From Point Cloud to City Model –
The 3D Data Management Lifecycle

GWF 2016 Workshop – 3DCityDB on Oracle

Hans Viehmann
Product Manager EMEA
ORACLE Corporation
May 26, 2016
CityGML in Practice
Using 3DCityDB on Oracle for City Modeling
Agenda

• Hans Viehmann (Oracle)
  – From Point Cloud to City Model – The 3D Data Management Lifecycle
  – Data Conflation, Processing, Publishing

• Stephan Plabst (M.O.S.S.)
  – Simplifying the Management of your 3D Data with Automatic and Scalable Workflows
  – Modeling, Maintenance, Change Detection

• Lutz Ross (virtualcitySYSTEMS)
  – CityGML-based SDIs – Implementation requirements and examples
  – Best Practices and Visualization
City Modeling

• Many business cases have become economically viable
  – Cost of 3D data collection has gone down significantly

• Leading to large scale projects, sometimes country-wide initiatives
  – Eg. in Poland, the Netherlands (3D Pilot NL, AHN-2), Germany (AdV), UK, Ireland, ...
  – Using LiDAR or Photogrammetry for data acquisition

• In Europe partly driven by EU mandates (eg. noise emission)
  – Requiring 3D data for simulation

• Lots of use cases, eg. in Smart Cities track
  – City and urban planning, citizen participation, city marketing, ...
  – Users in Local Government, Telco, Utilities, Public Transport, Public Safety, ...
Visualization
Simple graphical representation – data, not information

Screenshot courtesy of Rico Richter, HPI
Modeling and CityGML

Using the data requires objects

- Modeling required for any kind of analysis beyond visualization
  - Associating objects with geometry, topology, semantics, appearance, ...
- CityGML is established standard for this purpose
  - Information model to represent relevant 3D urban objects
  - Defining classes and relations for these objects
  - XML-based format to exchange and store data (GML3-based application schema)
  - Standardized by OGC, currently in version 2.0.0
  - Can be used to derive logical data model in databases
3DCityDB
Open source data model - www.3dcitydb.org

• Semantically rich, hierarchically structured model
• Five different Levels of Detail (LODs), including textures and facades
• Representation of generic and prototypical 3D objects
• Free, also recursive aggregation of geo objects
• Complex digital terrain models (DTMs)
• Management of large aerial photographs using SDO_GEORASTER objects
• Version and history management
• Matching/merging of building features
• Works with Oracle Spatial and Graph 10gR2, 11g and 12c
City of Berlin – 3D City Model
Implemented on Oracle with 3DCityDB

• 550000 buildings, reconstructed from 2D cadastre and LIDAR data
• Textures extracted from oblique aerial photography
• 2012 Oracle Spatial Excellence Award

Images courtesy of: TU Berlin, Institute for Geodesy and Geoinformation
Why use a database here?

Requires a spatially enabled database

• Data integration with other sources
  – Online availability, including history and metadata
  – Geo-referenced imagery, existing 3D structures, attributes,...

• Fast access to arbitrary part of data set
  – For processing or visualization

• General benefits of mature DBMS
  – Information lifecycle management – data administration, tuning
  – Scaleability – multi-processor support, clustering, ...
  – Executing data-intense logic where the data resides
Oracle Database with Spatial and Graph Option

Oracle Spatial and Graph

- "Points"
- "Lines"
- "Polygons"
- Web Services (OGC)
- SPARQL End Point
- Rasters
- Network Graphs
- Topologies
- RDF Semantic Graphs
- Geocoding, Routing
- Inferencing
- 3D
Spatial Processing in BGT BRAVO Project
Automating data conflation and quality assurance
Spatial Processing in BGT BRAVO Project
Automating data conflation and quality assurance
Creating a 3D model – elementary building blocks

• Simple Surfaces
  – Face = 3D Polygon

• Composite Surface
  – Multiple connected faces

• Simple Solid
  – Closed composite surface

• Composite Solid
  – Multiple connected simple solids

• Extrusion
  – Generating solids from 2D polygons
Other relevant database object types

- **TINs (SDO_TIN)**
  - Separating physical storage from logical structure to support huge datasets
  - Extraction, conversion to geometry
  - Compression for efficient storage

