Suggestions for Extending the OGC Styled Layer Descriptor (SLD) Specification into 3D – Towards Visualization Rules for 3D City Models

S. Neubauer
University of Applied Sciences FH Mainz, Germany

A. Zipf
University of Applied Sciences FH Mainz, Germany

ABSTRACT: 3D city models can be visualized on the web through OGC W3DS services. However this standard does not support a client-side definiton of visualization rules in order to have a homogeneous visualization of 3D scenes representing data from different servers. But for 2D maps on the web this goal is achieved using a specific language [specification] - the Styled Layer Descriptor (SLD) Specification. We present here an extension of the latest version of this specification into 3D as a separate profile and give first examples of implementing this 3D-SLD profile into our W3DS server implementation.

1. MOTIVATION

3D city models are becoming more and more popular. Urban data management benefits from this through a range of new possible applications. But of course this trend needs standards in order to allow interoperability between different cities. 3D visualizations of city or landscape models are being made interoperable by corresponding OGC standards. Standards are also being developed for 3d city models - CityGML seems especially promising regarding the exchange of 3D building models. Typically, within GIS there is a clean cut between raw geodata and visualization properties. This is an advantage because the same data can be used and displayed in multiple ways according to specific needs within each project or of each user. We argue, that this division should also apply to 3d city models. Until recently, in almost all cases the 3d model was already considered as a type of visualization. Typically graphic formats such as DXF, DWG, VRML and other special CAD formats were being used for representing the 3d data. Ideally we believe that it would make more sense if the raw data would only describe the geometry plus the semantic object classes with their respective attributes. This raw data then is completed with visualization rules - as it is the case in 2D GIS. In order to allow this clean cut, a separate format for the visualization regulation should be defined. This then can be applied to the various spatial features in different situations in different ways. This is already the usual way, how it is done for 2d web map. The visualization rules there are being expressed through the OGC Styled Layer Descriptor (SLD) specification. SLD offers many chances in this direction and makes it possible to integrate diverse data sources into a WMS and to style them consistently. It would be great if this would also apply to 3d data representing DEMs, 3D landscape and city models. The WPVS/W3DS/WTS specifications are similar to WMS and for WMS SLD is defined and gives many advantages. On the other hand raw data is represented using GML both in 2D and 3D and CityGML is an application schema of GML. For this reason, an extension of SLD into the third dimension is a step in the right direction. So far nothing of this sort is planned or realized by the OGC or others to our knowledge. However, there are contemplations about extending CityGML by further visualization elements. Unfortunately this would be in contrast to the desired effect of dividing raw data and visualization specifications. This extension would lead to a parallel visualization regulation within the OGC because 2d elements also have to be incorporated in 3d visualizations. For this reason, an SLD extension seems to be the more promising approach. This is not as simple as it may seem, for 3d visualizations are very complex and detailed extension are necessary. In this paper we would like to make first suggestions for further specifying SLD for 3d models. This is already being implemented in the 3D-GDI Heidelberg project. On the one hand the SLD file can be used for configuring a W3DS, on the other hand it can be used dynamically like a WMS in order to visualize the results from the client side. Some of the proposed extensions to SLD are presented as well as first results from implementing these within our W3DS server implementation.

