Oracle Spatial and Graph

Network Data Model Graph

ORACLE WHITE PAPER | NOVEMBER 2016

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Table of Contents

Introduction

Network Data Model graph (NDM) is a feature of Oracle Spatial and Graph that lets you store and analyze physical and logical networks used in industries such as transportation, utilities, and communications. NDM persistently manages the network connectivity in the database, while a Java API provides fast in-memory graph analytics, including shortest path, nearest neighbors, within cost, and reachability. It provides an open data model that simplifies network modeling, analysis, and management so that users can focus on application logic. Application information is separated from connectivity information so that the model can be applied to many network applications without customization. NDM further provides a constraint mechanism to let users guide analysis based on application rules and attributes.

NDM is designed to natively make the most of Oracle Database features for scalability, security, partitioning, and parallelism. It can exploit the processing power and bandwidth of Oracle Exadata Machine to obtain extreme performance capabilities that are orders of magnitude over what was previously possible.

Oracle Spatial and Graph NDM simplifies feature editing and analysis by providing a feature analysis function that associates real-world feature representations with network elements. Rather than working with database representations like “links” and “nodes”, users can interact with real-world features like valves and pipelines to model a water network, intersections and roads to model a road network, and power lines and transformers to model an electricity network. Feature modeling maintains these relationships through feature metadata, simplifying application development and maintenance.

NDM performs network analysis on large networks through its load-on-demand (LOD) approach. Instead of loading the whole network into memory, the network is first divided into manageable sub-networks (network partitions), and only partitions that are needed during analysis are loaded into memory. Loading and unloading of network partitions are automatically managed, thus removing memory as a limiting factor.

Oracle Spatial and Graph also supports modeling networks that include a time dimension. Users may associate time properties with nodes and links, and specify temporal inputs in network analysis queries. Most real-world networks have this time element. The fastest route at a non-congested time could be a very slow route at a congested time. Utility networks experience different demand loads based on seasonal demand and the time of day. NDM supports queries such as finding the fastest travel route for a specified time of day.

With NDM you can model and analyze multimodal transportation networks and compute the fastest paths on multimodal transportation networks. A multi-modal network usually consists of multiple networks of different transportation modes like car, bus, subway, and rail. In NDM, these networks are interconnected and treated conceptually as a single network.

In addition to determining connectivity relationships, the Network Data Model graph also provides the infrastructure for the Oracle Spatial and Graph routing engine to produce driving directions and other street network-based analysis. The Oracle Spatial and Graph routing engine is a road network routing service that provides the optimal route with driving directions between two or more locations including turn and truck-specific restrictions.

This paper provides an overview of the NDM feature modeling framework in Oracle Spatial and Graph. It explains how to include temporal information with traffic patterns information in routing and multi-modal routing, and explains how the routing engine uses the NDM model. Large-scale time/distance analysis and NDM API capabilities are also discussed.

Feature Modeling and Analysis

A feature is an object of interest in a network application that is associated with a node or link. For example, in a transportation network, features include exits and intersections (mapped to nodes), and highways and streets (mapped to links). Features are main subjects dealt with by network applications. Network elements such as nodes and links are mathematic representations that are used for network graph modeling and analysis.

NDM provides feature modeling and analysis support, which enables users to model and analyze networks in the terminology with which they are most familiar.

Features, Feature Layers and Other Feature Tables

A feature consists of one or more feature elements. A feature element is a point or line along the network. If it is a point, it can lie on a node or along a line; if it is a line, it can be a full link or a partial link (see Figure 1). A feature can have any of the feature types: a point on a node, a point on a link, multiple points, a line on a link, multiple lines on multiple links, a collection of points and lines, etc.

A feature layer corresponds to a table containing features that have the same set of attributes. For example, in a road network, there may be separate feature layers for restaurants and hotels (and perhaps other feature layers for other things of interest to travelers). A feature layer can have any combination of supported feature types.

A parent feature consists of features from one or more feature layers. For example, in an electrical network, the substation is a parent feature for the feature layers and for all its associated parts, such as joints, switches, and cables. When a child-parent relationship exists, features can form hierarchies.

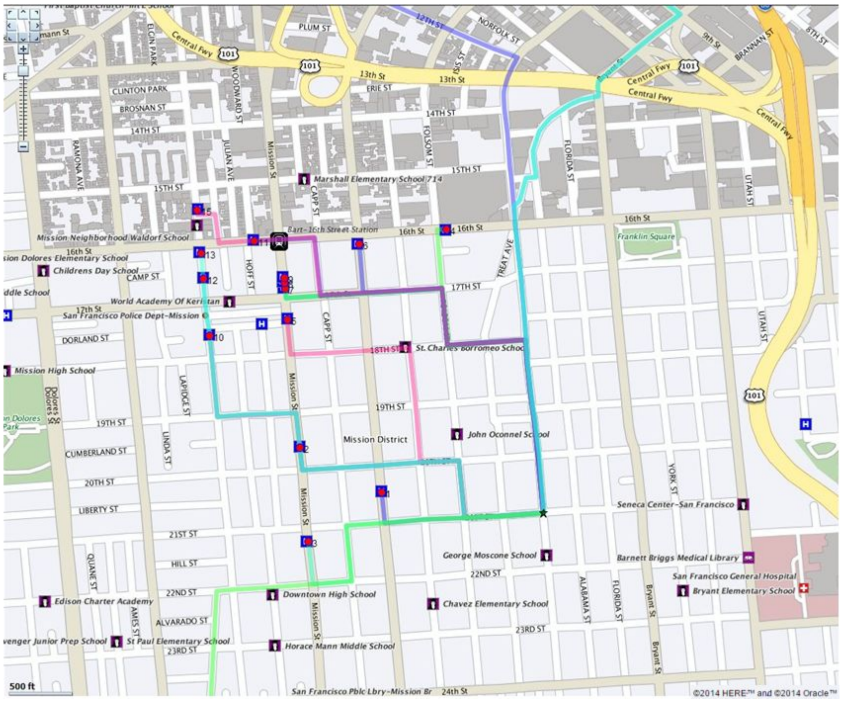


Figure 1. Hotels and restaurants as point features in NDM (data source: here.com)

The following tables are used for feature modeling in NDM:

- **Metadata for feature layers:** The metadata information of the feature layers is stored in USER_SDO_NETWORK_FEATURES.
  The feature metadata information includes: network name, feature layer name, feature layer ID, feature layer type, feature table, relation table, and hierarchy table.
- **Feature table:** The feature table contains NDM feature information on real world objects, their attributes, and their network element type.
- **Feature element relationships table:** The feature element relationships table contains information about the relationships between feature elements and network elements (nodes and links).
- **Feature hierarchy table:** The feature hierarchy table contains feature hierarchy information. Child features can belong to different parent feature layers.

