Towards 3D Spatial Data Infrastructures (3D-SDI) based on open standards – experiences, results and future issues

Jens Basanow, Pascal Neis, Steffen Neubauer, Arne Schilling, Alexander Zipf

University of Applied Sciences Mainz &University of Bonn (Cartography), Germany. <name>@geographie.uni-bonn.de

1 Introduction

Building up Spatial Data Infrastructures (SDI) has been an important and actively followed topic in geo research for years. It's also regarded in politics and by decision makers as leveraging technology for reducing time and cost in building up geo services for internal usage as well as for public information services. On the European level the new INSPIRE (Acronym for Infrastructure for Spatial Information in Europe) directive 2007/2/EC intends to lay down general rules for the implementation of national spatial data infrastructures for the purpose of environmental policies. On the technical side, SDIs must rely on open standards that are specified by the Open Geospatial Consortium (CS-W, WMS, WFS, WCS, WPS, OpenLS etc.)

Based on the theoretical background of INSPIRE and several discussion drafts of the OGC, we have implemented an SDI for the city of Heidelberg that comprises an array of established OGC services and some new proposed technologies that are required for the extension into the 3rd dimension. In this paper we will discuss the components that have been developed for a 3D SDI and some important aspects that need to be addressed to make this kind of infrastructure work. For the standard services we could use existing open source solutions, others must be extended or developed from the scratch, not to mention new techniques for the data preparation and integration. The components have been implemented for several projects with different goals but always ensuring the interoperability and having the reusability in mind.

The central part is the OGC Web3D Service (W3DS) which delivers the actual 3D data. The W3DS specification is currently in draft status and not yet adopted by the OGC. We will present our own implementation of this service and some implications that come up when considering different use cases. In our case it was important to use the W3DS not only for producing static scenes, but also to request the data piecewise in order to stream it to the client which implements a more dynamic visualization. This is due to large data amounts that are not comparable to 2D bitmaps delivered by a WMS.

A possible extension to the WMS is the support of the Styled Layer Descriptor (SLD) profile for controlling the appearance of maps. It is advisable to separate the geometry or geographic raw data from the visualization rules. Proposals have been made to include further visualization elements directly into CityGML. We suggest to use the SLD specification in combination with W3DS services. We describe how SLD can be extended in order to provide 3D symbolizations - e.g. for 3D points, linestrings, surfaces, and solids.

Other topics that need to be addressed in this context is an adequate way to describe our 3D data in a catalogue service, for which we examined different alternatives. Additionally we examine the integration of route services, for which the OGC OpenLS specification can be used, as we will show.

Finally we want to give an outlook on future research topics that arise from current trends such as Location Based Services or Service Oriented Architectures (SOA). We need to investigate how the presented concepts can be applied to mobile 3D navigation services, which have different requirements in terms of visualization and user guidance. In the long term, higher level concept for defining chains of web services within an SOA could be applied that help orchestrating the SDI services in a more flexible way. In particular, the Business Process Execution Language (BPEL) could be used for defining scenarios that are realized through chaining open GI services that constitute SDIs.

2 3D Data Management – an overview

Data management is at the heart of an SDI. A powerful database is necessary for managing and administering 3D data efficiently (Zlatanova & Prosperi 2005). Object-relational databases such as PostGIS or OracleSpatial have already been applied successfully for handling geographic information. A lot of work has been done in this respect already. Within this paper we only give a short overview on recent developments regarding standard-based data sources for 3D visualization services, such as the W3DS. In order to create a 3D data storage layer which can be used as a source for our W3DS implementation a thesis was prepared, testing open and commercial database capabilities regarding geometry models, export formats, availability, etc. The following products have been assessed:

iGeo3D as part of the deegree-framework allows the user to manage 3D geodata through the web. It is completely based on OGC standards. The 3D database scheme "CityFeatureStore" of the data storage module can be used for various database systems (Oracle, PostGreSQL/ PostGis). The degree-WFS offers write- and read access to the data and iGeoSecurity provides access protection. The CityGML format or multiple image formats can be used for exchanging the data.

The *City Model Administration open source toolkit* (*CAT3D*) was developed within the EU-project VEPS (Virtual Environmental Planning System) by the HfT Stuttgart. It can connect to different data sources and produce several output formats (VRML, KML, Shapefile). The architecture is modular so that additional data sources and formats can be supported by implementing the according modules.

A 3D extension by CPA Geoinformation for the commercial *SupportGIS* offers ISO/OGC conform 3D data storage and supports databases such as Oracle, PostgreSQL, MySQL and Informix. Together with SupportGIS-3DViewer and SupportGIS-Web3D a 3D platform is provided with a CityGML central database structure.

Within the VisualMap project (FhG IGD/EML) a database called *ArchiBase* was developed which allows the administration of various different 3D geodata. The modelling tools can come from applications such as 3D Studio Max as well as from GIS. The scheme has been realized for Oracle. The data can be managed via a graphical user interface and exchange formats can be in VRML 2.0 and XML. Principles which evolved from this work can be found within the CityServer3D.

The *CityServer3D* (Haist & Coors 2005) s a multi-layered application consisting of a spatial database, a server and a client application. The database manages 3D geometries in multiple levels of detail along with the corresponding metadata. The server as the core component provides different interfaces for importing and exporting various geodata formats. The data is structured by a meta-model and stored in a database.