2. 3D SPATIAL DATA INFRASTRUCTURES (3D-SDI) AND 3D CITY MODELS

Currently spatial data infrastructures (SDIs) are being built up at regional, national, as well as international level. They allow a decentralized organization of spatial data and the co-operative use of distributed services (Fitzke et al. 2004). However, the technology for integrating 3D geodata such as virtual city models in SDIs is still in the beginning, e.g. metadata as an important base of every SDI is often neglected in the context of 3D city models (Nonn and Zipf 2006). There are still many open questions regarding the interoperability between 3D spatial services and adequate workflows at public authorities. Quite a lot of municipalities are already building city models that have potential for various applications, most prominent being virtual city guides, but also city planning, disaster management, simulation of sound propagation, and others. Within the project "geospatial data infrastructure for Heidelberg" www.GDI-3D.de, we implement an 3D SDI for the city of Heidelberg. It has access to all available geodata sources and currently provides components for the visualization on Internet PCs. Also mobile devices such as PDAs are supported in early prototypes (Fischer et al 2005). We rely on the specifications of the Open Geospatial Consortium (OGC), which defines standards for GI web services that have been accepted internationally. Within the scope of SDI-3D we develop the client software for downloading and visualizating 3D city models. For desktop PCs we can use standard software (internet browser), but also specialized 3D Map vierwers have been implemented (Schilling et al. 2007). Todays desktop PCs have powerful GPUs, enough memory and advanced graphics cards, so that very realistic real time visualizations can be realized. Mobile devices have a quite limited performance and other restrictions, which requires the development of separate components that either display reduced models or even only perspective views rendered by the server and transmitted as ready raster images (Web Terrain Service) In addition CityGML, a geometric and semantic model for the detailed description of the built up environment will be also investigated – in particular as an export format. It is the a promising OGC proposal regarding the modeling, contents, and quality of 3D city models. It could enable urban planning managers, architects, mobile service providers, makers of navigation systems, and other actors to work with consistent data structures and exchange data more easily. The semantic information enables thematic queries, for instance for particular building types, and the analysis of geodata.

For visualization purposes of 3D scenes over the web a Web 3D Service (W3DS) has been proposed within OGC as discussion draft. It delivers 3D scenes (display elements) from 3D city or landscape models over the web using formats like VRML, X3D, GeoVRML or similar. This service has been implemented within the project 3D-SDI (www.gdi-3d.de) and it is used currently for the 3D Heidelberg City Model together with a custom Map 3D Viewer that supports some advanced features, such as streaming and encryption of the data. (Schilling et al 2007, Zipf et al 2007). Some more aspects are discussed in the next chapter.

3. THE OGC WEB 3D SERVICE (W3DS)

The specification for the delivery of perspective views of digital terrain models has already been accepted by the OGC, the Web Terrain Server (WTS) – it will be renamed Web Perspective View Service (WPVS) in future versions. Being an image based service, it does not support interactive applications very well. The Web 3D Service (W3DS) has been submitted to the OGC and goes one step further (OGC 2005). The parameters are similar to those of the WTS. Information on available 3D map layers and respective visualization styles is provided by the server using the *GetCapabilites* request, which self-describes the service. The *GetScene* request delivers complete 3D scenes in one of the well-known formats. VRML 2.0 is mandatory as a basic format, but also other formats can be used. The requested area is described as simple bounding box. Optional parameters include a point of interest, a point of camera and a *style* for each layer. The latter is based on the OGC Styled Layer Descriptor (SLD). This is a specification for the visualization of 2D maps on the web. We are currently extending this in order to allow for declarative styling of 3D scenes of landscape and city models. This is important as typical GIS data makes a distinction between geometric raw data on the one hand and visualization rules for that data in a separate format that can be combined to a visualization that includes the style rules and the geometric data.

4. POSSIBILITIES AND LIMITATIONS OF THE STYLE LAYER DESCRIPTOR

With SLD the OGC offers the possibility for extending the few functionalities of a basic WMS regarding map display. Without SLD the user is only able to display a data set as map layer as a whole in a pre-defined way. Additionally the user is limited to a set of styles defined by the server in a proprietary way. Only the names of the styles available to the user. This means for example that using WMS without SLD support does not offer the client any opportunity for building classes for the available data on the client side e.g. for thematic mapping. Instead one specified style is assigned to each layer. If the data needs to be classified (dividing the attribute information according to different classes) and then displaying this classification in different way within a WMS all possible variations would be needed to be available as individual layers – which would result in a complex configuration and data handling.

The only alternative is to define further methods which enable such a client-based fine-tuning on request, is necessary. SLD offers this possibility. Similar to a signature catalog, the geodata can be styled dynamically by the client requesting the maps. This way the visualization of maps with heterogeneous data sources becomes more flexible, as this data can be provided with the same visualization specifications and then displayed "on the fly" in a homogeneous way.