Using Feature Modeling and Analysis

You can model objects of interest on the network as features. A feature consists of one or many feature elements. A feature element can be a point, a link, or a partial link along the network. You can define feature layers on top of a network. For example, restaurants and hotels (Figure 1) can each be defined as a feature layer on a road network, and switches can be defined as a feature layer on an electrical network.

The following are the typical steps for using feature modeling in NDM:

1. Create a feature layer.

   For example, the points of interest (POIs) on a road network can be modeled as features. Each type of POI (hotels, restaurants, hospitals, schools, and so on) corresponds to one feature layer. The following example adds a feature layer for hotels:

```
sdo_net.add_feature_layer(
  'US_ROAD_NETWORK',   --network name
  'HOTEL',             --feature layer name
  2,                   --feature layer type: point on link
  'HOTEL_TAB',                 --feature table or view name
  'HOTEL_NET_REL',     --relation table or view name
  null);                --hierarchy table or view name
```

2. Register feature user data, if any application-specific feature attributes are potentially useful in feature analysis.

   Feature user data is registered by adding an entry in the XXX_SDO_NETWORK_USER_DATA view, just like registering the user data for network nodes or links, except that the TABLE_TYPE column is set to the name of the feature layer. The following example adds hotel name as user data for hotel features:

```
INSERT INTO USER_SDO_NETWORK_USER_DATA(
  network, table_type, data_name, data_type, category_id)
VALUES(
  'US_ROAD_NETWORK',    --network name
  'HOTEL',          --feature layer name
  'NAME',                   --user data name, i.e., name of the user data column
  'VARCHAR2',                        --user data type
  3);                       --user data category
```

3. Add, update, or delete features on the feature layer with SQL or PL/SQL API.
4. In some cases NDM is used with data supplied by third-party data vendors. For example, street network data may be used from vendors such as Nokia Here and TomTom. When the content in the feature tables, feature element relationship table, and/or feature hierarchy table is provided by a data supplier, you can skip this step.

Otherwise, call procedures in SDO_NET package, such as ADD_FEATURE, UPDATE_FEATURE, and DELETE_FEATURES to add, update or delete features in a feature layer. Then, perform feature analysis using the feature analysis functions included in the NDM Java API. The feature analysis functions include: Shortest paths between features, Nearest (reaching) features and Within (reaching) cost features.

Temporal Modeling and Analysis

Most networks exhibit a time-dependence. The ability to include a temporal dimension to the NDM model enhances the kinds of analysis often required by GIS and other applications. Travel times on roads vary depending on the time of day and networks may evolve over time based on seasonal needs or demands. Incorporating these temporal variations into network analysis makes the model more useful and accurate. For example, a user would be able to ask questions such as “Which route would take me fastest to my work location if I start from home at 8 AM?” or “When is the best time to go to the grocery store so that I spend the least time driving?”

Some of the time-dependent analysis can be achieved with the NDM constraint and cost calculator mechanism. This paper presents our approaches to handle two common applications that require temporal information: routing considering traffic patterns and multi-modal network routing.

Traffic Patterns

Traffic pattern data sets offer information about the actual speed that traffic travels on roads and highways during predefined time intervals. Network analysis using these traffic patterns allows computations to account for varying congestion levels on roads at different times of day. This essentially makes computations sensitive to the start time. For example, shortest path from point A to point B could be different for different start times at point A (Figure 2).

A link may be associated with up to seven patterns. This allows the model to represent speed patterns for each day of the week. It is not mandatory for links to have seven patterns; it is possible for links to have speed patterns for weekdays and weekends, thus associating each link with two patterns or having more speed patterns for holidays.