Currently we use the well known PostGIS extension to PostgreSQL for the 2D data and 3D points of the DEM, as well as VRML code snippets, but after some evaluation of the projects mentioned above we will extend the data management of our server to also support native 3D data types within the database as needed. Further the exchange of 3d city models through CityGML delivered by a WFS is another subject and already covered in work by the above mentioned projects, such as iGeo3D etc.

3 Towards a 3D Catalogue Service for 3D Metadata

Within an SDI it is important to record information about available data sets via metadata in order to enable find relevant data. Three metadata standards seemed most relevant for spatial data: the ISO 19115 along with its predecessors Dublin Core and CEN-TC287. The suitability of the current metadata standards for 3D spatial data was evaluated by Nonn et al. (2007). Additionally the authors investigated which enhancements or supplements might be needed by most important metadata specification for spatial data, ISO 19115, so that it can be used to describe 3D landscape and city models. An assessment was carried out seeking the highest possible sufficiency for 3D spatial data, city- and landscape models. For that the present OGC CityGML discussion paper (Gröger et al. 2006) was particularly assessed - especially regarding the question how to allow a semantic description of the structures within 3D city models.

As of today there is still no online object catalogue available for CityGML from which attribute values could be derived. If an online object catalogue containing code lists from CityGML was available online, it would not be necessary to put this kind of information directly into the ISO 19115 standard. Instead, a reference to the internet catalogue could be given. The feature type attributes contain an object type list, also linking the user to the specific part of the online catalogue.

For us this work paves the way for future discussions on the needs of 3D-SDI especially for 3D city models. Although current SDI developments focus on 2D spatial data, we think that in the long run a similar development is necessary for 3D data. Already a range of basic attributes in ISO 19115 also apply to 3D data. Even so we also found a need to add further specifications to the metadata catalogues. We have made first suggestions for possible ways to add these missing elements to the ISO 19115. We are aware that these suggestions are a first attempt and need further discussions. For first results see Nonn et al. (2007).

4 Scene-based Visualization with the Web3D Service

Regarding the portrayal of 3D information, a Web3D Service (W3DS) was proposed to the OGC as discussion draft (OGC 2005). The W3DS delivers 3D scenes of 3D city or landscape models over the web as VRML, X3D, GeoVRML or similar formats. The parameters are similar to those of the WPVS (Web Perspective View Service) which adds to the well known WMS interface parameters for camera position, view target etc. We have implemented a server that supports all of these parameters, but also provides some noteworthy techniques applied to a W3DS service for the first time in a standard-conformant way. For example in order to provide techniques that are already state of the art in computer graphics such as dynamic concepts like continuous LODs for triangle meshes or streaming of geometry parts, we have developed a sort of "pseudo-streaming" using an intelligent client-application and pre-processed DEM-tiles with different resolutions and sizes, that allows faster delivery of scenes when compared with typical implementations of the W3DS, that deliver only complete scenes in file documents, that cover the whole requested scene. A similar scenario has been introduced at Web3D 2002 (Schilling and Zipf 2002). Back then, there were no 3D OGC standards we could lean on for our scenario. This has changed and we now incorporated these standards into the project. The work presented here is embedded in a larger project that involves a several OGC Web Services (OWS), as well as several clients and the integration of various data sources.

As shown in figure 1, many requests from our Map3D-Client trigger a service chain involving separate OWS that are necessary to process the request. 2D maps are delivered by a Web Map Service (WMS) for proving overview maps of the region. The 3D information is provided by our Web 3D Service (W3DS) implementation. The Web Feature Service (WFS) standard is or will be the basis for both of these services. The WFS is already integrated for the 2D map data used by the WMS and will also be used to provide the data which is necessary for creating 3D scenes.

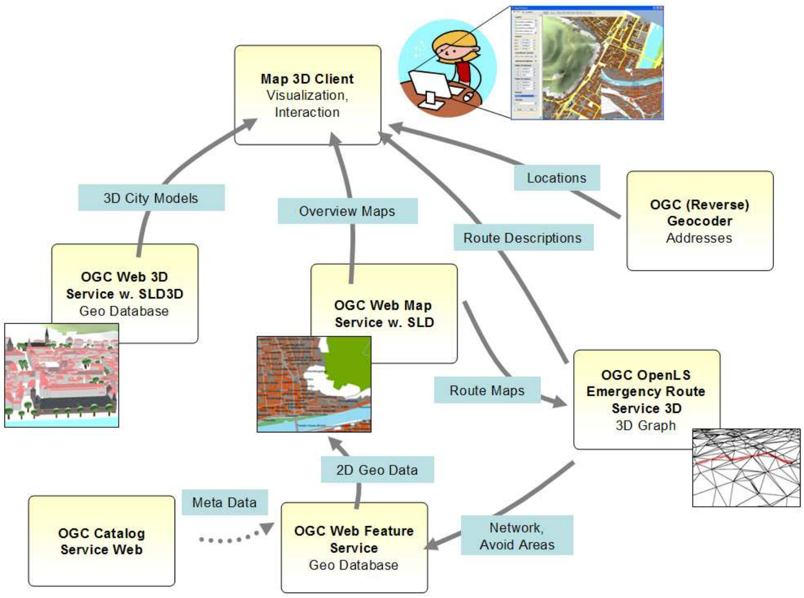

Fig. 1: Components and service chaining in our 3D-SDI (WPS to be added soon for pre-processing of terrain data).

