Open Geospatial Consortium Inc.

Date: 2009-10-13

Reference number of this document: OGC 09-042

Version: 0.0.1

Category: OpenGIS® Discussion Paper

Editor(s):

Steffen Neubauer (University of Bonn, Department of Geography)

Alexander Zipf (University of Bonn, Department of Geography)

3D-Symbology Encoding Discussion Draft

Warning

This document is not an OGC Standard. This document is an OGC Discussion Paper and is therefore not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, an OGC Discussion Paper should not be referenced as required or mandatory technology in procurements.

Document type:	OpenGIS® Discussion Paper
Document subtype:	Discussion Draft
Document stage:	Draft proposed version 0.0.1
Document language:	English

Contents

Page

Figures

Page

i. Preface

This document present an extension of the Symbology Encoding (SE) /Styled Layer Descriptor (SLD) specifications into 3D as a separate profile.

Suggested additions, changes, and comments on this draft report are welcome and encouraged. Such suggestions may be submitted by email message or by making suggested changes in an edited copy of this document.

ii. Document terms and definitions

This document uses the standard terms defined in Subclause 5.3 of [OGC 05-008], which is based on the ISO/IEC Directives, Part 2. Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

iii. Submitting organizations

The following organizations submitted this document to the Open Geospatial Consortium Inc.

Research Group Cartography,
Department of Geography
University of Bonn
Meckenheimer Allee 172
D-53115 Bonn
www.geographie.uni-bonn.de/karto

iv. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Company	Contact	Adress
Research Group Cartography, Department of Geography University of Bonn	A. Zipf	Meckenheimer Allee 172 53115 Bonn Germany
Research Group Cartography, Department of Geography University of Bonn	S. Neubauer	Meckenheimer Allee 172 53115 Bonn Germany

v. Revision history

Date	Release	Editor	Primary clauses modified	Description
2009-03-27	0.0.1	Steffen Neubauer		First internal 3D-SE version
2009-10-10	0.0.1	C. Reed	Various	Prepare for publication as Discussion Paper

vi. Changes to the OGC Abstract Specification

The OpenGIS® Abstract Specification does not require changes to accommodate the technical contents of this document.

vii. Future work

…

Foreword

This document includes 7 annexes; Annexes ABDEFG are normative, and Annexes C are informative.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium Inc. shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

Introduction

The importance of the visual portrayal of geographic data cannot be overemphasized. The skill that goes into portraying data (whether it be geographic or tabular) is what transforms raw information into an explanatory or decision-support tool. From USGS' topographic map series to NOAA and NIMA's nautical charts to AAA's Triptik, fine-grained control of the graphical representation of data is a fundamental requirement for any professional mapping community.

3D city models can be visualized on the web through OGC 3D Portrayal services (W3DS or WPVS/WTS). However these draft OGC standards do not support a client-side definition of visualization rules. The latter would for example allow realizing a homogeneous visualization of 3D scenes representing data from different servers. However for 2D maps on the web this goal is achieved through the Symbology Encoding (SE) /Styled Layer Descriptor (SLD) Specifications. We present an extension of the SE specification into 3D as a separate profile. We provide examples of the current realization of this 3D-SE profile within our W3DS implementation.

3D-Symbology Encoding Discussion Draft

1 Scope

This OpenGIS® Discussion Draft specifies the format of a scene-styling language for producing scenes with user-defined styling.

The following paragraphs will introduce an approach of extending SLD into the third dimension.

2 Compliance

Compliance with this standard shall be checked using all the relevant tests specified in Annex A (normative).

3 Normative references

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

ISO 19105:2000, Geographic information — Conformance and Testing

OGC 06-121r3, OpenGIS® Web Services Common Specification

OGC: OpenGIS® Symbology Encoding (SE) Implementation Specification 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: Filter Encoding Implementation Specification version 1.1.0 doc.nr. 04-095

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

OGC: Candidate OpenGIS® CityGML Implementation Specification (City Geography Markup Language) version 0.4.0 doc.nr. 07-062, pp. 21-32

In addition to this document, this standard includes a normative XML Schema Document file as specified in Annex B.

4 Terms and definitions

For the purposes of this standard, the definitions specified in Clause 4 of the OWS Common Implementation Specification [OGC 05-008] shall apply. In addition, the following terms and definitions apply.

4.1
client
software component that can invoke an operation from a server

4.2
coordinate reference system
coordinate system that is related to the real world by a datum [ISO 19111]

4.3
coordinate system
set of mathematical rules for specifying how coordinates are to be assigned to points [ISO 19111]

4.4
geographic information
information concerning phenomena implicitly or explicitly associated with a location relative to the Earth [ISO 19101]

4.5
interface
named set of operations that characterize the behaviour of an entity [ISO 19119]

4.6
layer
basic unit of geographic information that may be requested as a map from a server

4.7
map
portrayal of geographic information as a digital image file suitable for display on a computer screen

4.8
operation
specification of a transformation or query that an object may be called to execute [ISO 19119]

4.9
portrayal
presentation of information to humans [ISO 19117]

5 Conventions

5.1 Abbreviated terms

Most of the abbreviated terms listed in Subclause 5.1 of the OWS Common Implementation Specification [OGC 05-008] apply to this document, plus the following abbreviated terms.

3D-GDI	3D GeoSpatial Data Infrastructure
CAD	Computer Aided Design
CityGML	City Geography Markup Language
DEM	Digital Elevation Model
DWG	AutoCAD file format
DXF	Drawing Interchange Format
DTD	Document Type Definition
EPSG	European Petroleum Survey Group
GDI	Geodata infrastructure
GeoVRML	VRML with a extension for 3D geodata
GML	Geography Markup Language
HTTP	Hypertext Transfer Protocol
LBS	Location Based Services
LOD	Level of Detail
MIME	Multipurpose Internet Mail Extensions
OGC	Open Geospatial Consortium
POC	Point Of Camera
POI	Point Of Interest
RFC	Request for Comments
SDI	Spatial Data Infrastructure
SE	Symbology Encoding
SLD	Styled Layer Descriptor
VRML	Virtual Reality Modeling Language
W3DS	Web 3D Service

WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
WPVS	Web Perspective View Service
WTS	Web Terrain Service
X3D	Extensible 3D
XML	Extensible Markup Language
XSD	XML Schema Definition

6 3D-Symbology Encoding overview

6.1 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 data and services offered from different cities. A range of standards that are relevant for any kind of spatial data, and city models in particular, as well as web-based services, are being developed by the Open Geospatial Consortium (OGC). For the case of 3D city models CityGML (Gröger et al. 2007) seems especially promising regarding the exchange of the raw data. Within this paper we do not consider the exchange of raw data, but the interoperability of 3D scenes for interactive visualization of 3D city models. Typically, within GIS there is a clean cut between the raw geodata and the visualization properties. This is an advantage because the same data can be used and displayed in multiple ways according to the specific needs within different projects or for individual 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 itself. This is largely due to the fact, that typically graphic formats such as DXF, DWG, VRML and other special CAD formats were being used for representing the 3D data. We argue that it would be more beneficial regarding re-usability and interoperability if the raw data would only describe the typical feature properties: the geometry plus the semantic object classes with their respective attributes. In a further step this raw data then can be completed with visualization rules - as it is the case in 2D GIS (Shape + SLD). In order to allow this clean cut, a separate open format for the visualization regulation needs to be defined. This can then be applied to the various spatial features in different situations in different ways. This way it is already usually done for 2D (web) maps. In that case the visualization rules themselves are being expressed through the OGC Styled Layer Descriptor (SLD) specification (Müller 2007). SLD offers many chances. Next to allowing a client application or end user to define the style of a map, more importantly it makes it possible to integrate diverse data sources into one WMS map and to style them consistently within that map. For the same reasons it would be very beneficial if this mechanism could also be applied to 3D data representing DEMs, 3D landscape and city models. For visualizing 3D spatial data there exist two approaches within the OGC: Next to the mentioned W3DS there exist the WTS (Web Terrain Service) being renamed to WPVS (Web Perspective View Service). This returns an image of the 3D scene similar to the 2D WMS. Being an image based service, it does not support interactive applications very well. The W3DS works similar, returning already styled display elements, which only then are rendered to the screen on the client side, allowing a higher degree of user interaction.

For 2D WMS the visualization rules can already be expressed through SLD/SE, which gives many advantages. On the other hand geographical raw data is represented using GML (geography Markup Language) both in 2D and 3D. In particular 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. Recently also Ramos et al. (2007) support this direction of work. When planning and realizing such an extension we need to make sure, that existing standards are well integrated and that a backward-compatibility to the original standard (in this case the OGC Symbology Encoding) is maintained to the largest possible degree.

However, there are contemplations about extending CityGML by further visualization elements by extending the appearance model. As long as this only concerns facade textures, which can be seen as an alternative way to represent the geometry of the building, this can be still regarded as raw data (such as also a grid coverage can represent raw data). But it would be dangerous if this approach is taken too far by also considering styling and cartographic effects, as this would lead to a second, parallel visualization regulation within the OGC – in particular because 2D elements also have to be incorporated in 3d visualizations and for 2D we already have the existing SE specification. Unfortunately this also would be in contrast to the desired effect of dividing raw data and visualization specifications. For this reason, an extension of the existing Symbology Encoding as a profile, that incorporates the existing 2D SE specification seems to be the more promising approach. This is not as simple as it may seem: 3D visualizations can be very complex and detailed extensions are necessary. In this paper we would like to make first suggestions for further specifying SLD for 3D landscape and city models. This has been already implemented in the project 3D-GDI (3D GeoSpatial Data Infrastructure) www.gdi-3d.de. The result can be used in two ways: On the one hand the SE file can just be used for configuring a specific W3DS server, on the other hand it can be used dynamically on request in order to define the appearance of the 3D scene directly from the client side, such as it is the case with 2D WMS. Some of the proposed extensions to SE are presented within this paper, as well as first results from implementing these within our W3DS server.

6.2 3D Spatial Data Infrastructure (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 “3D geospatial data infrastructure for Heidelberg” we implement an 3D-SDI for the city of Heidelberg. In a follow-up project on mobile navigation with 3D city models (www.MoNa3D.de) (Coors and Zipf 2007) further investigation regarding optimized handling, compression and transmission of 3D city models as well as more semantic adequate route instructions are researched (Neis & Zipf 2008). In all cases 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 3D-SDI we develop both the server and the client software for downloading and visualizing 3D city models (Schilling et al 2007, 2008, Zipf et al 2007, Basanow et al 2007). A specialized

3D Map viewer - the “XNavigator” - has been implemented. It supports some advanced features, such as streaming and encryption of the 3D scene.

6.3 The OGC Web 3D Service (W3DS)

The Web 3D Service (W3DS) has been submitted to the OGC (OGC 2005) and delivers 3D scenes (display elements) from 3D city or landscape models over the web using formats like VRML, X3D, GeoVRML or similar. Currently an OGC working group is being formed that tries to harmonize this daft with the other 3D portrayal service drafts within the OGC – such as the Web Perspective View Service (WPVS) and Web Terrain Service (WTS). The request parameters are similar to each other. The W3DS GetScene request then delivers the actual complete 3D scenes in one of the well-known formats. VRML 2.0 is default as a basic format, but also other formats can be used. The requested area is described as simple bounding box. Optional parameters - amongst others - include a point of interest, a point of camera and a style for each layer. This style usually is only a reference (through a name) to a pre-configured style already known to the service through its internal configuration. In addition to this named styles it would be beneficial to be able to specify the details of the appearance of the style from the client side per request, as outlined above. This is where the Styled Layer Descriptor Specification (or more precisely its successor, the Symbology Encoding) comes into play. This will be explained in the next paragraphs.

6.4 Possibilities and Limitations of the Styled Layer Descriptor

The “Styled Layer Descriptor Implementation Specification 1.0.0” (SLD) has been split up into two documents within the OGC 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” (SE). This language can be used to portray the output of Web Map Servers (WMS), Web Feature Servers (WFS) and Web Coverage Servers (WCS) - and as we will see later: partially also of a Web 3D Service (W3DS). The second document “Styled Layer Descriptor profile of the Web Map Service” defines how the “Symbology Encoding (SE)” can be used in conjunction with WMS. We focus on SE in the following.

