Chapter 18
Serving CityGML via Web Feature Services in the OGC Web Services - Phase 4 Testbed

Eddie Curtis

18.1 Background

The OGC Web Services – Phase 4 (OWS-4) test-bed is an initiative under the OpenGeospatial Consortium's (OGC) interoperability programme in which 72 organisations collaborated to extend and demonstrate interoperability of internet based geospatial services. The main activities of the test-bed took place between June and December of 2006. These activities included the deployment and integration of a number of software and data components culminating in two demonstrations of the capabilities developed during the test-bed, as well as numerous reports detailing issues encountered and recommendations.

The test-bed used an emergency response scenario to exercise the capabilities of a variety of OGC conformant components. The scenario required numerous components and services to interoperate in order to provide emergency planners with information to coordinate the response to the hypothetical incident. The activity was divided into a number of threads each of which addressed particular set of technical issues affecting the scenario such as security, workflow and sensor web enablement. This paper will consider some of the findings from the thread which covering the integration of Computer Aided Design (CAD) systems, GIS (Geographical Information Systems) and Building Information Models (BIM), known as the CAD/GIS/BIM thread. This thread was concerned with issues related to exchanging information about buildings and 3D geometry between GIS analysts and building designers.

Snowflake Software, United Kingdom
eddie.curtis@snowflakesoftware.co.uk

18.2 CAD/GIS/BIM Integration

Whilst both GIS and CAD deal with spatial information the approaches used by each are shaped by different requirements and working practices. The GIS world emphasises the analysis of measured information i.e. data collected through survey, satellite imagery etc. whereas the CAD world emphasises the design and construction processes. However, there is clearly an overlap of concerns between geographical information users and the Architecture Engineering and Construction (AEC) industry, which makes extensive use of CAD.

There are a number of scenarios in which the interoperation of BIM and geographical information are of benefit.

In the site planning process for AEC it is useful to consider contextual information about the site in order to understand how a proposed building will interact with its environment. This could include aerial photography, terrain models, and nearby buildings and infrastructure. The ability of CAD systems to discover and import this information is therefore beneficial to the design process. Enabling CAD systems to act as clients to OGC web services is a means to achieve this.

Integration of BIM models for different sites requires the introduction of geographical concepts into the BIM models. Unless the separate BIM models are referenced to geographical coordinate systems it is not possible to relate the positions of objects in the separate models to each other. This type of integration is necessary to handle common infrastructure shared by buildings and can identify potential conflicts between construction projects.

It is also beneficial to bring BIM information into the GI world. Location based services could be extended from the street to building interiors. For example, fire-fighters could be provided with information about infrastructure such as electricity and water supplies both inside and outside a burning building. Geographical analysis and broad scale visualisation could also make use of BIM information allowing, for example, emergency planners to identify buildings prone to particular risks or suitable for conversion to emergency use. Since BIM models often contain highly detailed information it is not practical to create and maintain a single database containing building information for a whole city. On-the-fly integration of information held in a distributed, heterogeneous set of databases through OGC web services represents a more practical approach to making building information available to GIS applications.

18.3 CityGML

City Geography Markup Language (CityGML)[2] is an information model and GML application schema for the exchange of 3D city and landscape models. Originally developed by the members of the Special Interest Group 3D (SIG 3D) of the Geodata Infrastructure North-Rhine Westphalia (GDI NRW) initiative in Germany, CityGML has now been adopted as an OGC discussion paper with a view to it becoming an OGC best practices paper.

A key characteristic of CityGML is that it combines the ability to contain complex, geo-referenced 3D vector data along with the semantics associated with the data. In contrast to other 3D vector formats, CityGML is contains a rich, general purpose information model in addition to geometry and graphics content. The CityCML information model includes:

- Digital Terrain Models as a combinaton of triangulated irregular networks (TINs), regular rasters, break and skeleton lines, mass points
- Sites (currently buildings and bridges)
- Vegetation (areas, volumes, and solitary objects with vegetation classification)
- Water bodies (volumes and surfaces)
- Transportation facilities (both graph structures and 3D surface data)
- City furniture
- Generic City objects and attributes

For specific domain areas CityGML also provides an extension mechanism to allow the model to be enriched with additional properties and feature types. Targeted application areas explicitly include urban and landscape planning; architectural design; tourist and leisure activities; 3D cadastres; environmental simulations; mobile telecommunications; disaster management; homeland security; vehicle and pedestrian navigation; training simulators; and mobile robotics.