Further we have implemented the OpenLS Route Service Specification (as well as the OpenLS Utility Service (Geocoding and Reverse Geocoding)). The route calculation itself is done on a 2D network graph, however, the resulting route geometry is then replaced by 3D linestrings which are taken from the 3D network. This 3D network has been pre-calculated by mapping the 2D linestrings onto the Digital Elevation Model (DEM) so that the route segments will exactly follow the terrain including tunnels and bridges. This so called Route Service 3D (RS3D) uses exactly the same interface as the already standardized OpenLS Route Service, without needing to extend anything. Due to the more accurate representation of the route geometries because of the 3D extension, we get a lot more route segments. Practical tests showed that we needed to reduce the geometries further using horizontal and vertical generalization so that we can produce smooth visualizations and animations. Also a 2D overview map is being produced by our implementation of the OpenLS Presentation Service.

The OGC Catalog Service (CS-W) shown in figure 1 is based on the deegree framework and delivers metadata on the actual spatial data. Before we will add metadata to this service we conducted an investigation exploring if the relevant metadata standards such as Dublin Core or ISO 19115 are appropriate for describing 3D spatial data such as 3D city models (Nonn et al. 2007). The purpose of the CS-W then is to provide information through search functions, where to find GI services und spatial data within the spatial data infrastructure or in the Web. We have used these for 2D data in former projects, such as OK-GIS or geoXchange (Tschirner et al. 2005).

5 Streaming and different LODs of DEM using the W3DS

As mentioned earlier we have developed a "smart" Java3D client which uses pre-processed DEM-tiles served by the W3DS to satisfy state of the art computer graphics with respect to streaming. Further it uses different LODs when changing the field of view of the viewer. This was done using open standards by OGC as explained :

A high-precision (5 meter) DEM, covering an area of nearly 150 square kilometres, was divided into several groups of smaller, rectangular DEM pieces with different accuracies and point-densities. Each DEM-tile group represents one Level-of-Detail (LOD). This means that those tiles covering wide areas describe the surface more approximately than smaller tiles with a high point density. Each DEM tile is replaced by four smaller tiles in the next higher LOD. This allows the client to retrieve DEM-tiles in different LODs using the W3DS. A dynamic DEM can be processed by requesting the needed tiles in consideration of the viewer's position, the line of sight and the distance along the line of sight. All changes in the viewer's field of view or position causes a new series of W3DS-GetScene requests delivering new DEM-tiles. These tiles are then added to the scenegraph. Memory is saved by only displaying the tiles in the view and by removing all tiles outside of the view on the fly. An example of the results is available as a video screen capture showing the effect on the DEM when navigating the scene in real time. The videos are available from http://www.gdi-3d.de.

6 Standard- based Configuration of 3D Visualization through extensions of the Styled Layer Descriptor

In conventional GIS the raw data is typically separated from the visualisation properties. This provides the possibility of displaying the same data in multiple ways depending on the project use case or user preferences. So far this separation is not yet established in 3D GIS data since usually the 3D model is considered as a kind of visualization itself - including all appearance properties in itself. This is the case for all common graphic formats like DXF, 3DS, VRML, and other proprietary CAD formats. In the GIS world we strive to describe only the geometry and the object classes in the raw data sets and to store attribute data and display properties in different files, as it is the case with the most popular products.

As already been successfully implemented for 2D web maps by providing Styled Layer Descriptor (SLD) documents which define rules and symbols controlling the map appearance, the same should also be applied to 3D maps, respective city models. By using SLD it is also possible to integrate different data sources into a single rendering service like a WMS and to style all data consistently.

We propose an extension to the SLD specification in order to support 3D geometries and appearance properties. As of now, the approach in this direction is unique. However, there are considerations on extending CityGML by further visualization elements. If such an extension would also cover pure styling information this would undermine the desired separation of raw data and visualization rules. Therefore we need to be aware of already existing OGC specifications and incorporate them into new standards or simply extend the existing ones. In this case the part for styling polygons, lines, and points in SLD is partly useful also in 3D. Therefore an SLD extension seems to be a more promising approach. In the next sections we make some first suggestions for a SLD3D, which incorporates the standard SLD elements and some new elements which are only valid in 3D space. The SLD3D has been implemented and tested within the 3D-SDI Heidelberg project (Neubauer 2007, Neubauer & Zipf 2007). The SLD files are currently used for configuring the W3DS server, however, in the future it can be specified by the client as well in order to provide more flexible ways of interaction.

Relevant aspects of this extension can be categorized as follows:

- Rotation of elements for all three axes
- Displacements and positions are extended by Z
- SolidSymbolizer for object volume description
- SurfaceSymbolizer for defining surfaces with triangular meshes (tin)
- Integration of external 3D objects into the scene
- Defining material properties
- Billboards for 2D graphics
- 3D legends
- Lines displayed cylindrically (e.g. for routes)

Current WMSs can provide a choice of style options for the user; the W3DS however can only provide the style names and not what the portrayal will look like in the scene in more detail. The biggest drawback however is that the user has no way of defining his own styling rules. The ability for a human or machine client to define these rules requires a styling language that the client and server can both understand. This work focuses on the definition of such a language, called the 3D Symbology Encoding (3D SE). This language can be used to portray the output of Web 3D Services

3D-Symbology-Encoding includes the FeatureTypeStyle and CoverageStyle root elements which are taken from the standard Symbology Encoding. These elements contain all information for the styling, for example filters and different kinds of symbolizers. As the specification states, Symbolizers are embedded inside of Rules, which group conditions for styling features. A Symbolizer describes how a feature will appear on a map, respectively a 3D scene. The symbolizer has also graphical properties such as color and opacity.

