Oracle Spatial – Improving Quality of Data in New Zealand Post

David Thornburrow
New Zealand Post

Introduction

This paper is a real life example of how Oracle Spatial has made it possible for New Zealand Post (NZ Post) to advance confidently into the future with the best Data Management System, knowing that the address data is the highest level of quality possible. Following is a detailed account of NZ Post's approach and solution to Data Management Systems and an overview of Oracle Spatial with relevant examples.

Data Management Systems are only as good as the quality of data within them. NZ Post's address data was hampered by the lack of quality, consistent data, and limited postcode usage. This meant NZ Post remained dependent on manual mail sorting processes to meet delivery targets.

In order to tackle this situation NZ Post established a program of work to scope, design and build a robust data management system utilising spatial data and analysis tools to capture and maintain address data – National Postal Address Database (NPAD).

Oracle Spatial was the platform of choice to store and maintain address data. ESRI application was used to manage data, integrated with Oracle Spatial using ArcSDE. Bulk updates, external data loads and crystal reports make use of Oracle Spatial Operators and Functions. This enforces correct allocation of postcode suburb/city names while allowing the user to visually view the data. The solution has given NZ Post a reliable tool to maintain a core asset for a consistent, measurable and known level of data quality.

New Zealand Post

Every year NZ Post delivers over 1 billion items of mail to around two million delivery points in New Zealand. The postie is the last step in a hugely complex process that involves multiple levels of mail collection, sort, transport and delivery.

Like most postal authorities the world over, NZ Post had traditionally captured address data directly from a widely distributed delivery network applying only ad-hoc quality management. As a result NZ Post's ability to achieve maximum processing and delivery efficiencies and fully exploit commercial data opportunities has been impaired. In fact the lack of quality and consistent address data together with limited postcode usage meant that NZ Post remained heavily dependent on manual mail sorting to meet delivery targets. Mail re-processing resulting from rejections, defaults and miss-sort costs NZ Post dearly every year.

NPAD is a completely new model for address management within NZ Post. It is part of a wider initiative to create a more efficient mail sort capability and distribution network.

A single source of addressing truth was essential to support NZ Post today and in the future. NPAD was also important for two other reasons. NZ Post is about to launch a new postal address standard in New Zealand and an entirely new spatially modelled postcode system. For a country that is unfamiliar with postcodes as a concept, NPAD will play a crucial role in efforts to get New Zealanders to use and benefit from postcodes.

Implicit quality improvement processes including spatial capture, attribute scrubbing and the assigning of quality codes on all address data has been a key goal. Every address in NPAD has a consistent, measurable and known level of quality. In order to create an authoritative addressing database several objectives were identified:

- Creation of a uniform view of address data throughout NZ Post that complies fully with the NZ postal addressing standard.
- Removal of duplication of effort to manage NZ Post's core Delivery and Processing address information.
- Development of continuous quality improvement processes including robust capture, use and maintenance of addressing data.
- Verification of NZ Post address data against authoritative external sources of legal addresses (e.g. local authorities).
- Maximising the amount of mail that can be sorted down to delivery round / PO Box range by mail sorting technology.

- Reduction of rework and double handling of mail within NZ Posts network.

The aim of NZ Post for all software solutions is that the solutions contribute to a Service Orientated Architecture. This is one of the seven core architectural principles adopted in the design and build of NPAD:

1. Service Orientated Architecture
2. Loose Coupling
3. Performance
4. Business Continuity
5. Reuse
6. Ease of Update
7. Integration

