at level 0.

Links at level 0 are aggregated into "superlinks", i.e., links at level 1. This aggregation may be "lengthwise" (links with the same direction of travel may be aggregated), widthwise, or both, i.e., the constituents of a link at level 1 may be links aggregated lengthwise and/or widthwise from level 0.

Links at level N are aggregated into links at level N+1. This aggregation can only be lengthwise (links with the same direction of travel may be aggregated), i.e., the constituents of a link at level N+1 may be links aggregated lengthwise from level N.

Nodes and links at level 0 may be aggregated into nodes at level 1. This means that the constituents of a node at level 1 may be nodes and links at level 0.

Nodes and links at level N may be aggregated into nodes at level N+1. This means that the constituents of a node at level N+1 may be nodes and links at level N.

Widthwise aggregation can be applied only to level 0 links, aggregating them into (level 0) roads and level 1 links.

Figure 12 below shows aggregation relationships among these entities. All aggregation is drawn in the upward direction.

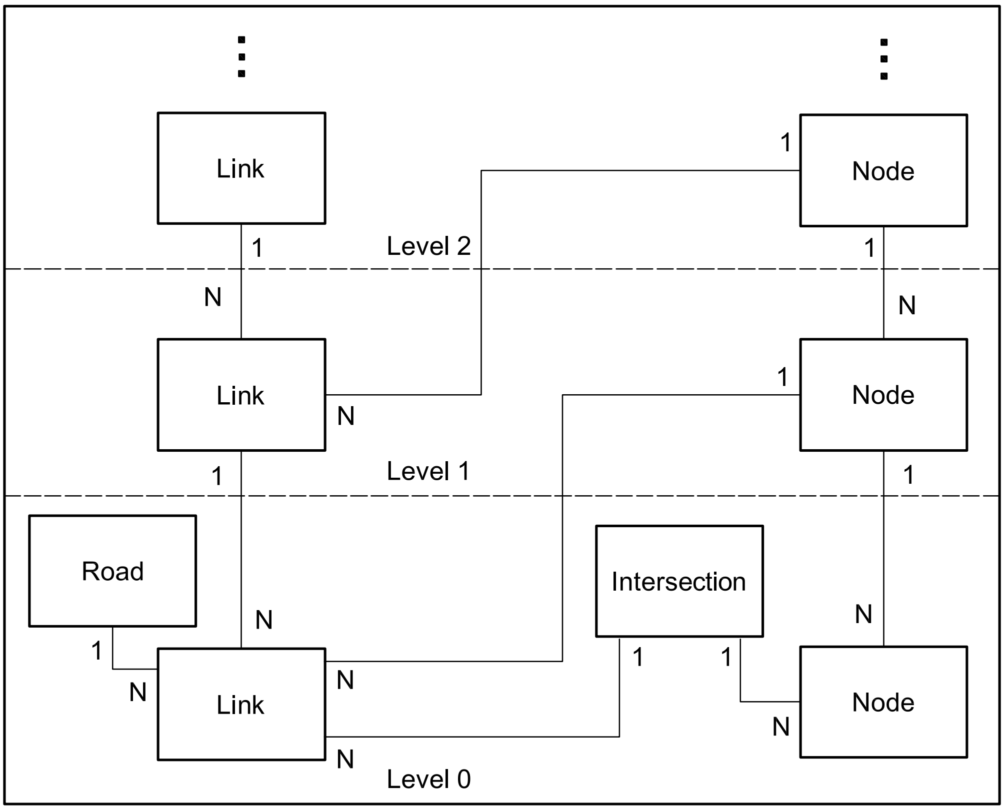


Figure 12 — Aggregation of transportation entities

6.3 Address Location entities

Address Location is determining a location based on information describing or naming the location, e.g. a postal address or the names of two crossing streets. In order to associate name information with the corresponding location(s) in the transportation network, appropriate data model entities are to be created and related with each other to allow flexible methods of accessing and examining data entries.

Generally, two directions for the “data access path” can be identified:

— *Geocoding*: determining a link or node of the transportation network by its address description;

— *Reverse geocoding*: determining an address description associated with a link or node.

Within the Logical Data Model for Address Location entities diagrammed in Figure 13, the two “ends” of the above-mentioned access paths are *name information* and *links & nodes* (part of the lowest level Logical Data Model for Transportation entities). Name information and nodes/links are connected by a flexible structure of other entities and relations.

For modelling name information, the following entity types can be distinguished:

— A *Place* represents any named area which in some kind of hierarchical structure (involving other *Places*) can serve to specify the location of a *Navigable Feature Name*. These places can be official place names, assigned by the government or postal service, or locally known names which people use to identify regions.

— A *Navigable Feature Name* represents the name of any location in the transportation network that can be identified by a name.