The 3D-SE could be used flexibly by a number of services or applications that style georeferenced information in 3D. It can be seen as a declarative way to define the styling of 3D-geodata independent of service interface specifications, such as W3DS.

6.1 PolygonSymbolizer

The PolygonSymbolizer describes the standard 2D style of a polygon including *Fill* for the interious and *Stroke* for the outline as defined in SLD. Additionally the 3D-SLD extension describes 3D features like BillboardPlacement.

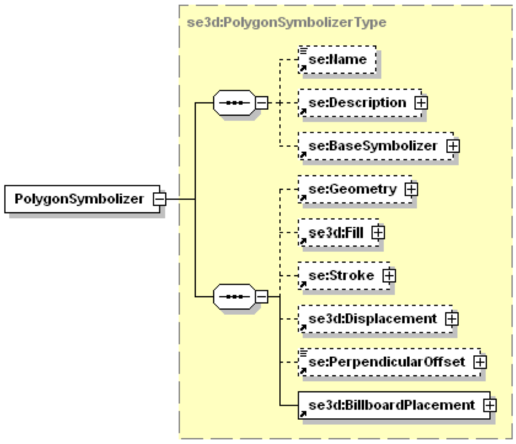

Fig. 2 - XML schema for the SLD-3D PolygonSymbolizer

6.2 LineSymbolizer

A 2D line can be represented in 3D as a pipe feature, with a certain radius and colour. The standard attributes from SLD-specification also can be set (StrokeWidth, StrokeType, etc…)

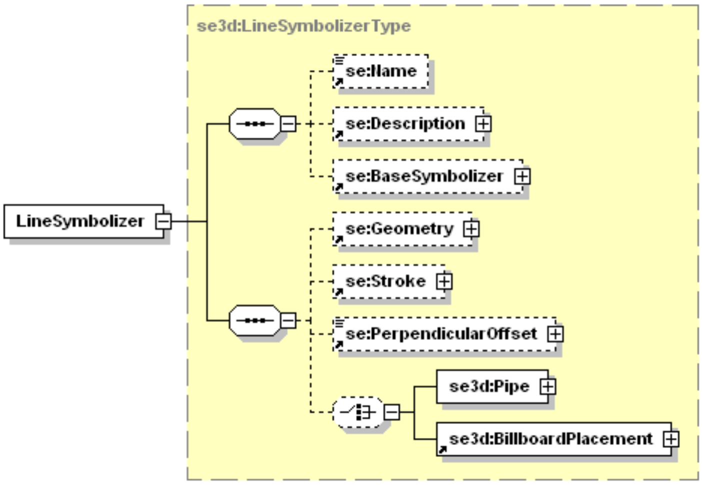

Fig. 3 – XML schema of the SLD-3D LineSymbolizer

6.3 BillboardPlacement

With the BillboardPlacement element 2D objects (text, images etc.) can be placed so that they face always towards the viewer. This is useful for icons, pixel graphics, signs, and other abstract graphics. BillboardPlacement contains 3 sub elements: AnchorPoint, Displacement, and Rotation. The syntax is:

```
<xsd:element name="BillboardPlacement" type="se3d:BillboardPlacementType">
  <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>
```

Fig. 4 - XSD schema of the SLD-3D BillboardPlacement

AnchorPoint

The 3D Symbology Encoding Anchor Point element is extended by the AnchorPointZ. The coordinates are given as floating-point numbers like AnchorPointX and AnchorPointY. These elements each have values ranging from 0.0 and 1.0. The default point is X=0.5, Y=0.5, Z=0.5 which is at the middle height and middle length of the graphic/label text. Its syntax is:

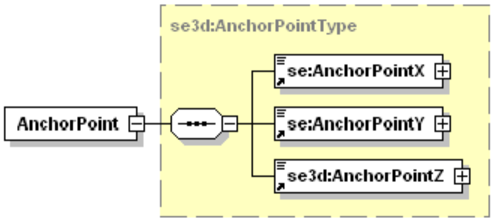

Fig. 5 - XSD schema of the SLD-3D AnchorPoint

Displacement

The Displacement is extended by an Z like the AnchorPoint. The default displacement is X=0, Y=0, Z=0. The schema is visualized in figure 7:

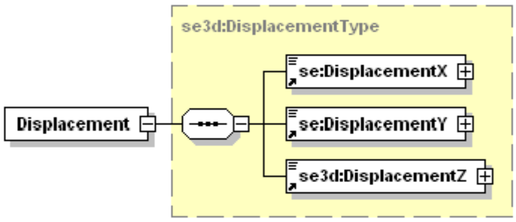

Fig. 6 - XSD schema of the SLD-3D Displacement

If Displacement is used in conjunction with Size and/or Rotation the graphic symbol can be scaled and/or rotated before it is displaced.

