

EXPERIMENTAL STUDIES FOR THE DEFINITION OF 3D GEOSPATIAL WEB SERVICES

Andrea Scianna ^a

^a ICAR C.N.R. (National Research Council - Italy) – c/o Dipartimento di Rappresentazione, viale delle Scienze, edificio 8 – University of Palermo, 90128 Palermo, Italy tel.+39 091 8031074– andrea.scianna@cnr.it

Commission IV, WG IV/5

KEY WORDS: geospatial, web, services gis

ABSTRACT:

The importance of geospatial information delivery, across Internet, is increasing more and more. But if in the last years was satisfying to get 2D geographic information, from Internet browsers, extracted by http servers supported by map servers, today users would acquire 3D information especially in some case or sectors as that of building cadastral applications or civil protection. In this case availability of 3D information could be very valuable.

But behind this need there exist many important aspects that today are only partially resolved as the definition of 3D geospatial web services that are connected and are dependent on some other aspects (i.e. the standardization of 3D geographic data models).

The experimentation carried out deals with these issues, trying to define a 3D web service in order to visualize and query by Internet Browser 3D model of the built environment. Actually there are few 3D data models (such as CityGML) whose standard definition process is not complete. Besides these models are built with main requirement of 3D visualization even if at different level of detail. But some sectors require a better use of geographic 3D information such as querying at different level of detail (such as at a level regarding the different building parts defined through their attributes) and 3D processing.

Based on a project named “Interoperability and cooperative management of geographic, dynamic, multidimensional and distributed data with Free and Open Source GIS: Management and use of distributed 3D data by open source Web-GIS software” funded by Italian Ministry of Instruction, University and Research as Program of Relevant National Interest (PRIN 2007), it is here illustrated a part of a process, that start from the construction of a Java plugin that initially consist in reading information directly from a relational database management server with spatial extension, and ends with the construction of an application server which is based the extraction of GML 3D data, all based on the existence of a 3D geospatial web service whose definition is one the last and main goal of the research.

1. INTRODUCTION

The research, of which this project is part, called “Interoperability and cooperative management of geographic, dynamic, multidimensional and distributed data with *Free and Open Source GIS: Management and use of distributed 3D data by open source Web-GIS software*” is one of Italian Projects of Relevant National Interest (*PRIN 2007*), funded by Minister of Instruction, University and Research; it started on September 2007.

The project deals with the development of an experimental platform useful to browse, query and navigate a 3D urban model by Internet browsers and the NET, whose data are managed by a *RDBMS* with spatial extension, made available by 3D web service. The data model is the geometrical-topological model, called *GIANT* defined in a previous research *PRIN 2005* by this research unit.

As reported by Gruen and Wang [3] (1999), the generation and management of 3-D city models became an important issue ... due to the increasing demands for a realistic presentation of the real world.

There are today many sectors which could advantage from a more deep knowledge of urban environments, as civil protection, public works management, urban planning, etc. [1].

To navigate, in 3D cartography models, usually *VRML* techniques and browsers are used. But this kind of navigation doesn't allow also full browsing and query of semantic data (attributes) as, with *SVG* extension for Internet browsers, it is possible to operate only in 2D.

With *GML* it is possible describe 3D world at different levels of detail, with topological description and according to a predefined schema file.

Recent proposal of 3D *GML* models as *CityGML* allows also to associate attributes to geometrical information and this is an optimal solution for transferring or transmitting GIS data.

Towards this direction is oriented the work of a research group at GISLAB (ICAR CNR at Dipartimento di Rappresentazione of University of Palermo), that since many years has been working on building a 3D model of digital cartography and instruments to browse it.

In OGC 05-019 Editors U.Quad, T.H. Kolbe [5], is defined a WEB 3D Service that operates as *Portrayal Service* in order to render a scene.

This web service doesn't answer with data in such a way to be accessed by clients for navigation inside the model or query or processing operations; the service allows rendering of scenes eventually composed by up to four different images, delivered from different data sources (3D data providers); besides the images provided from different sources could be composed

together, according to a specific pipeline in order to be presented as unique representation.

This kind of service gives an instantaneous view of a scene, substantially an image, and hence is not suitable for the needs of cadastral applications, civil protection or planning activities where it is necessary to go inside the scene, select view point and dynamically navigate inside the model, also querying it to know feature of real world objects.

The service documentation so demands to *WFS* to get full data for these last specific purposes.

