Web 3D Service (W3DS)
Status Report

Arne Schilling
University of Bonn, Germany

OGC TC Meeting 12/03/2008 Valencia
Web 3D Service (W3DS) Status Report

1. W3DS Introduction
2. New Extensions
3. OWS-6 DSS Thread
W3DS Status

Current Status: OGC Discussion Paper (OGC 05-019)

Released: February 2005
Version: 0.3.0

Editors: Udo Quadt, Thomas Kolbe

Presentations in TC Meetings:
January, 18th 2005, OGC TC meeting, New York City
November 7th 2005 OGC TC meeting, Bonn
**W3DS Scope**

W3DS delivers 3D scenes of a selected region that can be explored interactively.

**What is a Scene?**
1. A Scene is composed of data from one or multiple layers.
2. A Scene may also contain map elements (title, compass, scale bar, legend etc.) -> “3D map”, and predefined viewpoints.
3. A Scene must be provided in a CRS that can be used for visualization. **NOT WGS84.** Ideal: Cartesian coordinates. Large coordinates are also problematic.
4. A Scene is composed of “Display Elements” (geometries, triangles, materials, animations, lights, fog).
5. The structure of a Scene is not defined!
6. Semantics usually missing, depends on format.
Optimization Techniques for efficient rendering

- 1) Reduce the depth of the scene graph
   - Combine nested Transform Groups, remove unnecessary Groups
- 2) Reference to existing Materials and Appearances
   - Instead of defining the same Material repeatedly for each object, re-use already defined materials -> less memory consuming
- 3) sort objects according to Material (Display Lists)
   - Switching between materials during rendering is costly!
- 4) combine objects (3D Shapes) with same material
   - Instead of many 3D Shapes with each a geometry and a material: combine to few larger 3D Shapes. E.g. combine all roof geometries with same material into one larger geometry.
   - Structure of Scene gets lost!!!, no individual GIS objects any more!!
- 5) Create a Texture Atlas
   - Copy all textures into one large texture. Parts of the Texture Atlas are applied to the objects

In Java3D: “compiling”
Portrayal of CityGML

Client

W3DS

3D DB

WFS

X3D/VRML (ca. 17% of CityGML, GZIP compressed: 3%)

CityGML

processing unit

X3D

- CityGML Parser
- Feature Identification
- Geometry Optimizations
- Scene graph simplification
- Data synchronization
Portrayal Pipeline

e.g. WFS

W3DS

WMS WPVS
W3DS Operations

- GetCapabilities (mandatory)
  - very similar to WMS: layers, styles, bbox etc.
- GetScene (mandatory)
  - returns a 3D Scene / Scenegraph
W3DS Operations

- GetCapabilities (mandatory)
  - -> very similar to WMS: layers, styles, bbox etc.
- GetScene (mandatory)
  - -> returns a Scenegraph
- GetFeatureInfo (optional, Version 0.3.1)
  - -> returns attributes of selected features
- GetLayerInfo (optional, Version 0.3.1)
  - -> returns attribute names and values of selected layer
GetCapabilities Response

Scale Denominators

“The <MinScaleDenominator> and <MaxScaleDenominator> elements define the range of scales for which it is appropriate to generate a map of a Layer.”

“The scale denominator values are guidelines for clients, not firm limits.” (OGC 06-042)

<MinScaleDenominator>1e3</MinScaleDenominator>
<MaxScaleDenominator>1e6</MaxScaleDenominator>

1/6e06 – 1/12e06
1/500 – 1/3000
GetCapabilities Response

Levels of Detail
The optional <MinLevelOfDetail> and <MaxLevelOfDetail> elements describe the range of Levels of Detail that can be provided by the layer.