6.4 Material

Due to the more complicated lighting simulation in 3D, it is necessary to replace the simple colour fill element by a Material element describing the physical properties of a surface by simple means. The implementation follows the fixed function pipeline of OpenGL.

```
<xsd:element name="Material" type="se3d:MaterialType">
  <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="AmbientIntensity" type="xsd:double" minOccurs="0"/>
    <xsd:element name="Shininess" type="xsd:double" minOccurs="0"/>
    <xsd:element name="Transparency" type="xsd:double" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

Fig. 7 - XSD schema of the SLD-3D material-description

The annex shows a simplified basic SLD-3D document containing one NamedLayer and one UserStyle. Several of these can be defined inside the document. The examples of extensions given so far give only a first impression of the very large list of extensions to the original SLD schema (Neubauer& Zipf 2007). Further information and the full schema will be made public in late 2007. Currently these new 3D styles are implemented within our W3DS server.

7 Scene Integration and Server Architecture

The concept of the internal architecture of our W3DS implementation is shown in Fig 9. The service is intended to produce ready to use display elements. This means that all integration tasks can be done in advance because the display elements can be processed as far as possible before hand. For this reason we separate the functionality into a visualization server, delivering 3D scene graphs in a Web3D format, and a modelling or authoring engine, processing all the raw data into a completely integrated 3D data set of the area. This also prepares tiles that can be streamed to the client quickly.

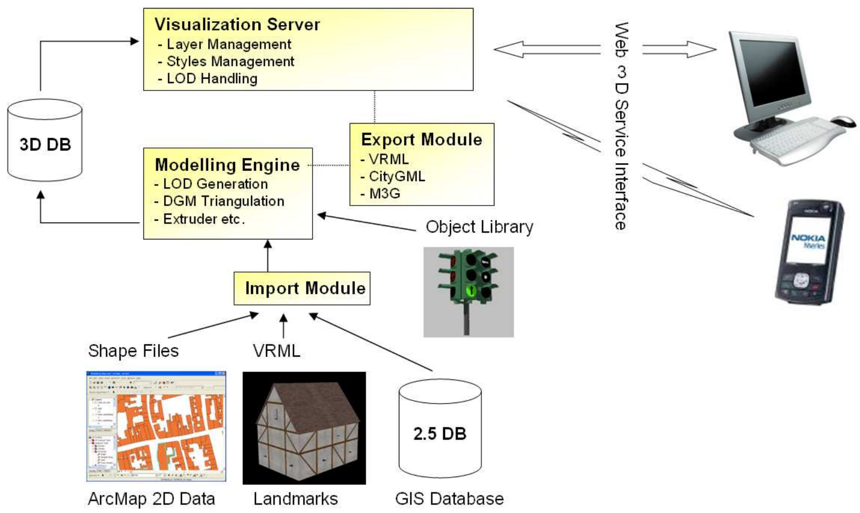

Fig. 8 - The Web3D Service is implemented as a visualisation Server

At the moment we are using Java3D objects for the internal data management in the visualization server. The export module encodes the objects into VRML syntax which is returned to the client as response to the GetScene request. In a first version we were storing all

Java3D objects in the server's java heap space. In practice however, we realized that this is insufficient for larger data sets so that we are now switching to a database implementation which holds all the Java3D objects. In the pre-processing step we integrate the different data sources like 2D GIS data, terrain data, point objects and VRML landmarks. Fig. 9 describes the integration process of the spatial data, from its original raw state to its on screen visualization. This data is imported from various sources. In a next step it is converted into 3D objects that compromise a 3D scene and is stored in a database for faster access. This 3D-geodatabase is the data source for the visualization server, which delivers 3D scenes by a W3DS conform web request and offers the possibility to export into various formats.

Fig. 9 - Thematic 2D layers integrated into the terrain model

We found the data integration of thematic 2D area information (forest, streets, water, etc) particularly interesting because these are not applied to the terrain as textures but are cut into the TIN triangulation. Fig. 10 shows how the original 2D layers are transformed into 3D geometries. For each layer we create one indexed face set that covers the terrain exactly. The original surface underneath is cut away. The downside is that larger geometries are produced. The advantage however, is that aliasing effects do not occur and that we do not need to transmit additional textures.

8 Standard-based 3D Route Planning within an 3D SDI

The OpenLS (OpenGIS Location Services) framework consists of five core services (OpenLS 2002). We have implemented already three of these services using Java (Neis et al 2006, Neis et al 2007). The Route Service (RS) allows the setting of various criteria such as start- and destination, time, distance, travel-type, one way street information as well as the possibility to add areas or streets which should be avoided (*AvoidAreas*). Following this standard we implemented a Route Service 3D (RS3D). The main difference is that the RS3D provides GML code that contains also Z values for all route geometries and instructions so that the result can be integrated into 3D landscapes without further calculations. Similar ideas have been proposed by Zlatanova and Verbree (2006).

The implementation is based on a complete 3D street network. For the actual route calculations we use the standard Dijkstra-Algorithm. The original 2D network graph and the 3D graph are topologically equal. However, we need to transform the network segments in order to reflect the terrain surface. We do this by mapping the 2D linestrings onto the DEM and adding new line vertices wherever an Edge of the terrain triangulation is crossed, so that it is exactly parallel to the surface. Depending on the terrain accuracy the amount of data for the network geometry increases, but also the correctness of the graph. Network segments representing bridges, tunnels, or underpasses currently still need to be adjusted manually, which is relevant for the visualization and route animation. The RS3D uses the existing OpenLS Location Utility Service for geocoding and the OpenLS Presentation Service for generating overview maps. This is a good example for service chaining within an

SDI. Our tests showed that this approach is faster than to make calculations of the geometries on the fly which takes a considerable amount of time.

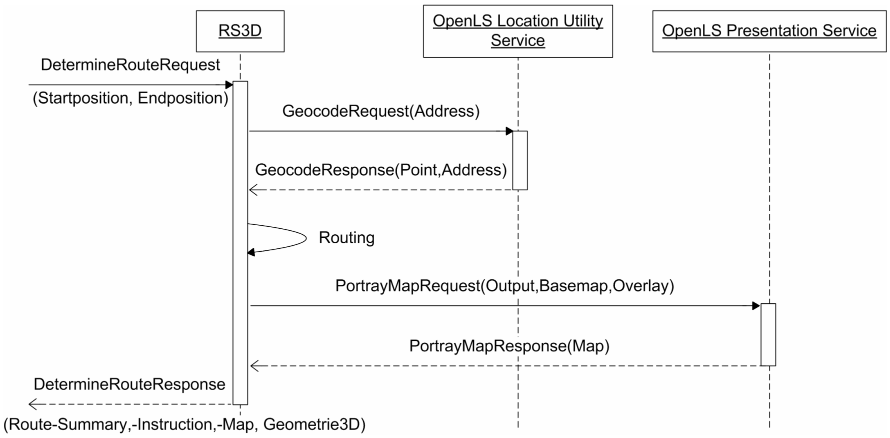

Fig. 10 - RS3D UML sequence diagram

9 Route Presentation within the W3DS Viewer