Usually, a server software suite, adapt to extract *GML* 2D data, is mainly composed by a *HTTP* server and a *MapServer* (as *UNM Mapserver*). The extraction generally consists of a *GML* file in which are listed feature classes and some instances of enumerated feature types with corresponding attributes. Rarely, topology is defined and the model extracted is only composed of geometry types defined in the standard *GML* schema by *OGC*.

But a cartography model requires the definition of specific schema (this is possible thanks to freedom and flexibility of *GML*), in which are defined all classes of objects coming from an abstraction of real word, derived from user needs, and possibly provided with a topological structure.

DATA ALLOCATION USING POSTGIS AND QUADS

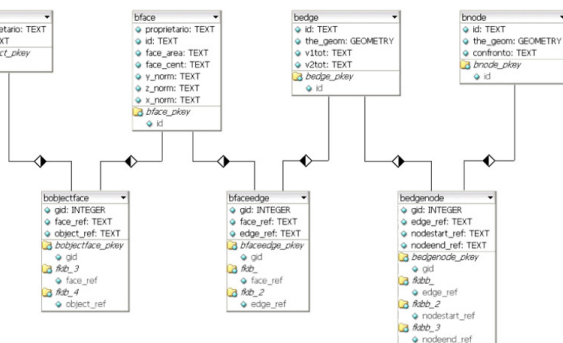


Figure 1. Querying spatial database

In this specific situation the main goal is to extract data, from a spatial database, compliant to the geometrical topological model *GIANT*. This is a cartographic model developed on the basis of some models proposed in Italy by Commissione Geodetica Italiana and following models as *GeoUML* by Intesa GIS (this last one proposed as 2D numerical cartography model).

In a previous step of this research it has been experimented a way to write data, according to some sections of *GIANT* model in a spatial database managed by *PostgreSQL* and *PostGIS*.

A full extraction of geospatial information, in a way compliant

with *GIANT* requires to build all the structures of the model inside a *PostGIS* database. At the present day the model is partially implemented, in order to write software modules of the 3D *WFS*. Spatial and topological data are defined as proposed in figure 1.

Following this way, a simple 3D model of buildings has been stored inside a *PostGIS* database, with a specific topological structure (see the data model of figure 2) and all this is the start point of this part of the research, whose goal consists in designing and writing a 3D web service software application similar, when possible, to *OGC WFS*.

The activity, here reported, deals with the development of a system useful to publish on Internet, geographical 3D data based on a *GML* 3 model.

The extracted data are presented as *GML/XML* data; to browse data in a graphical environment it is necessary to write a suitable software module, aspect that is the last goal of this research and that is at a stage of development even if a prototype is already available.

2. STRUCTURE OF THE WEB SERVICE

In general, a 3D web service for extraction of spatial vector data should:

- process client requests about the existence and capabilities of service (this is *GetCapabilities* operation of *OGC WFS*);
- describe feature types allocated inside a spatial database (this is *DescribeFeatureType* operation of *OGC WFS*);
- extract data from the spatial database when requested by the clients (this is *GetFeature* operation of *OGC WFS*).

Actually the 3D *WFS* should perform the subsequent operations:

1. process client *HTTP GET* or *POST* requests (compliant to *SOAP* or *XML*);
2. query the *RDBMS* to get information according to the type of request;
3. compose the *XML* answer, compliant to *GIANT* schema, putting together spatial information – with its topological structure – and semantic data ;
4. transmit *XML* data back to clients.

The proposed 3D *WFS* is defined as *OGC Basic WFS*. Compared with traditional 2D web service, *WFS* like, some differences arise:

1. the server replies with geometrical data furnished with the 3rd component (z coordinate) for characteristic points of each geometrical element; usually common map servers return data with only 2D components, as *UNM MapServer* does, or projecting data entities on xy plane;