Recently some work was done within the OGC regarding SLD: the “Styled Layer Descriptor Implementation Specification 1.0.0” has been split up into two documents to allow the parts that are not specific to WMS to be reused by other service specifications. The first of the new documents from the SLD 1.1.0 is the “Symbology Encoding Implementation Specification”. This language can be used to portray the output of Web Map Servers (WMS), Web Feature Servers (WFS) and Web Coverage Servers (WCS). The second document “Styled Layer Descriptor profile of the Web Map Service” defines how the “Symbology Encoding (SE)” can be used in conjunction with Web Map Services.

5. THE STRUCTURE OF SLD

The SLD specification uses xml schema definition (XSD) for defining the possible elements for symbolizing the map. The most important part of SLD is the paragraph about the *Rule*: here, the scale, fill color or line width or transparency information for displaying the layer can be defined. The element which is "at the bottom" of the SLD document is drawn at a later period and therefore covers the elements drawn first (because it has been drawn on top of the previously drawn layers). This way layer displaying priorities can be specified. While SLD still has some deficits (Brinkhoff 2004, Weiser and Zipf 2005), it offers interesting possibilities not only in the context of web-based mapping, but also personalized Location Based Services (LBS)(Zipf 2005).

The *Rule* element is of particular importance, because it is the first step for defining the classes: All necessary information regarding the classification and symbolization of the class can be found within that *Rule* element. For symbolization, the following possibilities are available within *SLD: PointSymbolizer, LineSymbolizer, PolygonSymbolizer, TextSymbolizer, (Raster-Symbolizer*). Within the respective symbol definitions, all necessary settings and properties for the corresponding classes are stored such as fill-, line-, text-, and point color, line width, text- and point sizes, transparency, fill type (filled with graphical fill elements or with reference to a bitmap) along with the line type (dashed or dotted etc.) (cp. Müller 2007).

6. INTEROPERABLE VISUALIZATION OF 3D CITY MODELS

3D scenes can be sent to a 3d viewer via the internet by using a W3DS. To date, W3DS (Web 3D Service) usage offers no possibilities to couple three dimensional geometries with SLD. This is because the current SLD has only been developed for 2d visualizations. Within a W3DS it has only been possible to specify pre-defined styles without SLD using fixed names. An extension to the W3DS draft (resp. WTS/WPVS) to support 3D-SLD seems sensible, as it would allow similar applications of 3D city model visualizations, as already being supported by SLD-WMS. Some interesting scenarios are possible: The cities or regions of X and Y could each host their own W3DS with their own city or region model. If a user would need navigation information for getting from X to Y, both cities or regions could be displayed according to the same visualization standards. This can be coupled with rout planning applications, such as Neis et al (2007). Another example is, that urban designers could highlight different designs or planning scenarios according to color, etc.

7. EXTENDING SLD INTO THE THIRD DIMENSION

The following paragraph will introduce our approach of extending SLD into the third dimension. We will inform about which extensions seem necessary or at least wishful. Below is a list of relevant aspects:

- Rotation of elements for all three axes
- Displacements and positions are extended by Z
- *SurfaceSymbolizer* for defining surface visualizations (eg. DEMs as TINs)
- *SolidSymbolizer* for object volume description
- Integration of external 3d objects into the scene
- Defining *material* properties
- Billboards
- 3D legends
- Extrusion of 2D-geometries
- Lines displayed cylindrical (e.g. for routing, etc)

The last two bullet points on the list need discussion, as they again mix geometry and styling, but are included for the sake of completeness. On the other hand we have left out of the list aspects on detailed definition of texture parameters, such as texture coordinates, or other parameters needed for example for synthetic textures, as we think this is a separate topic. We are focusing on thematic visualizations and filtering similar to the original aims of SLD in 2D.

The table 1 summarizes the suggestions:

