Accessing a 3D geo-DBMS using Web technology

M.E. de Vries and J.E. Stoter
Section GIS technology, Delft University of Technology
Delft, The Netherlands
Tel.: +31 15 2784268, Fax: +31 15 2782745
{m.e.devries|j.e.stoter}@geo.tudelft.nl

1 Introduction

Spatial data used in GISs is often part of a complete workflow. Therefore in many organizations there is a growing need for a central DBMS: a system in which spatial data and non-spatial data are maintained in one integrated environment. Since spatial data types have been implemented in (object-)relational DBMSs, many DBMSs are capable of maintaining spatial data in 2D.
Apart from the need for an integrated DBMS, there is also a trend of modeling the world in three dimensions. However, GIS applications, which use true 3D functionalities other than 3D visualizations are still rare. GIS and DBMS vendors seem to hesitate with full support for 3D. It seems that users are waiting for vendors to make the step to 3D, while vendors are waiting for an extensive request for 3D tools from their customers. In this respect, research into new 3D solutions can play a critical role (Stoter and Zlatanova, 2003a).
In (Stoter and Zlatanova, 2003b), it is described how 3D geo-information can be maintained in DBMSs within current techniques. The research presented in this paper focuses on a related issue: how to query and to visualize 3D geo-data that is organized in a DBMS. To explore current possibilities, we built two prototypes using different Web technology. The used tools and techniques are from outside the GIS domain. In this research the potentials are studied of using these tools and techniques in GIS and geo-info-science. The first prototype was built on a Microsoft platform, with ASP and MS Access. It uses VRML for visualization. The second prototype is based on XSQL (a Java servlet solution), Oracle and the new 3D format: X3D.
The aim of the prototypes is to show the possibilities and constraints of accessing 3D geo-information organized in a geo-DBMS, in order to contribute to 3D GIS developments. Related research can be found in (Nebiker, 2003), (Belfore, 2002) and (Luttermann, 1999).

We start with a short description on how 3D geo-objects can be modeled in a DBMS. These data models (a topological model and a geometrical model) were used in our prototypes. Section 3 contains a short description on VRML and X3D, the 3D formats used in the prototypes. In section 4 the prototypes are described, together with their functionalities. The paper ends with a conclusion on the prototypes and points out some topics for future research.

2 3D geo-objects in DBMS

GISs have evolved to spatial DBMSs in which spatial as well as non-spatial data is maintained in one integrated environment. To achieve this, mainstream DBMSs (Oracle (Oracle, 2001), IBM DB2 (IBM, 2000), Informix (Informix, 2000), Ingres (Ingres, 1994)) have implemented spatial data types and spatial operators more or less similar to the OpenGIS Consortium Simple Features Specification for SQL (OGC, 1999). The implementation consists of an SQL extension that supports storage, retrieval, query and update of simple spatial features (points, lines and polygons). OGC specifications (OGC, 2003) are until now 2D, as are the implementations of spatial data types in mainstream DBMSs. The only 3D functionalities in 3D that are available in DBMS are length and perimeter in 3D (PostGIS, 2003;

MapInfo, 2002) and spatial indexing in 3D (e.g. 3D R-tree in Oracle). Although current DBMSs do not support 3D objects, z-coordinates can be used to store 0D, 1D and 2D objects in 3D.
(Stoter and Zlatanova, 2003b) describe how 3D geo-objects can geometrically be maintained in DBMSs within current techniques. (Arens et al., 2003) implemented support for a 3D primitive (polyhedron) in a DBMS. This implementation has been based on the proposal of (Stoter and Van Oosterom, 2002). (Stoter and Zlatanova, 2003b) also describe a topological implementation of 3D geo-objects in DBMS based on the Simplified Spatial Model (Zlatanova, 2000). The prototypes presented in this paper are based on these DBMS models, which are implemented in Oracle Spatial 9i (Oracle, 2001).