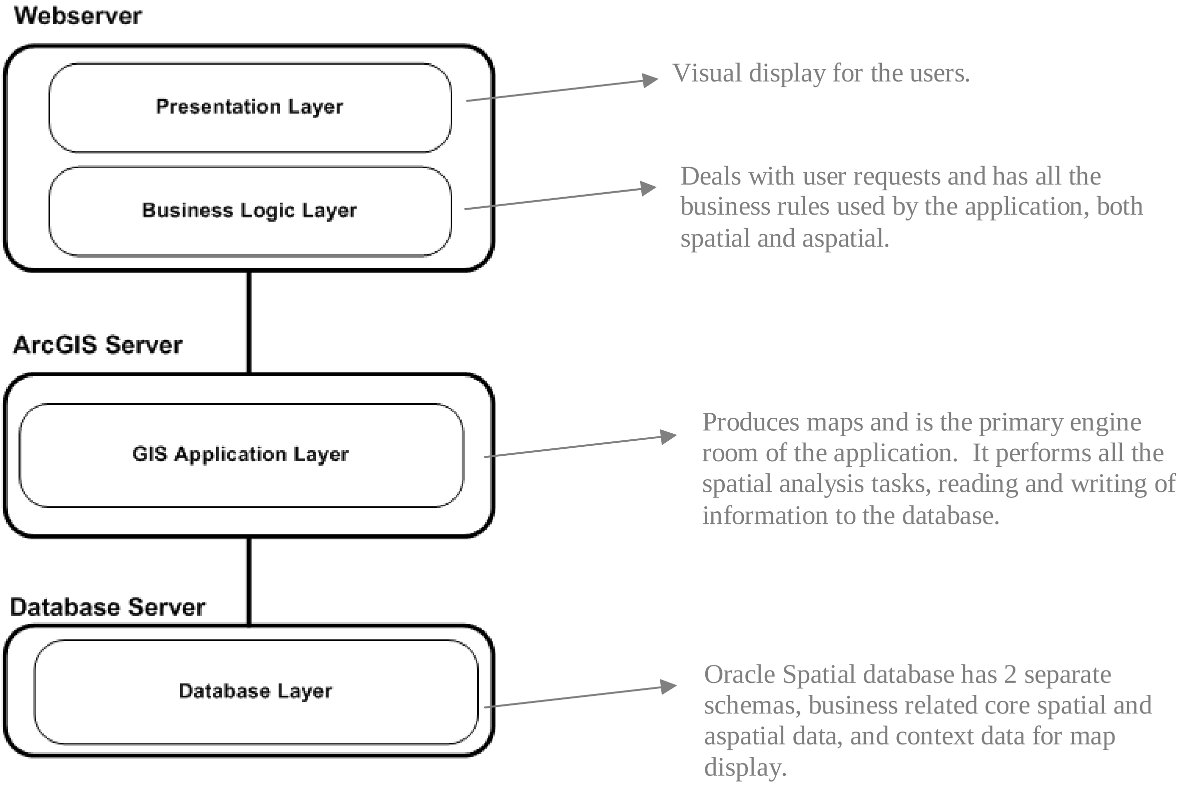
This ensures NZ Post can provide re-usable business services and capable of being extended to other business services in the future.

To meet this challenge NZ Post gathered the best local expertise in the industry with its own business experts and using the Rational Unified Process software development methodology, designed a solution. The outcome was an n-tier application comprising data layer (Oracle Database Server, ESRI ArcSDE Server), application or business logic layer (ESRI ArcGIS Server & business logic) and presentation layer (.Net application, Microsoft IIS HTTP Server).

An early decision was made to build a web based application utilising a services orientated architecture, ideally making use of web services to allow other, as yet undetermined systems, take advantage of the solution outputs.

NZ Post standards required the employment of an n-tier software architecture that separated the various layers of the solution. This resulted in both logical and physical separation as shown in Figure 1.

Figure 1 – NPAD Layer Description


Webserver
Presentation Layer
Business Logic Layer
ArcGIS Server
GIS Application Layer
Database Server
Database Layer
Visual display for the users.
Deals with user requests and has all the business rules used by the application, both spatial and aspatial.
Produces maps and is the primary engine room of the application. It performs all the spatial analysis tasks, reading and writing of information to the database.
Oracle Spatial database has 2 separate schemas, business related core spatial and aspatial data, and context data for map display.

Within the solution there are two main integration points, they are:

1. Integration of external data into the NPAD database
2. Application to Application data synchronisation

For Internal Application to Application (A2A) integration, NZ Post operates an Enterprise Application Integration (EAI) environment based upon Microsoft BizTalk server 2004. All A2A integration between

NPAD and its precursor, Walrus (Urban Addressing), is via the EAI environment. Business rules are applied to the generated message streams via this engine. This ability allows the translation from application to application to be externalised from the direct application and therefore promotes a higher level of abstraction and re-use. The standards around this environment are web services and XML. The application provides web services and/or xml messages based upon business rules internally in the solution.

