

Fruizione di modelli dati GIS 3D di ambiente urbano tramite Web Services

Andrea Scianna

ICAR C.N.R. (National Research Council -Italy), c/o DICA, viale delle Scienze, edificio 8, University of Palermo
90128 Palermo, Italy, Tel.+39 091 8031074, andrea.scianna@cnr.it

Abstract

Today are more and more required GIS applications based on full 3D data; fields like urban and town planning and pollution studies need 3D data, both for visualization purpose, as well as carry out many spatial analysis. Starting from these needs borns the idea to develop the research "Management and use of distributed 3D data by open source Web-GIS software" that is part of the Italian "PRIN 2007"* research project, aimed to build urban and suburban 3D models, and to interact with them using open source software only.

Buildings have been modeled upon the GIANT3D model (Geographical Interoperable Advanced Numerical Topological 3-Dimensional Model) developed in the research "PRIN 2004", regarding "Evolved structure of numerical cartography for Gis and Web-GIS".

Python scripts, activated by Blender, allow to store data into a spatial database implemented through PostgreSQL and PostGis, that could be a remote database somewhere on the net; all geometrical and topological information, implemented in the 3D model, are so transferred in PostGIS.

This information can be retrieved by Blender using other Python scripts, so Blender fully interacts, in by-directional way, with 3D data allocated in PostGIS. These data can be also accessed by many other clients, both directly using a database client, as using other protocols (like HTTP on the internet). Next step was to build a software server in order to supply a 3D web feature service WFS-OGC "like" permitting to access to 3D data through SOAP protocol on the basis of a request composed in XML.

The research end with building of an applet running on Internet browser, permitting to browse and query the 3D model of a city.

Riassunto

Oggi molti tipi di applicazioni richiedono che i dati contengano veri dati tridimensionali; campi come la pianificazione urbana e gli studi sull'inquinamento hanno bisogno di dati 3D, sia per scopi di visualizzazione, che per svolgere molte analisi spaziali. Dal manifestarsi di tali esigenze nasce l'idea dello sviluppo della ricerca "*Gestione e uso di dati distribuiti 3D open source web-GIS - fa parte progetto di ricerca italiano "PRIN 2007"*", finalizzata a costruire modelli 3D urbani e suburbani, e ad interagire con essi utilizzando solo software libero e open source sia per la costruzione e la strutturazione tridimensionale del modello 3D di ambiente urbano che per l'allocazione dei dati ed infine la sua fruizione.

Gli edifici sono stati modellati sul modello GIANT3D (Geographical Interoperable Advanced Numerical Topological 3-Dimensional Model) sviluppato nella ricerca "PRIN 2004", sulla "struttura

* PRIN 2007: "Interoperabilità e gestione cooperativa di dati geografici dinamici multidimensionali e distribuiti con strumenti GIS liberi e Open Source" – Principal investigator Paolo Zatelli.

* PRIN 2007: "Interoperabilità e gestione cooperativa di dati geografici dinamici multidimensionali e distribuiti con strumenti GIS liberi e Open Source" – Principal investigator Paolo Zatelli.

evoluta di cartografia numerica per GIS e Web-GIS".

Gli script di Python, attivati dal software Blender, permettono di archiviare tutte le informazioni geometriche e topologiche in un database spaziale, che potrebbe essere un database remoto da qualche parte in rete, tramite PostgreSQL e PostGIS.

Queste informazioni possono essere recuperate da Blender con altri script Python, in tal modo Blender interagisce pienamente con i dati 3D allocati in PostGIS sia in lettura che in scrittura. Questi dati possono essere accessibili anche da molti altri client, sia direttamente che tramite un client di database, che tramite l'utilizzo di altri protocolli (come HTTP su Internet). Il passo successivo è stato quello di costruire un server in grado di rendere disponibile un servizio WFS like 3D per accedere ai dati eseguendo una richiesta tramite protocollo soap formulata come sequenza XML.

La sperimentazione illustrata si è conclusa con la realizzazione di un'applet funzionante con i browser web tale da visualizzare ed interrogare tramite in remoto tramite Internet un modello 3D dell'ambiente costruito.

