CityGML DBMS storage
3DCityDB implementation

100% CityGML
TU Delft, March 14th, 2011

Javier Herreruela, Thomas H. Kolbe, Claus Nagel, Gerhard König
{javier.herreruela | thomas.kolbe | claus.nagel | gerhard.koenig}@tu-berlin.de
www.igg.tu-berlin.de
3D City Database
General Overview
Motivation for a 3D geo database in Berlin

- Repository for the official 3D city model
  - Complete representation of city topography and landscape
  - Data from various sources (cadastre, planning, architecture, utility networks, etc.)

- Usage of 3D city model for applications like
  - City and Urban Planning
  - Political Issues and Consulting, Civic Participation
  - Marketing, Service, Promotion for companies

- Basis for the Berlin 3D Spatial Data Infrastructure
  - Access through standardized OGC Web Services, Google Earth (KML), online streaming
Berlin 3D Spatial Data Infrastructure

### Services
- **WSS**
- **WCS**
- **WFS / WFS-T**
- **WPVS**
- **W3DS**

### Geodata systems
- **3D geo database**
  - 3D city model (CityGML)
  - DTM (CityGML)
  - Orthophotos
  - Version and history management

### Internet / Berlin governmental network
- **3D City Model Editor**
- **Geo-enabled applications**
- **Google Earth Browser**
- **CAD System (IFC)**
- **Direct access**

### Consumer / Producer
- **CityGML**
  - X3D, VRML, etc
- **CityGML**
  - KML
- **Java client**
  - KML/COLLADA exporter

- **WSS**
- **SQL**
Berlin 3D Spatial Data Infrastructure

**Internet / Berlin governmental network**

- 3D City Model Editor
- Geo-enabled applications
- Google Earth Browser
- CAD System (IFC)
- Direct access

**Services**

- CityGML
- X3D, VRML, etc
- CityGML
- KML
- CityGML

**Java client**

- KML/COLLADA exporter
- SQL

**3D geo database**

- 3D city model (CityGML)
- DTM (CityGML)
- Orthophotos
- Version and history management
3DCityDB V2 is a **free and Open Source 3D geo database** to store, represent, and manage virtual 3D city models

- **CityGML 1.0 (and 0.4.0) compliant**
  - Support for all CityGML feature types
  - Representation of feature geometry in all 5 LODs
  - Support for CityGML’s appearance model

- **Relational database schema for Oracle 10g R2 Spatial or higher**
  - Spatial data types and advanced processing of spatial data
  - Efficient storage and management of raster data
  - Sustainability through Workspace Manager (Version and History Management)
  - Connection possibilities to commercial GIS

- **Open Source** and released under the terms of the **LGPLv3**
  - Developed by IGG, TU Berlin in cooperation with IGG, Univ. of Bonn, lat/lon and 3DGeo (now: Autodesk)
  - direct 3DCityDB support in Autodesk LandXPlorer Professional
  - current developments are also supported by VirtualCitySystems
Functionality of the 3D City Database

**CityGML properties**

- Support for all thematic CityGML modules
- Compliant with CityGML version 1.0.0 & 0.4.0
- Flexible 3D geometries in LOD0 - 4
- Multi-representation of objects wrt. LODs, time, appearance
- Generic (prototypical) 3D objects and attributes
- (recursive) aggregation of objects
- Digital Terrain Models (DTM)
- Appearance information
- Referencing of exterior data sources

**additions**

- Version management and planning alternatives
- Exchange of DTM and aerial images (WebServices)
- Import and export of CityGML files
- Referencing of exterior data sources
- Support for all thematic CityGML modules
Development cycle of the 3D City Database

CityGML

XSD Schema
<xs:complexType name="CityModelType">
<xs:complexContent>
<xs:extension>
...
</xs:extension>
</xs:complexContent>
</xs:complexType>

UML Model

Model simplification

Simplified UML Model

Java binding (JAXB)

Schema-derived classes
public class CityModel {
    ...
}

SQL queries (Imp/Export Tool)

Export data

Import data

Database creation

Mapping classes to tables

Relational schema

SQL DDL statements (JDeveloper)
Development cycle of the 3D City Database

CityGML

XSD Schema
<xs:complexType name="CityModelType">
  <xs:complexContent>
    <xs:extension ...