Table 1: Proposed 3D Extensions of SLD

Element	Extension	Function
LegendGraphic	Graphic_3D	3d legend integration
	• ExternalGraphic_3D	As an external 3d object
	o Format_3D	File format
	o OnlineResource	Link to file
	• Mark_3D	As a simple 3d object
	o WellKnownName	Known object
	o Fill	Surfaces filled

	o Stroke • Opacity • Size • Rotation3D	Drawing edges Opacity Size 3D rotation
Rule	SolidSymbolizer • Geometry • Fill • Stroke	3d object description Geometry definition Surface filled Surface outline
	SurfaceSymbolizer • SurfaceGeometry o TIN o MassPointRelief o BreaklineRelief o RasterRelief • Fill • Stroke	2.5d surface description Geometry type Please refer to definition 2 Taken from City GML Break line DEM Raster DEM Surface fill Connect points
LineSymbolizer	Extrusion • Height • Offset • Base_Height • SurfaceGeometry	Lines to 3d object Height of extrusion Offset to base height Base height Surface geometry of base height
	Pipe • Radius • CssParameter	Lines to 3d objects Radius of cylinder Color
	Billboardplacement • AnchorPoint … • Displacement … • Rotation	Please refer to definition 1 Anchor position of billboard Displacement of anchor point Rotation of labels on the billboard
PolygonSymbolizer	Extrusion …	Polygons to boxes
	BillboardPlacement …	
PointSymbolizer	Graphic_3D …	Inclusion of 3d Models
	BillboardPlacement …	
	Extrusion …	Points to lines
TextSymbolizer	Depth	Text depth
RasterSymbolizer	BillboardPlacement …	
LabelPlacement	BillboardPlacement …	
Fill	Material • DiffuseColor • AmbientIntensity • SpecularColor • Shininess • Transparency • EmissiveColor	Defining Material Properties Diffuse color Ambient intensity Specular color Shininess Transparency Emissive Color
PointPlacement	Rotation3D • Rotation_X • Rotation_Y • Rotation_Z	Rotation of 3D points Rotation around X Rotation around Y Rotation around Z
AnchorPoint	AnchorPointZ	Anchor point in direction of Z
Displacement	DisplacementZ	Displacement in direction of Z

8. REALIZING THE SLD EXTENSION

The base for the extension is the Symbology Encoding Version 1.1.0 (2006-07-20) [OGC „05-077r4_Symbology_Encoding“]. The advantage of this new SLD version is the independent styling language with the XML-Namespace „se“ (Symbology Encoding). For the current test- and discussion phase we introduce a new XML-Namespace „se3d“ (Symbology Encoding 3D). This namespace is used to develop an independent 3D-SLD, which imports all existing elements from the existing symbology encoding. This means that these elements remain unchanged for the Symbology Encoding 3D (see fig. 1). Based on this, the individual elements can be extended by new attributes without changing the existing symbology encoding in a first step. This method can be seen in fig. 2 for the element „*Rule*“ and in fig. 3 for the extended *SurfaceSymbolizer*.

After further successful test results, these will be made available online soon, so the extensions can then be discussed in more detail. Currently our W3DS implementation of 3D-SLD for internal configuration is usable to a large degree. Also the dynamic usage of the 3D-SLD via a web request, with a reference to respective SLD in the URL, is already implemented for most of the presented elements.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- SE-3D schema -->
<xsd:schema
  targetNamespace="http://www.opengis.net/se3d"
  xmlns:se3d="http://www.opengis.net/se3d"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:se="http://www.opengis.net/se"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xsd:import
    namespace="http://www.opengis.net/ogc"
    schemaLocation="../../filter/1.1.0/filter.xsd"/>
  <xsd:import
    namespace="http://www.opengis.net/gml"
    schemaLocation="../../gml/3.1.1/base/feature.xsd"/>
  <xsd:import
    namespace="http://www.opengis.net/se"
    schemaLocation="../../se/1.1.0/FeatureStyle.xsd"/>
```

```
3D-SLD
  filter
    1.1.0
      expr.xsd
      filter.xsd
      filterCapabilities.xsd
      sort.xsd
  gml
    3.1.1
      base
      xlink
        xlinks.xsd
  se
    1.1.0
      common.xsd
      FeatureStyle.xsd
      Symbolizer.xsd
  se3d
    1.0.0
      se3d.xsd
```

Figure 1: 3D Extension of the Symbology Encoding