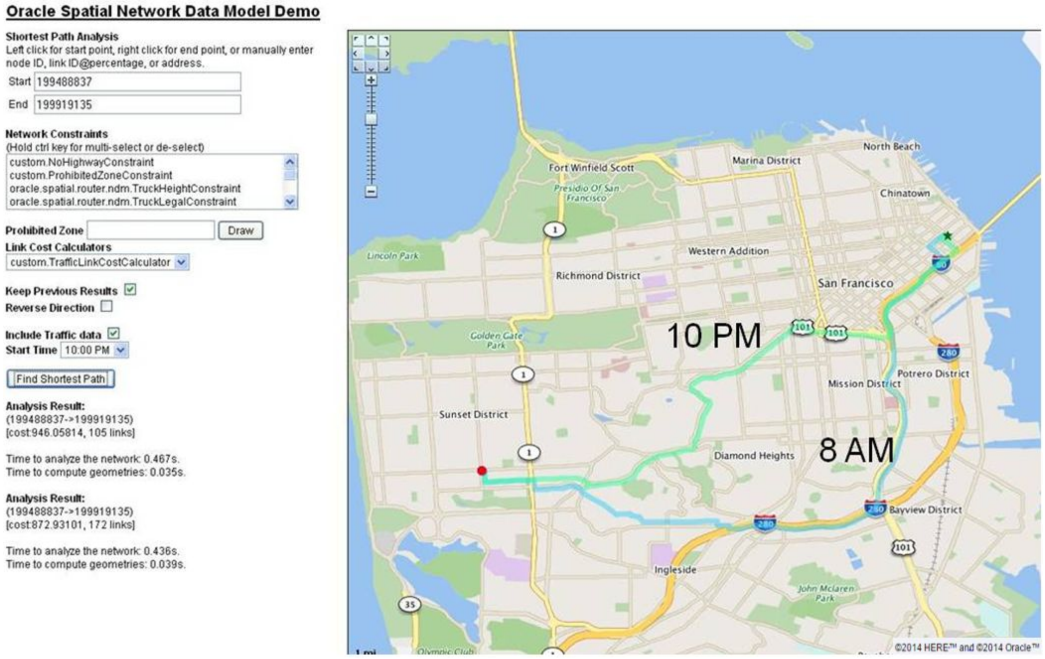
Intervals

A pattern is modeled as a set of speed values for a set of intervals. Non-overlapping intervals must entirely cover the 24-hours of a day and should be of equal duration. Granularity of the intervals should be such that as it divides the 24-hours day in an integer number of slots.

Samplings

There can be multiple sets of traffic patterns (called samplings), and links can be associated with multiple samplings.

A sampling group's pattern is based on parameters such as different granularities (number of time intervals) or classification (for example weekends/weekdays or working days/holidays) or traffic attribute used (speed or travel time).



Oracle Spatial Network Data Model Demo

Shortest Path Analysis
Left click for start point, right click for end point, or manually enter node ID, link ID@percentage, or address.

Start 199488837

End 199919135

Network Constraints
(Hold ctrl key for multi-select or de-select)
custom.NoHighwayConstraint
custom.ProhibitedZoneConstraint
oracle.spatial.router.ndm.TruckHeightConstraint
oracle.spatial.router.ndm.TruckLegalConstraint

Prohibited Zone

Link Cost Calculators
custom.TrafficLinkCostCalculator

Keep Previous Results ☑
Reverse Direction □

Include Traffic data ☑
Start Time 10:00 PM

Analysis Result:
(199488837->199919135)
[cost:946.05814, 105 links]

Time to analyze the network: 0.467s.
Time to compute geometries: 0.035s.

Analysis Result:
(199488837->199919135)
[cost:872.93101, 172 links]

Time to analyze the network: 0.436s.
Time to compute geometries: 0.039s.

Figure 2. Two optimal routes of the same start/destination at different start times.

Modeling Traffic Patterns with NDM

The following tables are used for modeling traffic patterns in NDM:

Traffic Pattern Metadata Table

This metadata table describes the organization of the traffic patterns for each network: classification (number of patterns per link), granularity (number of intervals per pattern) and the kind of attribute used.

Traffic Patterns Table

This table associates traffic patterns with individual network links. Each row contains a link ID, a sampling ID, and 7 speed series as an array (one for each day of the week).

Traffic Patterns User Data Table

This table contains the traffic patterns partition blobs information used in NDM analysis.

Using Traffic Patterns

Traffic patterns reflect the actual travel time at a certain time of day. To consider traffic delays, NDM uses a cost calculator as a function of the day and current time. During network analysis the link cost calculator must return the link cost (travel time) at the current time.

This involves the following steps:

- **Computing the current time:** This is computed by adding current cost to the start time (an input parameter).
- **Accessing the correct link speed series corresponding to the current time:** The day corresponding to the current time is computed and the index to the traffic pattern corresponding to this day is calculated.
- **Finding the right index for the current time:** Since the user data only stores the first value in a series of consecutive equal values, the index to access the speed value corresponding to the current time is found using a binary search.

Multimodal Network Modeling and Analysis

A multimodal network consists of multiple networks (called component networks) that have the capability to interact with one another. The component networks can differ in their key characteristics. The model for a multimodal network should have the ability to represent the specific features of components while providing a means to aggregate all component networks by modeling the interactions across networks.

An example of a multimodal network is a public transportation system in a metropolitan area. Public transportation systems typically consist of multiple bus and train networks, often managed by different agencies, and each network offering multiple routes. These multimodal transportation networks, more often than not, provide passengers the choice to transfer across modes or routes, thus facilitating an interaction among the networks.

Though transportation networks are commonly used as illustrative examples, multimodal networks are also extensively used to capture the structure of a wide variety of complex networks such as biological and utility networks. Multimodal networks may include multiple networks that belong to the same mode (such as multiple bus networks or multiple water supply networks) or multiple networks of dissimilar nature such as bus and train networks or a water network in conjunction with a sewage network and road network.

An NDM multimodal network consists of a set of component networks. Each component network represents a service network in a transportation mode like road networks, bus networks, rail networks, and pedestrian networks. The road network usually serves as a base component network to which all other components connect. Another component network contains transit data that includes the transit stop details, information on routes, and trip schedules that may be provided by the transit agency that operates the services. The multimodal network is an aggregation of all its component networks connected by transfer links that connect nodes of different modes .