UML Model

Model simplification

Simplified UML Model

Java binding (JAXB)

Schema-derived classes
public class CityModel {
  ...
}

SQL queries (Imp/Export Tool)

Import data

Export data

Database creation

Mapping classes to tables

Relational schema

SQL DDL statements (JDeveloper)
Simplifications of CityGML’s data model

- CityGML = Application independent Geospatial Information Model
  - Thematic model covers a broad range of application fields
  - Complex aggregations/relations within thematic modules
  - Comparable situation on the geometry level (GML3)

- Analysis of the requirements for the Berlin geo database:
  
  **A simplified schema is already sufficient**

- Adaption of CityGML’s data model
  - Simplification where possible → but still **99% compliant to CityGML**
  - Less number of database tables → simpler relational schema
  - Allows for more efficient querying/processing of database content
Examples for model simplification

- Multiplicities of attributes: $0..* \rightarrow 0..1$
- Cardinalities of associations (and also type of associations): $n:m \rightarrow 1:n \text{ resp. } n:1$
  - aggregation $\rightarrow$ composition
- Recursive aggregations: parent-id und root-id
- Conversion of data types: gml:CodeType, app:Color $\rightarrow$ String
  - GML geometry $\rightarrow$ SDO_GEOMETRY
- Project specific classes and attributes: Orthophotos, Address information, Metadata
Simplifications of CityGML’s geometry model

- CityGML is implemented as GML3 application schema
  - Only a subset of GML3 geometry classes is supported (straight lines, planar surfaces)
Derivation of the relational database schema

CityGML

XSD Schema
<xs:complexType name="CityModelType">
<xs:complexContent>
<xs:extension ...>

UML Model

Model simplification

Simplified UML Model

Mapping classes to tables

Relational schema

SQL DDL statements (JDeveloper)

Java binding (JAXB)

Schema-derived classes
public class CityModel {
...
}

SQL queries (Imp/Export Tool)

Import data

Export data

Database creation

C

3D City Database | CityGML and KML/COLLADA Import/Export Tool

March 16, 2011  Herreruela, Kolbe, Nagel, König | 3D City Database

Institute for Geodesy and Geoinformation Science
Derivation of the relational database schema

- Realisation of geometry model

### GML3 Geometry

<table>
<thead>
<tr>
<th>Geometry</th>
<th>isSolid</th>
<th>isComposite</th>
<th>isTriangulated</th>
<th>Oracle Geometry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polygon, Triangle, Rectangle</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>SDO_GEOMETRY</td>
</tr>
<tr>
<td>MultiSurface</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>NULL</td>
</tr>
<tr>
<td>Composite Surface</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>NULL</td>
</tr>
<tr>
<td>Triangulated Surface</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>NULL</td>
</tr>
<tr>
<td>Solid</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>NULL</td>
</tr>
<tr>
<td>MultiSolid</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>NULL</td>
</tr>
<tr>
<td>Composite Solid</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>NULL</td>
</tr>
</tbody>
</table>
Geometry representation

LOD1 building with appearance information

Representation in 3D geo database?
Geometry representation

```xml
<bldg:lod1Solid>
  <gml:Solid>
    <gml:exterior>
      <gml:CompositeSurface gml:id="lod1Surface">
        <gml:surfaceMember>
          <gml:Polygon gml:id="Left1">
            <gml:exterior>
              <gml:LinearRing gml:id="LeftRing1">
                <gml:posList srsDimension="3"> 0.0 0.0 0.0 10.0
                                              0.0 0.0 10.0 0.0 4.0 0.0 0.0 4.0 0.0 0.0 0.0
                </gml:posList>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </gml:surfaceMember>
        ...  
        <gml:surfaceMember>
          <gml:Polygon gml:id="Roof1">
            <gml:exterior>
              <gml:LinearRing gml:id="RoofRing1">
                <gml:posList srsDimension="3"> 0.0 0.0 4.0 10.0
                                              0.0 4.0 10.0 5.0 4.0 0.0 0.0 4.0 0.0 0.0 4.0
                </gml:posList>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </gml:surfaceMember>
      </gml:CompositeSurface>
    </gml:exterior>
  </gml:Solid>
</bldg:lod1Solid>
```

<table>
<thead>
<tr>
<th>SURFACE_GEOMETRY</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
</tbody>
</table>
Geometry representation

```xml
<building:lod1Solid>
  <gml:Solid>
    <gml:exterior>
      <gml:CompositeSurface gml:id="lod1Surface">
        <gml:surfaceMember>
          <gml:Polygon gml:id="Left1">
            <gml:exterior>
              <gml:LinearRing gml:id="LeftRing1">
                <gml:posList srsDimension="3"> 0.0 0.0 0.0 10.0
                                               0.0 0.0 10.0 0.0 4.0 0.0 0.0 4.0 0.0 0.0 0.0
              </gml:posList>
            </gml:LinearRing>
          </gml:exterior>
        </gml:Polygon>
      </gml:surfaceMember>
      ...
      <gml:surfaceMember>
        <gml:Polygon gml:id="Roof1">
          <gml:exterior>
            <gml:LinearRing gml:id="RoofRing1">
              <gml:posList srsDimension="3"> 0.0 0.0 4.0 10.0
                                               0.0 4.0 10.0 5.0 4.0 0.0 5.0 4.0 0.0 0.0 5.0 4.0
              </gml:posList>
            </gml:LinearRing>
          </gml:exterior>
        </gml:Polygon>
      </gml:surfaceMember>
    </gml:CompositeSurface>
  </gml:exterior>
</gml:Solid>
</building:lod1Solid>
```