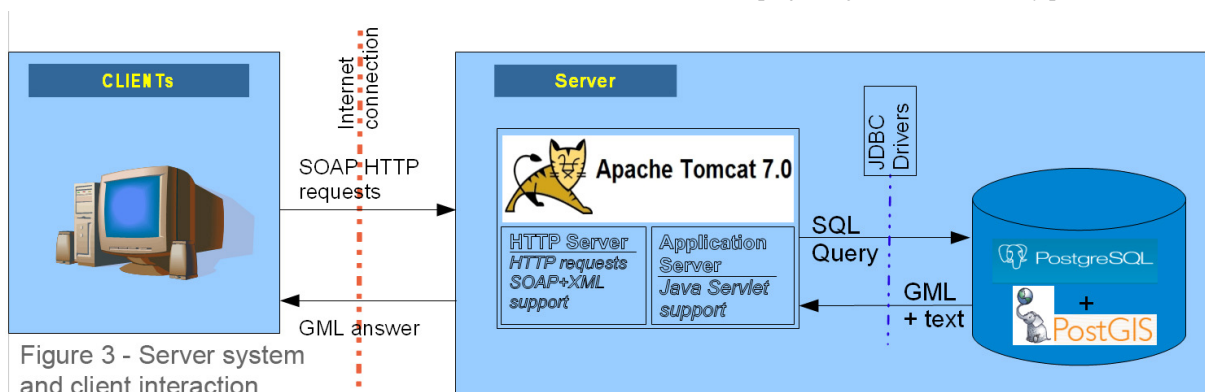


Figure 3 - Server system and client interaction

- topology data are written inside a *PostGIS* spatial database;
- the server replies with a *GML* file structured according a specific schema file; processing activity is performed by a Java servlet managed by Tomcat; usually map servers reply with *GML* data structured in a plane format according only to *OGC GML* Schema.

In order to query the server for selective extraction of spatial entities, an interface has been written in Java (see figure 2). This interface allows to send to server both spatial, non spatial and mixed queries. Another screen of the same application allows to browse graphical 3D representation produced by query processing.

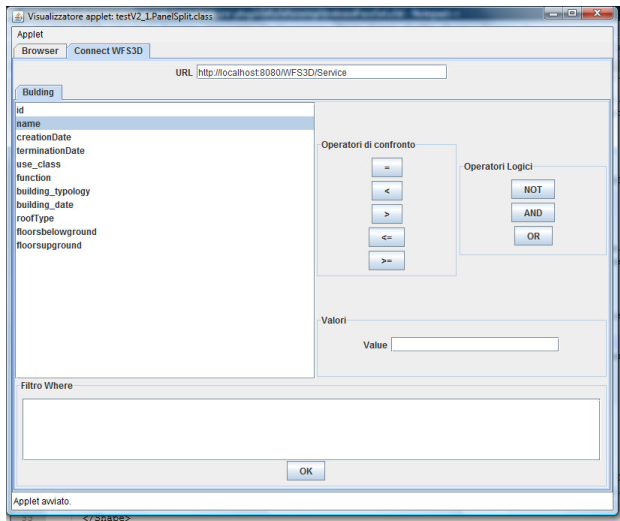


Figure 2. Querying spatial database

Operations available at the present development stage are:

- GetCapabilities;
- DescribeFeatureType;
- GetFeature.

2.1 Get Capabilities

As the *OGC* corresponding functions, it provides data general data on the service as well as information on class of features, that are present in the spatial database.

Except the name of the service this function is fully compliant to *OGC WFS* corresponding function. So this is similar to *OGC GetCapabilities* function.

Example of the request:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<GetCapabilities service="WFS3D"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
../wfs/1.1.0/WFS.xsd" xmlns="http://www.opengis.net/wfs"/>
```

Once the query is sent to the server, it replies with *XML* data as:

```
<?xml version="1.0" encoding="UTF-8"?>
<WFS3D_capabilities
xmlns:gml="http://www.opengis.net/gml"
...
```