3 VRML and X3D

In 1994 the Web3D Consortium launched VRML (Virtual Modeling Language), which became an international ISO standard in 1997. The basis for the development of VRML was to have a simple exchange format for 3D information. This format is based on the most used semantics of modern 3D applications: hierarchical transformations, illumination models, viewpoints, geometry, fog, animation, material characteristics and texture.
The development of VRML has stopped since the Web3D Consortium started to work on a XML version of VRML, in order to integrate with other web technologies and tools: X3D (eXtensible 3D). The specifications of X3D have only recently become available (May, 2003). In our research we use both X3D and VRML to visualize 3D geo-information. The data structure of a X3D document is very much comparable to the data structure of a VRML file. So as far as the underlying data model is concerned, X3D must be seen as a subset of VRML (Web3D, 2003). The difference lies in the notation (the syntax) used. While VRML is text, with accolades for structuring, X3D is coded in XML, with 'tags' for structuring. This is a major advantage for on-the-fly retrieval, because of the ease of use of XML in Internet applications.

4 Prototypes

The basic idea of the prototypes is to organize 3D geo-objects in a DBMS and to query them via an Internet browser. Geo-objects contain both spatial and non-spatial (administrative) information. The spatial information can be visualized after conversion into VRML or X3D and the non-spatial attribute information can be presented in (dynamic) HTML pages.
On a client request a connection is made to the DBMS and the spatial information of interest is selected from the DBMS and converted into X3D/VRM. A browser plug-in at the client side makes it possible to view the VRML or X3D output. VRML and X3D provide the possibility to start a script when a user clicks on an object. This functionality is used to retrieve the non-spatial information that is linked to a 3D geo-object. Via the VRML/X3D plug-in a request is sent to a (application) server. The server receives and interprets the incoming information and sends a HTML with the required information back to the browser.
For retrieving the spatial and the non-spatial information from the DBMS a technique is needed to communicate between a client and a database on a server. For this communication, several techniques are available such as ColdFusion, ASP.NET, ASP, JSP or PHP. The choice of the used technique is dependent on the used web server.

To show the possibilities to query 3D geo-objects via an Internet client, first a simple prototype was built, based on Microsoft technology. The aim of this first prototype was to study functionalities of accessing 3D geo-information using common Web technology. After good results of the first prototype, a second, more advanced prototype was built.

4.1 Prototype I: ASP, VRML and MS Access

In the first prototype only the non-spatial data can be dynamically retrieved from the database. For the spatial information, a (static) VRML-file is created beforehand containing all the 3D geo-objects in the data set. A Java program was written which converts the geographical information that is stored in the Simplified Spatial Model in the DBMS (Oracle) to a VRML file. For the prototype we used a dataset from a 3D cadastre project (Stoter and Ploeger, 2002). The dataset contains a building complex of the Central Station in The Hague, which is divided into 15 property units (objects). For every object attribute information (such as ownership) is stored in MS Access.

In this prototype Microsoft Internet Information Server 5.0 (IIS) is used as web server and ASP (Active Server Pages) for the communication between Internet client and database server. For the communication between the MS Access DBMS and ASP an ODBC connection is set up.
The operations in the prototype are performed at the server and not at the client side (see figure 1). For the prototype we use an interface of an HTML page consisting of two frames, one frame to display the VRML data and the second frame to show the attribute information. The user opens the VRML file in the browser. When the user clicks on an object, a URL string containing the unique key for the object is sent to the server. The ASP page connects to the DBMS, retrieves the requested attribute information and presents it in the bottom frame as dynamic HTML.