```xml
<xsd:element name="Rule" type="se3d:RuleType">
  <xsd:annotation>
    <xsd:documentation> A Rule is used to attach property/scale conditions to and group the
      individual symbols used for rendering. </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="RuleType">
  <xsd:sequence>
    <xsd:element ref="se:Name" minOccurs="0"/>
    <xsd:element ref="se:Description" minOccurs="0"/>
    <xsd:element ref="se3d:LegendGraphic" minOccurs="0"/>
    <xsd:choice minOccurs="0">
      <xsd:element ref="ogc:Filter"/>
      <xsd:element ref="se:ElseFilter"/>
    </xsd:choice>
    <xsd:element ref="se:MinScaleDenominator" minOccurs="0"/>
    <xsd:element ref="se:MaxScaleDenominator" minOccurs="0"/>
    <xsd:element ref="se3d:Symbolizer" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

Figure 2: se3d:*RuleType*

Figure 3: Different building types styled thematically using SLD 3D in our W3DS, different vegetation types have been styled with different textures defined by the SLD3D. (data source: Bureau of Surveying, Standtvermessungsamt Heidelberg)

In particular it has already been realized in our W3DS, that 3D objects like buildings can be selected using the SLD by a *Filter* within such a *Rule* - based on attribute values. The selected buildings than receive their specific visualization properties through the SLD also. A simple example of a first realization of such a thematic coloring based on SLD is presented below:

The *PolygonSymbolizer* has been changed by the extended elements *Fill* and *Displacement* as well as by the new elements *Extrusion* and *BillboardPlacement*. With the new *SurfaceSymbolizer* the appearance of the terrain model can be described according to the properties of the surface geometry. *BillboardPlacement* is also possible for the *RasterSymbolizer* element and has therefore been extended this way. For example instead of 3d tree models, pictures of trees can be added as billboards to enhance speed. For displaying geometries with volume, the newly introduced *SolidSymbolizer* can be applied. This refers to the geometry, fills the surfaces of the object and can describe edges.

```
<xsd:element name="SurfaceSymbolizer" type="se3d:SurfaceSymbolizerType"
  substitutionGroup="se3d:Symbolizer">
  <xsd:annotation>
    <xsd:documentation> A "Surface Symbolizer" specifies the rendering of a surface
      geometry, including its interior fill and border stroke. </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="SurfaceSymbolizerType">
  <xsd:complexContent>
    <xsd:extension base="se:SymbolizerType">
      <xsd:sequence>
        <xsd:element ref="se3d:SurfaceGeometry" minOccurs="0"/>
        <xsd:element ref="se3d:Fill" minOccurs="0"/>
        <xsd:element ref="se:Stroke" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

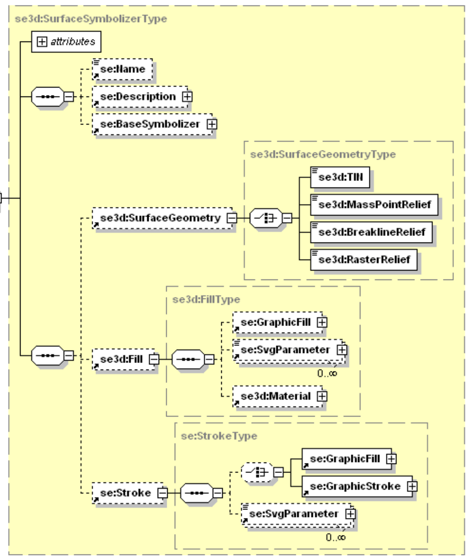

Figure 4: se3d:*SurfaceSymbolizerType*

Some more examples are presented: For instance the elements of the new element *Material* are similar to the VRML format. This way the properties of the surfaces can be described by the properties of the material. Included are *Diffuse*-, *Specular*-, *EmissiveColor*, *AmbientIntensity*, *Shininess* and *Transparency*.

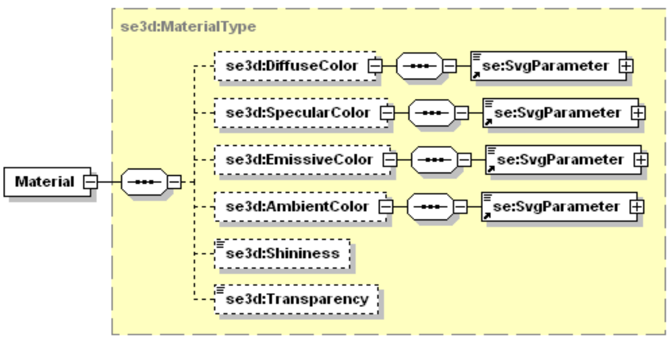

Figure 5: The se3d:MaterialType

For the color values of the *MaterialType*, the well known *SvgParameter* of SE is used. This means, that all of the colors are in a hexadecimal code as demonstrated in the example below. The diffuse color is dark red, the specular color is light red, the emissive color is zero because this is not an emitting source and the base color is also dark red. The shininess is 30% (0.3) and the material is not transparent (0.0).