```
<ows:ServiceTypeVersion>0.1</ows:ServiceTypeVersion>
<ows:Title>web service 3d gis</ows:Title>
<ows:Abstract>servizio che restituisce scene GIANT3D:
Application Schema GML_Based</ows:Abstract>
...
<ows:ServiceProvider>
<ows:ProviderName>LabGis_Palermo</ows:ProviderName>
<ows:ProviderSite>None</ows:ProviderSite>
<ows:ServiceContact>
<ows:IndividualName>Andrea Scianna</ows:IndividualName>
<ows:PositionName>Ricercatore</ows:PositionName>
<ows>ContactInfo>
...
</ows>ContactInfo>
<ows:Role>Contatto</ows:Role>
</ows:ServiceContact>
</ows:ServiceProvider>
<ows:OperationMetadata>
<ows:Operation name="GetCapabilities">
<ows:DCP>
<ows:HTTP>
<ows:Get
xlink:href="http://localhost:8080/GetCapabilitie
s/W3DS"/>
</ows:HTTP>
</ows:DCP>
...
<ows:Parameter name="Section">
...
</ows:Parameter>
</ows:Operation>
<ows:Operation name="DescribeFeatureType">
<ows:DCP>
<ows:HTTP>
<ows:Get
xlink:href="http://localhost:8080/GetCapabilities/W3DS?"/>
<ows:Post
xlink:href="http://localhost:8080/GetCapabilities/W3DS"/>
</ows:HTTP>
</ows:DCP>
<ows:Parameter name="AcceptFormats"/>
</ows:Operation>
<ows:Operation name="GetFeature">
<ows:DCP>
<ows:HTTP>
<ows:Get
xlink:href="http://localhost:8080/GetCapabilities/W3DS?"/>
<ows:Post
xlink:href="http://localhost:8080/GetCapabilities/W3DS"/>
</ows:HTTP>
</ows:DCP>
...
</ows:Parameter>
<ows:Parameter name="AcceptFormats">
<ows:Value>text/xml; subtype=gml/3.1.1</ows:Value>
</ows:Parameter>
</ows:Operation>
</ows:OperationMetadata>
<wfs:FeatureTypeList>
<wfs:FeatureType xmlns:bo="http://www.dirap.unipa.it">
<wfs:Name> Building </wfs:Name>
<wfs:Title>Building</wfs:Title>
<wfs:Abstract>descrizioneMappa</wfs:Abstract>
<wfs:NoSRS/>
<wfs:OutputFormats>
<wfs:Format>descrizioneMappa</wfs:Format>
```

```

</wfs:OutputFormats>
</wfs:FeatureType>
</wfs:FeatureTypeList>
</WFS3D_capabilities>

```

This service operates as *GET* or *POST* methods. In the section identified by *FeatureTypeList*, of the response, are reported classes of features, defined in the *GIANT* schema.

2.2 DescribeFeatureType

This operation operation generate, as response, a description of classes of features as described in the *GIANT* schema.

On the basis of a request as:

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<DescribeFeatureList xmlns:gia="http://www.dirap.unipa.it"
service="WFS3D"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
../wfs/1.1.0/WFS.xsd" xmlns="http://www.opengis.net/wfs">
<TypeName>giant3d:GIANT3DModel</TypeName>
</DescribeFeatureList>

```

The response of the service is for example:

```

<?xml version="1.0" encoding="UTF-8" ?>
- <schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:giant3d="http://www.dirap.unipa.it"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink"
elementFormDefault="qualified"
targetNamespace="http://www.dirap.unipa.it" version="0.1">
...
- <complexType abstract="true" name="_BuildingType">
- <annotation>
<documentation>This abstract type includes urban objects
classified as buildings (residential houses, public
buildings,schools, etc. . .). Its attributes are inherited by the
"Building" and "Building_Part" classes. A building can be
subdivided into different "Building_Part", that represent a
relevant portion with its identity."orizzontal_partitioning", that
is one of the attributes of the abstract class "_Building",
represents all building units that belong to each
building.</documentation>
</annotation>
- <complexContent>
- <extension base="_ConstructionType">
- <sequence>
<element minOccurs="0" name="use_class"
type="BuildingClassType" />
<element maxOccurs="unbounded" minOccurs="0"
name="function" type="BuildingFunctionType" />
<element maxOccurs="unbounded" minOccurs="0"
name="building_typology" type="BuildingUsageType" />
<element name="building_date" type="xs:gYear" />
<element name="roofType" type="RoofTypeType" />
<element minOccurs="0" name="consistsOfBuildingPart"
type="Building_PartPropertyType" />
<element name="floorsupground" type="xs:integer" />
<element name="floorsbelowground" type="xs:integer" />
<element name="ground_size" type="Ground_sizeType" />
<element name="maximum_size" type="Maximum_sizeType"
/>
<element name="crowning" type="CrowningType" />

```

```

<element maxOccurs="unbounded" name="terrain-
building_intersection" type="Terrain-
Building_intersectionType" />
<element minOccurs="0" name="orizzontal_partitioning"
type="orizzontal_partPropertyType" />
<element minOccurs="0" name="bounded"
type="_BoundedbyType" />
<element maxOccurs="unbounded" name="property_lands"
type="accessory_perimeterType" />
<element maxOccurs="unbounded" minOccurs="0"
name="Gisolated" type="gml:IsolatedPropertyType" />
<element maxOccurs="unbounded" minOccurs="0"
name="Gsurface" type="gml:ContainerPropertyType" />
</sequence>
</extension>
</complexContent>
</complexType>
<element abstract="true" name="_Building"
substitutionGroup="_Construction" type="_BuildingType" />
+ <complexType name="BuildingType">
- <complexContent>
- <extension base="_BuildingType">
<sequence />
</extension>
</complexContent>
</complexType>
<element name="Building" substitutionGroup="_Building"
type="BuildingType" />
</schema>

```

Examining the response it is possible to see that, the element *Building* is a complex type *BuildingType*. *BuildingType* describe the cartographic model of building as defined in *GIANT*.