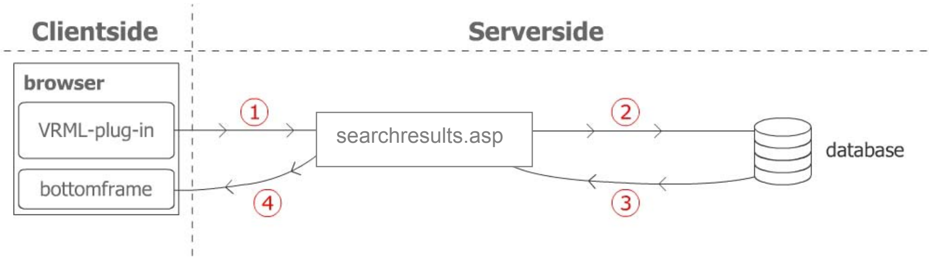


Figure 1: Model of the prototype

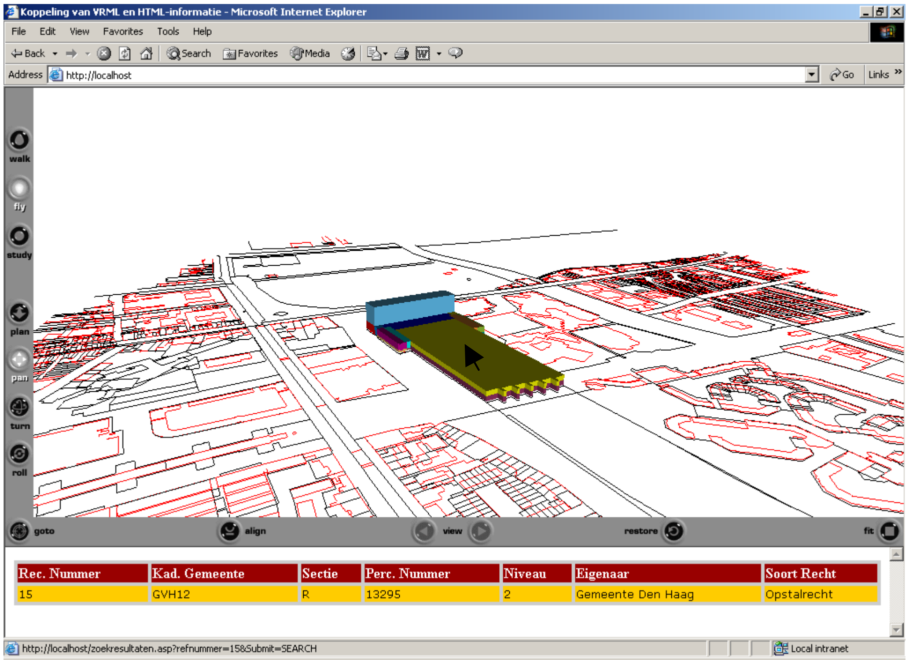
The URL string for object with id ‘5’ looks as follows:

http://localhost/ASP/searchresults.asp?id=5&submit=SEARCH

The URL’s are linked to the objects by storing the URL string with the relevant parameters with every object in the VRML file.
The ASP code contains a method Request.QueryString. This method filters the needed parameters from the URL string that is sent to the server. In this case the unique id of the object. The ASP page also contains a SELECT statement to get the requested information from the database. In this case:

```
SELECT Recnr, KadGemeente, Sectie, Perceelnr, niveau, eigenaar, soort_recht
FROM eigendom
WHERE Recnr = 'varName'
```

A screen dump of the prototype, using Cortona as VRML browser (see link in (NIST, 2003)), is shown in figure 2.



Rec. Nummer	Kad. Gemeente	Sectie	Perc. Nummer	Niveau	Eigenaar	Soort Recht
15	GVH12	R	13295	2	Gemeente Den Haag	Opstalrecht

Figure 2: Screen dump of first prototype (ASP and VRML)

This first prototype has three main disadvantages:

1. The transformation of spatial information into the 3D format is not performed on the fly.
2. The non-spatial data is stored in another DBMS than the spatial data. MS Access was selected for this prototype, to explore possible techniques.
3. MS Access does not support spatial data types, although 3D objects could be stored in a topological model in MS Access.

To overcome these disadvantages a second, more advanced prototype was built.

4.2 Prototype II: XSQL, X3D and Oracle

The second prototype is based on Oracle's XSQL Servlet and on X3D. The XSQL Servlet is part of Oracle's XML Developers Kit (or XDK), aimed at 'XML enabling' Oracle's DBMS technology (Thomas, 2003).

XSQL operates in combination with an XML Parser, an XSL processor (for the processing of XSLT stylesheets) and the XML SQL Utility (or XSU), also part of XDK. Servlets are Java classes that operate in a Web server environment. They process requests that are passed to them by the Web server, when HTTP requests are received from a client. In this respect XSQL performs the same role as ASP in the prototype we described in the previous section (server-side processing).