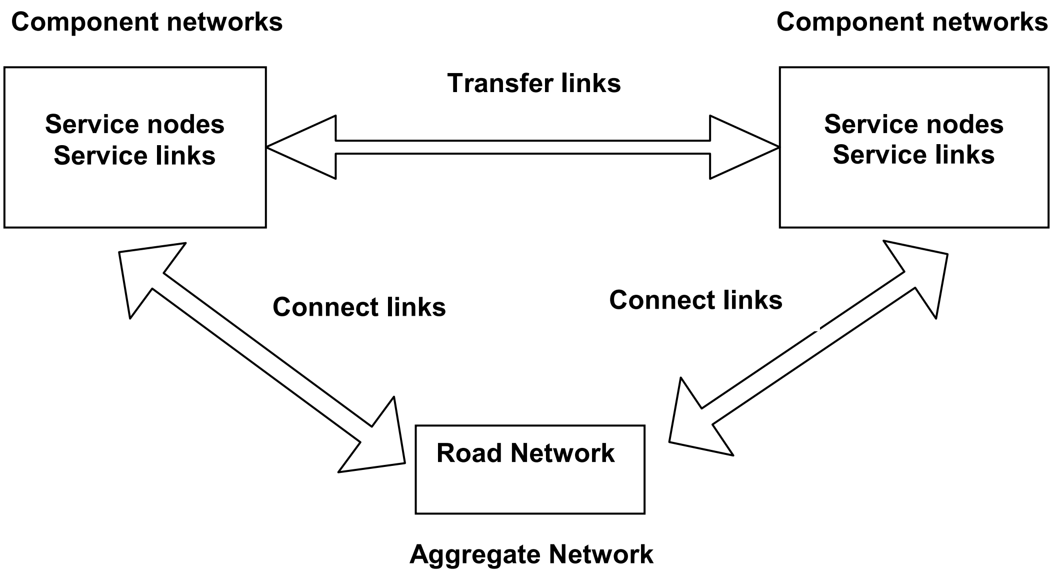


Figure 3: An aggregated multimodal network

Multimodal Network Modeling

NDM uses the same data model for multimodel networks as simple networks. However, multimodel networks involve additional concepts called Component Networks and Aggregated Networks and Associated Metadata Tables.

Component Networks

A component network is represented by NDM node and link tables.

Aggregated Networks

A unique feature of multimodal networks is their capability to interact with one another while maintaining their individual unique characteristics. To analyze a multimodal network it is often necessary to run analysis on the composite network consisting of the component networks. This composite network is formulated by aggregating the component networks; the interactions form the basis of this aggregation process. The combined network is referred to as aggregated network in this document. For example, in a multimodal transportation network that consists of multiple train and bus networks that offers facilities for commuters to transfer across multiple networks, it is often necessary to conduct route planning over the entire aggregated network. This aggregate network consists of all available component networks to make the capabilities of the entire network available to the user.

Metadata Tables

Metadata tables store information regarding the aggregated network and the component networks that are a part of the multimodal network representation. There should be one entry for each aggregate network. The metadata table for aggregate networks contains the following information:

NETWORK_NAME: Unique name of the aggregated multimodal network NODE_TABLE_NAME: Name of the node table of the aggregated network.

LINK_TABLE_NAME: Name of the link table of the aggregated network.

TRANSFER_LINK_TABLE_NAME: Name of the transfer links table.

LINK_TYPE_TABLE_NAME: Name of the link types table.

The metadata table for the component networks in each aggregate network contains the following information:

NETWORK_NAME: Unique name of the aggregated multimodal network.

SUBNETWORK_NAME: Unique name of the component network.

SUBNETWORK_ID: Identifier for the component network.

NODE_TABLE_NAME: Name of the service nodes table.

LINK_TABLE_NAME: Name of the service links table.

Modeling of Interactions across Components: Transfer Links

The interactions across networks provide the basis of aggregation across component networks. The facility to transfer from a bus route to a train route when stops are within walking distance is an example of such an interaction. A water network pumping station located on a road segment provides the relationship across a road network and water network; the pumping station is accessible from the road network by being located on a road segment. Such interactions across networks are modeled using links across network (or specifically nodes that represent the network entities that are related to each other) called transfer links. In the transportation network, transfer links connect the nodes representing the bus and train stops, and in the water network these links connect the pumping station to the road network nodes.

The transfer link table contains the following information:

LINK_ID: Number that uniquely identifies the link.

START_NODE_ID: Node ID of the start node of the link.

END_NODE_ID: Node ID of the end node of the link.

COST: Link Cost.

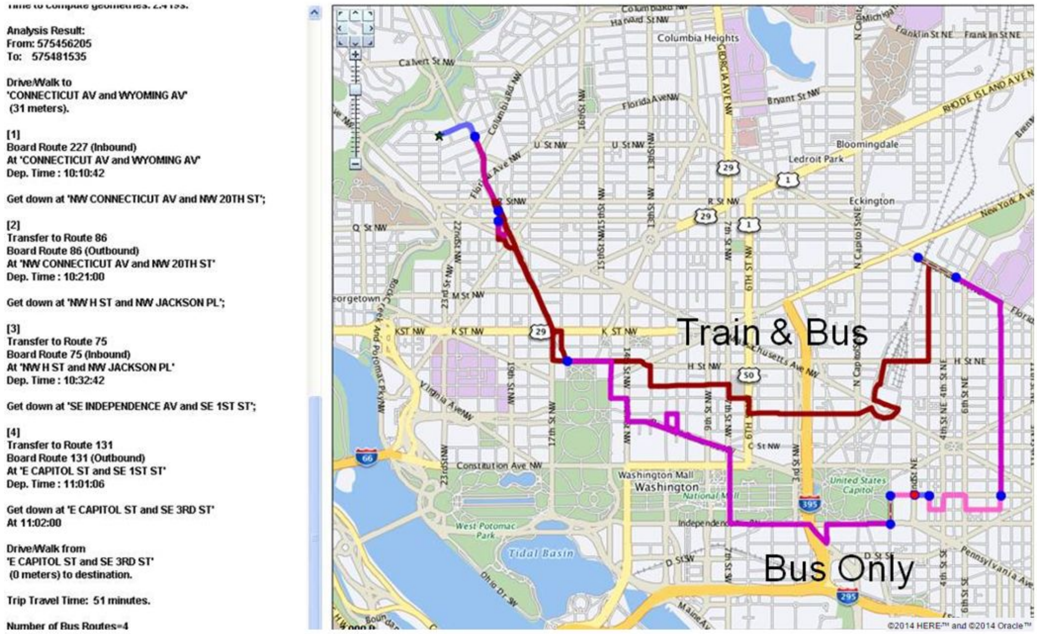
GEOMETRY: Line geometry of the link.

START_NETWORK_ID: Component network ID of start node.

END_NETWORK_ID: Component network ID of end node.