Using the OGC SE it becomes possible to extend the few functionalities of a basic WMS regarding map display. Without SE the user is only able to display a spatial data set as a whole in a pre-defined way as map layer. Additionally the user is limited to the pre-defined set of styles provided by the server in a proprietary way. Only the names of the styles are available to the user through the GetCapabilities request. How these styles actually look like is not explained, nor can the user change their appearance. This means that using WMS without SE support does not offer the client any opportunity for example for building classes for the available attribute 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 this classifications shall be displayed in different ways within a WMS it is necessary to provide all possible variations as individual layers within the service. This definitely results in a complex configuration and data handling. This is only one example of the many shortcomings of cartographic WMS styling without SE.

The only alternative is to define further methods which enable such a client-based fine-tuning on request. SE does offer 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.

While still not offering all possibilities a cartographer would dream of (cp. Brinkhoff 2004, Weiser & Zipf 2005, Dietze & Zipf 2007), SE is a big advantage and already provides a range of options for specifying more sophisticated cartographic styles for map layers and individual features within this layer. It offers interesting possibilities not only in the context of web-based mapping, but also personalized Location Based Services (LBS)(Zipf 2005).

6.5 The Structure of Symbology Encoding

The SE specification uses xml schema definition (XSD) for defining the possible XML elements for symbolizing the individual layers of a map. There exist two alternatives how geometry (the layers) and their appearance (Styles) can be defined. The first alternative is using NamedLayer (Fig. 1) and NamedStyle for referencing and thus using already defined and known layer and styles available at the server.

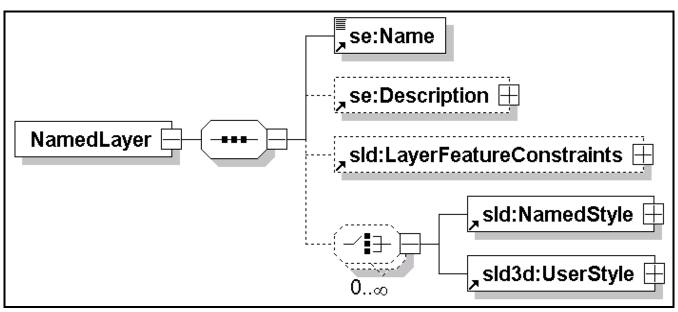

Fig. 1 NamedLayer Schema

The second alternative is to use the elements UserLayer (Fig. 3) and UserStyle. They allow defining a layer or a style dynamically through a set of attributes. These then can be send to the server which has to style the data according to that definition.

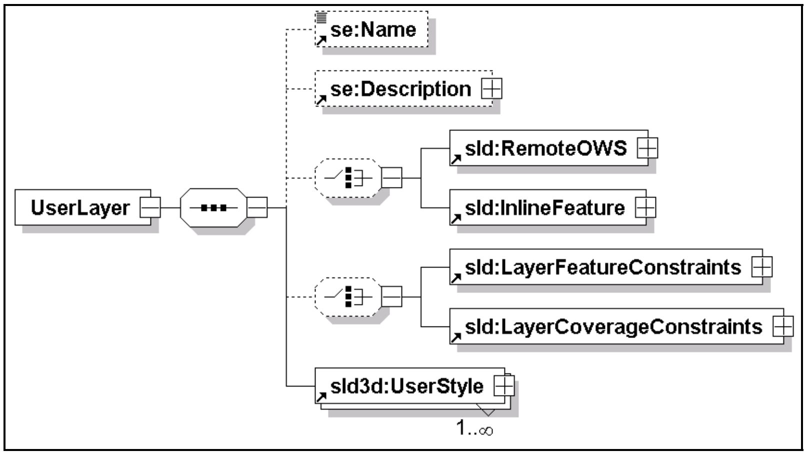

Fig. 2 UserLayer Schema

Further also combinations of NamedLayer and UserStyle can be sensible in order to define user-specific visualizations of data sets already available at and known to the server. The most important part of SE is the element Rule: here, a set of graphical properties of the referenced raw data such as ‘scale’, ‘fill color’, ‘line width’ or ‘transparency’ information for displaying the layer can be defined. All necessary information regarding the classification and symbolization of the class can be found within that Rule element. 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.

For selecting (filtering) features (their geometries) according to a range of properties (spatial information, thematic attributes or object IDs) the element Filter is used. This stems from the OGC Filter Encoding (OGC: Filter Encoding)(Fig. 5,4,5) and includes several possibilities for filtering features. This allows using typical GIS functionalities such as topological and thematic queries within the SE in order to define sets of features that shall be visualized in a distinct way. For symbolization, the following possibilities are available within the original SE: PointSymbolizer, LineSymbolizer, PolygonSymbolizer, TextSymbolizer and (RasterSymbolizer). 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).

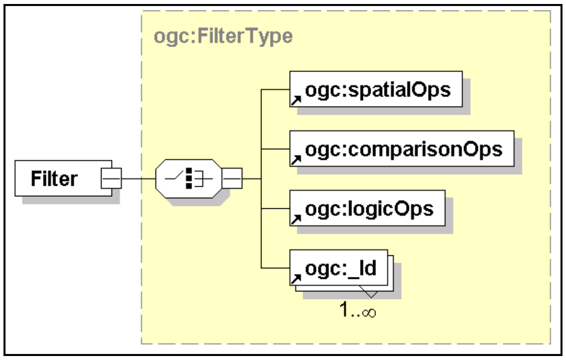

Fig. 3 Filter Types of the OGC Filter Encoding

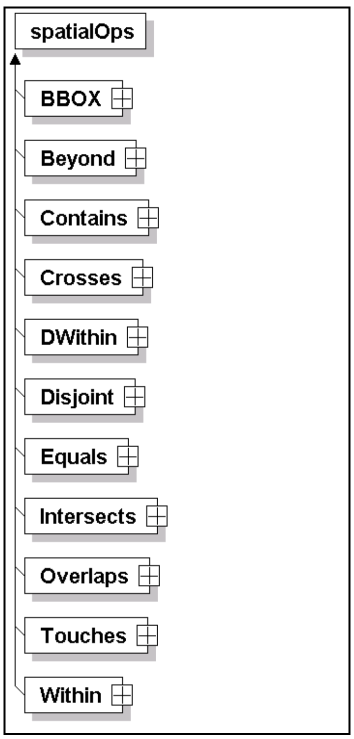

Fig. 4 spatialOps Type

Fig. 5 comparisonOps Type

6.6 Interoperable Visualization of 3D City Models

As explained 3D scenes of 3D city models can be sent to a 3D viewer via the internet by using a W3DS. In the current version of the W3DS specification draft there are no possibilities offered to couple 3D geometries with SE, as it has only been developed with 2D in mind.

An extension to the W3DS draft (resp. WTS/WPVS) to support 3D-SE 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 A and B could each host their own W3DS with their own city or region model. If a user would need a 3D scene that covers both cities A and B, e.g. in a 3D car navigation scenario, both cities and regions could be displayed according to the same visualization rules (Fig. 6). This is currently being coupled with route planning applications, such as Neis et al. (2007), Neis & Zipf (2008). Another example of the advantages of a 3D-SE is that urban planners can highlight different designs or planning scenarios using different styles (color) etc.

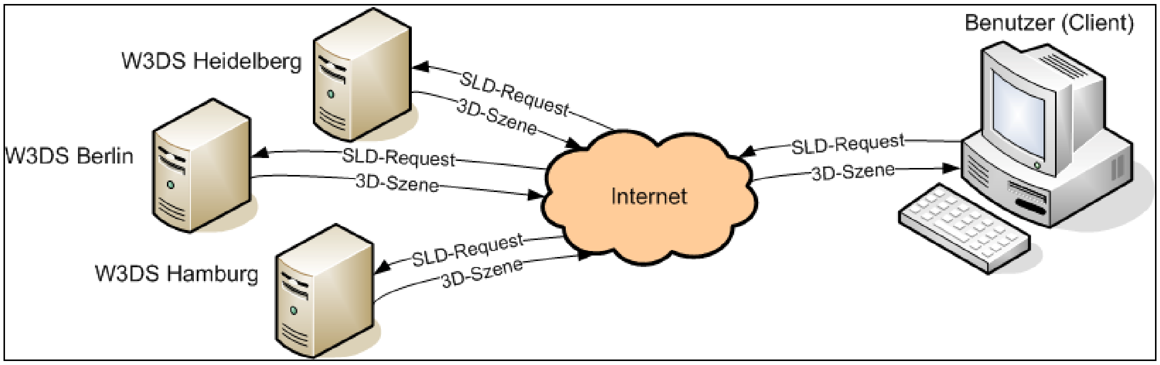

Fig. 6 Client accesses several W3DS servers and joins results. Visualization properties are controlled by client through sending requests including 3D-Symbol Encoding (SLD)

7 3D-Symbology Encoding common elements

7.1 Introduction – 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 we do regard necessary or at least wishful, after having studies a range of 3D graphic libraries and file formats such as Java3D, VRML, etc. Below is a list of relevant aspects we identified. All of these have been considered in a proposal for an XML schema definition representing the new 3D profile of the Symbology Encoding. As explained earlier we focus on cartographic representations, therefore omitting any reference to applying textures with texture coordinates to the model. But of course it is possible to use textures as fill materials as already specified in the original SE. If there is a need and common wish to include more sophisticated textures (either photo textures or parametric textures), then the proposed XML schema can easily be extended for that purpose. These are the extensions we define in the first version of SE-3D:

- Rotation of elements for all three axes
- Displacements and positions are extended by Z
- *SurfaceSymbolizer* for defining surface visualizations (eg. Contour, Elevation)
- *SolidSymbolizer* for object volume description
- Integration of external 3d objects into the scene
- Defining *material* properties
- LOD Handling
- Billboards
- 3D legends

- Lines displayed as cylindrical pipes (e.g. for routing, etc.)

In particular the last bullet point on the list needs discussion, as it could somehow mix geometry and styling, but it could be realized in an appropriate manner. 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 in Appendix G summarizes the suggestions of the above list with some more specific details.

7.2 Realizing the SLD extension

The base for the extension is the Symbology Encoding Version 1.1.0 (2006-07-20) [OGC „05-077r4“]. 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. 11). 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. 13 for the element „Rule“ and in

Fig. 15 XML schema se3d:SurfacesSymbolizerType

for the extended SurfaceSymbolizer.

In particular it is possible to use the OGC Filter Encoding functionalities for thematic filtering, to select 3D objects like buildings based on attribute values using the SE, as the SE can include a Filter within a Rule. The selected buildings than receive their specific visualization properties through the SE also. This has already been realized in our W3DS implementation. A simple 3D example of a first realization of such a thematic coloring based on SLD is presented in Fig. 7, where public buildings are depicted in red and other buildings in grey.

Fig. 7 Different building types are styled thematically using SLD 3D in a W3DS. (data sources: Bureau of Surveying, Stadtvermessungsamt Heidelberg, EML, own data)

```
<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"
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"/>
```

Fig. 8 Start of the XML schema of the 3D Extension of the Symbology Encoding

Currently our W3DS implementation of 3D-SE for internal configuration is usable to a large degree. Also the dynamic usage of the 3D-SE via a web request, with a reference to respective SLD in the URL, is already implemented for most of the presented elements.

7.3 Proposed 3D extensions of SLD

Now we will explain a selection of the changes, which we introduced to the original SE, when we defined the 3D-SE.

7.3.1 Rule

Within the *Rule* we also added the support to specify a particular Level of Detail (LOD) for a 3D scene (Fig. 13). This is useful if specific visual effects shall be realized and the generic LOD handling shall be overridden. An example is the case of 3D focus maps, as a extension of the original 2D focus maps (Zipf &Richter 2002).