— In addition to places, it is useful to identify locations through the use of alphanumeric *Postal Codes* assigned by the postal service. These codes are commonly found in house addressing and can be used as an unambiguous way of selecting places, and navigable feature names.

Three types of location are distinguished and specially modelled:

— *Road Section* (corresponding to a house number range along a named road);

— *CrossRoad* (two intersecting named roads);

— *Junction* (a junction carrying a name itself).

If a location has more than one name (e.g. official and colloquial names, official names in different languages), this shall be represented by multiple instances of the *Navigable Feature Name* entity.

A *Navigable Feature Name* (i.e. a name) does not directly reference to a link or node of the transportation network. In case of a Junction, the Junction entity refers to the corresponding node of the transportation network. When a *Navigable Feature Name* corresponds to a Road Section or a CrossRoad, additional information is related to the *Navigable Feature Name* in such a way that the combination of information components uniquely corresponds to links and/or nodes. For a Road Section, this means the combination of *Navigable Feature Name* plus House Number Range(s). For a CrossRoad, the Logical Data model maintains a set of references to all nodes and all links (bounding these nodes) which correspond to the combination of primary *Navigable Feature Name* and intersecting *Navigable Feature Name*.

Additionally, a *Postal Code* can be used in the search of places or *Navigable Feature Names* to restrict the search to a particular area.