Multimodal Network Analysis

The aggregate multimodal network is considered as a single network for the purpose of computing routes. Typical NDM constraints and cost calculators can be added to the multi-modal network just as with any networks. Figure 4 shows two routes with train-or-bus and bus-only constraints on a multi-modal network.



```
Time to compute geometries: 2.419s.

Analysis Result:
From: 575456205
To:   575481535

Drive/Walk to
'CONNECTICUT AV and WYOMING AV'
 (31 meters).

[1]
Board Route 227 (Inbound)
At 'CONNECTICUT AV and WYOMING AV'
Dep. Time : 10:10:42

Get down at 'NW CONNECTICUT AV and NW 20TH ST';

[2]
Transfer to Route 86
Board Route 86 (Outbound)
At 'NW CONNECTICUT AV and NW 20TH ST'
Dep. Time : 10:21:00

Get down at 'NW H ST and NW JACKSON PL';

[3]
Transfer to Route 75
Board Route 75 (Inbound)
At 'NW H ST and NW JACKSON PL'
Dep. Time : 10:32:42

Get down at 'SE INDEPENDENCE AV and SE 1ST ST';

[4]
Transfer to Route 131
Board Route 131 (Outbound)
At 'E CAPITOL ST and SE 1ST ST'
Dep. Time : 11:01:06

Get down at 'E CAPITOL ST and SE 3RD ST'
At 11:02:00

Drive/Walk from
'E CAPITOL ST and SE 3RD ST'
 (0 meters) to destination.

Trip Travel Time:  51 minutes.

Number of Bus Routes=4
```

Figure 4: Routes of difference preferences (bus only route and train and bus route)

Extending Routing Engine Capabilities with NDM

The routing engine in Oracle Spatial and Graph is a road network web routing service that provides the optimal route with driving directions and route geometry between two or more locations. This routing engine uses the NDM API.

Legal Turn Restrictions

The routing engine enforces turn restrictions based on both physical barriers (e.g: left turns are not allowed because of a median in the road) and legal turn restrictions between two edges. The routing engine understands restricted maneuvers that involve use of two or more links accessed in a particular order. A maneuver is subject to the restriction only if the route uses all links in that particular order. If the route did not access all links or accessed them in an order that is not restricted the restriction would not apply.

This kind of restriction requires that when a link with a restriction is considered, the current route must be backtracked to see if the path leading to the link matches a restriction. The restriction is enforced only if there is a matching path (Figure 5).

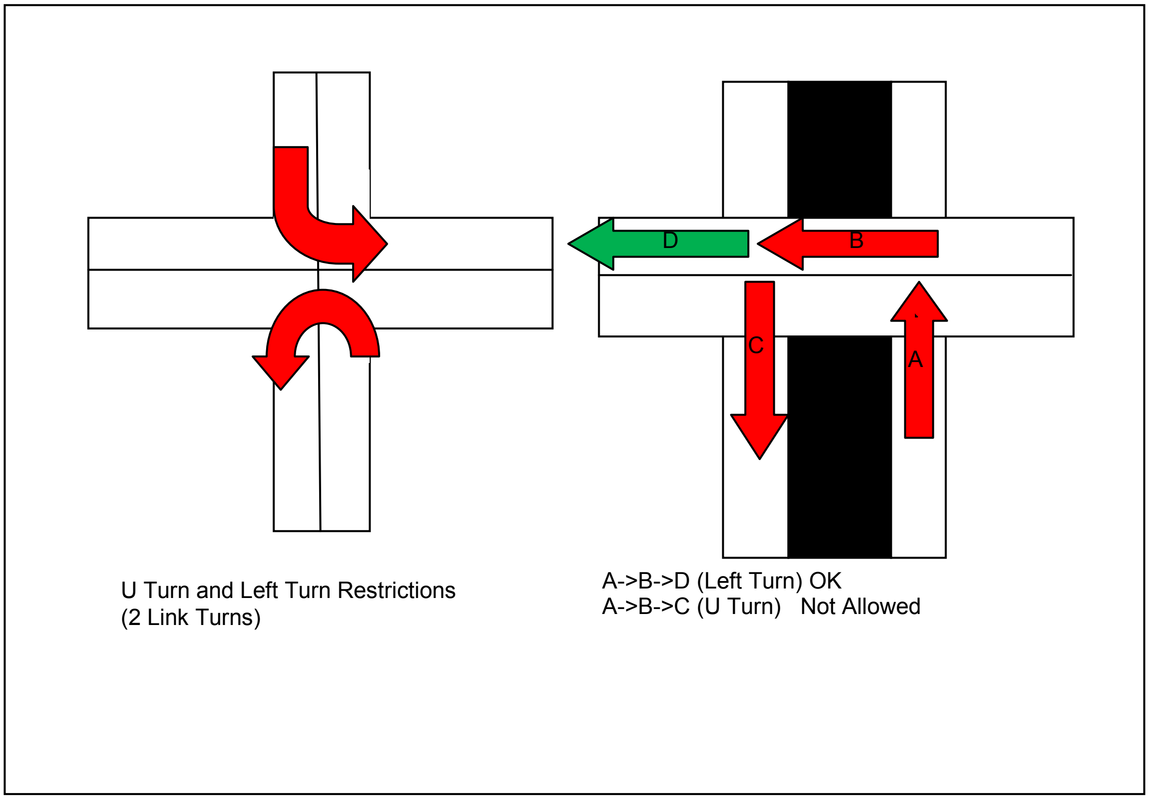

D
B
C
A
U Turn and Left Turn Restrictions
(2 Link Turns)
A->B->D (Left Turn) OK
A->B->C (U Turn) Not Allowed

Figure 5: Examples of Legal Turn Restrictions

Traveling Salesperson Routing

The routing engine implements a traveling salesperson algorithm which attempts to plan an optimal route to all the addresses when given a series of addresses and a set of constraints.