Introduzione

L'attività di ricerca "Aspetti di gestione e fruizione WebGis di dati tridimensionali distribuiti con strumenti Open Source", fa parte del progetto: Interoperabilità e gestione cooperativa di dati geografici dinamici multidimensionali e distribuiti con strumenti GIS liberi e Open Source. Essa è stata sviluppata secondo 3 linee principali;

- studio dell'ulteriore estensione del modello GIANT con l'aggiunta della descrizione del terreno, qui non trattato;
- studio per l'allocazione di informazioni geografiche 3D create con un software CAD/modellazione_solida, in un database relazionale per applicazioni geografiche gestito da PostgreSQL+PostGIS;
- studio per la realizzazione di un visualizzatore 3D in grado di permettere la consultazione grafica 3D e alfanumerica di informazioni geografiche di un predefinito modello, allocate in un file GML3 o in un database relazionale PostGIS.

Allocazioni di informazioni geografiche 3D in un database relazionale

La modellazione 3D e l'archiviazione dei dati in un Relational Database Management System con estensione spaziale, tramite software CAD e/o di modellazione solida è stata una delle fasi più complesse del progetto.

Sono stati esaminati diversi software Free e Open Source per la modellazione tridimensionale, con l'obiettivo principale di indagare sulla loro capacità nel costruire modelli tridimensionali, sulle possibilità di personalizzazione tramite un linguaggio di programmazione e sulla possibilità di trasferire i dati del modello tridimensionale insieme con gli attributi, che qualificano gli oggetti geometrici o complessi di oggetti geometrici come oggetti geografici, in un database geografico gestito da un RDBMS Open Source con estensione spaziale.

Per l'aspetto di modellazione del mondo reale sono stati in particolare oggetto di test i software K-3D (www.k-3d.org), Wings3d (www.wings3d.com), Blender (www.blender.org).

Dall'esame approfondito dei software sopra citati, è emerso che il più adatto per gli scopi prefissati era Blender. Ciò alla luce del fatto che Blender è un software di modellazione 3D Open Source molto potente, ben documentato, dotato di funzioni avanzate di modellazione, disponibile su molti sistemi operativi.

In Blender sono disponibili una vasta gamma di oggetti 3D (mesh poligonali, NURBS, curve di Bezier, B-spline, metaball,...).

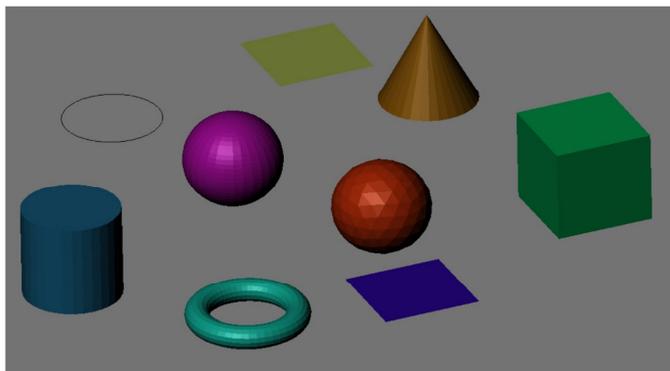


Figura 1 - Primitive in Blender

Una caratteristica fondamentale di Blender, è la possibilità di utilizzare script Python, dei veri e propri programmi, che funzionano come estensioni dello stesso Blender e permettono di effettuare numerose operazioni non incluse fra quelle standard. Una delle principali difficoltà nell'utilizzo di Blender, così come avviene per quasi tutti i software CAD, è la trasformazione delle primitive geometriche tipiche dei CAD in quelle dei tipiche dei GIS; Blender utilizza primitive di tipo "Mesh", "Curve" e "Surface". Per definire geometricamente delle entità geografiche è comunque possibile utilizzare primitive complesse del modellatore: è il caso ad esempio di una spline (o una NURBS) costituita da un unico punto o un paio di punti soltanto, per descrivere rispettivamente un punto o una linea. Quando poi ha luogo la scrittura sul database geografico, tramite gli script Python, gli oggetti o primitive geometriche Blender vengono archiviate nelle tabelle del database previa conversione nelle corrispondenti primitive GIS (nel caso specifico quelle aventi conformazione simile e compatibili in termini di archiviazione con il database geografico). L'estensione geografica del software di database relazionale permette infatti di allocare nelle tabelle e campi del DB alcuni tipi predefiniti di primitive geometriche atte a rappresentare gli oggetti territoriali.