CityGML provides a model at multiple levels of generalisation. These can be used individually within a model or multiple levels of representation can be modelled together. The levels are:

- LOD 0 – Regional, landscape
- LOD 1 – City, region
- LOD 2 – City districts, projects
- LOD 3 – Architectural models (exterior), landmarks
- LOD 4 – Architectural models (building interiors)

Since CityGML deals with buildings and constructions there is clearly some overlap with BIM. However, the two information models are different in scope. CityGML stops far short of the level of detail supported by BIM. A BIM can contain detail down to the level of component parts within individual fixtures such as doors and windows. CityGML LOD4 provides a level of detail suitable for a "walkable" model of a building for simulation or space analysis purposes. However, CityGML provides for modelling of the building context including roads, street furniture, terrain, vegetation etc. BIM models are concerned only with the building. The two models are complementary with

CityGML providing a geographical view of buildings in their context, and BIM providing a detailed view of buildings and their construction.

18.4 The OWS-4 Scenario

The OWS-4 test-bed required a variety of systems and data to interoperate to solve a test scenario. In this scenario a ‘dirty bomb’ has detonated in a port. After identifying and analysing the problem the emergency planners decide that an emergency field hospital will be required and begin looking for a suitable building to convert to use as a hospital. There are a number of criteria which the building must meet including access to an airstrip, and space requirements for an operating theatre.

Identification of a suitable building requires a number of OGC services to provide information about the site such as aerial photography of the site, terrain models etc. In order to assess the space requirements of the building a 3D model of the building is needed. This is supplied from a Web Feature Server (WFS)[4] providing a CityGML model of the site. A hangar building at a nearby airport is identified as being suitable for the field hospital.

The hangar must be modified for use as a hospital. Here the focus shifts from GIS analysis to CAD design. An operator uses a CAD client to access the BIM model for the hangar via the internet. The operator modifies the hangar model to convert it for use as a hospital. The modified hangar model is made available as both a BIM and CityGML model, thus enabling the GIS analysts to see the modified hangar as CityGML as they did with the original hangar model.

18.5 CityGML WFS Challenges

Serving CityGML via a WFS presents a number of technical challenges arising from the characteristics of the CityGML model.

The CityGML model is untypical of GML application schemas in the level of complexity of the data model. CityGML makes extensive use of complex data types for properties and nesting of features within feature collections. Consequently CityGML data can contain very deeply nested data structures.

The geometry types supported in relational databases are often more limited than the range of geometry types used in CityGML. For example, TINs and 3D solid geometries are not supported in Oracle 10g. This presents an obstacle to WFS implemented on top of a relational database.

At high levels of detail data volumes can become large, even for a small geographical area.

18.6 Snowflake CityGML WFS

CityGML datasets for the test-bed scenario were created by the Forschungszentrum Karlsruhe institute. The Forschungszentrum Karlsruhe team developed a software tool for the conversion of building information models encoded using Industry Foundation Classes (IFC)[1] building models to CityGML. This tool carries out a number of mapping operations between IFC and CityGML models. The tool maps IFC classes to CityGML classes e.g. an IfcSpace becomes a CityGML interiorRoom. The tool also converts geometry from the local coordinate systems used within the IFC model to geographic coordinate systems.

The Snowflake CityGML WFS was created by deploying Snowflake's GO Publisher. GO Publisher is a data translation engine which translates from relational databases to XML. In order to stand up the GO Publisher WFS the CityGML data produced by Forschungszentrum Karlsruhe was loaded into an Oracle data model using a GML bulk loading tool called GO Loader. This tool has a similar translation capability to GO Publisher but translates from GML to relational models.

Oracle's SDO geometry types were used to store the geometry and conventional relational structures were used to hold the non-spatial properties. Relationships such as the containment relationships between buildings and rooms were represented as table joins in the database. GO Publisher's graphical user interface was then used to configure a translation from the relational model to the CityGML schema. GO Publisher tools were used to bundle this translation with the GO Publisher software into a Web ARchive (WAR) file for deployment within an application server. The CityGML WFS was deployed by uploading this WAR file into the Tomcat application server.