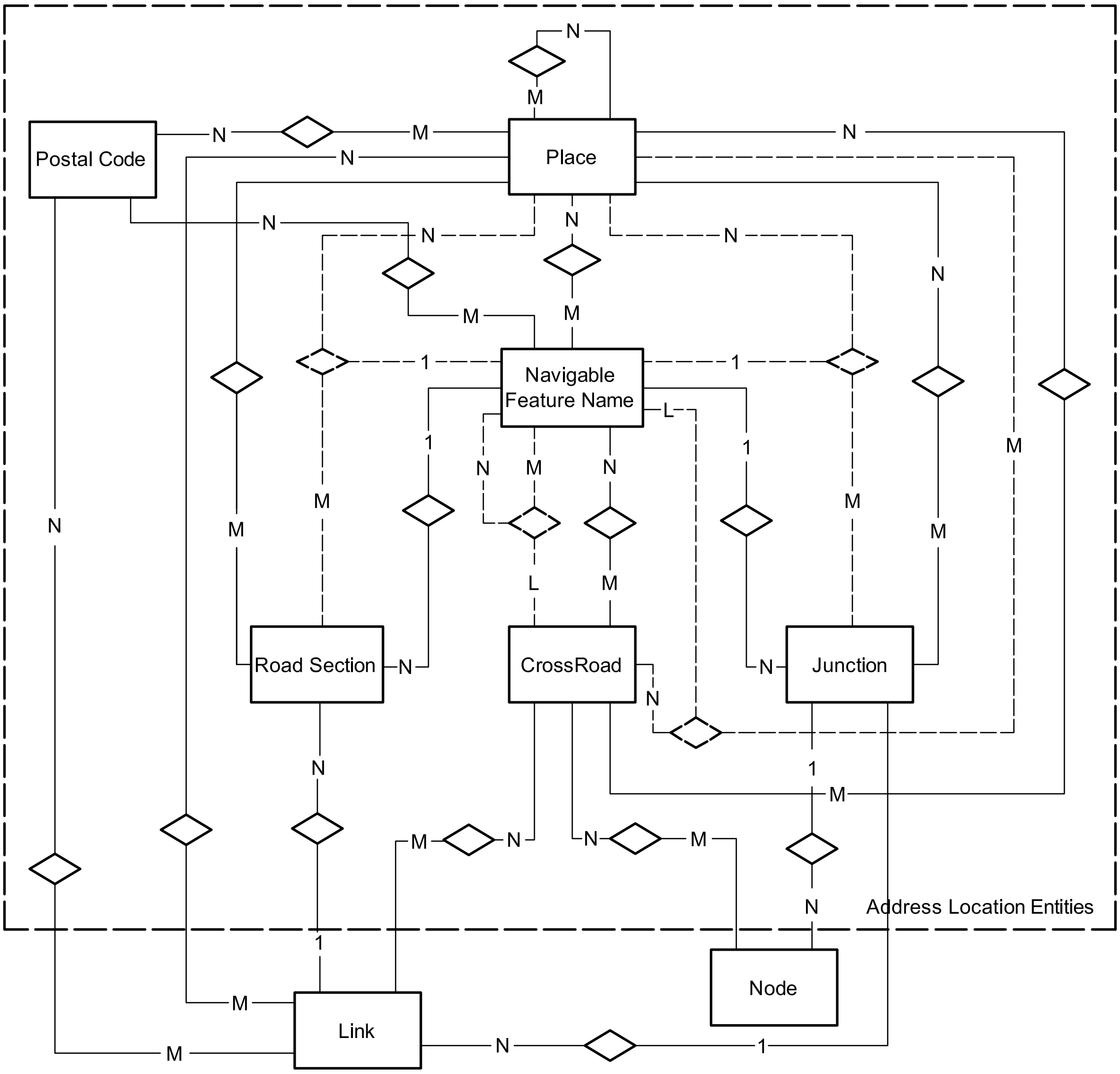


Figure 13 — Logical Data Model of Address Location entities

6.4 Service/POI entities

The logical data model for Service/POI entities shall cover the whole range of services including Third Party Data (TPD) and Branded Third Party Data (BTPD). See Figure 14.

TPD and BTPD are considered special types of service information. All the relationships in the LDM for services apply to TPD services as well. There may be multiple TPD entities (from different vendors) for a particular real-world service. It should be possible for the multiple TPD entities to be related to each other.

Conceptually, the BTPD consists of three components:

— fielded BTPD records for the set of attributes to be queried on;

— some form of metadata for each service type which describes the field contents, format, and range of values for the attributes which can be queried;

— presentation data at least including the attributes not used for queries and formatting information.

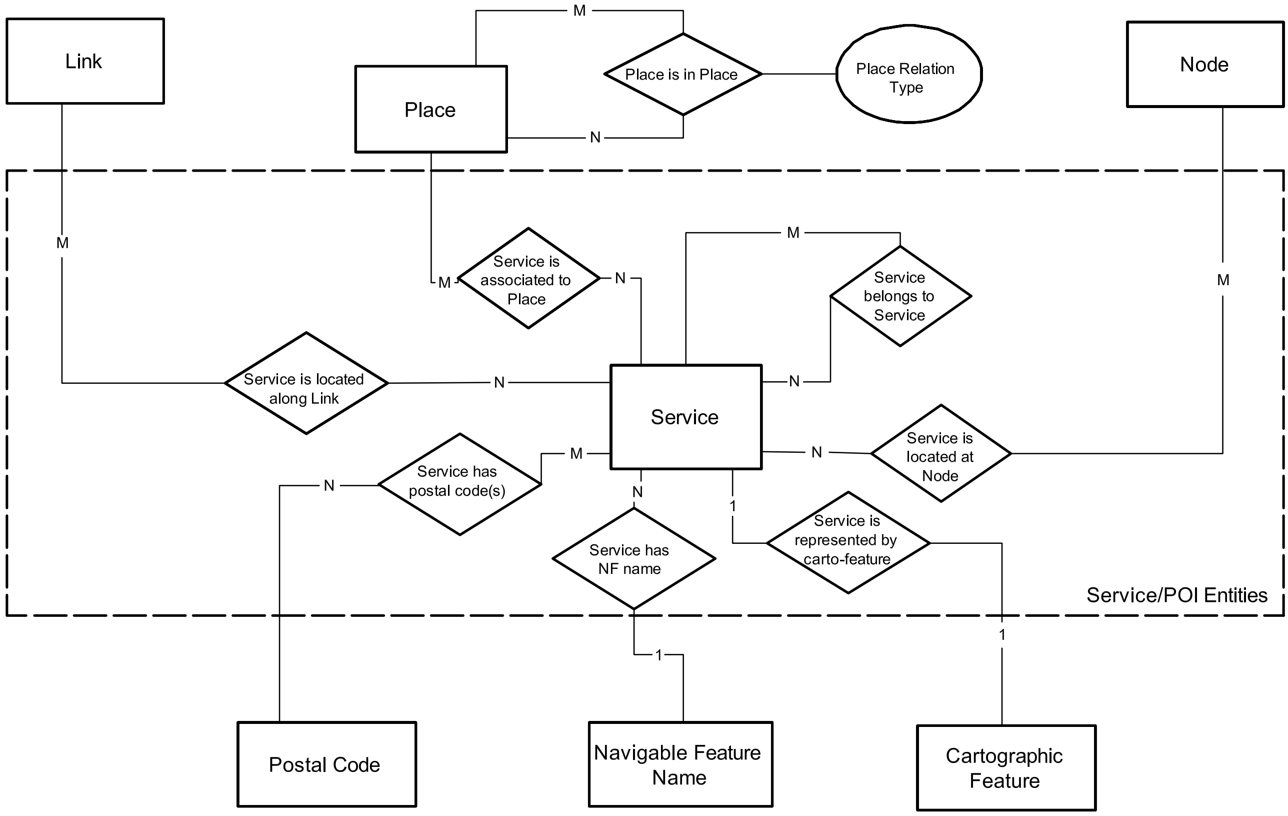


Figure 14 — Logical Data Model for Service/POI entities

TPD to TPD relationships are modelled as shown in Figure 15 below. Figure 15 is fully contained in the "Service" box shown in the centre of Figure 14 above.