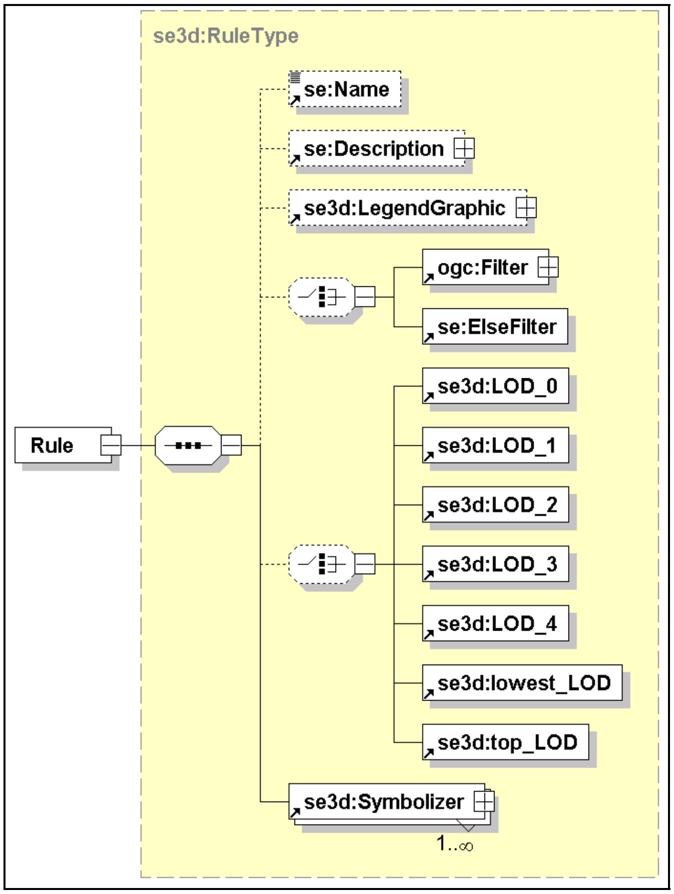

Fig. 8 se3d: RuleType

For displaying volume geometries, the newly introduced *SolidSymbolizer* can be applied. This refers to the geometry, fills the surfaces of the object and can describe edges.

With the new *SurfaceSymbolizer* the appearance of the terrain model can be described according to the properties of the surface geometry such as isolines, steepness, aspect etc.

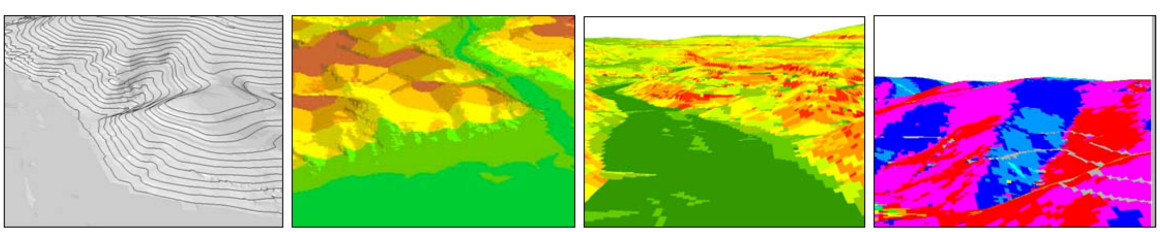

Fig. 9 Possibilities of se3d:SurfaceSymbolizerType

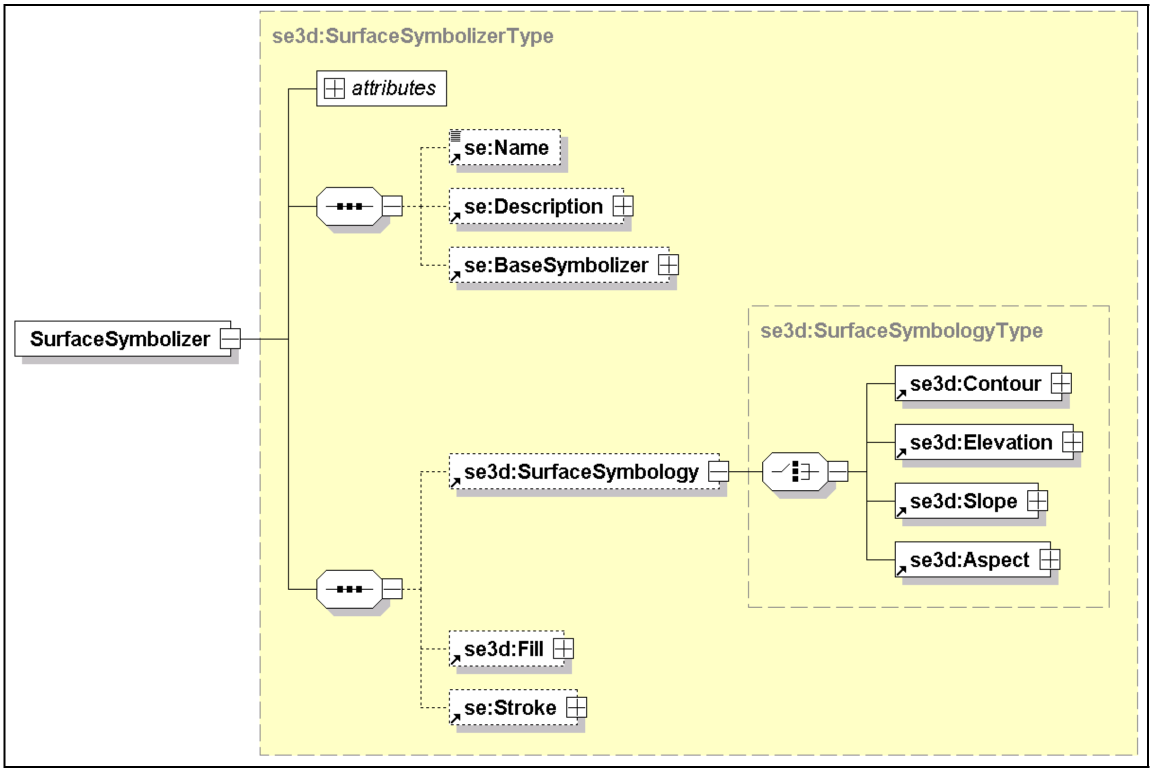

Fig. 10 XML schema se3d:SurfacesSymbolizerType

7.3.2 Point Symbolizer

We introduce also new possibilities for displaying points, since at the position of a point a full 3D geometry object (e.g. a 3D symbol) or billboards can be used. For example at a point representing the position of a traffic light, it is now possible to reference a full 3D object, e.g. a VRML file, which represents the traffic light.

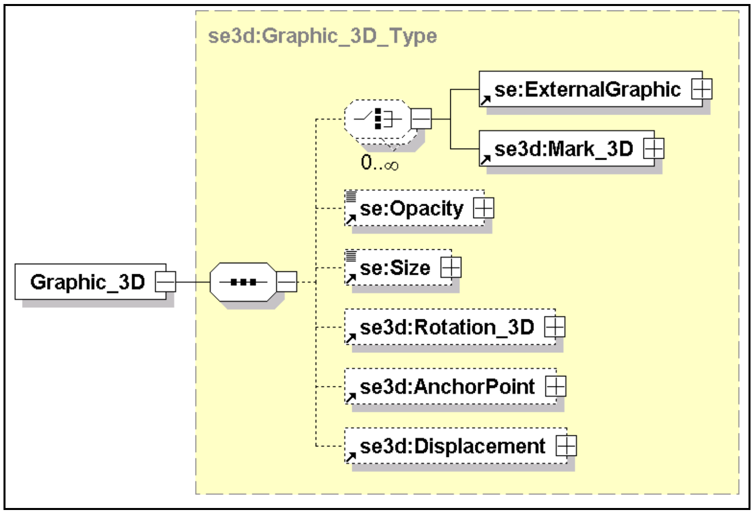

Fig. 11 XML structure of Graphic_3D

7.3.3 Line Symbolizer

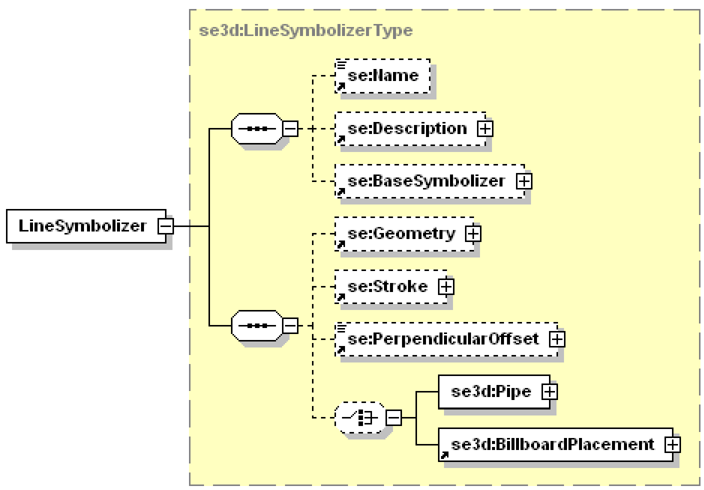

Fig. 12 XML schema se3d:LineSymbolizerType

7.3.4 Polygon Symbolizer

The *PolygonSymbolizer* has been changed by extending its child elements *Fill* and *Displacement*. The new element *BillboardPlacement* has been added.

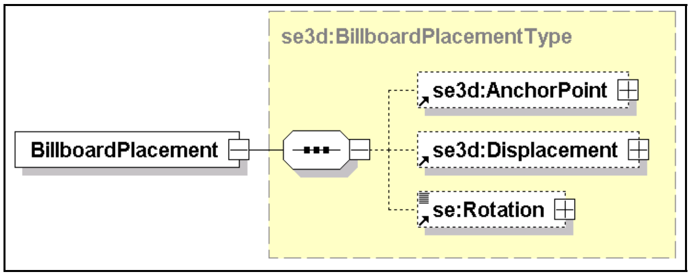

Fig. 13 schema for se3d:BillboardPlacementType

7.3.5 Text Symbolizer

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 label.

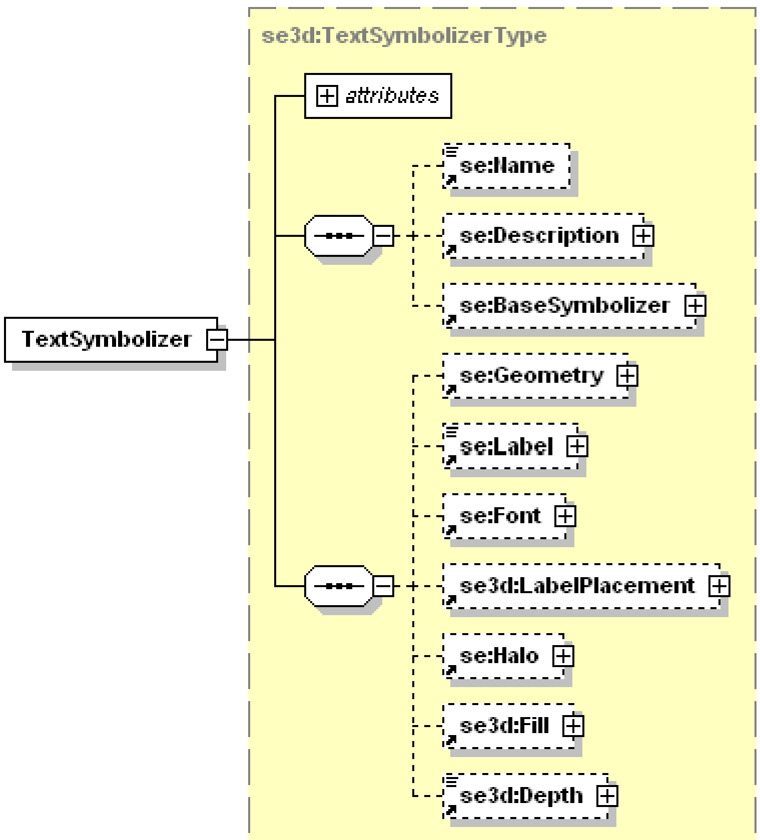

Fig. 14 schema se3d:TextSymbolizerType

7.3.6 Raster Symbolizer

BillboardPlacement is also possible for the *RasterSymbolizer* element, which has therefore been extended this way. For example instead of using full 3D tree models for representing trees also just pictures of trees can be added as billboards in order to enhance speed.

7.3.7 Fill

Some more examples are presented: For instance the elements of the new element *Material* are similar to the VRML/Java3D 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* (Fig. 16).