```
<se3d:Material>
        <se3d:DiffuseColor>
                <se:SvgParameter name="">#AA0B00</se:SvgParameter>
        </se3d:DiffuseColor>
        <se3d:SpecularColor>
                <se:SvgParameter name="">#333333</se:SvgParameter>
        </se3d:SpecularColor>
        <se3d:EmissiveColor>
                <se:SvgParameter name="">#000000</se:SvgParameter>
        </se3d:EmissiveColor>
        <se3d: AmbientColor>
                <se:SvgParameter name="">#AA0B00</se:SvgParameter>
        </se3d: AmbientColor>
        <se3d:Shininess>0.3</se3d:Shininess>
        <se3d:Transparency>0.0</se3d:Transparency>
</se3d:Material>
```

The *ShadingModel* element provides the choice between *GouraundShading* and *FlatShading*. Within 3d programming this is a type of surface fill.

- *Flat-Shading* is the easiest shading form. Every face of the bent geometry is displayed according to a calculated color, depending on the position of the light (not a continuous color ramp). The surfaces seem flat and faint because the light reflections, shadows, transparencies, etc. are not included into the calculation.
- *Gouraud-Shading* interpolates the corner points. This way softer color ramps are generated between polygons. Gouraud-Shading can only display matt surfaces, which scatter the light evenly and randomly into all directions. For this reason the objects seem to have a plastic like appearance.

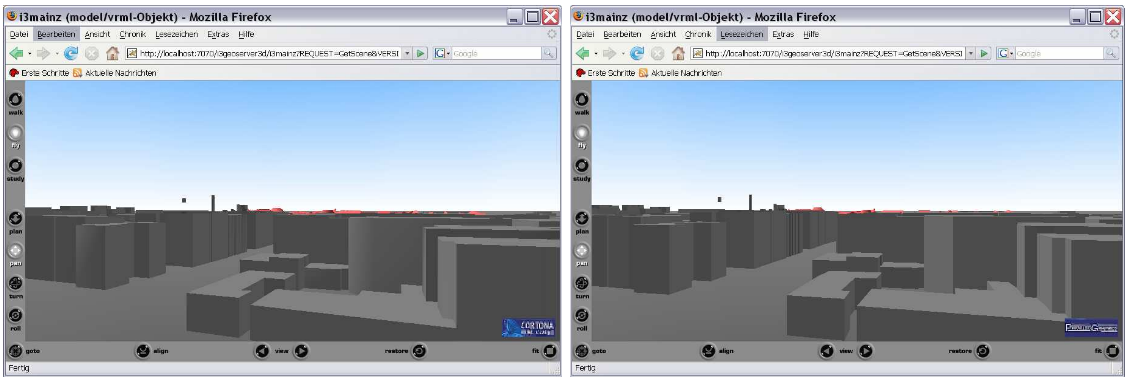

Figure 6: 3D city model delivered from our W3DS with Gouraud shading and Flat shading, specified through the *ShadingModel* within the SLD-3D.

There are also new possibilities for point display since 3d geometries or billboards can be used. Points can also be extruded. For example with 3d geometries a traffic light can be included as a VRML model as well as attribute based objects.

The proposal suggests that it is also possible for a 3d graphic to include external graphics and simple 3d geometries. It is a further possible to position the graphic at a certain 3d point and to then rotate and move this around any of the three axes.

The *Rotation_3D* element can deliver the angle in degrees for all three axes in a Cartesian Coordinate System. Also, the already existing elements *AnchorPoint* and *Displacement* are extended by Z. By using the se3d element *WellKnownName*, geometries familiar to the server can be referred to. These geometries could include sphere, cylinder, cone, and cube. By applying the Fill element from the se3d it is possible to describe the geometries by material.

Billboard is a technique, which gives the impression that 2D objects appear to be 3 dimensional. This can be done by turning the object, so it is always facing the viewer. An advantage is that only the front side has to be available and polygons needed for the other sides don´t have to exist. In the simplest case, only a surface with a texture is necessary to give the impression of a 3d object. The rotation is responsible for the direction that the object is facing. Because a 3D scene consists of 3D symbols which have to be described in a legend, the new element called *Graphic_3D* has been integrated also into the element *LegendGraphic*. Further the *TextSymbolizer* element describes the texts in the 3D world. By using new optional attributes there is a wide range of possibilities for doing so. New elements are *LabelPlacement* and *Fill*, along with text depth. This can be used to define a 3D text.