This response is obtained from a *HTTP POST* Request.

2.3 GetFeature

This function allows extract, on the basis of query sent as *HTTP POST* Request, some 3D elements stored inside the spatial database.

A request could be:

```

<?xml version="1.0" encoding="UTF-8"?>
<GetObject xmlns="http://www.opengis.net/wfs"
xmlns:giant3d="http://www.dirap.unipa.it"
xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
service="WFS3D"
xsi:schemaLocation="http://www.opengis.net/wfs
../wfs/1.1.0/WFS.xsd">
<Query TypeName="Building">
<ogc:Filter xmlns="http://www.opengis.net/ogc">
<ogc:PropertyIsEqualTo>
<ogc:PropertyName>name</ogc:PropertyName>
<ogc:Literal>edificio1</ogc:Literal>
</ogc:PropertyIsEqualTo>
</ogc:Filter>
</Query>
</GetObject>

```

The server replies with data of entities, the user asked for, coming from the query string composed in the interface. The result is as the subsequent:

```

<?xml version="1.0" encoding="UTF-8"?>

```

```
<GIANT3DModel xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.dirap.unipa.it">
```

--- real world objects section from GIANT---

```
<Building gml:id="1">
  <gml:name>edificio1</gml:name>
  <Gsurface>
  <gml:TopoSolid>
  <gml:DirectFace gml:orientation="+
    type="symple" xlink:href="#59217e9f-2128-
    11df-a213-005056c00008"/>
  . . .
  </gml:TopoSolid>
  </Gsurface>
</Building>
```

--- nodes section ---

```
<gml:Node gml:id="8da46de6-9660-40a1-98dc-
55db63086ced">
  <gml:pointProperty>
  <gml:Point srsName="EPSG:3004">
  <gml:coordinates>1,1,-1</gml:coordinates>
  </gml:Point>
  </gml:pointProperty>
</gml:Node>
<gml:Node gml:id="95288bf1-9937-45b7-b0ef-
f7b5dfbe8061">
. . .
. . .
```

--- edges section ---

```
<gml:Edge gml:id="46660ddf-98d9-4ac2-8eac-
612bf000b4b3">
  <gml:DirectNode type="symple" xlink:href="#8da46de6-9660-
40a1-98dc-55db63086ced"/>
  <gml:DirectNode type="symple" xlink:href="#95288bf1-9937-
45b7-b0ef-f7b5dfbe8061"/>
</gml:Edge>
. . .
```

--- faces section ---

```
<gml:Face gml:id="59217e9f-2128-11df-a213-
005056c00008">
  <gml:DirectEdge gml:orientation="+ type="symple"
  xlink:href="#46660ddf-98d9-4ac2-8eac-612bf000b4b3"/>
  <gml:DirectEdge gml:orientation="+ type="symple"
  xlink:href="#2b8f349f-b274-4bff-b224-326b6f07a0e7"/>
  <gml:DirectEdge gml:orientation="+ type="symple"
  xlink:href="#c2c59be1-698d-4333-831b-041a2d596af2"/>
  <gml:DirectEdge gml:orientation="-" type="symple"
  xlink:href="#5fe4b399-cec3-4404-aa03-b1c1115558c2"/>
</gml:Face>
. . .
</GIANT3DModel>
```

Data provided by *GetFeature* are structured in a format compliant to cartographic *GIANT* model.

So in general inside the dataset two main parts exist:

- the first part with the description of entities of real world as established in the *GIANT* model;
- a second part in which topological and geometrical information are reported.

Topology is defined by *xlink*: attributes, according to *GIANT* model.

3. THE DEVELOPMENT

As foreseen in the project proposal, all system is composed and built by are free and open source software (FOSS).

The development of WEB dynamic applications has been carried out in *Eclipse IDE Enterprise Edition*, with the support of *J2EE* and related plug-ins.

Some functions for parsing client request are derived from *GeoTools*, while for the management of XML information has been Java language with the support of *JAXP* libraries; *Java* is the main language used for the development of the whole system.

As said before, Tomcat (see Figure 3) has the role of HTTP and servlet container (acting as web server and middleware or applications server software), while as Relational database Management System has been used PostgreSQL with PostGIS spatial extension.