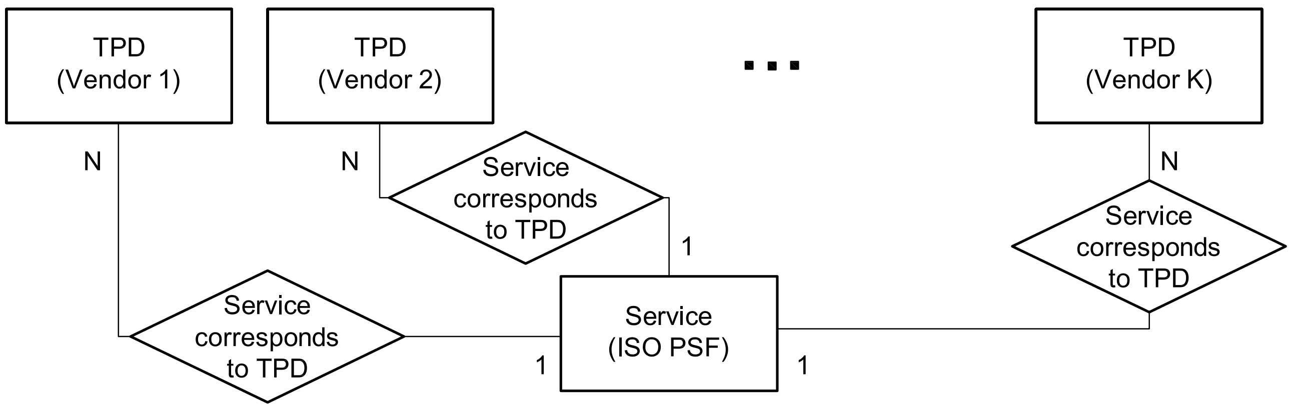


Figure 15 — TPD ⇔ TPD relationship

6.5 Cartographic entities

6.5.1 General

The Map Display function is used to display a map of a specified geographic area. An application may display maps to the end-user. The application may also accept end-user input that references the map display (such as from a point and click device).

An application may display Point Features, Line Features, Area Feature, Cartographic Text and Symbols for a specified geographic area. This may include roads, physical features, administrative boundaries, and names for all of those. Text and symbols can be positioned on a display to annotate this map.

The Map Display function provides cartographic data that can be used to display a map of any application-specific arbitrarily-oriented rectangle in the database. The data model consists of three basic entities to support a variety of map drawing styles (see Figure 16):

— Cartographic Features,

— Cartographic Text, and

— Symbols.

To facilitate data access speed, the Map Display application groups cartographic data into map levels. The higher levels contain only the more significant cartographic features (also based on the scale of the cartographic map). To facilitate map drawing, the Cartographic Features and Cartographic Text have to be organized by “drawing-order”.