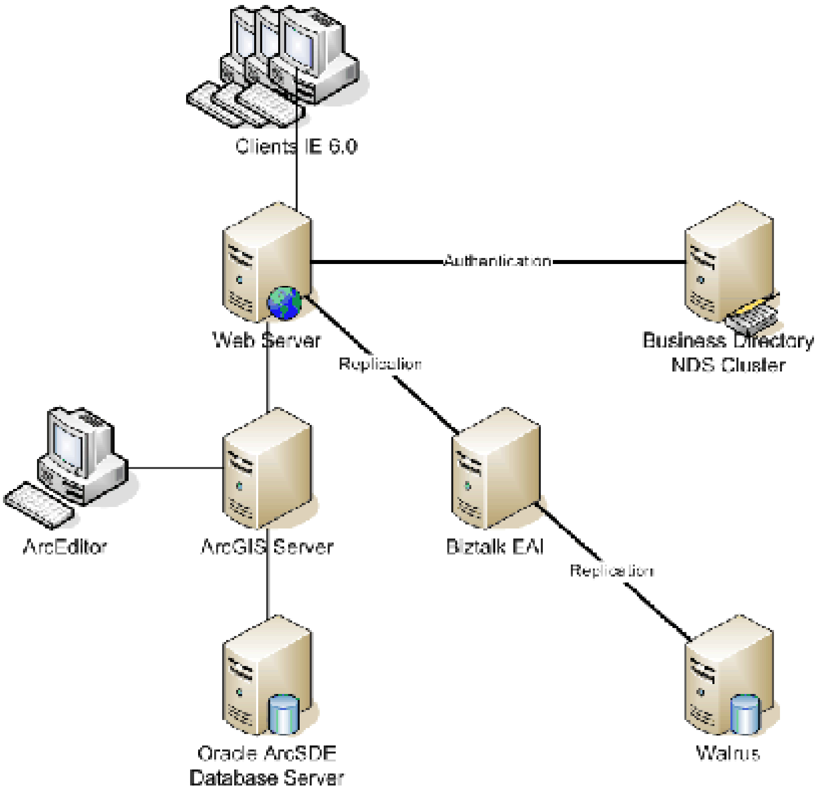
A key requirement of the solution is that NPAD is not to be affected by outages of related systems and therefore the system should be designed to allow for this scenario by using asynchronous processes to replicate data to Walrus.

Replication between NPAD and Walrus is two way and a replication worker component was designed to operate at the business logic layer and consumed web services exposed by Biztalk to pass addressing information that was in turn passed to Walrus. A handshake between the replication worker and Biztalk ensured that the data was successfully passed.

The replication worker also exposed a webservice which in turn is consumed by Biztalk to receive confirmation messages & legacy maintained data back from Walrus

Figure 2 shows the relationship between the NPAD technology stack and surrounding systems.