The default behavior of the Oracle Spatial and Graph routing engine is to compute a route between a start point, zero or more intermediate points, and an end point. When multiple points are involved in a route, routes are calculated based on the order as presented to the routing engine and no attempt is made to optimize the ordering of these points.

When using traveling salesperson, the routing engine supports optimizing the order in which the points are visited based on the most efficient path. Only points that are not fixed are considered candidates for reordering.

Constraints the Oracle Spatial and Graph routing engine considers are:

- an open route with a fixed start point
- an open route with a fixed end point
- an open route with a fixed start and end point
- a closed route with a single point acting as both the start and end point

Driving Direction API

This interface is being provided for inclusion in NDM. The feature allows NDM users to implement routing applications using their own specialized sets of constraints. These routing applications can be run against the Oracle Spatial and Graph routing engine schema [6]; the resulting sub-paths can be passed to the new interface to generate an XML document containing the route's driving directions. The driving directions can be generated in English, French, German, Italian or Spanish. The Java API to generate driving directions is in class oracle.spatial.network.apps.router.RouterDirectionsGenerator.

The following shows the driving directions generated by the driving direction API of a shortest path returned from the NDM API:

Step 1. Start out on CORNWALL ST (Going East) 0.1 mi (0.2 min

Step 2. Turn RIGHT onto ARGUELLO BLVD (Going South) 0.1 mi (0.2 min)

Step 3. Turn LEFT onto EUCLID AVE (Going East) 0.3 mi (0.7 min)

Step 4. Turn RIGHT onto SPRUCE ST (Going South) 0.2 mi (0.4 min)

Step 5. Turn LEFT onto GEARY BLVD (Going East) 0.3 mi (0.7 min)

Step 6. Turn RIGHT onto MASONIC AVE (Going South) 0.3 mi (0.6 min)

Step 7. Turn LEFT onto TURK ST (Going East) 0.2 mi (0.4 min)

Step 8. Turn RIGHT onto LYON ST (Going South) 0.1 mi (0.2 min)

Step 9. Turn LEFT onto GOLDEN GATE AVE (Going East) 0.1 mi (0.2 min)

Step 10. Turn RIGHT onto BAKER ST (Going South) 0.5 mi (1.3 min)

Step 11. Stay STRAIGHT to go onto BUENA VISTA AVE E (Going Southeast) 0.2 mi (0.4 min)

Step 12. Turn LEFT onto DUBOCE AVE (Going East) 0.1 mi (0.3 min)

Step 13. Turn RIGHT onto CASTRO ST (Going South) 0.1 mi (0.3 min)

Step 14. Turn LEFT onto 14TH ST (Going East) 0.6 mi (1.4 min)

Step 15. Turn RIGHT onto RAMONA AVE (Going South) 0.1 mi (0.6 min)

Step 16. Turn LEFT onto 15TH ST (Going East) 0.1 mi (0.1 min)

Large Scale Drive Time/Distance Analysis

Geomarketing is the integration of geographical intelligence into various aspects of marketing including sales, distribution, and customer relationships. It is often used in the process of planning and implementing marketing activities. Typical geomarketing examples are site selection and geo-targeting. Based on travel distance/time from customers or competitors, business decisions are made to determine the best store locations and most effective product promotions for target customers. This kind of analysis usually requires huge amounts of route computation and could take days for large numbers of customers.

The challenges of performing this kind of analysis include:

- The number of customers is big (tens of millions).
- The number of stores in tens of thousands.
- The customer-oriented marketing requires routes from each customer to all stores.

The traditional approach to performing these calculations is to generate spatial buffers around stores as a primary filter and then compute the optimal route (shortest path query) for each customer to its nearest store (nearest neighbor query).

NDM introduces the network buffer representation (Figure 6) that captures the coverage and cost information on top of network representation. With network buffers NDM can perform large scale drive time and distance analysis in an efficient way. NDM supports both in-memory and persistent network buffers. Once network buffers have been generated, the coverage and cost information is ready. The coverage information quickly answers if a given point is covered in a network buffer or not. The actual cost can be further computed with cost information.

Network buffers are especially suitable for large scale drive time/distance queries. The cost information is accurate as it is based on the actual network representation (unlike spatial approximation!)