Route presentations in 3D can be done using either dynamically updated texture maps containing a line string with certain width, colour, and pattern, or using 3D geometries like tubes or others along the route. We chose the latter alternative, since texturing is already used for the terrain styling. The 3D line string is extruded as a pipe with a certain distance to the ground. Switching to a new route quickly is no problem since the Web3D Service delivering the city model and the route service are independent and the scenegraph part containing the route can be replaced. The waypoints are displayed as 2D labels on top of the screen. By computing key frames for every node and connecting them by a Spline interpolator (KBRot-PosScaleSplinePathInterpolator) we have created a route animation that moves the viewpoint along the route in some distance. Unfortunately the initial route linestring, which fits to the terrain surface perfectly, is not very useful for generating animations. Small features or errors in the TIN, such as little bumps that are not visually dominant, are preserved and lead to jerky movements in the animations. This occurs because the animation moves along a segmented line and the camera view changes at every small segment. Also, the network graph that is used for the route computation contains sharp corners at intersections. While this is sufficient for 2D maps there are difficulties when applied to 3D maps. Therefore, we implemented an additional simplification that filters small features in order to receive a smooth animation. The results can be seen from the video captures on www.heidelberg-3d.de

Fig. 11 - 3D routing in the W3DS-Viewer

10 Future Issues in 3D SDIs

10.1 Orchestrating 3D Web Services

A major goal for developing complex applications based on OGC Webservices (OWS) that represent a spatial data infrastructure is to provide flexibility and reusability. Through the aggregation of standardized base services, complex functions and workflows of a certain granularity can be achieved. These new aggregated functionalities can then also be used as web services on their own. In order to to avoid the need to program the aggregation of several independent OWS by hand a higher level solution has been proposed. This alternative is called "Web Service Orchestration" (WSO) through standardized higher level languages, such as the Business Process Execution Language (BPEL). The promise of WSO is to provide an easy and flexible way to link services in given patterns, especially through configuration. This can be realized through so called orchestration scripts. Their configuration can then be carried out with graphical tools instead of hard-coded programming. The BPEL scripts will be executed in corresponding WSO engines.

First experiences using WSO or BPEL in the context of OGC service have been reported recently (Weiser and Zipf 2007). Discussions within the OGC about service oriented architectures with OWS already started in 2004 (OGC 04-060r1). Although there were some earlier considerations to adjust the architecture towards compatibility to common web services (OGC 03-014) the OWS2 paper (OGC OWS2) offered the first helpful suggestions in this direction. Similar methods and ideas recently have been discussed by Kiehle et al. (2006) and Lemmens et al. (2006). Some results from the OWS4 initiative can be found in the recent internal OGC discussion paper "OWS4 Workflow IPR" (OGC 06-187). The proof-of-concept evaluation presented in Weiser and Zipf (2007) shows that it is possible to create an added-value by combining and aggregating OGC Web Services. However it only makes sense to use orchestration where a continuous service chain without human intervention is given. This is why it is necessary to find stable standard chains (small parts of a larger workflow that can act as modular building blocks) as well as to do research on assembling BPEL scripts in an even more dynamic way, even for non-technical users. At the moment developers still have to face a lot of small technical problems when trying to realize WSO for OWS (Weiser et al 2007). Especially user interface concepts are needed in order to ease the highly dynamic orchestration of OWS on the fly. Then it would also allow in principle to represent such service chains as presented figure 1 through the use of WSO-technologies like BPEL. A major obstacle at the moment is the missing SOAP interface for the relevant OWS. But this will change in the foreseeable future, as SOAP interfaces now need to be added to every new version of an OGC specification.

10.2 Future 3D Navigation on Mobile Devices

When talking about 3D route planning it makes sense to use this in a 3D environment. We presented preliminary work on the Mainz Mobile 3D system, a PDA based navigation system, that was also capable of sending W3DS requests to the W3DS server and displaying the VRML scene returned (Fischer et al. 2006). Recently a new project started together with several partner including Hft Stuttgart and companies like Navigon, Teleatlas, GTA Geoinformatics, Heidelberg Mobil and others. The project is called "Mobile Navigation with 3D city models" (MoNa3D, http://www.mona3d.de) and extends mobile navigation systems into 3D (Coors & Zipf 2007). Improved 3D navigation through semantic route descriptions, using landmarks will be investigated. 3D visualizations are important for both pedestrian and vehicle navigation systems. 3D city models allow the integration of landmarks into the route description. Regarding 3D one has also to consider indoor navigation – also here the choice and visualization of the right landmarks is important (Mohan & Zipf 2005). Most navigation systems today only offer direction and distance information. This is not sufficient in order to provide the user with ideal orientation information. Studies from cognitive psychology have shown that directions using landmarks have been rated higher. We aim at gaining more knowledge towards optimizing 3D navigation information to provide the user with relevant orientation details. This could lead to safer navigation through reduced stress. In order to achieve sustainable outcomes, the needed 3D city models for navigation support have to be available within a functioning 3D geodata infrastructure (3D-GDI), such as http://www.3d-gdi.de.

11 Conclusion and Outlook

In this paper we have discussed the first outcomes of a research project which concentrates on the implementation of the next generation of spatial infrastructures based on open standards that are currently in the discussion phase within the OGC. We showed how to use these standards in order to develop a 3D SDI. A demo application using an OpenLS-based 3D route service has been introduced.