Figure 2 - NPAD and Surrounding Systems

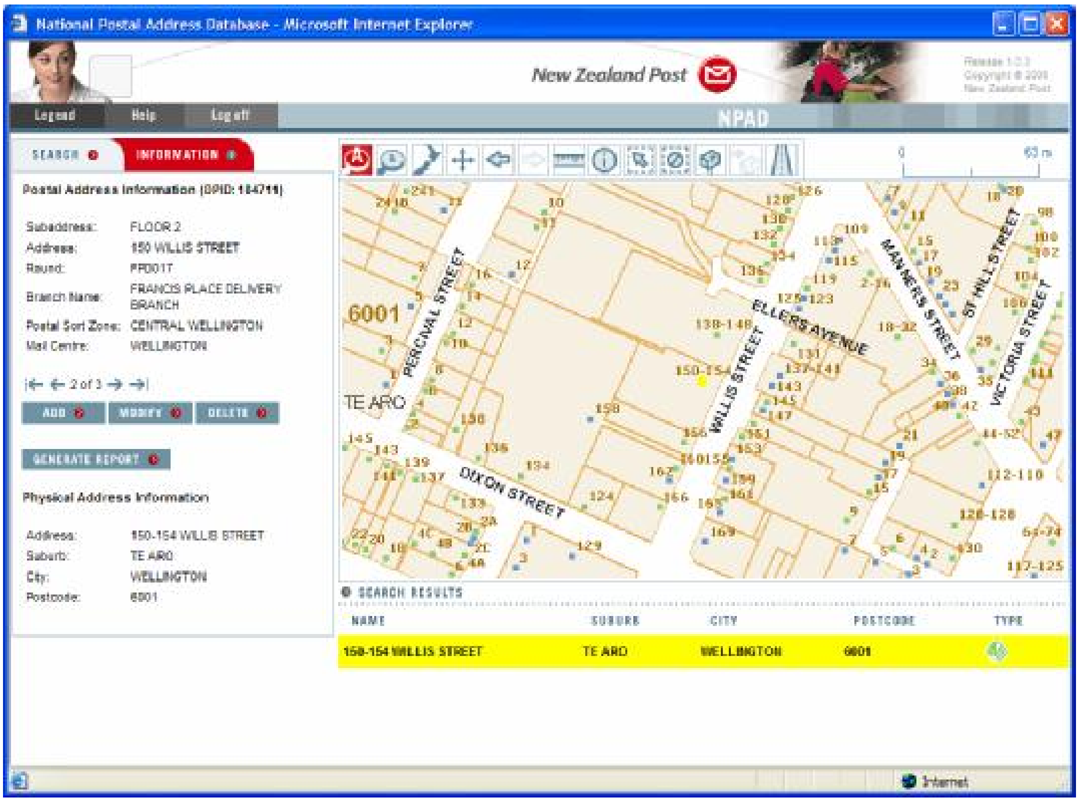


The application interface was designed with ease of use in mind and a focus on rapidly completing routine business processes with a high degree of accuracy ensuring data quality. Key aspects of the interface design include search and retrieval of address and street features, postal address capture, street capture, and movement of postal addresses.

Predictive search technology was deployed to enable rapid and accurate retrieval of information and dramatically improves the success rate for entering in correct data for searching purposes. Comprehensive reports are generated in pdf format that include the current map display and relevant detail depending on the report selected. This enables users and managers alike to visualise information in a way that has not previously been possible.

Figure 3 shows the interface being used to find a specific address to be updated. Map navigational tools are provided to further refine the map display if necessary.

Figure 3 - NPAD Interface



National Postal Address Database - Microsoft Internet Explorer

Postal Address Information (DPID: 184711)

Subaddress:	FLOOR 2
Address:	150 WILLIS STREET
Round:	FP001T
Branch Name:	FRANCIS PLACE DELIVERY BRANCH
Postal Sort Zone:	CENTRAL WELLINGTON
Mail Centre:	WELLINGTON

2 of 3

Physical Address Information

Address:	150-154 WILLIS STREET
Suburb:	TE ARO
City:	WELLINGTON
Postcode:	6001

SEARCH RESULTS

NAME	SUBURB	CITY	POSTCODE	TYPE
150-154 WILLIS STREET	TE ARO	WELLINGTON	6001	

Oracle Spatial

Oracle Spatial is an integrated set of functions and procedures designed to make spatial data management easier and more natural to users of location-enabled and Geographic Information System (GIS) applications. Oracle Spatial provides users with direct access to spatial data using interfaces and utilities such as JDBC, XML, JSQL, OCI, PL/SQL, and SQL*Plus. Oracle Spatial is also fully integrated with Oracle Tools and Applications.

Oracle Spatial requires Oracle9i Enterprise Edition and incurs extra licensing costs. Oracle Spatial is the full fledged enterprise wide spatial information system. Oracle also provides Oracle Locator which is a cut down version f Oracle Spatial and requires only 9i standard or enterprise editions. It offers a subset of the Oracle Spatial capabilities at no extra cost.

Spatial data is any data with a location component. GIS applications are generally aimed at a totally geographic view of the data. They include data capture, data editing, map generation, image processing, image classification, data conversion, data conflation and data analysis. The location-enabled application uses only a minimal part of the geographic processing capability to enhance an otherwise non-geographic application. Once this data is stored in an Oracle database, it can be easily managed, maintained and related to all the other data stored in the database.

Oracle Spatial consists of the following components:

- A schema (mdsys) that prescribes the storage, syntax, and semantics of supported geometric data types.
- Spatial indexing mechanisms.
- A set of spatial operators and functions for performing area-of-interest queries, spatial join queries, and other spatial analysis operations.
- Administrative utilities.