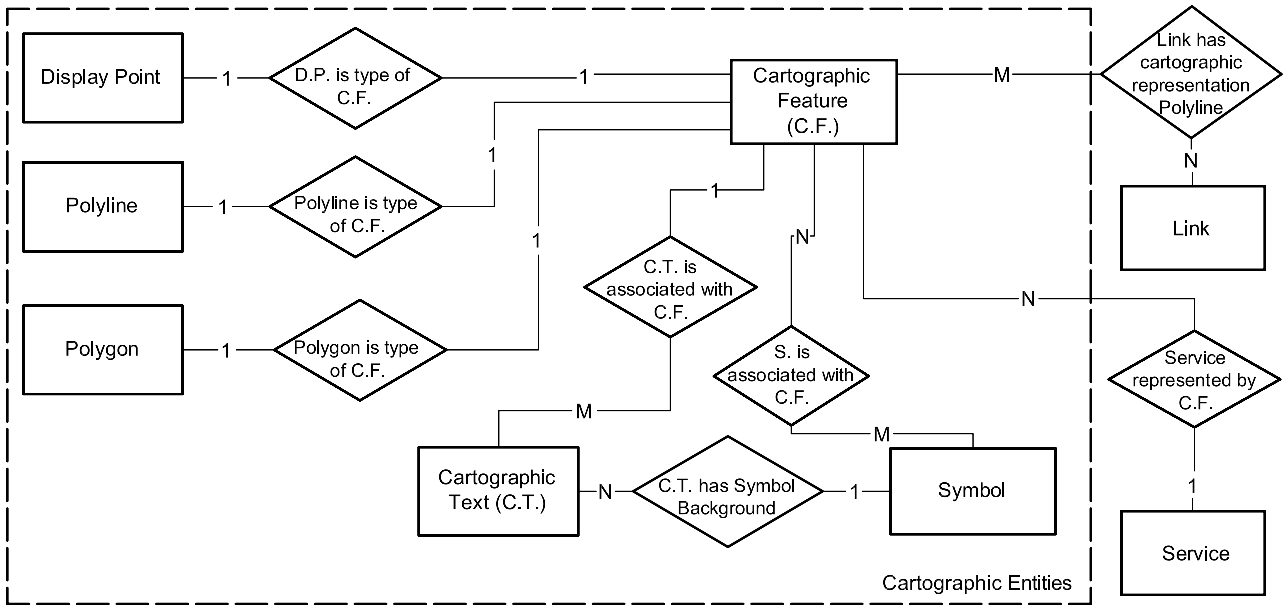


Figure 16 — Data Model for Cartographic entities

6.5.2 Cartographic Feature

The Cartographic Feature consists of three entity types:

— the *Display Point* entity, which is used to represent Services, POIs and Signposts. Depending on the level of generalization, a *Display Point* may also be used to represent an Area Feature;

— the *Polyline* entity, which is used to represent Line Features such as Road Elements. A cartographic polyline does not necessarily correspond to a single Road Element or Line Feature. Depending on the level of generalization, a *Polyline* may also be used to represent an Area Feature. For Map Display data, topological connectivity is not relevant. One cartographic polyline can correspond to many Line Features;

— the *Polygon* entity, which is used to represent Area Features such as parks and lakes. To aid in polygon filling, the shape points of the polygon are returned in fixed order for the outer boundary of the polygon and the enclaves.

An Area Feature may be represented as multiple *Polygons*. A Line Feature may be represented as multiple *Polylines*. If a feature is split, new points are created and, for polygons, new boundary lines are added to close it. Each of these virtual entities contains attributes that are specific to that particular type of entity. These attributes provide the application with the information it needs to determine the entity's display characteristics (e.g. positions).

6.5.3 Cartographic Text

The Cartographic Text entity is used to store the name text that is associated with all or part of a cartographic feature. In addition to the text, this entity may contain a suggested location, orientation, language code, priority (or importance), and suggested scale ranges that can be used to position the text on the display map. Cartographic Text entities are language-dependent and different Cartographic Text entities can be associated to the same Cartographic Feature for different languages.

6.5.4 Symbol

A Symbol entity is a graphic associated with a cartographic feature.

6.6 Dynamic Traffic Information entities

Dynamic Traffic Information entities are used to deliver real-time traffic conditions. These conditions may be used for dynamic route calculations. In addition, the traffic information may be displayed along with the rest of the map display functionality on an informational basis only.

The basic Dynamic Traffic Information entity is the *Traffic Location entity.* In the real world this may correspond to area locations, linear locations, e.g. part of the road, or point locations, e.g. an intersection of a position along a road.

The *Traffic Location entity* may refer to either the *link* transportation entity or the *place* address location entity. A *Traffic Location entity* may be on several different *links,* and *links* may contain several different *Traffic Location entities.* A *place* may contain many *Traffic Location entities* and a *Traffic Location entity* may be in many *places.*

The data model for Dynamic Traffic Information entities is illustrated in Figure 17 below:

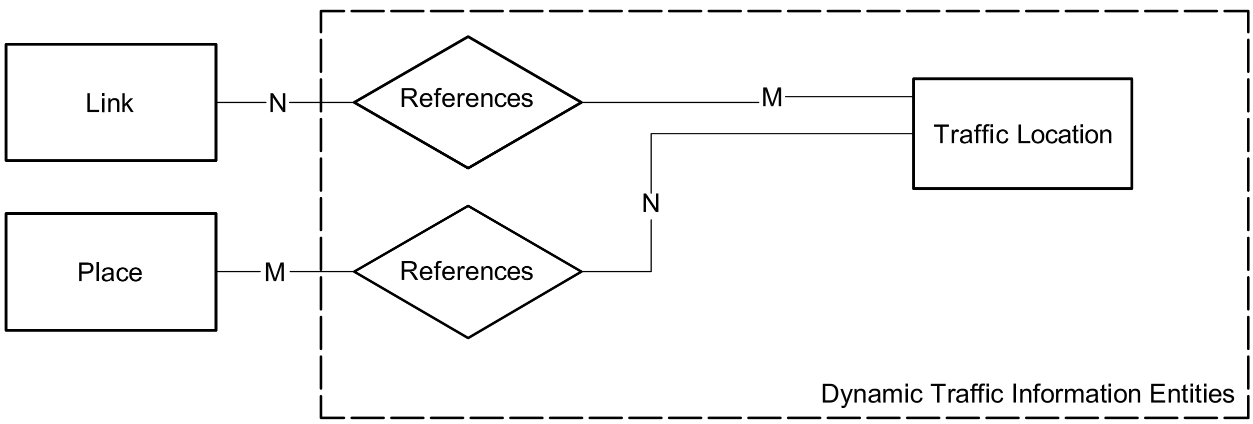


Figure 17 — Data Model for Dynamic Traffic Information entities

7 Logical Data Organization

7.1 Global architecture

Figure 18 shows the conceptual view of the global LDO architecture and the identified logical building blocks therein. A general description of the role and concept of each building block is given in 7.1.1.

This conceptual view does not anticipate any particular parcelling method to be used for a building block or level.

NOTE It is an application issue to determine which levels will be used for the Route Planning and Guidance applications.

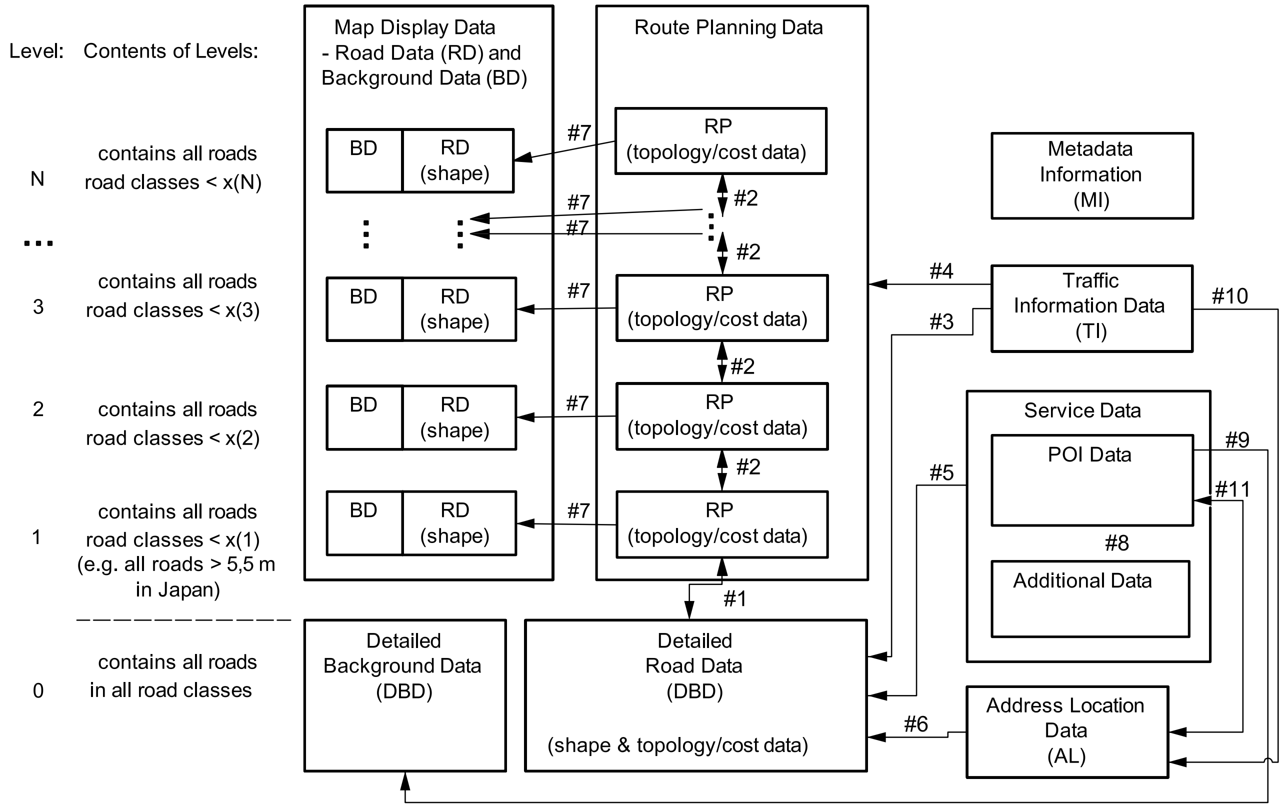


Figure 18 — Logical Building Blocks of the LDO

7.1.1 Logical Building Blocks of the PSF

This subclause provides a general description of the role and concept of the identified building blocks.

7.1.1.1 Detailed Road Data

The Detailed Road Data (DRD) block contains the lowest generalization level with the most detailed transportation elements, and is used for calculating routes, performing map matching, deriving route guidance instructions and drawing roads on the screen (as far as road geometry is concerned). Therefore DRD contains the full road geometry, including shape points as well as topology and associated travel “cost” data.