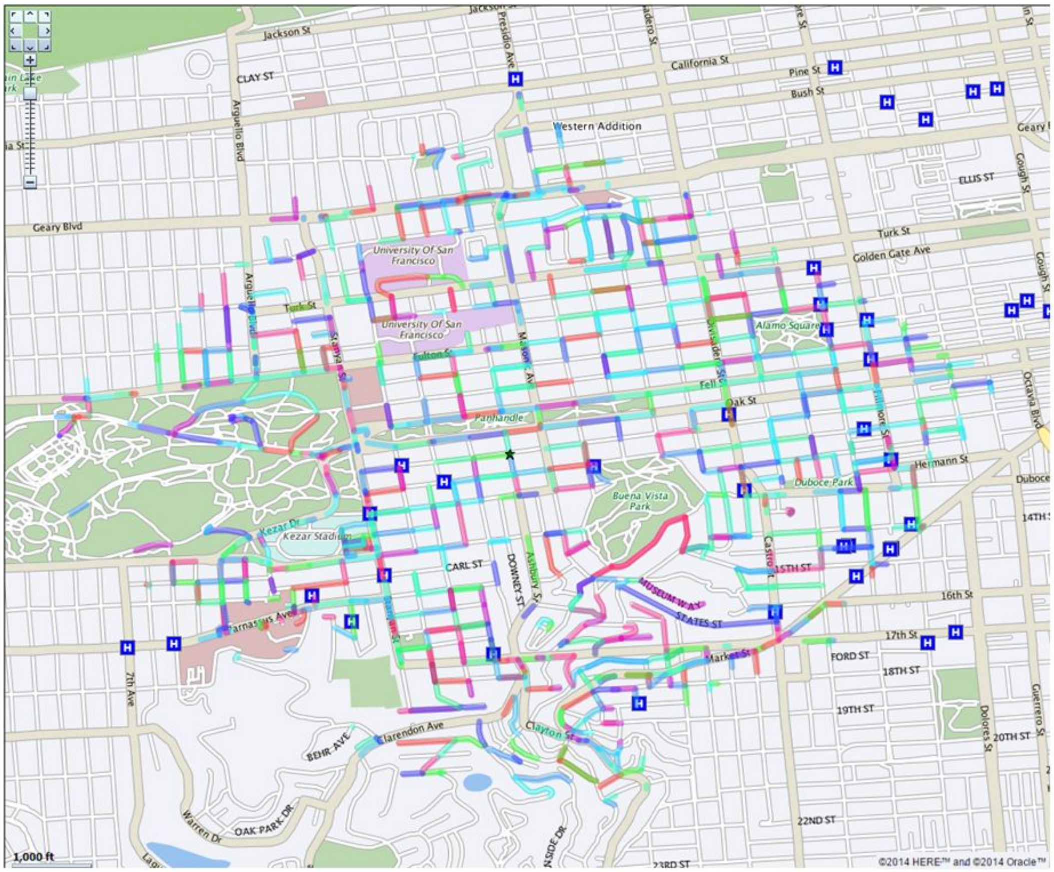


Figure 6: A network buffer (in color lines) of distance 2 KM from the target store.

Each network buffer contains the following information:

Buffer_ID (for different network buffers), Radius(cost), Direction(forward or backward); the coverage and cost information for each network buffer.

Buffer_ID, Link_ID, Start_Percentage, End_Percentage, Start_Cost, End_Cost.

If the network buffers are persistent in database as tables, minimum cost between any points and buffer centers can be effectively and accurately computed as simple table joins. No spatial joins/operators or route computations are needed.

NDM XML API

NDM provides a Java API for network analysis. Many customers use XML API for web service application development. The NDM XML API supports constraints and cost calculators in their Java implementation.

The following XML request illustrates a shortest path request with a prohibited links and nodes constraint (Figure 7).

```
<?xml version="1.0" ?>
<networkAnalysisRequest
   xmlns="http://xmlns.oracle.com/spatial/network"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:gml="http://www.opengis.net/gml">
  <networkName>NAVTEQ_SF</networkName>
  <shortestPath>
    <startPoint>
      <nodeID>199520844</nodeID>
    </startPoint>
    <endPoint>
      <nodeID>199652792</nodeID>
    </endPoint>
    <constraint>
      <className>custom.ProhibitedZoneConstraint</className>
      <parameters>
<prohibitedLinks>198948086 -198948086 198888450 -198888450 198855600 -198855600 -198839651 198839651
198855359 -198855359 198991077 -198843906 198843906 -198991077 -198784737 198784737 199002315 -
199002315 -198774825 198774825 199112417 -199112417 915150608 -915150608 -199101044 199101044
915150609 -915150609 -199019754 198947228 -198947228 199019754 199069927 -199069927 199108382 -
199108382 -199031133 199031133 198961680 -198961680 -198993224 198993224 198917460 -198917460 -
198797704 -198933059 198933059 198797704 -198868150 -198884568 198868150 198884568 198647330 -
198647330 198905052 -198905052 -199094290 199094290 -198610152 198610152 198943159 -198943159
199122328 -199122328 -198950504 198950504 -199042723 198883566 199042723 -198883566
</prohibitedLinks>
```

```
<prohibitedNodes>199459625 199514127 199742411 199722431 199788589 199686046 199424489 199491652
915150607 199381179 199413306 199481657 199474081 199623184</prohibitedNodes>
   </parameters>
  </constraint>
  <subPathRequestParameter>
   <cost> true </cost>
   <isFullPath> true </isFullPath>
   <startLinkIndex> true </startLinkIndex>
   <startPercentage> true </startPercentage>
   <endLinkIndex> true </endLinkIndex>
   <endPercentage> true </endPercentage>
   <pathRequestParameter>
    <cost> true </cost>
    <isSimple> true </isSimple>
    <startNodeID>true</startNodeID>
    <endNodeID>true</endNodeID>
    <noOfLinks>true</noOfLinks>
    <linksRequestParameter>
     <onlyLinkID>true</onlyLinkID>
    </linksRequestParameter>
    <nodesRequestParameter>
     <onlyNodeID>true</onlyNodeID>
    </nodesRequestParameter>
   </pathRequestParameter>
  </subPathRequestParameter>
 </shortestPath>
</networkAnalysisRequest>
```

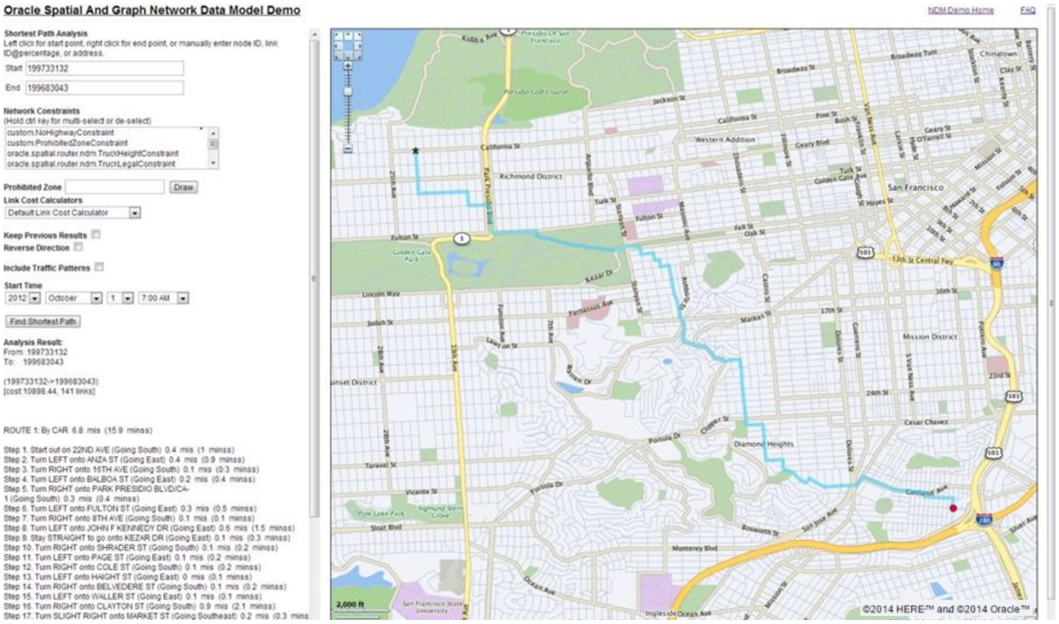
Figure 7: Sample XML shortest path request with prohibited links and nodes constraint (constraint information is shown in bold)