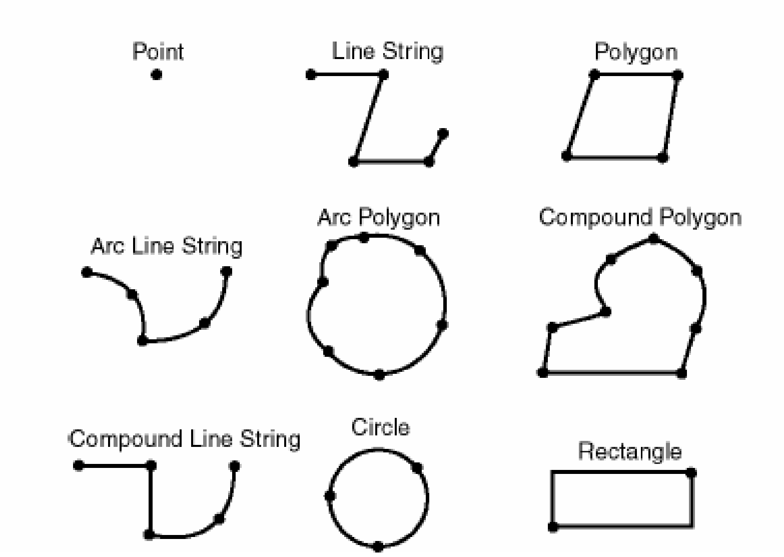
Before creating Oracle Spatial data types, the mdsys schema must be created. This mdsys user stores the spatial data type “sdo_geometry”. There is no spatial table, only an ordinary Oracle table that contains one or more SDO_GEOMETRY columns. The sdo_geometry object will contain an entire geometry in a single row, a single column of an Oracle table.

The spatial component of a spatial feature is the geometric representation shape in some coordinate space. This is referred to as its geometry.

Geometry Types

A geometry is an ordered sequence of vertices that are connected by straight line segments or circular arcs. The semantics of the geometry are determined by its type. Oracle Spatial supports several primitive types and geometries composed of collections of these types, including two-dimensional:

- Points and point clusters
- Line strings
- *n*-point polygons
- Arc line strings (All arcs are generated as circular arcs.)
- Arc polygons
- Compound polygons
- Compound line strings
- Circles
- Optimized rectangles



Data Model

The spatial data model is a hierarchical structure consisting of elements, geometries, and layers, which correspond to representations of spatial data. Layers are composed of geometries, which in turn are made up of elements.

Query Model

Oracle Spatial uses a two-tier query model to resolve spatial queries and spatial joins. The term is used to indicate that two distinct operations are performed to resolve queries. The output of the two combined operations yields the exact result set. The two operations are referred to as primary and secondary filter operations.

The primary filter permits fast selection of candidate records to pass along to the secondary filter. The primary filter compares geometry approximations to reduce computation complexity and is considered a lower-cost filter. Because the primary filter compares geometric approximations, it returns a superset of the exact result set.

The secondary filter applies exact computations to geometries that result from the primary filter. The secondary filter yields an accurate answer to a spatial query. The secondary filter operation is computationally expensive, but it is only applied to the primary filter results, not the entire data set.

Oracle Spatial uses a spatial index to implement the primary filter. Oracle Spatial does not require the use of both the primary and secondary filters.

Creating Spatial Columns

Below is an example of adding a spatial component to a postcode table, note that it uses the standard Oracle table creation syntax, but uses the new Oracle data type mdsys.sdo_geometry.

```
create table postcode
(
   postcode varchar2(4) not null,
   source_id integer not null,
   ps_zone_id integer,
   last_mod_date date default sysdate not null,
   user_id varchar2(30),
  spatial_postcode mdsys.sdo_geometry
)
tablespace npad_static_spatial_tables
/
```

Sdo_geomtry object has 5 fields, 2 number types and 3 are object types:
sdo_gtype number
sdo_srid number
sdo_point sdo_point_type
sdo_elem_info mdsys.sdo_elem_info_array
sdo_ordinates mdsys.sdo_ordinates_array

Oracle Spatial defines these data types as:
Sdo_point_type as OBJECT (
X number,
Y number,
Z number);

Sdo_elem_info array as varray (1048576) of number;

Sdo_ordinates_info array as varray (1048576) of number;

Before storing spatial data within a table, insert the sdo_geomtry metadata. dba_sdo_geom_metadata contains metadata information for all spatial tables. Users are responsible for populating this metadata. For each spatial column, insert an appropriate row into the user_sdo_geom_metadata view. Oracle Spatial populates two other views, all_sdo_geom_metadata and dba_sdo_geom_metadata.