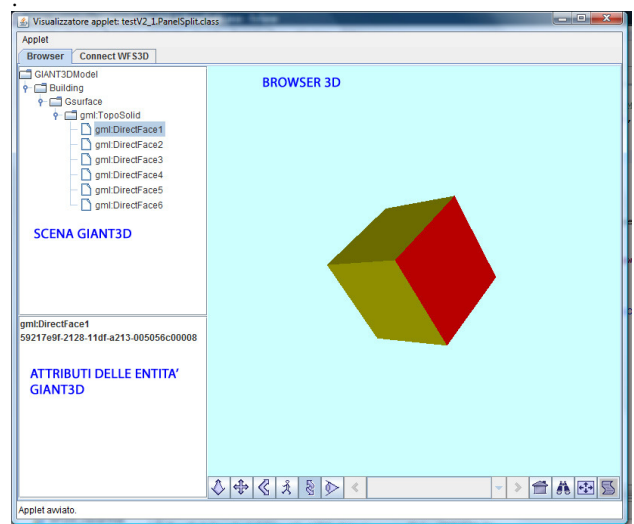


Figure 4. Browsing of 3D model with a first version of 3D Java browser

4. CONCLUSION AND FURTHER DEVELOPMENTS

The work, here presented, is in progress. The last goal of the project is the development of the Java applet that can be downloaded by the client, querying the server for *GML* 3D data; according this way, the client processes data transforming them in a suitable format for graphical browsing, as shown in the experimental version of a browser, the research group is going to be completed (see Figure 4). This graphical browser is written in Java; it executes the conversion from *GML* response - compliant to *GIANT* - to *X3D* before visualize the model. The browser is written to operate as applet running in the development environment.

It acts as thick client as it executes some heavy operations.

5. REFERENCES AND/OR SELECTED BIBLIOGRAPHY

Reference from Proceedings, Books and other Literature:

[1] A. Altmaier, T. H. Kolbe (2003) – “Applications and Solutions for Interoperable 3D Geo-visualization” - Proceedings of the Photogrammetric Week 2003, Stuttgart, Germany

[2] Chen T. Khuan, A. Abdul-Rahman, S. Zlatanova (2008) - “New 3D data type and topological operations for Geo-DBMS” - In: P. van Oosterom, S. Zlatanova, F. Penninga and E. Fendel (Eds.); Advances in 3D Geoinformation Systems, Springer, 2008, pp. 279-311

[3] A. Gruen, X. Wang (1999) – “CyberCity Spatial Information System (CC-SIS): A new concept for the management of 3-D city models in a hybrid GIS” - 20th Asian Conference on Remote Sensing”, Nov. 22-25, 1999, Hong Kong, China.

[4] Held, G, A. A. Rahman and S. Zlatanova (2004) – “Web 3D GIS for urban environments”, in: Proceedings of the International Symposium and Exhibition on Geoinformation 2004 (ISG2004), 21-23 September, CDROM, 12 p.

[5] Open Geospatial Consortium Inc. - Web 3D Service - OGC™ document: OGC 05-019, date: 2005-02-02, Version: 0.3.0, Category: OGC™ Discussion Paper, Editors: Udo Quadt, Thomas H. Kolbe

[6] A. Scianna, A. Ammoscato, R. Corsale (2008) – “GIANT3D: experimentations on a new 3D data model for GIS” – Proceedings of XXI ISPRS CONGRESS - WgS IV/1, BEIJING, July 2008, issn 16821750, CRC Press (Taylor & Francis Group);

[7] A. Scianna, A. Ammoscato (2010) – “3D GIS DATA MODEL USING OPEN SOURCE SOFTWARE”, in: ISPRS Archive Vol. XXXVIII, Part 4-8-2-W9, "Core Spatial Databases - Updating, Maintenance and Services – from Theory to Practice. Haifa, Israel, 15-17 March, 2010, HAIFA: University of Haifa, vol. ISPRS Archive Vol. XXXVIII.

References from websites:

- GeoTools - <http://geotools.org/>

- PostGIS documents – <http://www.postgis.org>

- PostgreSQL - <http://www.postgresql.org>

- CityGML – <http://www.citygml.org>

- Open Geospatial Consortium – <http://www.ogc.org>

- Open Geospatial Web Service Common:
<http://www.opengeospatial.org/standards/common>

- WebServices : <http://www.w3.org/2002/ws/>