Once deployed within the application server the Snowflake CityGML WFS was able to process requests to the WFS operations getCapabilities, describeFeatureType and getFeature. On receiving a getFeature request GO Publisher translates the WFS filter in the request into an SQL query using the translation configured prior to deployment. The resulting SQL query contains all conditions from the WFS filter including both spatial and non-spatial operations. The SQL query is run against the Oracle database containing the city model. GO Publisher then translates the resulting SQL records into GML, again using the translation configured prior to deployment. The resultant GML is then streamed back to the client. The data is returned in a compressed stream if appropriate.

By carrying out the two directional translation (WFS filter to SQL followed by relational data to GML) GO Publisher is able to make use of the scalability, performance and robustness of the underlying Oracle database. The application server can create multiple instances of the WFS in order to deal with concurrent requests for data. These technologies therefore provide a highly scalable platform for the WFS.

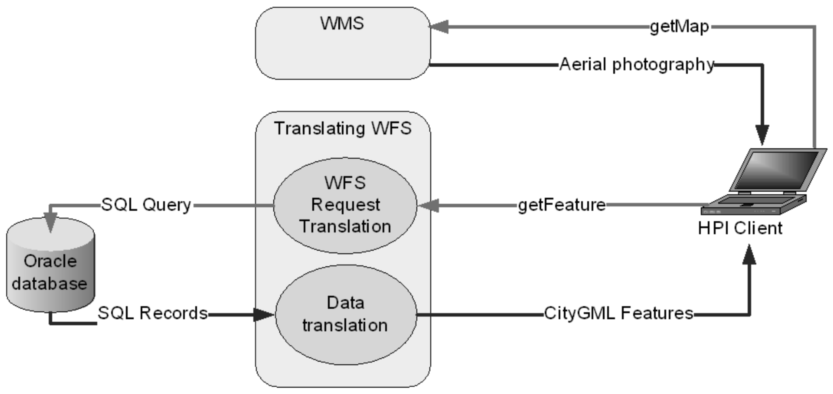

Fig. 18.1 Data requests and response from the client application

18.7 Findings

The testbed proved the feasibility of serving CityGML through the WFS interface. A client application developed by the Hasso-Plattner Institute successfully connected to the WFS, retrieved data and displayed it. The demonstration also showed the utility of connecting to OGC web services as the client was able to connect to several different services provided independently by different organisations around the world and to integrate the data into a single view. For example, the client took a terrain model from the CityGML WFS and draped aerial photography from a Web Coverage Server across the terrain. This allowed the analyst to build up a picture of the situation at the airport by drawing on a variety of independent sources of information.

The CityGML schema is much more complex than those usually deployed in WFS. The unusual level of complexity did not cause problems in the interaction between the client and the Snowflake CityGML WFS. The underlying information model of CityGML was known to the client, so the client was able to form meaningful request and correctly interpret the response.

A specific example of the data complexity is that CityGML buildings are made up of component parts which also contain component parts. A building may be made up of wall surfaces, each of which can contain windows and doors. All of these are objects in their own right with their own identity, properties and geometries. When the client requested the hangar building, the server returned the hangar and also returned some of the component parts of the hangar (walls, doors etc.) even though these were not explicitly requested.

For different parts of the scenario different combinations of objects were required. In particular the spaces (rooms) within the hangar were required for analysis of the building but later on, after editing of the spaces for use as a hospital, the spaces were supplied from a different server whilst the walls and windows continued to be supplied from the Snowflake CityGML WFS. This was handled by setting up a number of alternative WFSs which served different variations of the content. These included different combinations of optional properties of the CityGML model suitable for the different circumstances. Because of the translation capabilities of the Snowflake GO Publisher product this could be done without duplication of the data in the underlying database. Several different translations from the underlying database were set up with some translations omitting some of the optional CityGML properties. The client was thus able to get different views into the single underlying city model by connecting to different WFSs.