<MinLevelOfDetail>CityGML:1</MinLevelOfDetail>
<MaxLevelOfDetail>CityGML:4</MaxLevelOfDetail>
Putting together terrain tiles of different sizes and LODs

Extended LOD Concept

<table>
<thead>
<tr>
<th>Tile Level</th>
<th>Size</th>
<th>CityGML LOD</th>
<th>Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>0.5 km</td>
<td>4 - indoor</td>
<td>Bus Stop</td>
</tr>
<tr>
<td>15</td>
<td>1 km</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>2 km</td>
<td>2</td>
<td>Street Level</td>
</tr>
<tr>
<td>13</td>
<td>5 km</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>10 km</td>
<td>0 - landscape</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>20 km</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>39 km</td>
<td></td>
<td>City</td>
</tr>
<tr>
<td>9</td>
<td>78 km</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>156 km</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>313 km</td>
<td></td>
<td>State</td>
</tr>
<tr>
<td>6</td>
<td>625 km</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1250 km</td>
<td></td>
<td>Country</td>
</tr>
<tr>
<td>4</td>
<td>2500 km</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>45°</td>
<td></td>
<td>Continent</td>
</tr>
<tr>
<td>2</td>
<td>90°</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>180°</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>360°</td>
<td></td>
<td>Globe</td>
</tr>
</tbody>
</table>

Misund, Grandlund, Kolas
# GetScene Parameters

<table>
<thead>
<tr>
<th>URL parameter</th>
<th>Required/Optional/Conditional</th>
<th>annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>VERSION=&lt;version&gt;</td>
<td>R</td>
<td>requested version</td>
</tr>
<tr>
<td>REQUEST=GetScene</td>
<td>R</td>
<td>requested operation</td>
</tr>
<tr>
<td>SRS=namespace:identifier</td>
<td>R</td>
<td>spatial reference system</td>
</tr>
<tr>
<td>POI=&lt;point_of_interest&gt;</td>
<td>C</td>
<td>x,y,z point coordinates according to SRS</td>
</tr>
<tr>
<td>PITCH=&lt;pitch&gt;</td>
<td>C</td>
<td>angle of inclination [degree]</td>
</tr>
<tr>
<td>YAW=&lt;yaw&gt;</td>
<td>C</td>
<td>azimuth [degrees]</td>
</tr>
<tr>
<td>ROLL=&lt;roll&gt;</td>
<td>O</td>
<td>rotation around viewing vector [degree]</td>
</tr>
<tr>
<td>DISTANCE=&lt;distance&gt;</td>
<td>C</td>
<td>distance POI to POC [meter]</td>
</tr>
</tbody>
</table>
### GetScene Parameters cont’d

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POC=x, y, z</td>
<td>C</td>
<td>x,y,z coordinates of camera according to SRS</td>
</tr>
<tr>
<td>AOV=&lt;angle_of_view&gt;</td>
<td>C</td>
<td>angle of view [degree]</td>
</tr>
<tr>
<td>BBOX=xmin,ymin,xmax,ymax</td>
<td>R</td>
<td>2d bounding box</td>
</tr>
<tr>
<td>MINHEIGHT=&lt;lower_limit&gt;</td>
<td>O</td>
<td>displaying objects with height ( \geq ) lower_limit according to SRS</td>
</tr>
<tr>
<td>MAXHEIGHT=&lt;upper_limit&gt;</td>
<td>O</td>
<td>displaying objects with height ( \leq ) upper_limit according to SRS</td>
</tr>
<tr>
<td>LAYERS=&lt;layer list&gt;</td>
<td>O</td>
<td>comma separated list of 3D object sets</td>
</tr>
<tr>
<td>STYLES=&lt;style list&gt;</td>
<td>O</td>
<td>comma separated list of styles for each layer</td>
</tr>
<tr>
<td>FORMAT=&lt;format&gt;</td>
<td>R</td>
<td>MIME type of output</td>
</tr>
<tr>
<td>TIME=&lt;date_and_time&gt;</td>
<td>O</td>
<td>date and time</td>
</tr>
<tr>
<td>EXCEPTIONS=&lt;excepttype&gt;</td>
<td>O</td>
<td>exception format</td>
</tr>
<tr>
<td>TRANSLATE=x,y,z</td>
<td>C</td>
<td>translation vector that is applied to all 3D coordinates</td>
</tr>
<tr>
<td>ENVIRONMENT=on/off</td>
<td>O</td>
<td>switch on/off background elements like sky or light source</td>
</tr>
<tr>
<td>BGCOLOR=&lt;color&gt;</td>
<td>O</td>
<td>background color</td>
</tr>
<tr>
<td>BGIMAGE=&lt;image url&gt;</td>
<td>O</td>
<td>URL of background image</td>
</tr>
</tbody>
</table>
New GetScene Parameters in 0.3.1