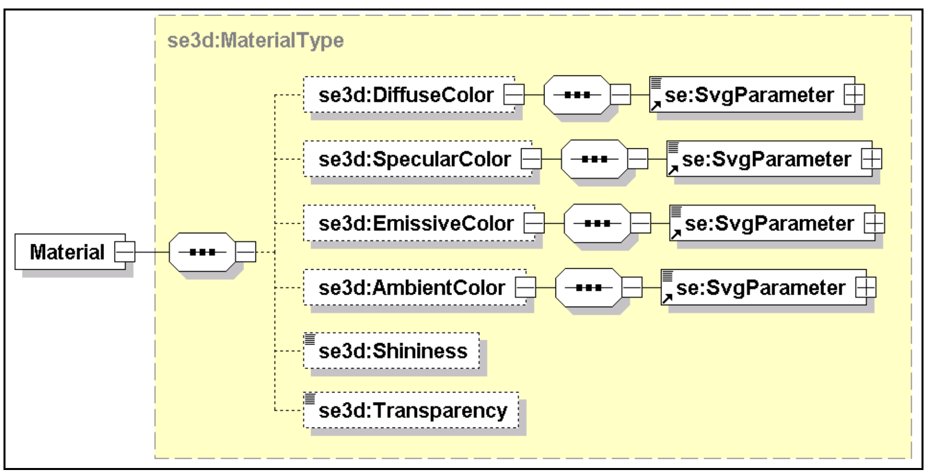

Fig. 15 schema se3d:MaterialType

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 (Fig. 18). 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>
```

Fig. 16 XML exa,mple for the usage of 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 (Fig. 20).

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 (Fig. 20).

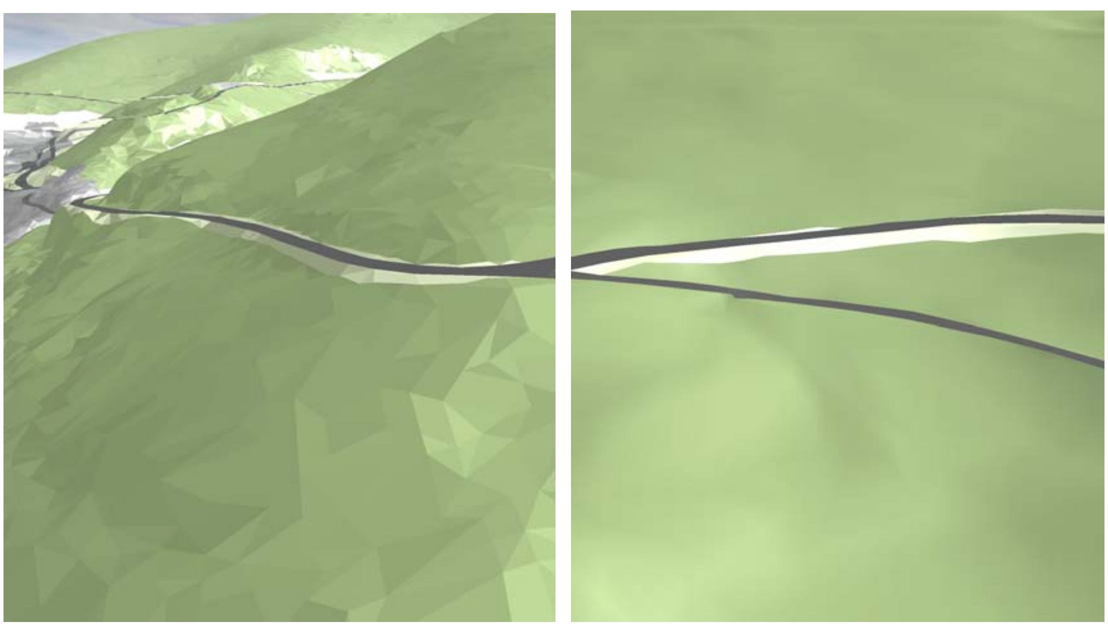

Fig. 17 Terrain model delivered from our W3DS with Flat shading (left) and Gouraud shading (right), as specified through the ShadingModel within the SE-3D

7.3.8 Legend Graphic

Because a 3D scene representing a 3D map 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*.

7.3.9 Point Placement

It is a further possible to position the graphic at a certain 3d point and then to rotate and move this object around any of the three axes. This is realized through the following mechanism:

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 element *Fill* from SE 3D, it becomes possible to describe the geometries by material.

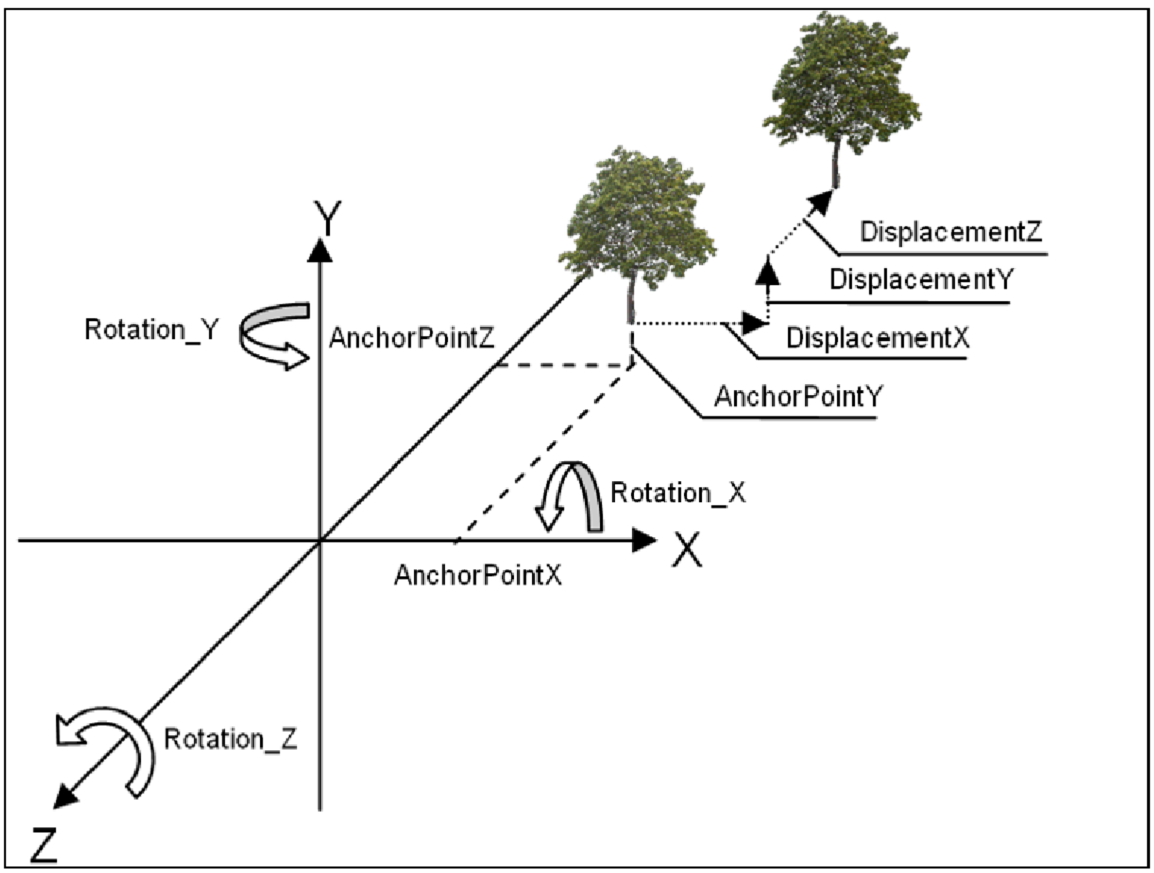

Fig. 18 New types of rotation and displacements in 3D

7.3.10 Label Placement

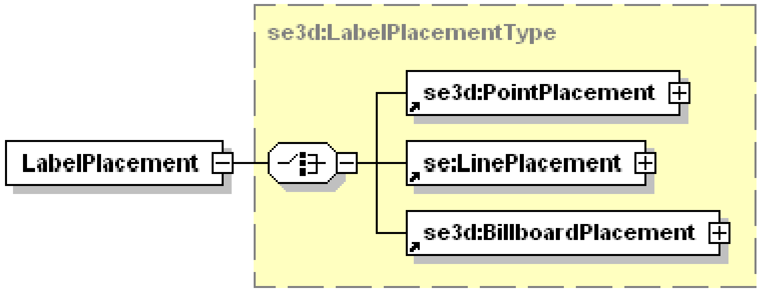

Fig. 19 LabelPlacementType

7.3.11 Anchor Point

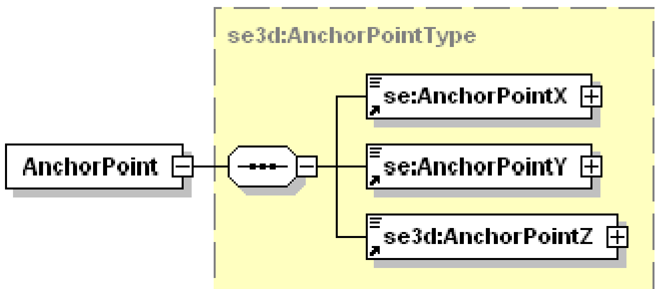

Fig. 20 AnchorPointType

7.3.12 Displacement

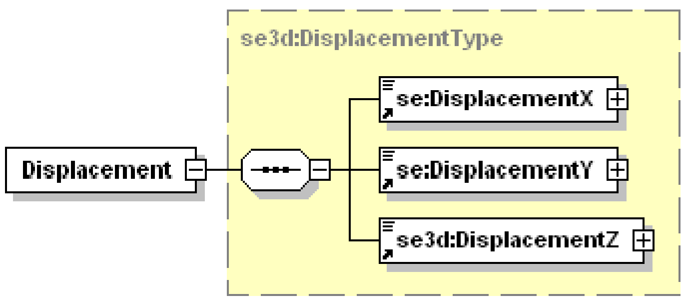

Fig. 21 DisplacementType

8 3D Styled Layer Descriptor Profil of the Web 3D Service

8.1 Introduction

In order to make the SE XML documents usable, the request to the W3DS needs to be extended in comparison to its original version according to the latest specification draft (v.0.4). It is no longer enough to support the parameter “style”, that only can hold a reference to a already known style, but it is necessary to give a reference to the SE document itself. This can be done in three different versions, similar to the possibilities of the WMS-SLD specification. New parameters need to be introduced in the W3DS specification which can be used interchangeable, with benefits and shortcomings for different uses. Further a XML schema for supporting POST requests has been created, and in addition to the existing GetScene request further request types have been added as explained below. But first we introduce the three versions of transmitting the SE document to the W3DS. Therefore the following parameters are introduced.

8.2 Version 1: &SLD=

This parameter holds as value a URL as reference to a SE document. The SE document can be on any web-accessible server defined through the URL. This version is used when the document has actually been created and stored already somewehre. It is transferred through a Get-Request, so usable for most of the usual situations. The parameter may replace the existing parameters LAYER and STYLE. They are no longer needed, as the information about LAYER and STYLEs now is included in the SE document referenced through the URL.

8.3 Version 2: &SLD_BODY=

This version is similar to the first one, as being used within the Get-Request and as it can replace the parameter LAYER and STYLE. But this parameter does not carry a reference to a URL where the SE document resides, but carries the SE document itself as text. This might be a benefit where the SE document is generated dynamically on the fly, e.g. through a web-based client-side symbol editor and when there is no need to store the generated SE document somewhere persistently. On the other hand the drawback is, that XML documents tend to become quite large and that all this text needs to fit in the original GET-Request. This is a problem as web-browser and web-server implementations usually have a limit on the length of the URL they can work with. Therefore it is only appropriate in situations where only small SE files are being produced and where there is no need to reuse them somewhere later, which would require storing them somewhere.

8.4 Version 3: <StyledLayerDescriptor>

This tag allows to use the POST-Request instead of the GET-Request. When using POST it is possible to send a XML file. The main benefit is that it is then possible to send also very large files within the request, even when the files have been created on the fly.

8.5 New request types

We defined a XML-Schema of the GetScene-Request to the W3DS for supporting http POST-requests. This schema is included in APPENDIX A XMLSchema "GetScene – Methode Post".

New request types to the W3DS in addition to GetScene are GetFeatureInfo and DescribeLayer. These requests are helpful for querying further information about individual layers or even features. The information send back by the DescribeLayer request can be used to dynamically define your own styles for the next scene on the client side, as this information is needed for generating appropriate filters such as in the case of the WMS.

9 Conclusion and outlook

In this paper we discussed the first outcomes of a research project that deals with the realization 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 Coors 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.

Fig. 22 Realizing thematic filtering with 3D-SE: highlighting of public buildings in red (LOD 1), see Appendix C Example 2 for SE document.

Fig. 23 DEM with textures according to attributes of the land use classification (LOD 2), see Appendix C Example 4 for SE document.

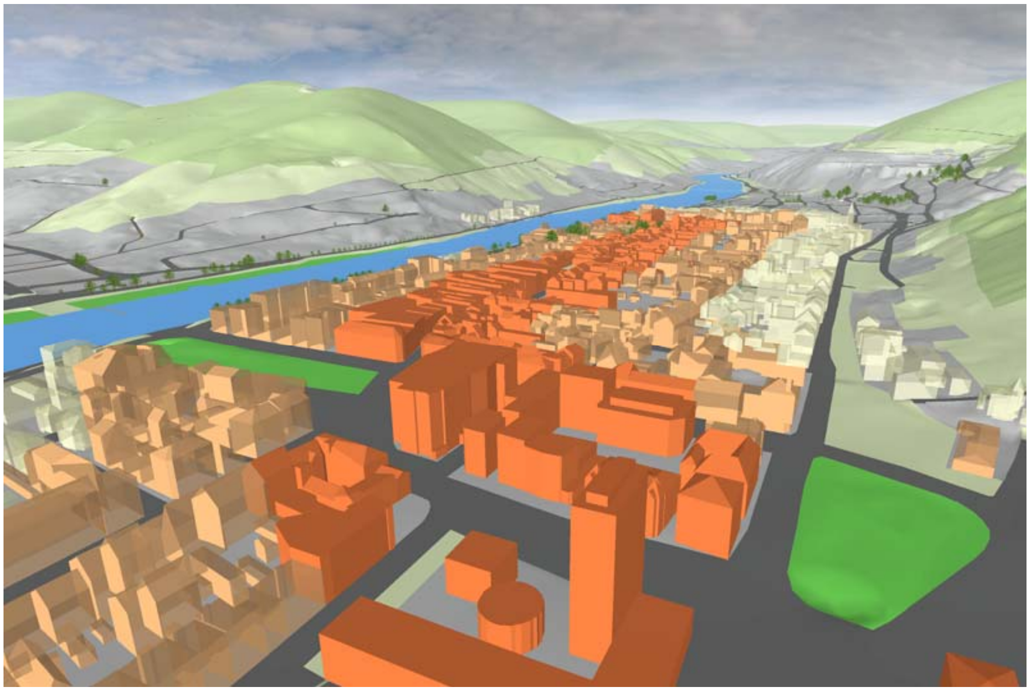

Fig. 24 XNavigator W3DS-Viewer with W3DS scene styled by a focus SLD (highlighting buildings within distinct buffers around a calculated route through different versions of red colour dependent from the distance to the route)

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 has been extended with most of the proposed 3D-SE functionality. 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 realization have been reported.

Now we will be able to evaluate the approach and then we can find out if all combinations of the SE visualization specification are sufficient or which changes have to be made. Whatever the syntax to be a future SE 3D will need to provide mechanisms to define 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 visualizations with more contrast. Also the usefulness of providing Filter-functionality for 3D scences has been demonstrated. If supported by different W3DS SE 3D also will enhance interoperability by allowing to integrate 3D data from several W3DS servers into one scene with the same visualization style.

10 Acknowledgements

We thank all colleagues, in particular Arne Schilling and Jens Basanow for their stimulating cooperation, help and input. This work has been supported by the Klaus-Tschira Foundation (KTS) GmbH, Heidelberg. Data has been provided by ourself, Europan Media Laboratory – EML, Heidelberg and Bureau of Surveying, City of Heidelberg.

Annex A
(normative)

Abstract test suite

In each Implementation Standard document, Annex A shall specify the Abstract Test Suite, as specified in Clause 9 and Annex A of ISO 19105. That Clause and Annex specify the ISO/TC 211 requirements for Abstract Test Suites. Examples of Abstract Test Suites are available in an annex of most ISO 191XX documents, one of the more useful is in ISO 191TBD. Note that this guidance may be more abstract than needed in an OpenGIS® Implementation Standard.

Inclusion of the Abstract Test Suite is expected in version 1.0.0 of each OGC Implementation Standard. In earlier versions, the following paragraph can be used:

An abstract test suite is not provided in this version of this Implementation Standard, but will be provided in version 1.0.0.

Annex B
(normative)

XML Schema Documents

In addition to this document, this standard includes several normative XML Schema Documents. These XML Schema Documents are bundled in a zip file with the present document. After OGC acceptance of a Version 1.0.0 of this standard, these XML Schema Documents will also be posted online at the URL http://schemas.opengeospatial.net/TBD/1.0.0. In the event of a discrepancy between the bundled and online versions of the XML Schema Documents, the online files shall be considered authoritative.

The abilities now specified in this document use specified XML Schema Documents included in the zip file with this document. These XML Schema Documents combine the XML schema fragments listed in various subclauses of this document, eliminating duplications. These XML Schema Documents are named:

filter.xsd

feature.xsd

FeatureStyle.xsd

owsCommon.xsd

StyledLayerDescriptor.xsd

These XML Schema Documents use and build on the OWS common XML Schema Documents specified [OGC 05-008], named:

ows19115subset.xsd

owsCommon.xsd

owsDataIdentification.xsd

owsExceptionReport.xsd

owsGetCapabilities.xsd

owsOperationsMetadata.xsd

owsServiceIdentification.xsd

owsServiceProvider.xsd

All these XML Schema Documents contain documentation of the meaning of each element and attribute, and this documentation shall be considered normative as specified in Subclause 11.6.3 of [OGC 05-008].

Annex C
(informative)

Example XML documents

C.1 Introduction

This annex provides more example XML documents than given in the body of this document.

C.2 Example 1