The CityGML model contains 5 levels of detail (LOD). In the testbed scenario the WFS client made a series of request for data including some buildings at LOD2 (building shapes with roof shapes) and the hangar at LOD4 (an architectural model including interior rooms, walls and doors). This mixing of LOD allowed the client to build up a view with increased level of detail for the buildings of interest and lower levels of details for the buildings which were requested for context. This created a composite view with LOD in different areas of the model customised, on-the-fly, to the task at hand. The CityGML LOD concept proved useful because it allows the level of detail available in a model to be specified easily in both the data and the description of a data set. This scenario showed that by providing CityGML data through a WFS the LOD concept adds further value by allowing the client to select data at a mixture of levels of detail appropriate to the scenario.

A number of 3D geometry types were served by the WFS in addition to the solid geometries of the buildings. A detailed solid geometry for a helicopter object was served. This geometry is a particularly large geometry in GML consisting of many hundreds of surface patches. A terrain model in the form of a TIN (Triangulated Irregular Network) was also served from the WFS. Both of these geometry objects required the WFS to return large GML files in response to the WFS request. Initially this cause performance problems since the files took approximately 9 minutes to return via the network available for the demonstration, thus preventing the WFS from being used in an interactive manner. This was overcome by using compressed streams to return the data. Both gzip and zip streams were implemented on the Snowflake CityGML WFS and the Hasso-Plattner client was enhanced to read these streams. The zip algorithm compressed the GML output to approximately 5% of its uncompressed size (a total data volume of 24.5 MB was compressed to 1.3 MB). This reduced the time for requesting, receiving and displaying the CityGML data from approximately 9 minutes to around 20 seconds, allowing the service to be demonstrated live from a remote server during the final demonstration of the OWS4 test-bed.

Filters were used within the WFS requests to select features spatially. Filters containing 2D bounding-box geometries were used to select features with 3D solid geometries. The spatial test was carried out by testing the

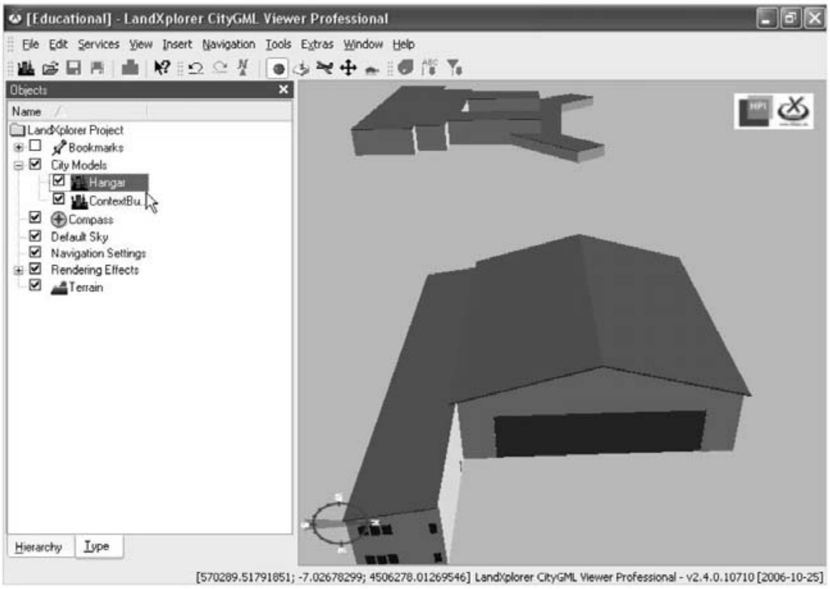
[Educational] - LandXplorer CityGML Viewer Professional
File Edit Services View Insert Navigation Tools Extras Window Help
Objects
Name
LandXplorer Project
Bookmarks
City Models
Hangar
ContextBu
Compass
Default Sky
Navigation Settings
Rendering Effects
Terrain
Hierarchy
Type
[570289.51791851; -7.02678299; 4506278.01269546] LandXplorer CityGML Viewer Professional - v2.4.0.10710 [2006-10-25]

Fig. 18.2 The LoD4 hangar building in the foreground has architectural detail such as windows and doors whilst LoD2 building in the background does not