The DRD block contains the following elements for Route Guidance:

- Signpost information;
- Landmarks;
- cross-reference to Navigable Feature Names.

This information is organized for optimal determination of Route Guidance information after a route has been determined.

7.1.1.2 Detailed Background Data

The Detailed Background Data (DBD) block is used to generate a map for display on the screen. It contains the lowest generalization level with the most detailed cartographic entities except for navigable road objects. For map display, DBD complements the DRD block, which contains the associated road network data.

The DBD building block only contains cartographic features (display points, polylines, and polygons) and cartographic text (representing street names, place names, etc.). It does not contain links, nodes, roads or intersections.

7.1.1.3 Map Display data

The Map Display data (MD) block contains multiple levels of map display related data. Each level corresponds to a different map scale (or map scale range) to be generated. MD is used for displaying a map on the screen.

Within each level, two (sub-)blocks can be identified, namely Road Data (RD) and Background Data (BD). This conceptual differentiation corresponds to the distinction between road and non-road related data in the lowest level, i.e. between DRD and DBD. On higher levels, however, this distinction is less obvious since both (sub-)blocks contain shape and display information only.

7.1.1.4 Route Planning data

The Route Planning data (RP) block contains multiple levels of road network data which correspond to a decreasingly dense road network for each higher level. RP is used for calculating a route. RP contains topology and the geometry of nodes, however no shape points between nodes.

The number of levels in the RP block may be different from the number of levels in the MD block.

7.1.1.5 Address Location data

The Address Location data block contains the information necessary to relate address location entities to the road network. This is used when selecting a destination by city name, street name, address, junction name, postal code, or crossroad, or when determining the address of a specific location (reverse geo-coding).

7.1.1.6 Service data

The Service data block contains information regarding services and POI entities. It includes the information necessary to select a service as a destination by place, location, service type or other attributes. It may also contain detailed information about services, such as telephone numbers and opening hours. Within the Service data block, two sub-blocks can be identified, namely POI Data and Additional Data.

In the POI Data block the POI/Services records are stored. It contains a specification set of attributes such as POI Category, Official Names, Address, Phone Numbers, etc., as it is typically available from Map providers.

The Additional Data Block contains descriptive and editorial information on POIs and Services that is additionally available from Third Party Providers. This includes for example Price Categories, Opening Times, Ranking, Images etc.

7.1.1.7 Traffic Information data

The Traffic Information data block contains the references used to relate external references (such as RDS-TMC codes, or VICS codes) to the appropriate transportation entities.

7.1.1.8 Metadata

The Metadata information block contains information specific to the region (such as country information or content-specific data information). The application may expose some data to the end-user (e.g. content-specific language information), use it for performance tuning (e.g. using maximum count information for run-time buffer allocation), or use it to support forward compatibility (e.g. the code translation tables).

Data contained in metadata include minimum bounding rectangle of database coverage, compression tree references, maximum counts for runtime buffer allocation, scaling factors, level count, country references, version number(s), attribute information, and language information. In addition, codes and values for time zones, POI types, features, language, affix, and bitmaps are stored in metadata. Further, metadata contain node cost table(s), translation table(s), and locale data. Finally, metadata contain data specified in the metadata subclauses (5.1.5, 5.2.6, 5.3.7, 5.4.6, 5.5.7 and 5.6.6) of this Technical Specification.

7.1.2 References between Building Blocks

References between LDO Building Blocks are as defined in Table 1. They are included here, rather than in Figure 18, for visual clarity. References may be realised in the PDO by one or more distinct cross-references.

The "RS" column indicates references related to the Road Section entity, which is both an entity and a reference between building blocks.

There are references from each Logical Building Block to the metadata block that are not reflected in Figure 18. These references are necessary to properly interpret the data in the other logical building blocks.