So far mostly data management and visualization issues for 3D spatial data have been discussed. The next step for 3D SDI not yet mentioned so far in this paper will be the standard conform geoprocessing of 3D data. Within the OGC a draft version of a new specification regarding processing of arbitrary spatial data is under development. This so called Web Processing Service (WPS) (see also Kiehle et al. 2006) still has some way to go, but we have first experiences with this draft have been gained through implementations of specific processing algorithms (e.g. spatial join and aggregation) (Stollberg 2006, &Stollberg and Zipf 2007) within the project OK-GIS (open disaster management with free GIS, www.ok-gis.de). From these we are confident, that a range of preprocessing steps needed within our scenarios in the 3D-SDI Heidelberg can be realized using this new standard.

Acknowledgements

This work has been supported by the Klaus-Tschira-Foundation (KTS) GmbH Heidelberg within the project www.heidelberg-3d.de. All spatial data by courtesy of the Bureau of Surveying Heidelberg and the European Media Laboratory (EML) Heidelberg. We thank all coworkers, in particular A. Weiser, B. Stollberg, U. Nonn and C. Mark for their efforts.

12 References

Abdul-Rahman, A.; Zlatanova, S.; Coors, V. (Eds.): Innovations in 3D Geo Information Systems Springer: Lecture Notes in Geoinformation and Cartography , 2006, 760 p.

Chen L, Wassermann B, Emmerich W, Foster H (2006): Web Service Orchestration with BPEL. Dept. of Computer Science, University College London

Coors, V. and Zipf, A, (Eds.)(2004): 3D-Geoinformationssysteme. Grundlagen und Anwendungen. Wichmann - Hüthig Verlag. Heidelberg. 525 pages.

Coors, V. and Zipf, A. (2007 accepted): MoNa 3D --- Mobile Navigation using 3D City Models. LBS and Telecartography 2007. Hongkong.

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

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

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

Gröger, G., Kolbe, T.H., Czerwinski, A. (2006): OpenGIS City Geography Markup Language (CityGML), Implementation Specification Version 0.3.0, Discussion Paper, OGC Doc. No. 06-057

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

iGeo3D: www.lat-lon.de/latlon/portal/media-type/html/language/de/user/anon/page/default.psml/js_pane/sub_produkte_deegree-igeo3d

ISO/TC 211 Geographic information/Geomatics. ISO reference number: 19115 (2002).

Kiehle C; Greve K & Heier C (2006): Standardized Geoprocessing – Taking Spatial Data Infrastructures one step further. Proceedings of the 9th AGILE International Conference on Geographic Information Science. Visegrád, Hungary.

Lemmens R, Granell C, Wytzisk A, de By R, Gould M, van Oosterom P (2006): Semantic and syntactic service descriptions at work in geo-service chaining. Proc. of the 9th AGILE Int. Conference on Geographic Information Science. Visegrád, Hungary

Mohan, S. K., Zipf, A. (2007): Improving the support of indoor evacuation through landmarks on mobile displays. 3rd International Symposium on Geoinformation for Disaster Management. Toronto, Canada.

Müller, M. (ed): OpenGIS Symbology Encoding Implementation Specification version 1.1.0 doc.nr. 05-077r4 http://www.opengeospatial.org/standards/symbol

Neis, P. (2006): Routenplaner für einen Emergency Route Service auf Basis der OpenLS Spezifikation. Diploma Thesis. University of Applied Sciences FH Mainz.

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

Neubauer, S., Zipf, A. (2007 accepted): Suggestions for Extending the OGC Styled Layer Descriptor (SLD) Specification into 3D – Towards Visualization Rules for 3D City Models, Urban Data Management Symposium. UDMS 2007. Stuttgart. Germany.

Nonn, U., Dietze, L. 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 Geograpic Information Laboratories for Europe. Aalborg, Denmark.

Nougeras-Iso, J., P.R. Muro-Medrano, F. J. Zarazaga-Soria (2005): Geographic Information Metadata for Spatial Data Infrastructures - Resources, Interoperability and Information Retrieval. Springer. 2005, XXII, 264 p

OGC 2005. Web 3D Service. OGC Discussion Paper, Ref. No. OGC 05-019.

OGC: OWS2 Common Architecture. Hrsg. OGC. RefNum. OGC 04-060r1; Vers. 1.0.0; Status: OGC Discussion Paper.

Open Geospatial Consortium Inc. OWS 1.2 SOAP Experiment Report. Hrsg. OGC. RefNum. OGC 03-014; Vers. 0.8; Status: OGC Discussion Paper.

Open Geospatial Consortium Inc. OWS-4 Workflow IPR. Hrsg. OGC. RefNum OGC 06-187; Vers. 1.0.0; 2007-09-14 Status: internal OGC Discussion Paper.

OPENLS: OGC Open Location Services Version 1.1 http://www.opengeospatial.org/functional/?page=ols.

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

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

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

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

Schilling, A., Zipf, A. (2002): Generation of VRML City Models for Focus Based Tour Animations - Integration, Modeling and Presentation of Heterogeneous Geo-Data Sources. In Web3D Conference, 9-12.03.2003, Saint Malo, France.

Stollberg, B. (2006): Geoprocessing in Spatial Data Infrastructures. Design and Implementation of a Service for Aggregating Spatial Data with the Web Processing Service (WPS). Diploma Thesis. University of Applied Sciences FH Mainz.

Stollberg, B. & Zipf, A. (2007 accepted): OGC Web Processing Service Interface for Web Service Orchestration - Aggregating geo-processing services in a bomb thread scenario. W2GIS 2007: Web&Wireless GIS Conference 2007. Cardiff, UK.

SLD: SLD-Specification (OGC): http://www.opengeospatial.org/docs/02-070.pdf

SupportGIS-3d: www.supportgis.de/Dip2/SupportGIS/3D/SupportGIS-3D.pdf

Weiser, A., Zipf, A. (2007): Web Service Orchestration of OGC Web Services (OWS) for Disaster Management. GI4DM 2007. 3rd International Symposium on Geoinformation for Disaster Management. Toronto, Canada.

Zipf, A., Tschirner, S. (2005): Finding GI-datasets that otherwise would have been lost - GeoXchange - a OGC standards-based SDI for sharing free geodata. 2nd International Workshop on Geographic Information Retrieval (GIR 2005) at the Fourteenth ACM Conference on Information and Knowledge Management (CIKM). November, 2005. Bremen, Germany. Springer Lecutre Notes in Computer Science.

Zlatanova, S. and E. Verbree (2005): The third dimension in LBS: the steps to go. In: Geowissenschaftliche Mitteilungen, Heft Nr. 74, 2005, pp. 185-190.

Zlatanova, S. and D. Prosperi (Eds) (2005): Large-scale 3D Data Integration CRC Press. 256 pages.

13 Annex Example Document of SLD-3D Extension