- **Point Clouds (SDO_PC)**
  - Separating physical storage from logical structure to support huge datasets
  - Extraction, conversion to geometry
  - Processing, eg. create TIN, generate contour lines

- **Georeferenced Raster (SDO_GEORASTER)**
  - Separating physical storage from logical structure to support huge datasets
  - Supporting any kind of gridded data
  - Ingesting lots of formats through GDAL
  - Image processing, mosaicking
  - Pyramiding for fast rendering
  - Raster algebra (for NDVI, TCT, or bespoke algorithms) and analytics
Dutch eScience research project on Massive Point Clouds
Project Consortium, led by Peter v. Oosterom, TU Delft

• TU Delft
• NLeSC (Netherlands eScience Center)
• Oracle Corporation (NEDC)
• Rijkswaterstraat
• Fugro B.V.
• AHN-2 Dataset of the entire country
• 6 -10 pts/m²  ➞ 640 billion pts
• 60,185 LAZ files, 987 GB in total, 11.64 TB uncompressed

Image courtesy of: PDOK, NL
Scalability testing with AHN-2 dataset

Performance and best practices

• Testing on Oracle Engineered System (Exadata X4-2)
• Loading all 640bn points from LAZ files in 4:39h
• Using Hybrid Columnar Compression
  – Requires 2.24TB of storage in database, compared to 0.97TB in LAZ files or 12TB (uncompressed) LAS
• Massive parallelization of processing
• Even in single user database environment usually outperforms file system
• Available as Cloud Service
How can this be used?

• Derivation of 3D models
  – Classification, conflation with data from other sources

• Web-based or service-based rendering
  – Visual inspection, etc.
  – using the full resolution of the dataset or parts thereof (pyramiding)

• Large scale data dissemination

• In-database processing and analytics, specifically change detection
  – Change detection in multi-temporal point clouds
  – Particularly for buildings and vegetation
Visualization and interactive analysis
Publish KML in the XDB Repository

```
declare
  result boolean;
  kmlDoc xmltype;
begin
  SELECT xmlElement ("kml",
    xmlAttributes ("http://www.opengis.net/kml/2.2" as "xmlns"),
    xmlElement ("Document",
      xmlElement ("Style",
        xmlAttributes ("BuildingStyle" as "id"),
        xmlElement ("LineStyle", xmlElement ("width", '1'), xmlElement ("color", 'ff0000ff'))),
      xmlElement ("PolyStyle", xmlElement ("color", '7d0000ff'))) ),
  xmlagg (xmlElement ("Placemark",
    xmlElement ("name", 'Building || omlid'),
    xmlElement ("styleUrl", '#BuildingStyle'),
    xmlType (sdo_util.toKMLGeometry (snap_to_ground (sdo_cs.transform (geom, 4327), 0)))) ),
  INTO kmlDoc
FROM buildings_ext;
result := dbms_xdb.createResource ('/public/Buildings/buildings_ext.kml', kmlDoc);
end;
```
Render in KML Viewer
Publish as Linked Data
Using RDF Graph support in Oracle Spatial and Graph

- Support for RDF, OWL, SPARQL, ... is integral part of database
- Using database as triple store
- Allowing RDF views on relational tables
- Standards-compliant SPARQL endpoint, GeoSPARQL support
- SPARQL queries in SQL possible
- Inferencing engine inside database

Semantic Modeling

Protégé
Summary

Managing 3D Data in 3DCityDB on Oracle

• „live“ access to 3D data for visualization and analysis
• Integrated management of all kinds of 2D and 3D data
  – Single source of truth for geospatial and associated attribute data
• Large scale „in-situ“ processing
  – Object recognition, data analysis, ...
• Management and publishing of data and metadata
  – Making data useable through open standards
• Taking advantage of cost saving database functionality
  – Compression, information lifecycle management, ...
More resources

• Further information on oracle.com
  – [www.oracle.com/goto/spatial](http://www.oracle.com/goto/spatial)

• Blogs

• Developer forums on OTN

• LinkedIn community
  – „Oracle Spatial and Graph“ group

• Google+ community
  – „Oracle Spatial and Graph SIG“
Integrated Cloud
Applications & Platform Services