```xml
<?xml version="1.0" encoding="UTF-8"?>
<StyledLayerDescriptor  xmlns="http://www.opengis.net/sld3d"
                        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                  xmlns:sld3d="http://www.opengis.net/sld3d"
                        xmlns:sld="http://www.opengis.net/sld"
                                  xmlns:se3d="http://www.opengis.net/se3d"
                        xmlns:se="http://www.opengis.net/se"
                                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                                  xmlns:ogc="http://www.opengis.net/ogc"
                         xsi:schemaLocation="../3d-sld/0.1.0/StyledLayerDescriptor.xsd"
                        version="0.1.0">
<NamedLayer>
   <se:Name>Gebaeude_LOD1</se:Name>
    <UserStyle>
      <se:Name>Usability</se:Name>
      <se3d:FeatureTypeStyle>
        <se3d:Rule>
          <Filter xmlns="http://www.opengis.net/ogc">
            <ogc:FeatureId fid="building.34890898"/>
            <ogc:FeatureId fid="building.34894858"/>
            <ogc:FeatureId fid="building.34891279"/>
            <ogc:FeatureId fid="building.34894864"/>
            <ogc:FeatureId fid="building.34895371"/>
            <ogc:FeatureId fid="building.34892242"/>
          </Filter>
          <se3d:PolygonSymbolizer>
            <se3d:Fill>
              <se3d:Material>
                <se3d:DiffuseColor>
                  <se:SvgParameter name="">#32CD32</se:SvgParameter>
                </se3d:DiffuseColor>
                <se3d:AmbientColor>
                  <se:SvgParameter name="">#333333</se:SvgParameter>
                </se3d:AmbientColor>
                <se3d:Shininess>0.3</se3d:Shininess>
              </se3d:Material>
              <se3d:ShadingModel>FlatShading</se3d:ShadingModel>
            </se3d:Fill>
          </se3d:PolygonSymbolizer>
        </se3d:Rule>
        <se3d:Rule>
          <se:ElseFilter/>
          <se3d:PolygonSymbolizer>
            <se3d:Fill>
              <se3d:Material>
                <se3d:DiffuseColor>
```

```
                    <se:SvgParameter name="">#D8D4C9</se:SvgParameter>
                  </se3d:DiffuseColor>
                  <se3d:Shininess>0.3</se3d:Shininess>
                </se3d:Material>
                <se3d:ShadingModel>FlatShading</se3d:ShadingModel>
              </se3d:Fill>
            </se3d:PolygonSymbolizer>
          </se3d:Rule>
        </se3d:FeatureTypeStyle>
      </UserStyle>
    </NamedLayer>
</StyledLayerDescriptor>
```

C.3 Example 2

```
<StyledLayerDescriptor xmlns="http://www.opengis.net/sld3d" [ ... ] version="0.1.0">
  <NamedLayer>
  <se:Name>Gebaeude_LOD1</se:Name>
  <UserStyle>
      <se:Name>Kartographisch</se:Name>
      <se3d:FeatureTypeStyle>
        <se3d:Rule>
          <se:ElseFilter/>
          <se3d:PolygonSymbolizer>
            <se3d:Fill>
              <se3d:Material>
                <se3d:DiffuseColor>
                  <se:SvgParameter name="">#D8D4C9</se:SvgParameter>
                </se3d:DiffuseColor>
                <se3d:AmbientColor>
                  <se:SvgParameter name="">#333333</se:SvgParameter>
                </se3d:AmbientColor>
                <se3d:Shininess>0.3</se3d:Shininess>
              </se3d:Material>
              <se3d:ShadingModel>FlatShading</se3d:ShadingModel>
            </se3d:Fill>
          </se3d:PolygonSymbolizer>
        </se3d:Rule>
        <se3d:Rule>
          <Filter xmlns="http://www.opengis.net/ogc">
            <PropertyIsLike escapeChar="" singleChar="" wildCard="">
              <PropertyName>objkey</PropertyName>
              <Literal>11</Literal>
            </PropertyIsLike>
          </Filter>
          <se3d:PolygonSymbolizer>
            <se3d:Fill>
              <se3d:Material>
                <se3d:DiffuseColor>
                  <se:SvgParameter name="">#F69292</se:SvgParameter>
                </se3d:DiffuseColor>
                <se3d:AmbientColor>
                  <se:SvgParameter name="">#333333</se:SvgParameter>
                </se3d:AmbientColor>
                <se3d:Shininess>0.3</se3d:Shininess>
              </se3d:Material>
              <se3d:ShadingModel>FlatShading</se3d:ShadingModel>
            </se3d:Fill>
          </se3d:PolygonSymbolizer>
        </se3d:Rule>
        <se3d:Rule>
          <Filter xmlns="http://www.opengis.net/ogc">
            <PropertyIsLike wildCard="" singleChar="" escapeChar="">
              <PropertyName>objkey</PropertyName>
              <Literal>13</Literal>
            </PropertyIsLike>
          </Filter>
          <se3d:PolygonSymbolizer>
```



```
            <se3d:Fill>
              <se3d:Material>
                <se3d:DiffuseColor>
                  <se:SvgParameter name="">#FFF8D4</se:SvgParameter>
                </se3d:DiffuseColor>
                <se3d:AmbientColor>
                  <se:SvgParameter name="">#333333</se:SvgParameter>
                </se3d:AmbientColor>
                <se3d:Shininess>0.3</se3d:Shininess>
              </se3d:Material>
              <se3d:ShadingModel>FlatShading</se3d:ShadingModel>
            </se3d:Fill>
          </se3d:PolygonSymbolizer>
        </se3d:Rule>
      </se3d:FeatureTypeStyle>
    </UserStyle>
  </NamedLayer>
</StyledLayerDescriptor>
```

C.4 Example 3

```
<StyledLayerDescriptor =http://www.opengis.net/sld3d [ … ] version="0.1.0">
  <NamedLayer>
<se:Name>Baeume</se:Name>
<UserStyle>
      <se:Name>BaumKomplex</se:Name>
      <se3d:FeatureTypeStyle>
        <se3d:Rule>
          <se3d:PointSymbolizer>
            <se3d:Graphic_3D>
               <se:ExternalGraphic>
                 <se:OnlineResource xmlns:ref="http://www.w3.org/1999/xlink"
                                    ref:href="http://MyServerURL/W3DS/protos/Baum_komplex.wrl"
                                    ref:title="Baum_komplex"/>
                 <se:Format>wrl</se:Format>
               </se:ExternalGraphic>
          </se3d:Graphic_3D>
          </se3d:PointSymbolizer>
        </se3d:Rule>
      </se3d:FeatureTypeStyle>
    </UserStyle>
  </NamedLayer>
</StyledLayerDescriptor>
```

C.5 Example 4

```
<StyledLayerDescriptor xmlns="http://www.opengis.net/sld3d" [ … ] version="0.1.0">
 <se:Name>SLD3D_integrated_DEM</se:Name>
  <NamedLayer>
    <se:Name>integrated_DEM</se:Name>
    <UserStyle>
      <se:Name>KartoTextur</se:Name>
      <se3d:FeatureTypeStyle>
        <se3d:Rule>
          <se:Name>Baubloecke</se:Name>
          <se:Description>
            <se:Title>Baubloecke</se:Title>
            <se:Abstract>areas with the landuse Baubloecke</se:Abstract>
          </se:Description>
          <Filter xmlns="http://www.opengis.net/ogc">
            <PropertyIsLike wildCard="" singleChar="" escapeChar="">
              <PropertyName>landuse</PropertyName>
              <Literal>Baubloecke</Literal>
            </PropertyIsLike>
          </Filter>
          <se3d:PolygonSymbolizer>
```