The metadata view has four columns:
table_name varchar2(32),
column_name varchar2(32),
diminfo mdsys.sdo_dim_array,
srid number

Using the example of creating a postcode as a spatial polygon.

```
insert into user_sdo_geom_metadata values (
        'postcode',
        'spatial_postcode',
        mdsys.sdo_dim_array (mdsys.sdo_dim_element('x', 1945500, 3070500,
0.001),mdsys.sdo_dim_element('y', 5278500, 6818600, 0.001)),
        335872);
```

The table_name column contains the name of a feature table, such as postcode, that has a column of type sdo_geometry.

The column_name column contains the name of the column of type sdo_geometry. for the postcode table, this column is called spatial_postcode.

The diminfo column is a varying length array of an object type, ordered by dimension, and has one entry for each dimension. the sdo_dim_array type is defined as follows:

The sdo_dim_element type is defined as:

```
sdo_dim_element as object (
sdo_dimname varchar2(64),
sdo_lb number,
sdo_ub number,
sdo_tolerance number);
```

The sdo_dim_array instance is of size *n* if there are *n* dimensions. that is, diminfo contains 2 sdo_dim_element instances for two-dimensional geometries, 3 instances for three-dimensional geometries, and 4 instances for four-dimensional geometries. Each sdo_dim_element instance in the array must have valid (not null) values for the sdo_lb (lower bound ordinate measure), sdo_ub (upper bound ordinate measure), and sdo_tolerance attributes. The postcode table is a two-dimensional geometry, that has a co-ordinate range of 1945500, 3070500 for the x and 5278500, 6818600 for the y with a tolerance of 0.001 (0.1 millimetres).

The srid number column should contain either the srid value for the coordinate system for all geometries in the column, or null if no specific coordinate system should be associated with the geometries. For the postcode table srid 335872 is used which stands for “New Zealand Map Grid”.

Indexing of Spatial Data
The spatial index, like any other index, provides a mechanism to limit searches, but in this case based on spatial criteria such as intersection and containment. A spatial index is needed to, find objects within an indexed data space that interact with a given point or area of interest (window query) and find pairs of objects from within two indexed data spaces that interact spatially with each other (spatial join).

The entries in the spatial index are dependent on the location of the geometries in a coordinate space, but the index values are in a different domain. Index entries may be ordered using a linearly ordered domain, and the coordinates for a geometry may be pairs of integer, floating-point, or double-precision numbers.

Creating spatial indexes is performed by defining the indextype as mdsys.spatial_index. A spatial index can be created simply by using the indextype mdsys.spatail_index.

> *create index postcode_sidx on postcode(spatial_postcode) indextype is mdsys.spatial_index;*

With most indexes it is ideal to pass some of the optional parameters. A full list of parameters can be found in the “Oracle Spatial Users Guide”.

Here is an example of a spatial postcode table, with a spatial column spatial_postcode.

> *create index postcode_sidx on postcode(spatial_postcode) indextype is mdsys.spatial_index*
> *parameters (' sdo_indx_dims=2 layer_gtype="collection"');*

Sdo_indx_dims=2 causes the first two dimensions to be indexed. Since this is a 2-dimensional object we only want the first 2 dimensions to be indexed.
Layer_gtype="collection" geometry is a heterogeneous collection of elements, the polygons in the collection can be disjoint. Collection is a superset that includes all other types.

Spatial Relations and Filtering
Oracle Spatial uses secondary filters to determine the spatial relationship between entities in the database. The spatial relation is based on geometry locations. The most common spatial relations are based on topology and distance.

The distance between two spatial objects is the minimum distance between any points in them. Two objects are said to be within a given distance of one another if their distance is less than the given distance.

To determine spatial relations, Oracle Spatial has several secondary filter methods:

- The sdo_relate operator evaluates topological criteria.
- The sdo_within_distance operator determines if two spatial objects are within a specified distance of each other.
- The sdo_nn operator identifies the nearest neighbours for a spatial object.