### SURFACE_GEOMETRY

<table>
<thead>
<tr>
<th>ID</th>
<th>GMLID</th>
<th>PARENT_ID</th>
<th>ROOT_ID</th>
<th>IS_SOLID</th>
<th>IS_COMPOSITE</th>
<th>GEOMETRY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UUID</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>SDO_GEOMETRY</td>
</tr>
<tr>
<td>2</td>
<td>lod1Surface</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>SDO_GEOMETRY</td>
</tr>
<tr>
<td>3</td>
<td>Left1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>SDO_GEOMETRY</td>
</tr>
<tr>
<td>4</td>
<td>Front1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>SDO_GEOMETRY</td>
</tr>
<tr>
<td>5</td>
<td>Right1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>SDO_GEOMETRY</td>
</tr>
<tr>
<td>6</td>
<td>Back1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>SDO_GEOMETRY</td>
</tr>
<tr>
<td>7</td>
<td>Roof1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>SDO_GEOMETRY</td>
</tr>
</tbody>
</table>
Geometry representation

```
<bldg:lod1Solid>
  <gml:Solid>
    <gml:exterior>
      <gml:CompositeSurface gml:id="lod1Surface">
        <gml:surfaceMember>
          <gml:Polygon gml:id="Left1">
            <gml:exterior>
              <gml:LinearRing gml:id="LeftRing1">
                <gml:posList srsDimension="3"> 0.0 0.0 0.0 10.0 0.0 0.0 10.0 0.0 4.0 0.0 0.0 4.0 0.0 0.0 0.0
              </gml:posList>
            </gml:LinearRing>
          </gml:Polygon>
        </gml:surfaceMember>
        ...
        <gml:surfaceMember>
          <gml:Polygon gml:id="Roof1">
            <gml:exterior>
              <gml:LinearRing gml:id="RoofRing1">
                <gml:posList srsDimension="3"> 0.0 0.0 4.0 10.0 0.0 4.0 10.0 5.0 4.0 0.0 5.0 4.0 0.0 0.0 4.0
              </gml:posList>
            </gml:LinearRing>
          </gml:Polygon>
        </gml:surfaceMember>
      </gml:CompositeSurface>
    </gml:exterior>
  </gml:Solid>
</bldg:lod1Solid>
```
Geometry representation

```xml
<bldg:lod1Solid>
  <gml:Solid>
    <gml:exterior>
      <gml:CompositeSurface gml:id="lod1Surface">
        <gml:surfaceMember>
          <gml:Polygon gml:id="Left1">
            <gml:exterior>
              <gml:LinearRing gml:id="LeftRing1">
                <gml:posList srsDimension="3">
                  0.0 0.0 0.0 10.0
                  0.0 0.0 10.0 0.0 4.0 0.0 0.0 4.0 0.0 0.0 0.0
                </gml:posList>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </gml:surfaceMember>
        ...
        <gml:surfaceMember>
          <gml:Polygon gml:id="Roof1">
            <gml:exterior>
              <gml:LinearRing gml:id="RoofRing1">
                <gml:posList srsDimension="3">
                  0.0 0.0 4.0 10.0 4.0 0.0 0.0 4.0 0.0 5.0 4.0
                </gml:posList>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </gml:surfaceMember>
      </gml:CompositeSurface>
    </gml:exterior>
  </gml:Solid>
</bldg:lod1Solid>
```

<table>
<thead>
<tr>
<th>SURFACE_GEOMETRY</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
</tbody>
</table>
Assignment of surface appearances

```xml
<app:appearanceMember>
  <app:Appearance>
    <app:theme>Summer</app:theme>
    ...
    <app:surfaceDataMember>
      <app:ParameterizedTexture gml:id="roofTexture">
        <app:imageURI>roof.png</app:imageURI>
        <app:wrapMode>wrap</app:wrapMode>
        <app:target uri="#Roof1">
          <app:TexCoordList>
            <app:textureCoordinates ring="#RoofRing1">
              0.0 0.0 1.0 0.0 1.0 1.0 0.0 1.0 0.0 0.0
            </app:textureCoordinates>
          </app:TexCoordList>
        </app:target>
      </app:ParameterizedTexture>
    </app:surfaceDataMember>
  </app:Appearance>
</app:appearanceMember>
```