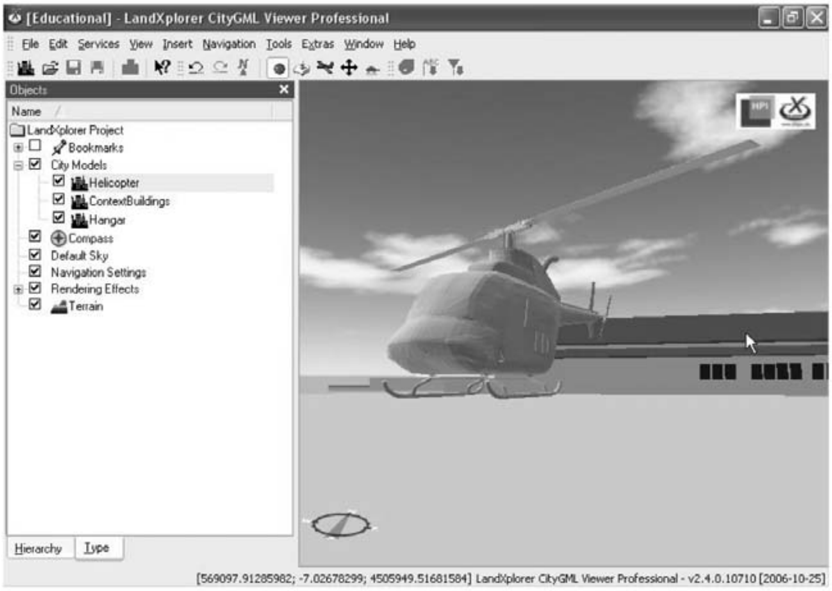
[Educational] - LandXplorer CityGML Viewer Professional
File Edit Services View Insert Navigation Tools Extras Window Help
Objects
Name
LandXplorer Project
Bookmarks
City Models
Helicopter
ContextBuildings
Hangar
Compass
Default Sky
Navigation Settings
Rendering Effects
Terrain
Hierarchy
Type
[569097.91285982; -7.02678299; 4505949.51681584] LandXplorer CityGML Viewer Professional - v2.4.0.10710 [2006-10-25]

Fig. 18.3 This screenshot shows illustrates the large number of surface patches and vertices in the helicopter geometry

interaction of the bounding-box with the 2D horizontal projection of the 3D geometries. This proved to be an effective interpretation of the filter even though the third dimension was ignored for purposes of the query. This is because the objects of the city model, although 3D, are distributed around the ground surface. The horizontal distribution of the objects is therefore great in relation to the vertical distribution. Consequently the degree of selection offered by 2D queries is high and corresponds to the use-cases of the test bed.

The limitations of the Oracle 10g geometry types was overcome by storing geometry in existing data structures but changing the interpretation of those structures on generating GML geometries. GML solid geometries where generated by interpreting the polygons within an Oracle multi-polygon as faces in the boundary of a GML solid. This approach to geometry translation is discussed in more detail in the paper 'Extending 2D Interoperability Frameworks to 3D' [3].

18.8 Conclusions

Despite the complexity of the CityGML model the WFS interface proved to be suitable for providing web based access to the city model.

Initial performance problems relating to the verbose nature of XML and GML were solved using compression and the WFS interface was shown to be a practical solution for large and complex models including large geometries.

The ability to select and filter data via the web allowed the client to build up a model specific to the problem in hand combining high levels of detail for buildings of particular interest and less detail for contextual objects. The scalable nature of the Snowflake CityGML WFS makes it feasible for a client to draw on very large city models by delegating the task of selecting and filtering the data to the remote server.

References

1. International Alliance for Interoperability (2006) Industry Foundation Classes,
`http://www.iai-international.org/Model/R2x3\_final/index.htm`
2. Gröger G, Kolbe TH, Czerwinski A (2006) City Geography Markup Language, OGC document 06-057r1
3. Mueller H, Curtis E (2005) Extending 2D Interoperability Frameworks to 3D, Paper presented to the International Workshop on Next Generation 3D City Models, Bonn, Germany, 2005
4. OGC (2005) Web Feature Service Implementation Specification, OGC document 04-094

Bibliography

OGC (2004) Filter Encoding Implementation Specification, OGC document 04-095

OGC (2007) OGC Web Services Architecture for CAD GIS and BIM, OGC Interoperability Program Report

OGC (2007) OpenGIS Geography Markup Language (GML) Implementation Specification, OGC document 03-105r1