Three simple uses of spatial operators and functions:

sdo_relate returns all address where that spatially interact with postcode 6002. This uses Oracle primary filter to first filter the postcode table to postcode 6002 and then perform the more spatial intensive operation using the secondary filter.

```
select ba.address
from base_address ba, postcode pc
where pc.postcode = 6002 and
  sdo_relate(ba.spatail_address,pc.spatial_postcode,
  'mask=anyinteract querytype=window') = 'true');
```

sdo_distance returns the distance between an address and the road it is linked with. Note all these operators have optional parameters, below the tolerance has been defined and the unit value.

```
select ba.address, sdo_geom.sdo_distance(ba.spatial_address, rs.spatail_road_segment, 0.001,
'unit =km')
from base_address ba, road_segment rs
where ba.road_seg_sufi = rs.road_seg_sufi;
```

sdo_nn returns the nearest neighbour between an address and the next closest address. This can be useful when comparing delivery attributes of addresses within an area. The below operator returns the 6 closest neighbours, a subquery is also used to return the distance of the addresses.

```
select   ba.address,  ba2.address,  (sdo_nn_distance(1) * 1000) as distance
from  base_address   ba, base_address   ba2
where  ba.addr_sufi != ba.addr_sufi and
sdo_nn(ba2.spatial_address, ba.spatail_address, 'sdo_num_res=6 unit=km', 1) = 'TRUE';
```

Geocoding

Geocoding is the process of converting tables of address data into standardised address, location, and possibly other data. The result of a geocoding operation is the pair of longitude and latitude coordinates that correspond with the input address or location.

A geocoded address has the option to perform proximity or location queries using a spatial engine, such as Oracle Spatial, or demographic analysis using tools and data from Oracle's business partners. In addition, geocoded data can be used with other spatial data such as block group, postal code, and county code for association with demographic information. Results of analyses or queries can be presented as maps, in addition to tabular formats, using third-party software integrated with Oracle Spatial.

Oracle Spatial is integrated with all major geocoding service providers. The usual and recommended approach for application developers is to use the API for the geocoding provider to obtain a geocoded result (longitude/latitude coordinate pair) for an address, and then use these coordinates to construct an mdsys.sdo_ geometry object for input to a spatial operator, function, or procedure.

Primary Benefits Associated with Oracle Spatial

Previously, spatial data was stored in file-based systems or proprietary applications. This resulted in separate databases for geographic features and attributes. By storing spatial and business information within an Oracle database, users benefit from the scalability, reliability, openness and performance of Oracle Server Technology. Other benefits include:

- Better data management for spatial data. Users gain access to fully functional spatial information systems based on industry standards (OpenGIS Consortium (OGC)) with an open interface to their data (e.g. SQL).

- Spatial data is now stored in enterprise-wide DBMS, thereby spatially enabling many more applications.

- Reduced complexity of systems management by eliminating the hybrid architecture of traditional GIS-based data management schemes.

- Proprietary data structures are avoided by using an open SQL platform thereby allowing for the seamless integration of GIS and spatial stores.

- Enables Internet and Mobile Location services.

Oracle Spatial and NZ Post

For NZ Post NPAD incorporates a spatial data representation model for all address elements including addresses (point), suburb/locality, city, postcode, postal sort zone (polygon), street (polyline). This approach enables NZ Post to apply address elements in a consistent and coordinated manner and facilitate exchange of data with local government and Land Information New Zealand. The model also enables NZ Post to match postal addresses to legal addresses provided by local authorities.

The principal benefits from the aforementioned address management methodology are simply the ability to apply key address elements that have a fundamental location element in a consistent manner using either point-in-polygon or nearest logic. The other parameters that are less objectively defined such as sub-address details are restricted through the application interface to pick-lists and open fields containing business rules around data type and domain.

A consequence of this approach is the high degree of reliance NZ Post place on all polygon data in NPAD including suburb/locality and city data. Apart from a few exceptions, suburb and locality information in New Zealand are typically not well demarcated by councils. To get around this situation NZ Post has partnered with the New Zealand Fire Service to define common-use suburb and locality boundaries. To produce this definition NZ Post has used a large volume of mail item images captured from mail sorting machines, to identify common use suburb and locality. These data have enabled the generation of a common use depiction of suburb and localities across the country.

Summary

This project has demonstrated the benefits of applying a geo-spatial data model and quality management application for address data management. Address data being fundamentally spatial by nature lends itself to this model, however the quality of the spatial layers that underpin the data model is crucial to overall success. The service oriented nature of the solution architecture was key to the project and in particular the ability to interface to existing systems managing related information. For NZ Post, NPAD provides a solid foundation on which to build improved business processes and offer an improved level of service.

References

Oracle Spatial - User's Guide and Reference
Release 9.2
March 2002
Oracle

Postal Address Data Management – Adding Quality With Spatial Technology
Matt Lythe and Chris Malcolm
July 2005
New Zealand Post

Copyright ©2005 by David Thornburrow