An XSQL 'page' is a combination of XSQL tags and SQL statements. The fragment below shows an XSQL page with a select statement that returns all rows in a table (parameters are used to make the statement generic). A where-clause for an attribute query or a spatial query, or any other standard SQL statement is possible as well.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="{@xsldoc}.xsl"?>
<mymap connection="{@con}" xmlns:xsql="urn:oracle-xsql"  >

  <xsql:query rowset-element="" max-rows="5000" >
     select t.{@geom}.sdo_gtype as gtype,
            t.{@geom}.sdo_elem_info  as info,
            t.{@geom}.sdo_ordinates as geom,
           {@idcol} as id
     from {@table} t
  </xsql:query>
```

An XSQL page has an .xsql extension. When the web server receives a request for an .xsql document, the page is passed to the XSQL servlet. The page is processed by the servlet: a connection to the database is made and the select statement is sent to the DBMS.
The result set that comes back from the database is already in XML format. The second step is then to transform the 'raw' XML stream into a X3D or VRML output stream.
Because of the XML syntax of X3D, the transformation from Oracle to X3D can easily be handled by XSLT stylesheets. Therefore in this prototype we used X3D. With the media-type set to "model/x3d+xml" the transformed output stream is recognized by the browser as X3D, so that the right viewer plug-in (we used BS Contact, see link in (NIST, 2003)) can be activated for visualization.
The way the XML output stream is transformed into X3D depends on the way the 3D geo-information is stored: when the data is topologically structured, a different XSLT stylesheet must be used then in the case of a non-topological model, when the geometry of each 3D object is stored in the object-record itself (Oosterom et al., 2002). In both cases however the same basic XSQL technology can be used.

Getting the attribute information follows the same principle as in the first prototype: with the <Anchor/> construct of VRML/X3D a URL can be called when one clicks on an object, in this case the URL of another XSQL-page. The value and attribute name of the link-id is passed to the XSQL page, together with the name of the Oracle table that has to be queried:

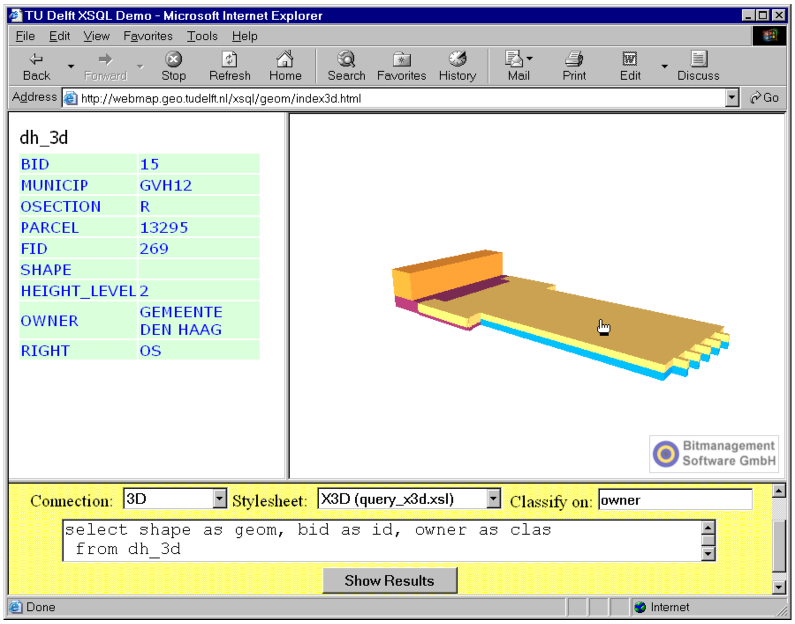
```
<Anchor parameter="target=left"
     url="fieldinfo.xsql?table=dh_3d&amp;idcol=bid&amp;id=1&amp;con=st">
  <Shape>
   <IndexedFaceSet convex="false" solid="false"
     ...
  </Shape>
</Anchor>
```

The XSQL servlet takes care of most of the steps in the retrieval process: establishing a JDBC connection to the database, sending the SQL queries to the database and - most important for our purpose - reformatting the SQL response (the result set of the query) into an XML output stream. This output stream can then be presented either as (dynamic) HTML (for attribute data), as SVG (for 2D spatial data) or - as in this prototype - as X3D, for 3D visualization.
The configuration used in this second prototype is: Apache as Web server software with Tomcat 4 as servlet container, XSQL for the server side processing, HTML and JavaScript for the user interface, and BS Contact as X3D viewer plug-in.

The second prototype (figure 3) differs from the first in a number of respects:

1. X3D is used as 3D graphic format instead of VRML
2. The transformation into the 3D format is performed on the fly, with real-time access to Oracle.

3. Spatial and non-spatial data are not in separate databases or database systems. Both 3D spatial and administrative data is stored in Oracle (in this case in the same Oracle table, but that is not necessary).


TU Delft XSQL Demo - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address
Go
Bitmanagement Software GmbH
Show Results
Done
Internet

Address: http://webmap.geo.tudelft.nl/xsql/geom/index3d.html

dh_3d

BID	15
MUNICIP	GVH12
OSECTION	R
PARCEL	13295
FID	269
SHAPE	
HEIGHT_LEVEL	2
OWNER	GEMEENTE DEN HAAG
RIGHT	OS

Connection: 3D Stylesheet: X3D (query_x3d.xsl) Classify on: owner