With the enhanced XML API, NDM provides a simple PL/SQL function to send the XML request through HTTP Post method and to return the response as an XMLTYPE. This enables database users to use PL/SQL to call the deployed NDM XML service.

NDM Demo Kits

A web-based viewer (Figure 8) is provided as a demo to help users visualize load-on-demand analysis results. This viewer is implemented in a 3-tiered architecture with the JavaScript GUI (web browser) built with Oracle MapViewer JavaScript API to handle user interactions, NDM analysis engine deployed in the middle tier as a java servlet, and the network data stored in the database backend.

To install the viewer, first download sample network data from HERE. Once the sample network is populated and registered in the database, deploy the demo ear file in WebLogic Server. To use the viewer, first enter database connection information for the database where the network data model (including partition information) is stored. You need to have a MapViewer service running on the same machine as your NDM servlet. For information on how to install and use MapViewer, see the Oracle Fusion Middleware User's Guide to Oracle MapViewer. You can perform various analysis operations on the selected network and use Oracle Map viewing functions for visualizing analysis results. A text area prints out analysis result and statistics. Figure 8 shows a shortest path analysis result.



Oracle Spatial And Graph Network Data Model Demo

Shortest Path Analysis
Left click for start point, right click for end point, or manually enter node ID, link ID@percentage, or address.

Start: 199733132
End: 199683043

Network Constraints
(Hold ctrl key for multi-select or de-select)

- custom.NoHighwayConstraint
- custom.ProhibitedZoneConstraint
- oracle.spatial.router.ndm.TruckHeightConstraint
- oracle.spatial.router.ndm.TruckLegalConstraint

Prohibited Zone

Link Cost Calculators: Default Link Cost Calculator

Keep Previous Results □
Reverse Direction □

Include Traffic Patterns □

Start Time: 2012 October 1 7:00 AM

```
Analysis Result:
From: 199733132
To:   199683043

(199733132->199683043)
[cost:10898.44, 141 links]


ROUTE 1: By CAR 6.8 mis (15.9 minss)

Step 1. Start out on 22ND AVE (Going South) 0.4 mis (1 minss)
Step 2. Turn LEFT onto ANZA ST (Going East) 0.4 mis (0.9 minss)
Step 3. Turn RIGHT onto 16TH AVE (Going South) 0.1 mis (0.3 minss)
Step 4. Turn LEFT onto BALBOA ST (Going East) 0.2 mis (0.4 minss)
Step 5. Turn RIGHT onto PARK PRESIDIO BLVD/CA-
1 (Going South) 0.3 mis (0.4 minss)
Step 6. Turn LEFT onto FULTON ST (Going East) 0.3 mis (0.5 minss)
Step 7. Turn RIGHT onto 8TH AVE (Going South) 0.1 mis (0.1 minss)
Step 8. Turn LEFT onto JOHN F KENNEDY DR (Going East) 0.6 mis (1.5 minss)
Step 9. Stay STRAIGHT to go onto KEZAR DR (Going East) 0.1 mis (0.3 minss)
Step 10. Turn RIGHT onto SHRADER ST (Going South) 0.1 mis (0.2 minss)
Step 11. Turn LEFT onto PAGE ST (Going East) 0.1 mis (0.2 minss)
Step 12. Turn RIGHT onto COLE ST (Going South) 0.1 mis (0.2 minss)
Step 13. Turn LEFT onto HAIGHT ST (Going East) 0 mis (0.1 minss)
Step 14. Turn RIGHT onto BELVEDERE ST (Going South) 0.1 mis (0.2 minss)
Step 15. Turn LEFT onto WALLER ST (Going East) 0.1 mis (0.1 minss)
Step 16. Turn RIGHT onto CLAYTON ST (Going South) 0.9 mis (2.1 minss)
Step 17. Turn SLIGHT RIGHT onto MARKET ST (Going Southeast) 0.2 mis (0.3 mins
```

Figure 8. NDM Demo Kits

Conclusion

Network Data Model graph lets you store and manage large volumes of network data to perform network-based analyses such as finding the shortest path between two nodes, finding all nodes within some distance from a node, and discovering the nearest neighbors to a node. The feature and time modeling enhancements to NDM in Oracle Spatial and Graph \simplify application development and support real-world analysis by moving complex spatial logic into the database.

The ability to model objects of interest along the network as features, such as restaurants and hotels on a road network, and switches on an electrical network, enables users to use terminology with which they are familiar. Temporal modeling and multimodal network modeling support allow for the modeling of transportation networks that consist of multiple modes such as buses, subways, and commuter rail lines.

To learn more about the Network Data Model graph and other Oracle Spatial and Graph technologies visit: http://www.oracle.com/technetwork/database/options/spatialandgraph/

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

- blogs.oracle.com/oracle
- facebook.com/oracle
- twitter.com/oracle
- oracle.com

Integrated Cloud Applications & Platform Services

Oracle Spatial and Graph Network Data Model Graph
November 2016
Author: Jack Chenghua Wang
Contributing Authors: Huiling Gong, Betsy George, Chuck Freiwald