Level of Detail
- Objects of one layer may be available in multiple LODs, especially building models (block, w. roof shape, textured, detailed façade model, indoor model)
- Terrain layer may be also available in multiple resolutions (from global earth surface to very detailed local models showing ditches, dikes etc.

Parameters: LOD=<string> ("qualifier:number")
   LOD_SELECTION=<string> ("equals" | "equals_or_smaller")
LOD Selection

LOD 4

LOD 3

LOD 2

LOD 1

LOD 0
LOD Selection

LOD 4

LOD 3

LOD 2

LOD 1

LOD 0

LOD=2
LOD_SELECTION=equals
LOD Selection

LOD 4

LOD 3

LOD 2

LOD 1

LOD 0

LOD=3
LOD_SELECTION=equals_or_smaller
LOD Selection

Examples

LOD=1
LOD_SELECTION=equals

LOD=2
LOD_SELECTION=equals

LOD=2
LOD_SELECTION>equals_or_smaller
New GetScene Parameters in 0.3.1

Spatial Constraint
- Currently:
  a) BBOX=xmin,ymin,xmax,ymax
  b) MINHEIGHT=<double>, MAXHEIGHT=<double>

We need more flexibility!
- Using WKT: BOUNDS=WKT (Polygon | PolyhedralSurface Z)

Examples:
BOUNDS=Polygon(((x,y, x,y,.....x,y), (x,y, x,y,.....x,y)) Polygon with Hole

BOUNDS=PolyhedralSurface Z (((x,y,z, x,y,z, x,y,z)), ((x,y,z, x,y,z, x,y,z)), ..... ((x,y,z, x,y,z, x,y,z)))
Condition: Surface must be closed!

Part of SFS 1.2 (OGC 06-103r3)
Spatial Selection Method
Currently: not defined! Depends on implementation

Parameter SELECTION_METHOD=<string> (“intersection” | “by_center” | “crop”)

Values:
- a) **intersection**: default operation, select features that touch, or intersect or are within BOUNDS
  - -> useful for static maps
- b) **by_center**: select features with center point being within BOUNDS
  - -> useful for selecting buildings displayed on a virtual globe
- c) **crop**: modify geometry of selected features (by intersection method)
  so that parts lying outside BOUNDS are cut away
  - -> useful for focus maps or subsets of terrain
New GetScene Parameters in 0.3.1

Focus Map: Multi-Resolution Terrain
New GetScene Parameters in 0.3.1

**Styled Layer Descriptors (SLD)**

- Enables user-defined Styling. Styles are defined as Styled Layer Descriptors. The Symbology Encoding (SE) needs extensions in order to style 3D objects.

One of 3 alternative parameters possible:

**Parameters:**

- **SLD=<string>: URL reference to SLD document**
- **SLD_BODY=<string>: inline SLD Document in GET request**
- **StyledLayerDescriptor=<xml>: inline SLD Document in POST request**

Defined in SLD profile of the WMS IS (OGC 05-078r4)
GetFeatureInfo Request

The GetFeatureInfo operation is designed to provide clients of a W3DS with more information about features within a scene that is currently displayed.
GetLayerInfo Request

Style Editor in OpenJUMP
GetLayerInfo Request

The purpose of the GetLayerInfo request is to collect information on the available attribute names and the values in the attribute table of a specific layer.