```
select shape as geom, bid as id, owner as clas
 from dh_3d
```

Figure 3: Screen dump of second prototype (XSQL and X3D)

The second prototype is a more platform-independent solution than the first prototype: because of the Java servlet technology it can be implemented on both Microsoft and Unix web servers. And although the XSQL Servlet is part of Oracle's XML toolkit, it can also be used to connect to other databases than Oracle (MySQL, PostgreSQL etc.), provided these databases can be accessed via JDBC connections.

5 Conclusion and future research

Geo-DBMSs have become a major topic in geo-science. Spatial data types for 2D are currently available in mainstream DBMSs and now geo-science is focusing on support for 3D, 4D and topology in DBMSs. The next step is to make the data organized in a geo-DBMS available within and between organizations and for individual users via Web technology.

This paper contributes to these developments, since it illustrates how 3D visualization techniques and techniques to query DBMSs via a web server can be combined. We showed how 3D geometry stored in an Oracle database can be converted into VRML or X3D, and how the 3D objects can be presented in a 'simple' Internet browser together with their non-spatial attribute information. Future work will focus on a number of research issues.

Although first experiences with the prototypes look promising with respect to performance, serious tests on larger data sets will be set up. Fast rendering of 3D objects is of course critical when displaying data via the Internet. One option is the use of compression techniques. It is also interesting to look into possible performance improvements when an XML SAX parser is used instead of the DOM parser that XSQL uses now. Unlike DOM parsers a SAX parser does not load the whole output stream into memory first, which is an advantage in case of large data sets. Other strategies have to do with navigation and zoom-levels. First data with low Level of Detail can be presented, and only when the user zooms in the Level of Detail is increased (see also Kofler, 1998, Pasman and Jansen 2002). This touches the

fundamental research issue of storing multi-representations in DBMSs or performing generalizations on the fly.
Also querying the topological structured data instead of the geometry is a topic of further research. Generating a geometric realization (for which three tables need to be queried) before converting it to X3D might be critical for performance.
Another issue is that of visualization constraints. X3D and VRML environments were originally meant to visualize all kinds of objects. Therefore, constraints should be implemented in the prototypes to meet specific conditions for visualizing geo-information, e.g. one should no be able to turn the data set upside-down.
Another topic is how to combine different geo-datasets in one environment, and how to combine different data sets stored in federated DBMSs in one environment. Besides technical aspects (how to access data stored in separate DBMSs) also cartographic aspects (e.g. color) should be considered. This requires that not only spatial and non-spatial information of spatial objects are maintained in DBMS but also characteristics such as physical properties of objects (texture, material, color), behavior (e.g. on-click-open) and different Levels Of Detail representations.

The prototypes described in this paper form a contribution to accessing 3D geo-objects organized in a DBMS. Our experiences show potentials for the use of common Web technology in geo-info-science and more specific in 3D GIS applications, which should be further elaborated in future research.