```
            <se3d:Fill>
          <GraphicFill xmlns="http://www.opengis.net/se">
                <Graphic>
                  <ExternalGraphic>
                    <OnlineResource xmlns:ns1="http://www.w3.org/1999/xlink"
                      ns1:href="textures/alterasphalt.jpg"/>
                    <Format> jpg</Format>
                  </ExternalGraphic>
                </Graphic>
              </GraphicFill>
              <se3d:Material>
                <se3d:DiffuseColor>
                  <se:SvgParameter name="fill">#FFFCE1</se:SvgParameter>
                </se3d:DiffuseColor>
                <se3d:SpecularColor>
                  <se:SvgParameter name="fill">#FFFFFF</se:SvgParameter>
                </se3d:SpecularColor>
                <se3d:AmbientColor>
                  <se:SvgParameter name="fill">#FCF8D3</se:SvgParameter>
                </se3d:AmbientColor>
                <se3d:Shininess>0.1</se3d:Shininess>
              </se3d:Material>
            </se3d:Fill>
          </se3d:PolygonSymbolizer>
        </se3d:Rule>
        <se3d:Rule>
          <se:Name>Strassen</se:Name>
         [ … ]
        </se3d:Rule>
        <se3d:Rule>
          <se:Name>Bahn</se:Name>
    [ … ]
        </se3d:Rule>
        <se3d:Rule>
          <se:Name>Waldflaechen</se:Name>
    [ … ]
        </se3d:Rule>
        <se3d:Rule>
          <se:Name>Gruenflaechen</se:Name>
    [ … ]
        </se3d:Rule>
        <se3d:Rule>
          <se:Name>Wasserflaechen</se:Name>
    [ … ]
        </se3d:Rule>
      </se3d:FeatureTypeStyle>
    </UserStyle>
  </NamedLayer>
</StyledLayerDescriptor>
```

Annex D
(normative)

XMLSchema “GetScene – Methode POST”

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema  xmlns:sld="http://www.opengis.net/sld"
		xmlns:sld3d="http://www.opengis.net/sld3d"
		xmlns:ogc="http://www.opengis.net/ogc"
		xmlns:se="http://www.opengis.net/se"
		xmlns:se3d="http://www.opengis.net/se3d"
		xmlns:gml="http://www.opengis.net/gml"
		xmlns:ows="http://www.opengis.net/ows"
		xmlns:xsd="http://www.w3.org/2001/XMLSchema"
		targetNamespace="http://www.opengis.net/sld3d"
		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"/>
	<xsd:import namespace="http://www.opengis.net/ows" schemaLocation="../../ows/1.0.0/owsCommon.xsd"/>
	<xsd:include schemaLocation="StyledLayerDescriptor.xsd"/>
	<xsd:element name="GetScene" type="sld3d:GetSceneType"/>
	<xsd:complexType name="GetSceneType">
		<xsd:sequence>
			<!-- LAYERS and STYLES parameters are handled by sld:StyledLayerDescriptor-->
			<xsd:element ref="sld3d:StyledLayerDescriptor" minOccurs="0"/>
			<xsd:element name="FORMAT" type="sld3d:OutputType"/>
			<xsd:element name="SRS" type="xsd:string"/>
			<xsd:element name="BBOX" type="ows:BoundingBoxType"/>
			<xsd:element name="POI" type="gml:PointPropertyType" minOccurs="0"/>
			<xsd:element name="PITCH" type="gml:AngleType" minOccurs="0"/>
			<xsd:element name="YAW" type="gml:AngleType" minOccurs="0"/>
			<xsd:element name="ROLL" type="gml:AngleType" minOccurs="0"/>
			<xsd:element name="DISTANCE" type="xsd:double" minOccurs="0"/>
			<xsd:element name="POC" type="gml:PointPropertyType" minOccurs="0"/>
			<xsd:element name="AOV" type="gml:AngleType" minOccurs="0"/>
			<xsd:element name="MINHEIGHT" type="xsd:double" minOccurs="0"/>
			<xsd:element name="MAXHEIGHT" type="xsd:double" minOccurs="0"/>
			<xsd:element name="LAYERS" type="sld3d:LayerListType" minOccurs="0"/>
			<xsd:element name="STYLES" type="sld3d:StyleListType" minOccurs="0"/>
			<xsd:element name="TRANSLATE" type="gml:PointPropertyType" minOccurs="0"/>
			<xsd:element name="TIME" type="xsd:dateTime" minOccurs="0"/>
			<xsd:element name="EXCEPTIONS" type="sld3d:ExceptionsType" minOccurs="0"/>
			<xsd:element name="ENVIRONMENT" type="xsd:boolean" minOccurs="0"/>
			<xsd:element name="BGColor" type="sld3d:ColorType" minOccurs="0"/>
			<xsd:element name="BGIMAGE" type="xsd:anyURI" minOccurs="0"/>
		</xsd:sequence>
		<xsd:attribute name="version" type="xsd:string" fixed="0.1.0"/>
	</xsd:complexType>
	<xsd:complexType name="LayerListType">
		<xsd:sequence>
			<xsd:element name="LayerName" type="xsd:string" maxOccurs="unbounded"/>
		</xsd:sequence>
	</xsd:complexType>
	<xsd:complexType name="StyleListType">
		<xsd:sequence>
			<xsd:element name="StyleName" type="xsd:string" maxOccurs="unbounded"/>
		</xsd:sequence>
	</xsd:complexType>
	<xsd:simpleType name="OutputType">
		<xsd:restriction base="xsd:string">
			<xsd:enumeration value="VRML"/>
			<xsd:enumeration value="X3D"/>
		</xsd:restriction>
	</xsd:simpleType>
	<xsd:simpleType name="ExceptionsType">
		<xsd:restriction base="xsd:string">
			<xsd:enumeration value="XML"/>
			<xsd:enumeration value="INIMAGE"/>
			<xsd:enumeration value="BLANK"/>
		</xsd:restriction>
	</xsd:simpleType>
	<xsd:complexType name="ColorType">
		<xsd:sequence>
			<xsd:element name="red" type="xsd:hexBinary"/>
			<xsd:element name="green" type="xsd:hexBinary"/>
```



```
                <xsd:element name="blue" type="xsd:hexBinary"/>
            </xsd:sequence>
        </xsd:complexType>
</xsd:schema>
```


Annex E
(normative)

3D-SLD XML-Schema-Definition

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
	xmlns:sld="http://www.opengis.net/sld"
	xmlns:se3d="http://www.opengis.net/se3d"
	xmlns:se="http://www.opengis.net/se"
	xmlns:xsd="http://www.w3.org/2001/XMLSchema"
	xmlns:sld3d="http://www.opengis.net/sld3d"
	targetNamespace="http://www.opengis.net/sld3d"
	elementFormDefault="qualified">
	<xsd:import namespace="http://www.opengis.net/sld" schemaLocation="../../sld/1.1.0/StyledLayerDescriptor.xsd"/>
	<xsd:import namespace="http://www.opengis.net/se3d" schemaLocation="../../se3d/1.0.0/se3d.xsd"/>
	<xsd:import namespace="http://www.opengis.net/se" schemaLocation="../../se/1.1.0/FeatureStyle.xsd"/>
	<!-- ********************************************************************** -->
	<xsd:annotation>
		<xsd:documentation> STYLED LAYER DESCRIPTOR version 0.1.0</xsd:documentation>
	</xsd:annotation>
	<xsd:element name="StyledLayerDescriptor">
		<xsd:annotation>
			<xsd:documentation> A StyledLayerDescriptor is a sequence of styled layers, represented
				at the first level by NamedLayer and UserLayer elements. </xsd:documentation>
		</xsd:annotation>
		<xsd:complexType>
			<xsd:sequence>
				<xsd:element ref="se:Name" minOccurs="0"/>
				<xsd:element ref="se:Description" minOccurs="0"/>
				<xsd:element ref="sld:UseSLDLibrary" minOccurs="0" maxOccurs="unbounded"/>
				<xsd:choice minOccurs="0" maxOccurs="unbounded">
					<xsd:element ref="sld3d:NamedLayer"/>
					<xsd:element ref="sld3d:UserLayer"/>
				</xsd:choice>
			</xsd:sequence>
			<xsd:attribute name="version" type="se3d:VersionType" use="required"/>
		</xsd:complexType>
	</xsd:element>
	<!-- ********************************************************************** -->
	<xsd:annotation>
		<xsd:documentation> LAYERS AND STYLES </xsd:documentation>
	</xsd:annotation>
	<xsd:element name="NamedLayer">
		<xsd:annotation>
			<xsd:documentation> A NamedLayer is a layer of data that has a name advertised by a WMS / W3DS.
			</xsd:documentation>
		</xsd:annotation>
		<xsd:complexType>
			<xsd:sequence>
				<xsd:element ref="se:Name"/>
				<xsd:element ref="se:Description" minOccurs="0"/>
				<xsd:element ref="sld:LayerFeatureConstraints" minOccurs="0"/>
				<xsd:choice minOccurs="0" maxOccurs="unbounded">
					<xsd:element ref="sld:NamedStyle"/>
					<xsd:element ref="sld3d:UserStyle"/>
				</xsd:choice>
			</xsd:sequence>
		</xsd:complexType>
	</xsd:element>
	<xsd:element name="UserLayer">
		<xsd:annotation>
			<xsd:documentation> A UserLayer allows a user-defined layer to be built from WFS and WCS
				data. </xsd:documentation>
		</xsd:annotation>
		<xsd:complexType>
			<xsd:sequence>
```



```
				<xsd:element ref="se:Name" minOccurs="0"/>
				<xsd:element ref="se:Description" minOccurs="0"/>
				<xsd:choice minOccurs="0">
					<xsd:element ref="sld:RemoteOWS"/>
					<xsd:element ref="sld:InlineFeature"/>
				</xsd:choice>
				<xsd:choice minOccurs="0">
					<xsd:element ref="sld:LayerFeatureConstraints"/>
					<xsd:element ref="sld:LayerCoverageConstraints"/>
				</xsd:choice>
				<xsd:element ref="sld3d:UserStyle" maxOccurs="unbounded"/>
			</xsd:sequence>
		</xsd:complexType>
	</xsd:element>
	<xsd:element name="UserStyle">
		<xsd:annotation>
			<xsd:documentation> A UserStyle allows user-defined styling and is semantically
				equivalent to a WMS named style. External FeatureTypeStyles or CoverageStyles can be
				linked using an OnlineResource-element </xsd:documentation>
		</xsd:annotation>
		<xsd:complexType>
			<xsd:sequence>
				<xsd:element ref="se:Name" minOccurs="0"/>
				<xsd:element ref="se:Description" minOccurs="0"/>
				<xsd:element ref="sld:IsDefault" minOccurs="0"/>
				<xsd:choice maxOccurs="unbounded">
					<xsd:element ref="se3d:FeatureTypeStyle"/>
					<xsd:element ref="se3d:CoverageStyle"/>
					<xsd:element ref="se:OnlineResource"/>
				</xsd:choice>
			</xsd:sequence>
		</xsd:complexType>
	</xsd:element>
	<xsd:element name="IsDefault" type="xsd:boolean"/>
</xsd:schema>
```


Annex F
(normative)

3D-SE XML-Schema-Definition