9. CONCLUSION AND OUTLOOK

In this paper we discussed the first outcomes of a research project that concentrates on the implementation of the next generation 3D spatial data infrastructures (3D-SDI) with a focus on 3D city models. It is one of the first implementations of a 3D web service that enables the delivery of 3D city and landscape models. Others include e.g. CityServer3D (Haist and Corrs 2005) or CAT3D (Coors&Bogdahn 2007). In contrast to many existing and well working proprietary client server solutions this is based on open standards that are currently in the discussion phase in OGC and will be supporting the 3D-SLD we have introduced here. Many internet map services and also car navigation systems show that going 3D is the next logical step.

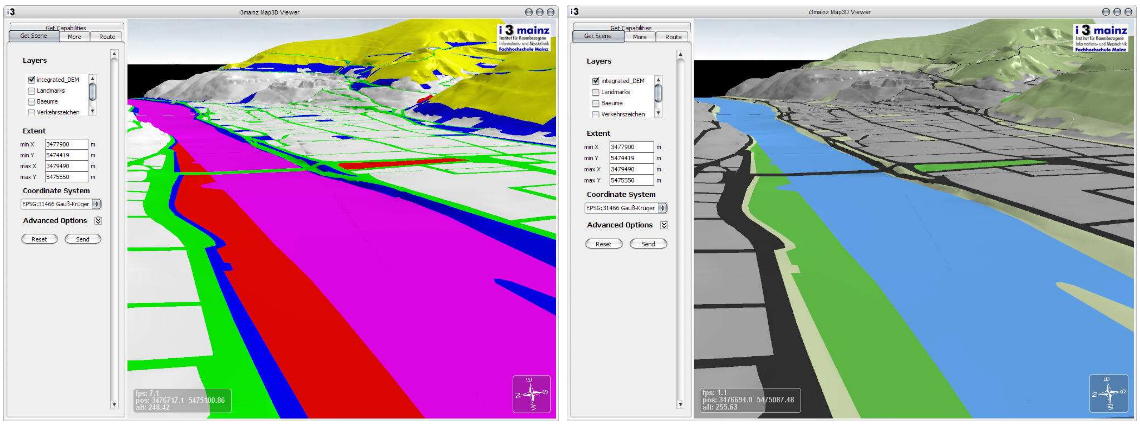

Figure 5: i3mainz 3D Viewer with 3DWS scene styled by different SLDs using the same geometry.

The additions and thoughts introduced here are a first attempt. They are currently being tested on the 3d city model of Heidelberg using the W3DS implemented in the project. This W3DS is currently being extended with the proposed 3D-SLD functionality. At the moment a range of elements are already supported by this implementation. The implementation of the proposed 3D SLD has been realized in two steps: Firstly, the SLD file is a fixed configuration file containing defined styles for the server and the client can choose between the defined styles dynamically. Then, there is also the possibility of delivering an SLD file, created entirely by the client, dynamically to the W3DS via the *GetScene* query. First results of this implementation have been reported.

Now we will be able to evaluate the approach and then we can find out if all combinations of the SLD visualization specification are sufficient or which changes have to be made. This is currently being investigated within a diploma thesis (Neubauer 2007) within our project 3D-SDI Heidelberg (www.heidelberg-3d.de). SLD 3D will enable the definition of custom 3D map styles (colors, patterns, textures, 3D marks etc.), so that the display can be adjusted to the client's requirements. For example a mobile display might need more colorful models with more contrast. If supported by different W3DS this will enhance interoperability by allowing to integrate 3D data from several W3DS servers into one scene with the same visualization style.

Acknowledgements
We thank all colleagues, in particular Arne Schilling and Jens Basanow for their help and input. This work has been supported by the Klaus-Tschira Foundation (KTS) gGmbH, Heidelberg.

10. REFERENCES