```xml
<?xml version="1.0" encoding="UTF-8"?>
<sld3d:StyledLayerDescriptor xmlns="http://www.opengis.net/sld3d"
  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:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/sld3d
  ..\3D-SLD\3d-sld\0.1.0\StyledLayerDescriptor.xsd"
  version="0.1.0">
  <se:Name>3D-SLD_1</se:Name>
  <se:Description>
    <se:Title>3D StyledLayerDescriptor 1</se:Title>
  </se:Description>
  <NamedLayer>
    <se:Name>Strassen</se:Name>
    <se:Description>
      <se:Title>Streets and Paths</se:Title>
      <se:Abstract>Polygon Layer</se:Abstract>
    </se:Description>
    <UserStyle>
      <se3d:FeatureTypeStyle>
        <se3d:Rule>
          <se3d:PolygonSymbolizer>
            <se3d:Fill>
              <se3d:Material>
                <se3d:DiffuseColor>
                  <se:SvgParameter name="fill">#050505</se:SvgParameter>
                </se3d:DiffuseColor>
                <se3d:SpecularColor>
                  <se:SvgParameter name="fill">#0f0f0f</se:SvgParameter>
                </se3d:SpecularColor>
                <se3d:EmissiveColor>
                  <se:SvgParameter name="fill">#000000</se:SvgParameter>
                </se3d:EmissiveColor>
                <se3d:AmbientIntensity>0.3</se3d:AmbientIntensity>
                <se3d:Shininess>0.0</se3d:Shininess>
                <se3d:Transparency>0.0</se3d:Transparency>
              </se3d:Material>
            </se3d:Fill>
          </se3d:PolygonSymbolizer>
        </se3d:Rule>
      </se3d:FeatureTypeStyle>
    </UserStyle>
  </NamedLayer>
</sld3d:StyledLayerDescriptor>
```