<table>
<thead>
<tr>
<th>TEXTUREPARAM</th>
<th>SURFACE_GEOMETRY_ID</th>
<th>IS_TEXTUREPARAMETERIZATION</th>
<th>WORLD_TO_TEXTURE_COORDINATES</th>
<th>SURFACE_DATA_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>1</td>
<td>-</td>
<td>0.0 0.0 1.0 0.0 1.0 1.0 0.0 1.0</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Diagram](roof.png)
3D City Database
CityGML Import/Export Tool
Creation of an Import/Export Tool

CityGML

XSD Schema
<xs:complexType name="CityModelType">
   <xs:complexContent>
      <xs:extension
   ...
</xs:complexType>

UML Model

Simplified UML Model

Model simplification

Java binding (JAXB)

Schema-derived classes
public class CityModel {
   ...
}

SQL queries (Imp/Export Tool)

Import data

Export data

Mapping classes to tables

Relational schema

SQL DDL statements (JDeveloper)

Database creation

Model simplification

Simplified UML Model

Mapping classes to tables

Relational schema

SQL DDL statements (JDeveloper)
Creation of an Import/Export Tool: Overview

**Data import**

1. **CityGML input file**
   - CityGML input file
2. **Read CityGML**
   - CityModel cityModel1 = new CityModel();
3. **Features**
   - CityModel cityModel1 = new CityModel();
4. **Database import**

**Application core**

1. **Xsd Schema**
   - Xsd Schema
   - Schema derived classes
   - public class CityModel {
     ...
   }
2. **Java binding**
3. **Write CityGML**
   - CityModel cityModel1 = new CityModel();
4. **Features**
   - CityModel cityModel1 = new CityModel();
5. **Database export**

**Data export**

1. **Oracle database**
2. **Import functionality**
3. **Export functionality**
4. **Instance of**
   - citygml4j
   - Xsd Schema
   - <xs:complexType name="CityModelType">
   - <xs:complexContent>
   - <xs:extension ...>
5. **Java binding**
6. **Instance of**
   - Schema derived classes
   - public class CityModel {
     ...
   }
7. **Write CityGML**
8. **Features**
   - CityModel cityModel1 = new CityModel();
9. **Data export**

**Application core**

1. **CityGML input file**
2. **Read CityGML**
3. **Features**
4. **Database import**
(Some) Characteristics of the Import/Export Tool

- Support for CityGML files of arbitrary size (>>4GB)
- Concurrency of data processing through multithreading
  - High performance on standard platforms
- Matching functionality
  - Identify and merge corresponding representations of the same building object within the database
- Support for XLink references (also within BRep geometries)
- Filter options enable user-defined import and export
  - GML ID, GML name
  - Bundled import and export for data tiling (classified by IDs or Bounding Boxes)
  - Selection of object classes
Performance data

- All tests executed on an Intel® Xeon® QuadCore, Win7 64-Bit, 12 GB RAM.
- 3D CityDB Server: 2 Intel® Xeon® QuadCore, Enterprise Linux RedHat (Kernel: 2.6.18), 32 GB RAM, 4 SAS disks (146 GB) and 16 SSD disks (64GB), Oracle 10.2.0.4.0

<table>
<thead>
<tr>
<th>City Model</th>
<th>Size in GB</th>
<th>LoDs</th>
<th>Import time without Textures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cologne</td>
<td>7.71</td>
<td>LoD1</td>
<td>37 Min</td>
</tr>
<tr>
<td>Berlin</td>
<td>6.23</td>
<td>LoD2, LoD3</td>
<td>20 Min</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>City Model</th>
<th>Size in GB</th>
<th>LoDs</th>
<th>Export time without Textures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cologne</td>
<td>7.71</td>
<td>LoD1</td>
<td>12 Min</td>
</tr>
<tr>
<td>Berlin</td>
<td>6.23</td>
<td>LoD2, LoD3</td>
<td>8 Min</td>
</tr>
</tbody>
</table>
3D City Database
Import/Export Tool Demo
What is available?

http://opportunity.bv.tu-berlin.de/software

- **3D City Database (current version 2.0.3)**
  - Oracle SQL scripts and PL/SQL functions
  - Comprehensive documentation

- **3D City Database Import/Export Tool (current version 1.2.2)**
  - Executable Java binaries
  - Complete source code
  - Comprehensive documentation
  - **KML/COLLADA exporter to be released in 2nd quarter 2011**

- **citygml4j (current version 1.0)**
  - Java class library and API for reading and writing CityGML datasets
  - Library files for Java5 and Java6
  - Source code, comprehensive documentation, tutorials