```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- SE-3D schema -->
<xsd:schema
	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"
	targetNamespace="http://www.opengis.net/se3d"
	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"/>
	<!-- ********************************************************************* -->
	<xsd:annotation>
		<xsd:documentation> Symbology Encoding FEATURE STYLE </xsd:documentation>
	</xsd:annotation>
	<xsd:element name="FeatureTypeStyle" type="se3d:FeatureTypeStyleType">
		<xsd:annotation>
			<xsd:documentation>
				A FeatureTypeStyle contains styling information specific to one feature type.
			</xsd:documentation>
		</xsd:annotation>
	</xsd:element>
	<xsd:complexType name="FeatureTypeStyleType">
		<xsd:sequence>
			<xsd:element ref="se:Name" minOccurs="0"/>
			<xsd:element ref="se:Description" minOccurs="0"/>
			<xsd:element ref="se:FeatureTypeName" minOccurs="0"/>
			<xsd:element ref="se:SemanticTypeIdentifier" minOccurs="0" maxOccurs="unbounded"/>
			<xsd:choice maxOccurs="unbounded">
				<xsd:element ref="se3d:Rule"/>
				<xsd:element ref="se:OnlineResource"/>
			</xsd:choice>
		</xsd:sequence>
		<xsd:attribute name="version" type="se3d:VersionType"/>
	</xsd:complexType>
	<!-- ********************************************************************* -->
	<xsd:element name="CoverageStyle" type="se3d:CoverageStyleType">
		<xsd:annotation>
			<xsd:documentation>
				A CoverageStyle contains styling information specific to one Coverage offering.
			</xsd:documentation>
		</xsd:annotation>
	</xsd:element>
	<xsd:complexType name="CoverageStyleType">
		<xsd:sequence>
			<xsd:element ref="se:Name" minOccurs="0"/>
			<xsd:element ref="se:Description" minOccurs="0"/>
			<xsd:element ref="se:CoverageName" minOccurs="0"/>
			<xsd:element ref="se:SemanticTypeIdentifier" minOccurs="0" maxOccurs="unbounded"/>
			<xsd:choice maxOccurs="unbounded">
				<xsd:element ref="se3d:Rule"/>
				<xsd:element ref="se:OnlineResource"/>
			</xsd:choice>
		</xsd:sequence>
		<xsd:attribute name="version" type="se3d:VersionType"/>
	</xsd:complexType>
	<!-- ********************************************************************* -->
	<xsd:annotation>
		<xsd:documentation> SE3D Symbolizer version 0.0.1 </xsd:documentation>
	</xsd:annotation>
```

```
<xsd:element name="Symbolizer" type="se:SymbolizerType" abstract="true"/>
<!-- ********************************************************************************** -->
<xsd:annotation>
	<xsd:documentation> Legend Graphic </xsd:documentation>
</xsd:annotation>
<xsd:element name="LegendGraphic" type="se3d:LegendGraphicType"/>
<xsd:complexType name="LegendGraphicType">
	<xsd:choice>
		<xsd:element ref="se:Graphic"/>
		<xsd:element ref="se3d:Graphic_3D"/>
	</xsd:choice>
</xsd:complexType>
<!-- ********************************************************************************** -->
<xsd:element name="Mark_3D" type="se3d:Mark_3D_Type">
	<xsd:annotation>
		<xsd:documentation>
			A "Mark 3D" specifies a geometric shape and applies colouring to it.
		</xsd:documentation>
	</xsd:annotation>
</xsd:element>
<xsd:complexType name="Mark_3D_Type">
	<xsd:sequence>
		<xsd:choice minOccurs="0">
			<xsd:element ref="se3d:WellKnownName"/>
			<xsd:sequence>
				<xsd:choice>
					<xsd:element ref="se:OnlineResource"/>
					<xsd:element ref="se:InlineContent"/>
				</xsd:choice>
				<xsd:element ref="se:Format"/>
				<xsd:element ref="se:MarkIndex" minOccurs="0"/>
			</xsd:sequence>
		</xsd:choice>
		<xsd:element ref="se3d:Fill" minOccurs="0"/>
		<xsd:element ref="se:Stroke" minOccurs="0"/>
	</xsd:sequence>
</xsd:complexType>
<xsd:element name="WellKnownName" type="xsd:string"/>
<!-- ************************************************************************** -->
<xsd:element name="Graphic_3D" type="se3d:Graphic_3D_Type">
	<xsd:annotation>
		<xsd:documentation>
			A "Graphic_3D" specifies or refers to a "graphic Symbolizer" with 3D-Format, size, and colouring.
		</xsd:documentation>
	</xsd:annotation>
</xsd:element>
<xsd:complexType name="Graphic_3D_Type">
	<xsd:sequence>
		<xsd:choice minOccurs="0" maxOccurs="unbounded">
			<xsd:element ref="se:ExternalGraphic"/>
			<xsd:element ref="se3d:Mark_3D"/>
		</xsd:choice>
		<xsd:element ref="se:Opacity" minOccurs="0"/>
		<xsd:element ref="se:Size" minOccurs="0"/>
		<xsd:element ref="se3d:Rotation_3D" minOccurs="0"/>
		<xsd:element ref="se3d:AnchorPoint" minOccurs="0"/>
		<xsd:element ref="se3d:Displacement" minOccurs="0"/>
	</xsd:sequence>
</xsd:complexType>
<!-- ****************************************************************************** -->
<xsd:element name="Rotation_3D" type="se3d:Rotation_3D_Type">
	<xsd:annotation>
		<xsd:documentation>
			Three dimensional rotations allow rotation around all three axes of a coordinate system.
		</xsd:documentation>
	</xsd:annotation>
</xsd:element>
<xsd:complexType name="Rotation_3D_Type">
	<xsd:sequence>
		<xsd:element ref="se3d:Rotation_X" minOccurs="0"/>
		<xsd:element ref="se3d:Rotation_Y" minOccurs="0"/>
```



```xml
			<xsd:element ref="se3d:Rotation_Z" minOccurs="0"/>
		</xsd:sequence>
	</xsd:complexType>
	<xsd:element name="Rotation_X" type="se:ParameterValueType"/>
	<xsd:element name="Rotation_Y" type="se:ParameterValueType"/>
	<xsd:element name="Rotation_Z" type="se:ParameterValueType"/>
	<!-- ********************************************************************************* -->
	<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>
	<!-- ********************************************************************************* -->
	<xsd:element name="SolidSymbolizer" type="se3d:SolidSymbolizerType" substitutionGroup="se3d:Symbolizer">
		<xsd:annotation>
			<xsd:documentation>
				A "Solid Symbolizer" specifies the rendering of a object geometry, including its interior fill and
				border stroke.
			</xsd:documentation>
		</xsd:annotation>
	</xsd:element>
	<xsd:complexType name="SolidSymbolizerType">
		<xsd:complexContent>
			<xsd:extension base="se:SymbolizerType">
				<xsd:sequence>
					<xsd:element ref="se:Geometry" minOccurs="0"/>
					<xsd:element ref="se:Stroke" minOccurs="0"/>
					<xsd:element ref="se3d:CoverFill" minOccurs="0"/>
					<xsd:element ref="se3d:VolumeFill" minOccurs="0"/>
				</xsd:sequence>
			</xsd:extension>
		</xsd:complexContent>
	</xsd:complexType>
	<!-- ********************************************************************************* -->
	<xsd:element name="CoverFill" type="se3d:FillType">
		<xsd:annotation>
			<xsd:documentation>
				CoverFill is for a SolidSymbolizer and describe only the hull of the geometry.
			</xsd:documentation>
		</xsd:annotation>
	</xsd:element>
	<!-- ********************************************************************************* -->
	<xsd:element name="VolumeFill" type="se3d:FillType">
		<xsd:annotation>
			<xsd:documentation>
				VolumeFill is for a SolidSymbolizer and describe the inside of the geometry.
			</xsd:documentation>
		</xsd:annotation>
	</xsd:element>
	<!-- ********************************************************************************* -->
	<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:Surface Symbology" minOccurs="0"/>
                                <xsd:element ref="se3d:Fill" minOccurs="0"/>
                                <xsd:element ref="se:Stroke" minOccurs="0"/>
                        </xsd:sequence>
                </xsd:extension>
        </xsd:complexContent>
</xsd:complexType>
<!-- ********************************************************************************* -->
<xsd:element name="SurfaceSymbology" type="se3d:SurfaceSymbologyType">
        <xsd:annotation>
                <xsd:documentation>
                        A "SurfaceSymbology" states the type of the cartographic visualisation.
                </xsd:documentation>
        </xsd:annotation>
</xsd:element>
<xsd:complexType name="SurfaceSymbologyType">
        <xsd:choice>
                <xsd:element ref="se3d:Contour"/>
                <xsd:element ref="se3d:Elevation"/>
                <xsd:element ref="se3d:Slope"/>
                <xsd:element ref="se3d:Aspect"/>
        </xsd:choice>
</xsd:complexType>
<!-- ********************************************************************************* -->
<xsd:element name="Contour" type="se3d:ContourType">
        <xsd:annotation>
                <xsd:documentation>
                        A "Contour" define the Contour of an Surface.
                </xsd:documentation>
        </xsd:annotation>
</xsd:element>
<xsd:complexType name="ContourType">
        <xsd:sequence>
                <xsd:element name="Interval" type="xsd:int" minOccurs="0"/>
                <xsd:element name="BaseHeight" type="xsd:double" minOccurs="0"/>
                <xsd:element ref="se:SvgParameter" minOccurs="0"/>
                <!-- Color -->
        </xsd:sequence>
</xsd:complexType>
<!-- ********************************************************************************* -->
<xsd:element name="Elevation" type="se3d:ElevationType">
        <xsd:annotation>
                <xsd:documentation>
                        A "Elevation" define the face elevation with a graduated color ramp.
                </xsd:documentation>
        </xsd:annotation>
</xsd:element>
<xsd:complexType name="ElevationType">
        <xsd:sequence>
                <xsd:element ref="se:ExternalGraphic" minOccurs="0"/>
                <xsd:element name="Classes" type="xsd:int" minOccurs="0"/>
        </xsd:sequence>
</xsd:complexType>
<!-- ********************************************************************************* -->
<xsd:element name="Slope" type="se3d:SlopeType">
        <xsd:annotation>
                <xsd:documentation>
                        A "Slope" defines a Visualisation with the Slope and colored Surface with the result.
                </xsd:documentation>
        </xsd:annotation>
</xsd:element>
<xsd:complexType name="SlopeType">
```

```
        <xsd:sequence>
            <xsd:element ref="se:ExternalGraphic" minOccurs="0"/>
            <xsd:element name="Classes" type="xsd:int" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <!-- ********************************************************************************* -->
    <xsd:element name="Aspect" type="se3d:AspectType">
        <xsd:annotation>
            <xsd:documentation>
                A "Aspect" defines a Visualisation with the Aspect and colored Surface with the result.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:complexType name="AspectType">
        <xsd:sequence>
            <xsd:element ref="se:ExternalGraphic" minOccurs="0"/>
            <xsd:element name="Classes" type="xsd:int" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <!-- ********************************************************************************* -->
    <xsd:element name="LineSymbolizer" type="se3d:LineSymbolizerType" substitutionGroup="se3d:Symbolizer">
        <xsd:annotation>
            <xsd:documentation>
                A LineSymbolizer is used to render a "stroke" along a linear geometry.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:complexType name="LineSymbolizerType">
        <xsd:complexContent>
            <xsd:extension base="se:SymbolizerType">
                <xsd:sequence>
                    <xsd:element ref="se:Geometry" minOccurs="0"/>
                    <xsd:element ref="se:Stroke" minOccurs="0"/>
                    <xsd:element ref="se:PerpendicularOffset" minOccurs="0"/>
                    <xsd:choice minOccurs="0">
                        <xsd:element name="Pipe">
                            <xsd:complexType>
                                <xsd:sequence>
                                    <xsd:element name="Radius"/>
                                    <xsd:element ref="se:SvgParameter" minOccurs="0"/>
                                </xsd:sequence>
                            </xsd:complexType>
                        </xsd:element>
                        <xsd:element ref="se3d:BillboardPlacement"/>
                    </xsd:choice>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <!-- ********************************************************************************* -->
    <xsd:element name="BillboardPlacement" type="se3d:BillboardPlacementType">
        <xsd:annotation>
            <xsd:documentation>
                "Billboard Placement" is a possibility to be placed geometry on a Billboard.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:complexType name="BillboardPlacementType">
        <xsd:sequence>
            <xsd:element ref="se3d:AnchorPoint" minOccurs="0"/>
            <xsd:element ref="se3d:Displacement" minOccurs="0"/>
            <xsd:element ref="se:Rotation" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <!-- ********************************************************************************* -->
    <xsd:element name="AnchorPoint" type="se3d:AnchorPointType">
        <xsd:annotation>
            <xsd:documentation>
                An "AnchorPoint" identifies the location for positioning it relative to a point geometry.
            </xsd:documentation>
        </xsd:annotation>
```