Table 1 — References between LDO building blocks

ID	LDO Building Block #1	Reference Type	LDO Building Block #2	Comment	RS
1	DRD	↔	RP	bi-directional references between Links and Nodes of level 0 (Detailed Road Data) to Links and Nodes of level 1 (Route Planning data) necessary for performing a route calculation	
2	RP	↔	RP	bi-directional references between Links and Nodes of level k (Route Planning data) to Links and Nodes of level k+1 (Route Planning data) necessary for performing a route calculation	
3	TI	→	DRD	references from a Traffic Location entity to corresponding Links and Nodes (in Detailed Road Data level) in order to consider dynamic traffic information in the route calculation process	
4	TI	→	RP	references from a Traffic Location entity to corresponding Links and Nodes in order to consider Dynamic Traffic Information in the route calculation process	
5	DRD	↔	RP	bi-directional references between Links and Nodes of level 0 (Detailed Road Data) to Links and Nodes of level 1 (Route Planning data) necessary for performing a route calculation	
6	RP	↔	RP	bi-directional references between Links and Nodes of level k (Route Planning data) to Links and Nodes of level k+1 (Route Planning data) necessary for performing a route calculation	
7	TI	→	DRD	references from a Traffic Location entity to corresponding Links and Nodes (in Detailed Road Data level) in order to consider Dynamic Traffic Information in the route calculation process	
8	TI	→	RP	references from a Traffic Location entity to corresponding Links and Nodes in order to consider Dynamic Traffic Information in the route calculation process	
9	Service Data	→	DRD	references from Service entities to their corresponding Links and Nodes in order to allow a route calculation to a user-selected Service/POI	
10	AL	→	DRD	references from Address Location entities to their corresponding Links and Nodes in order to transform an address input into a destination for route calculation	
11	RP	→	RD	references from Links to Polylines (in the higher layers) in order to enable highlighting a route on the screen	
12	POI Data	→	Additional Data	references from Service entities to corresponding Third Party Information	
13	POI Data	→	DBD	references between cartographic display points and Service entities	
14	TI	→	AL	references from a Traffic Location entity to corresponding Places in order to identify the location of Dynamic Traffic Information	
15	AL	↔	POI Data	References from Place entities to Service entities to support “search service in place”. Bi-directional references between Place and Service (and type) entities to support “search service and type available in place” and “get place from service and type”. Bi-directional references between Postal Code and Service entities to support “search service in postal code” and “get postal codes from service”.	

7.2 Detailed Road Data

This data block contains records for the following LDM entities:

- Condition;
- Intersection;
- Landmark;
- Link;
- Node;
- Road;
- Shape point;
- Signpost.

These LDM entities, and their relationship with other entities, are defined in Clause 6. Data structure for a particular entity can be made up of several existing entities' data structures.

7.3 Detailed Background Data

This data block contains records for the following LDM entities:

- Cartographic feature;
- Cartographic text;
- Display point;
- Polygon;
- Polyline;
- Symbol.

These LDM entities, and their relationship with other entities, are defined in Clause 6. The symbol entity's data record is stored in metadata.

7.4 Map Display Data

This data block contains records for the following LDM entities:

Table 2 — Map Display Data entities

	BD Building Block	RD Building Block
Cartographic feature	✓	✓ (road objects only)
Cartographic text	✓	✓ (cartographic text associated with navigable feature names only)
Display point	✓	
Polygon	✓	
Polyline	✓	✓ (road objects only)
Symbol	✓	

These LDM entities, and their relationship with other entities, are defined in Clause 6. The symbol entity's data record is stored in metadata.

7.5 Route Planning data

This data block contains records for the following LDM entities:

— Condition;

— Link;

— Node.

These LDM entities, and their relationship with other entities, are defined in Clause 6.

7.6 Address Location data

This data block contains records for the following LDM entities:

— Crossroad;

— Junction;

— Navigable feature name;

— Place;

— Postal code;

— Road section.

These LDM entities, and their relationship with other entities, are defined in Clause 6.

Note that "the Road Section entity provides a logical relationship between a link, a navigable feature name (representing a name associated with the link), a place, and a house number range"[3]. So, it is both an LDM entity and a reference between LDM entities (and, hence, LDO building blocks). For that reason, it is described in 7.1.2.

7.7 Service Data

7.7.1 POI Data

This data block contains records for the following LDM entity:

— Service

This LDM entity, and its relationship with other entities, is defined in Clause 6.

3) Requirements and Logical Data Model for Physical Storage Format (PSF) and API and Logical Data Organization for PSF used in ITS Database Technology.

7.7.2 Additional Data

This data block contains records for the following LDM entity:

— Third Party Data

This LDM entity, and its relationship with other entities, is defined in Clause 6.

7.8 Traffic Information

This data block contains records for the following LDM entity:

— Traffic Location

This LDM entity, and its relationship with other entities, is defined in Clause 6.

7.9 Metadata

This data block contains records for the following LDM entity:

— Symbol

This LDM entity, and its relationship with other entities, is defined in Clause 6.

Bibliography

[1] Car IN Format, http://www.siemensvdo.com

[2] KIWI Format Specification version 1.2.2 (JIS D0810)4), http://www.jsa.or.jp/default_english.asp

[3] SDAL Format (versions 1.3-1.8), NAVTEQ, http://www.navtech.com/sdalformat/features.html

[4] Siemens AF Format, http://www.siemensvdo.com

[5] TRAVELPILOT Format, http://www.blaupunkt.com

4) Japanese Industrial Standard.

ICS 03.220.01; 35.240.60

Price based on 54 pages