References

Ames, Andrea L. et al. (1997), The VRML 2.0 sourcebook, second edition, USA, John Wiley & Sons, Inc., 1997.

Arens, C., J.E. Stoter and P.J.M. van Oosterom (2003), Modelling 3D spatial objects in a GEO-DBMS using a 3D primitive, AGILE conference, April 2003, Lyon, France

Belfore, L.A. (2002), An architecture supporting live updates and dynamic content in VRML based virtual worlds. Symposium on Military, Government and Aerospace Simulation 2002 (MGA 2002), San Diego, California, April 2002, 138-143.

IBM (2000), IBM DB2 Spatial Extender User's Guide and Reference. special web release edition.

Informix (2000), Informix Spatial DataBlade Module User's Guide. http://www-3.ibm.com/software/data/informix/pubs/library/dbm.html

Ingres (1994), INGRES/Object Management Extension User's Guide, Release 6.5 (1994). CA-OpenIngres.

Kofler, M. (1998), R-trees for the visualisation of large 3D GIS Database, Ph.D. thesis, Technical University, Graz, Austria, 1998.

Luttermann, H. and M. Grauer (1999), Using interactive, temporal visualizations for WWW-based presentation and exploration of spatio-temporal data. Spatio-Temporal Database Management 1999, 100-118.

MapInfo (2003), http://www.mapinfo.com

Nebiker, S. (2003), Support for visualisation and animation in a scalable 3D GIS environment - motivation, concepts and implementation. ISPRS Commission V Working Group 6, Workshop on "Visualization and Animation of Reality-based 3D Models", February 2003, Engadin, Switzerland

NIST (2003), http://cic.nist.gov/vrml/vbdetect.html

OGC (2003), OpenGIS specifications, http://www.opengis.org/techno/specs.htm

OGC (1999), OpenGIS Simple Features Specification for SQL. Revision 1.1, OpenGIS Project Document 99-049.

Oosterom, P.J.M. van, J.E. Stoter, C.W. Quak, and S. Zlatanova(2002), The Balance Between Geometry and Topology. In: Advances in Spatial Data Handling, 10th International Symposium on Spatial Data Handling, Dianne Richardson and Peter van Oosterom (eds.), Springer-Verlag, Berlin, 2002, pp. 209-224.

Oracle (2001), Oracle Spatial User's Guide and Reference Release 9.0.1 Part Number A88805-01, June 2001

Pasman, W, W. A. van der Schaaf, R L Lagendijk, F W Jansen (1999), Low latency rendering and positioning for mobile augmented reality. In: B. Girod, H. Niemann, H.P. Seidel (eds.); Proceedings Vision Modeling and Visualization '99 (Erlangen, Nov. 1999), Infix, Sankt Augustin, 1999, p. 309-315.

PostGIS (2003), http://postgis.refractions.net

Stoter, J.E. and P.J.M. van Oosterom (2002), Incorporating 3D geo-objects into a 2D geo-DBMS. In: XXII FIG International Congress and the ACSM-ASPRS Conference and Technology Exhibition, April 19-26 2002, Washington D.C., USA, 12 p. CD-rom.

Stoter, J.E. and H.D. Ploeger (2002), Multiple use of space: current practice and development of a 3D cadastre. In: Proceedings of UDMS '02 23rd Urban Data Management Symposium, Prague, Czech Republic, 1-4 October 2002.

Stoter, J.E. and S. Zlatanova (2003a), 3D GIS, where are we standing?, paper submitted to ISPRS workshop on spatial and temporal data modelling, Quebec City, Canada, October, 2003.

Stoter, J.E. and S. Zlatanova (2003b), Visualising and editing of 3D objects organised in a DBMS, EUROSDR workshop: Rendering and visualisation, January 2003, Enschede, The Netherlands

Thomas, M. D. (2003), Oracle XSQL : Combining SQL, Oracle Text, XSLT, and Java to Publish Dynamic Web Content, Wiley 2003

Zlatanova, S. (2000), 3D GIS for urban development, PhD thesis, TUGraz, Austria, ITC publication, The Netherlands

Web3D (2003), http://www.web3d.org/fs_specifications.htm