```xml
</xsd:element>
<xsd:complexType name="AnchorPointType">
	<xsd:sequence>
		<xsd:element ref="se:AnchorPointX"/>
		<xsd:element ref="se:AnchorPointY"/>
		<xsd:element ref="se3d:AnchorPointZ"/>
	</xsd:sequence>
</xsd:complexType>
<xsd:element name="AnchorPointZ" type="se:ParameterValueType"/>
<!-- ********************************************************************************** -->
<xsd:element name="Displacement" type="se3d:DisplacementType">
	<xsd:annotation>
		<xsd:documentation>
			A "Displacement" gives X ,Y and Z offset displacements to use for rendering a text label, graphic or
			other Symbolizer near a point.
	</xsd:documentation>
	</xsd:annotation>
</xsd:element>
<xsd:complexType name="DisplacementType">
	<xsd:sequence>
		<xsd:element ref="se:DisplacementX"/>
		<xsd:element ref="se:DisplacementY"/>
		<xsd:element ref="se3d:DisplacementZ"/>
	</xsd:sequence>
</xsd:complexType>
<xsd:element name="DisplacementZ" type="se:ParameterValueType"/>
<!-- ********************************************************************************** -->
<xsd:element name="PolygonSymbolizer" type="se3d:PolygonSymbolizerType"
			substitutionGroup="se3d:Symbolizer">
	<xsd:annotation>
		<xsd:documentation>
			A "PolygonSymbolizer" specifies the rendering of a polygon or area geometry, including its interior
			fill and border stroke.
		</xsd:documentation>
	</xsd:annotation>
</xsd:element>
<xsd:complexType name="PolygonSymbolizerType">
	<xsd:complexContent>
		<xsd:extension base="se:SymbolizerType">
			<xsd:sequence>
				<xsd:element ref="se:Geometry" minOccurs="0"/>
				<xsd:element ref="se3d:Fill" minOccurs="0"/>
				<xsd:element ref="se:Stroke" minOccurs="0"/>
				<xsd:element ref="se3d:Displacement" minOccurs="0"/>
				<xsd:element ref="se:PerpendicularOffset" minOccurs="0"/>
				<xsd:element ref="se3d:BillboardPlacement"/>
			</xsd:sequence>
		</xsd:extension>
	</xsd:complexContent>
</xsd:complexType>
<!-- *************************************************************************************** -->
<xsd:element name="PointSymbolizer" type="se3d:PointSymbolizerType" substitutionGroup="se3d:Symbolizer">
	<xsd:annotation>
		<xsd:documentation>
			A "PointSymbolizer" specifies the rendering of a "graphic Symbolizer" at a point.
		</xsd:documentation>
	</xsd:annotation>
</xsd:element>
<xsd:complexType name="PointSymbolizerType">
	<xsd:complexContent>
		<xsd:extension base="se:SymbolizerType">
			<xsd:sequence>
				<xsd:element ref="se:Geometry" minOccurs="0"/>
				<xsd:element ref="se:Graphic" minOccurs="0"/>
				<xsd:choice minOccurs="0">
					<xsd:element ref="se3d:Graphic_3D"/>
					<xsd:element ref="se3d:BillboardPlacement"/>
				</xsd:choice>
			</xsd:sequence>
		</xsd:extension>
	</xsd:complexContent>
```



```
</xsd:complexType>
<!-- ********************************************************************************************** -->
<xsd:element name="TextSymbolizer" type="se3d:TextSymbolizerType" substitutionGroup="se3d:Symbolizer">
	<xsd:annotation>
		<xsd:documentation>
			A "TextSymbolizer" is used to render text labels according to various graphical parameters.
		</xsd:documentation>
	</xsd:annotation>
</xsd:element>
<xsd:complexType name="TextSymbolizerType">
	<xsd:complexContent>
		<xsd:extension base="se:SymbolizerType">
			<xsd:sequence>
				<xsd:element ref="se:Geometry" minOccurs="0"/>
				<xsd:element ref="se:Label" minOccurs="0"/>
				<xsd:element ref="se:Font" minOccurs="0"/>
				<xsd:element ref="se3d:LabelPlacement" minOccurs="0"/>
				<xsd:element ref="se:Halo" minOccurs="0"/>
				<xsd:element ref="se3d:Fill" minOccurs="0"/>
				<xsd:element ref="se3d:Depth" minOccurs="0"/>
			</xsd:sequence>
		</xsd:extension>
	</xsd:complexContent>
</xsd:complexType>
<!-- ********************************************************************************************** -->
<xsd:element name="Depth" type="se:ParameterValueType"/>
<!-- ********************************************************************************************** -->
<xsd:element name="RasterSymbolizer" type="se3d:RasterSymbolizerType"
		substitutionGroup="se3d:Symbolizer">
	<xsd:annotation>
		<xsd:documentation>
			A "RasterSymbolizer" is used to specify the rendering of raster/matrix-coverage data (e.g., satellite
			images, DEMs).
		</xsd:documentation>
	</xsd:annotation>
</xsd:element>
<xsd:complexType name="RasterSymbolizerType">
	<xsd:complexContent>
		<xsd:extension base="se:SymbolizerType">
			<xsd:sequence>
				<xsd:element ref="se:Geometry" minOccurs="0"/>
				<xsd:element ref="se:Opacity" minOccurs="0"/>
				<xsd:element ref="se:ChannelSelection" minOccurs="0"/>
				<xsd:element ref="se:OverlapBehavior" minOccurs="0"/>
				<xsd:element ref="se:ColorMap" minOccurs="0"/>
				<xsd:element ref="se:ContrastEnhancement" minOccurs="0"/>
				<xsd:element ref="se:ShadedRelief" minOccurs="0"/>
				<xsd:element ref="se:ImageOutline" minOccurs="0"/>
				<xsd:element ref="se3d:BillboardPlacement" minOccurs="0"/>
			</xsd:sequence>
		</xsd:extension>
	</xsd:complexContent>
</xsd:complexType>
<!-- ********************************************************************************************** -->
<xsd:element name="LabelPlacement" type="se3d:LabelPlacementType">
	<xsd:annotation>
		<xsd:documentation>
			The "LabelPlacement" specifies where and how a text label should be rendered relative to
			a geometry. The present mechanism is poorly aligned with CSS/SVG.
		</xsd:documentation>
	</xsd:annotation>
</xsd:element>
<xsd:complexType name="LabelPlacementType">
	<xsd:choice>
		<xsd:element ref="se3d:PointPlacement"/>
		<xsd:element ref="se:LinePlacement"/>
		<xsd:element ref="se3d:BillboardPlacement"/>
	</xsd:choice>
</xsd:complexType>
<!-- ********************************************************************************************** -->
<xsd:element name="Fill" type="se3d:FillType">
```

```
        <xsd:annotation>
            <xsd:documentation>
                A "Fill" specifies the pattern for filling an area geometry. The allowed SvgParameters are:
                "fill" (color) and "fill-opacity".
            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:complexType name="FillType">
        <xsd:sequence>
            <xsd:element ref="se:GraphicFill" minOccurs="0"/>
            <xsd:element ref="se:SvgParameter" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element ref="se3d:Material" minOccurs="0"/>
            <xsd:element ref="se3d:ShadingModel" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="ShadingModel" type="se3d:ShadingModelType"/>
    <xsd:simpleType name="ShadingModelType">
        <xsd:restriction base="xsd:token">
            <xsd:enumeration value="GouraudShading"/>
            <xsd:enumeration value="FlatShading"/>
        </xsd:restriction>
    </xsd:simpleType>
    <!-- ********************************************************************************************* -->
    <xsd:element name="Material" type="se3d:MaterialType">
        <xsd:annotation>
            <xsd:documentation>
                The "Material" consists of attributes which real materials can define.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:complexType name="MaterialType">
        <xsd:sequence>
            <xsd:element name="DiffuseColor" minOccurs="0">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="se:SvgParameter"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="SpecularColor" minOccurs="0">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="se:SvgParameter"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="EmissiveColor" minOccurs="0">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="se:SvgParameter"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="AmbientColor" minOccurs="0">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element ref="se:SvgParameter"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="Shininess" type="xsd:double" minOccurs="0"/>
            <xsd:element name="Transparency" type="xsd:double" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <!-- ***************************************************************************** -->
    <xsd:element name="PointPlacement" type="se3d:PointPlacementType">
        <xsd:annotation>
            <xsd:documentation>
                A "PointPlacement" specifies how a text label or geometry should be rendered relative to
                a geometric point.
            </xsd:documentation>
```



```
		</xsd:annotation>
	</xsd:element>
	<xsd:complexType name="PointPlacementType">
		<xsd:sequence>
			<xsd:element ref="se3d:AnchorPoint" minOccurs="0"/>
			<xsd:element ref="se3d:Displacement" minOccurs="0"/>
			<xsd:element ref="se3d:Rotation_3D" minOccurs="0"/>
		</xsd:sequence>
	</xsd:complexType>
	<!-- ********************************************************************** -->
	<xsd:simpleType name="VersionType">
		<xsd:annotation>
			<xsd:documentation>
				The "VersionType" merely restricts the version string that may be used with XML documents based
				on this schema.
			</xsd:documentation>
		</xsd:annotation>
		<xsd:restriction base="xsd:string">
			<xsd:enumeration value="0.1.0"/>
		</xsd:restriction>
	</xsd:simpleType>
</xsd:schema>
```


Annex G
(normative)

Table "Proposed 3D Extensions of SLD"

Element	Extension	Function
LegendGraphic	Graphic_3D • ExternalGraphic_3D o Format_3D o OnlineResource • Mark_3D o WellKnownName o Fill • Opacity • Size • Rotation3D	3d legend integration As an external 3d object File format Link to file As a simple 3d object Known object Surfaces filled Opacity Size 3D rotation
Rule	SolidSymbolizer • Geometry • CoverFill • VolumeFill	3d object description Geometry definition Surface filled Volume filled
	SurfaceSymbolizer • SurfaceSymbology o Contour o Elevation o Slope o Aspect • Fill	2.5d surface description Cartography Symbology of Landscape Surface contour Surface elevation Surface slope Surface aspect Surface fill
	LOD Handling • LOD_0 • LOD_1 • LOD_2 • LOD_3 • LOD_4 • Lowest_LOD • Top_LOD	Level of Detail Handling LOD 0 LOD 1 LOD 2 LOD 3 LOD 4 Lowest LOD level of the Sever Top LOD level of the Server
LineSymbolizer	Pipe • Radius • CssParameter	Lines to 3d objects Radius of cylinder Color
	Billboardplacement • AnchorPoint … • Displacement … • Rotation	Billboard Anchor position of billboard Displacement of anchor point Rotation of labels on the billboard
PolygonSymbolizer	BillboardPlacement …	Billboard
PointSymbolizer	Graphic_3D …	Inclusion of 3d Models
	BillboardPlacement …	Billboard
TextSymbolizer	Depth	Text depth
RasterSymbolizer	BillboardPlacement …	Billboard
LabelPlacement	BillboardPlacement …	Billboard
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

Bibliography

1. 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.

2. CityGML: www.citygml.org

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

4. Coors, V. and Zipf, A. (2007): MoNa 3D --- Mobile Navigation using 3D City Models. 4th International Symposium on LBS and Telecartography 2007. Hong Kong.

5. Dietze, L. and Zipf, A. (2007): EXTENDING OGC STYLED LAYER DESCRIPTOR (SLD) FOR THEMATIC CARTOGRAPHY - Towards the ubiquitous use of advanced mapping functions through standardized visualization rules. 4th Int. Symp. on LBS and Telecartography 2007. Hong Kong.

6. 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.

7. 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.

8. Gröger, G., Kolbe, T. H., Czerwinski, A. (2007): City Geography Markup Language. Version 0.4.0 (CityGML), OGC Best Practices Paper. Open Geospatial Consortium, Doc. No. 07-062 2007.

9. 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

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

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

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

13. OGC: Filter Encoding Implementation Specification version 1.1.0 doc.nr. 04-095

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

15. OGC: Candidate OpenGIS® CityGML Implementation Specification (City Geography Markup Language) version 0.4.0 doc.nr. 07-062, pp. 21-32

16. Ramos, H. F., Reitz, T., Haist, J. (2007): Symbology Encoding for 3D GIS – an approach to extend 3D city model visualization to GIS visualization (UDMS 2007), Stuttgart, Germany

17. 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.

18. Neis, P. and Zipf, A. (2008 submitted): Generating 3D Focus Maps for the (mobile) Web - an interoperable approach. In: International Journal of Location Based Services (JLBS). (Special Issue on Telecartography).

19. 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.

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

21. 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. AGILE 2007. International Conference on Geographic Information Science of the Association of Geographic Information Laboratories for Europe (AGILE). Aalborg, Denmark.

22. 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.

23. 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.

24. Zipf, A., J. Basanow, P. Neis, S. Neubauer, A. Schilling, (2007): 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.

25. Zipf, A. and Richter, K.F. (2002): Using Focus Maps to Ease Map Reading. Developing Smart Applications for Mobile Devices. In: Künstliche Intelligenz (KI) (Artificial Intelligence). Special Issue: Spatial Cognition. 04/2002. 35-37.