<table>
<thead>
<tr>
<th>URL parameter</th>
<th>Required/Optional/Conditional</th>
<th>annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>VERSION=0.3.1</td>
<td>M</td>
<td>Request version.</td>
</tr>
<tr>
<td>REQUEST=GetLayerInfo</td>
<td>M</td>
<td>Request name.</td>
</tr>
<tr>
<td>LAYER=&lt;layer&gt;</td>
<td>M</td>
<td>One Layer</td>
</tr>
<tr>
<td>COLUMNNAME=&lt;column list&gt;</td>
<td>O</td>
<td>Comma-separated list of one or more column to be queried.</td>
</tr>
<tr>
<td>FORMAT=output_format</td>
<td>M</td>
<td>Return format of feature information (MIME type).</td>
</tr>
</tbody>
</table>
GetLayerInfo Request

GetLayerInfo request:

http://www.myserver.de/W3DS?REQUEST=GetLayerInfo&VERSION=0.3.1&LAYER=Terrain&FORMAT=text/xml

GetLayerInfo response:

```xml
<GetLayerInfo>
  <Layer>
    <Name>Terrain</Name>
    <Attribute>
      <Name>id</Name>
    </Attribute>
    <Attribute>
      <Name>landuse</Name>
    </Attribute>
  </Layer>
</GetLayerInfo>
```
GetLayerInfo Request

GetLayerInfo request:

http://www.myserver.de/W3DS?REQUEST=GetLayerInfo&VERSION=0.3.1&LAYER=Terrain&COLUMNNAME=landuse&FORMAT=text/xml

GetLayerInfo response:

<GetLayerInfo>
  <Layer>
    <Name>Terrain</Name>
    <Attribute>
      <Name>landuse</Name>
      <Values>
        <Value>Bahn</Value>
        <Value>Baubloecke</Value>
        <Value>Gruenflaechen</Value>
        <Value>null</Value>
        <Value>Strassen</Value>
        <Value>Waldflaechen</Value>
        <Value>Wasserflaechen</Value>
      </Values>
    </Attribute>
  </Layer>
</GetLayerInfo>
Information Retrieval Concepts

W3DS is a Portrayal Service

- Result is optimized for efficient visualization
- No guarantee on the internal structure of the scene
- Access to attribute and meta data through additional server requests (GetFeatureInfo operation, OpenLS geocoder, directory service)
- Attribute are stored as tables, classic GIS Feature concept
- All CityGML tags can be transformed into attribute tables (including ID, address, object key, etc.)
Scope
The focus for DSS in OWS-6 builds on portrayal, WMS Tiling, and integrated client work from OWS-3, OWS-4 and OWS-5, with additional work on 3D visualization and integration of the built environment and landscape. This thread will encompass these capabilities and task areas:

- ISO 19117 and OGC SLD Portrayal
- 3D Portrayal of GML with Fly-through
- Outdoor and indoor 3D route services
- WMS performance (tiling)
- Integrated Client for multiple OWS services
OWS-6 Testbed DSS Thread

Portrayal of CityGML with 3D Fly-through

Client

W3DS

3D DB

WFS

X3D/VRML (ca. 17% of CityGML, GZIP compressed: 3%)

- CityGML Parser
- Feature Identification
- Geometry Optimizations
- Scene graph simplification
- Data synchronization
OWS-6 Testbed DSS Thread

3D Indoor and Outdoor Routing

University of New South Wales, Sydney, Australia

Bldg XY

Level 5

Room 234

Level 8

Room 68

Connector to Australian Road Transportation Network (RTN)

J2EE/Oracle/Silva/Java/Xmarc
Mobile [SMS/WAP/XML]

(c) Xmarc AsiaPac Pty Ltd 2002
OWS-6 Testbed DSS Thread

Integrated Client for multiple OWS Services: XNavigator (Uni Bonn Karto)
Integrated Client for multiple OWS Services: Aristoteles (Uni Bonn IGG)
OWS-6 Testbed DSS Thread

Sensor Data Integration
OWS-6 Testbed DSS Thread

Sensor Data Integration
The End

Thank you for your Attention