Brinkhoff, T. (2005): Towards a Declarative Portrayal and Interaction Model for GIS and LBS. Proceedings 8th Conference on Geographic Information Science (AGILE 2005), Estoril, Portugal, 2005, pp. 449-458.

CityGML: www.citygml.org

Coors, V. and Bogdahn, J. (2007): City Model Administration Toolkit (CAT3D) http://www.multimedia.fht-stuttgart.de/veps/CAT3D/cat3d.htm

Fischer, M., Basanow, J., Zipf, A. (2006): Mainz Mobile 3D - A PDA based client for the OGC Web 3D Service and corresponding server. International Workshop on 3D Geoinformation 2006 (3DGeoInfo'06). Kuala Lumpur. Malaysia.

Fitzke, J; Greve, K; Müller, M. and A. Poth (2004): Building SDIs with Free Software - the deegree Project. In: Proceedings of GSDI- 7, Bangalore, India.

Haist, J.; Coors, V. (2005): The W3DS-Interface of Cityserver3D. In: Kolbe, Gröger (Ed.); European Spatial Data Research (EuroSDR) u.a.: Next Generation 3D City Models. Workshop Papers : Participant's Edition. 2005, pp. 63-67

Müller, M. (ed): OpenGIS Symbology Encoding Implementation Specification version 1.1.0 doc.nr. 05-077r4

OGC: OpenGIS® Styled Layer Descriptor (SLD) Implementation Specification version 1.0 doc.nr. 02-070

OGC: Styled Layer Descriptor Profile of the Web Map Service Implementation Specification version 1.1 doc.nr. 05-078

OGC: Web 3D Service. OGC Discussion Paper, Ref. No. OGC 05-019.

Schilling, A., Basanow, J., Zipf, A. (2007): VECTOR BASED MAPPING OF POLYGONS ON IRREGULAR TERRAIN MESHES FOR WEB 3D MAP SERVICES. 3rd International Conference on Web Information Systems and Technologies (WEBIST). Barcelona, Spain. March 2007.

Neis, P., A. Schilling, A. Zipf (2007): 3D Emergency Route Service (3D-ERS) based on OpenLS Specifications. GI4DM 2007. 3rd International Symposium on Geoinformation for Disaster Management. Toronto, Canada.

Neubauer, S. (2007 working title): Visualisierungsvorschriften für 3D Stadtmodelle auf Basis von SLD. Diplomarbeit. FH Mainz. (In work.)

Nonn, U. and A. Zipf (2007): Metadata for 3D City Models - Analysis of the Applicability of the ISO 19115 Standard and Possibilities for further Amendments. Accepted for AGILE 2007. International Conference on Geographic Information Science of the Association of Geographic Information Laboratories for Europe (AGILE). Aalborg, Denmark.

Weiser, A., A. Zipf (2005): A visual editor for OGC SLD files for automating the configuration of WMS and mobile map applications. In: 3rd Symposium on Location Based Services and TeleCartography. Vienna. Austria. 2005. Springer.

Zipf, A. (2005): Using Styled Layer Descriptor (SLD) for the dynamic generation of user- and context-adaptive mobile maps - a technical framework. In: 5th International Workshop on Web and Wireless Geographical Information Systems (W2GIS). Lausanne, Switzerland. Springer Lecture Notes in Computer Science. Heidelberg, Berlin.

Zipf, A., J. Basanow, P. Neis, S. Neubauer, A. Schilling, (2007 accepted): Towards 3D Spatial Data Infrastructures (3D-SDI) based on Open Standards - experiences, results and future issues. "3D GeoInfo07". ISPRS WG IV/8 International Workshop on 3D Geo-Information: Requirements, Acquisition, Modeling, Analysis, Visualization. Delft, NETHERLANDS.