Nei CAD o nei Software di modellazione solida inoltre, tranne qualche eccezione, ai diversi oggetti geometrici non sono attribuibili le informazioni sui sistemi di riferimento geografico; nel caso specifico, il Sistema di Riferimento Geografico può essere definito in Blender come attributo di default di ogni primitiva geometrica che rappresenta un oggetto geografico, quindi allocato nelle tabelle del DB con lo script python.

Interazione tra Blender e i geodatabase utilizzando script Python

I database consentono generalmente di gestire grosse moli di dati archiviati sulle memorie di massa con tecniche di accesso diretto alle informazioni, gestendo efficientemente i diversi accessi, con diversi utenti contemporaneamente, ognuno con le relative autorizzazioni. Il gestore di database carica in memoria solo la parte dei dati consultata al momento a differenza dei software CAD che caricano tutti i dati che fanno parte del progetto in memoria centrale. I software GIS, dei quali un componente fondamentale è il gestore di database, si comportano sostanzialmente come i gestori di database.

Il problema da risolvere era come trasferire i dati geometrici e gli attributi, definiti in Blender per caratterizzare gli oggetti geometrici come geografici, nelle tabelle del database geografico.

A tal fine, è stata condotta un'ampia ricerca bibliografica, da cui è emerso che, per quanto si è avuto modo di apprendere, a livello internazionale, l'allocazione di dati 3D era stata sperimentata precedentemente solo con il database server MySQL e solamente nell'ambito di due progetti: il primo denominato "Blenderpeople", il secondo un progetto beta, curato da Nathan Letwory sviluppatore di Blender.

Sulla base di quanto fatto precedentemente con MySQL nei due progetti di cui prima, si è sperimentata analoga tecnica di archiviazione dei dati con il database server Postgresql e PostGIS

(estensione per database geografici di PostgreSQL).

Per permettere l'interazione tra Blender e Postgresql son stati scritti dei moduli in Python, che allo stato attuale permettono di:

- esportare oggetti definiti con Blender all'interno del database PostGIS;
- leggere oggetti allocati nel database PostGIS
- far ridisegnare a Blender tali entità (punti, linee, poligoni) eseguendo la conversione tra oggetti geometrici la cui scrittura è ammessa dal gestore di database e i corrispondenti oggetti utilizzabili in Blender. In relazione a ciò, poichè Blender non gestisce direttamente punti e linee, i punti sono disegnati come una curva di Bezier con il solo punto di partenza, mentre le polilinee sono disegnate come una curva di Bezier con curvatura nulla tra i vertici.

L'archiviazione delle entità geometriche e degli attributi (ovvero degli oggetti geografici) nel database si avvale delle funzioni per convertire il blocco dati da scrivere nel database da WKT (*Well Known Text*) a WKB (*Well Known Binary*) e viceversa.

Una volta messo a punto il sistema di archiviazione si è passati a studiare le possibilità di archiviazione delle relazioni topologiche fra le primitive geometriche necessarie per costruire oggetti geometrici complessi quindi oggetti geografici. Questo al fine di permettere ad esempio l'archiviazione di edifici caratterizzati da oggetti 3D elementari che possono rappresentare componenti edilizi e parti di edifici, come pareti, parti di coperture, contorni a terra, aree di sedime, zone di un edificio anche aventi consistenza di subalterni catastali (nel modello Giant è prevista la possibilità di definire partizioni orizzontali di edifici collegabili a record contenenti informazioni catastali).



Figura 2 - Relazioni tra dati e software nel nostro progetto

In relazione alla scrittura di dati geometrici strutturati topologicamente sono stati in particolare sperimentati due modelli:

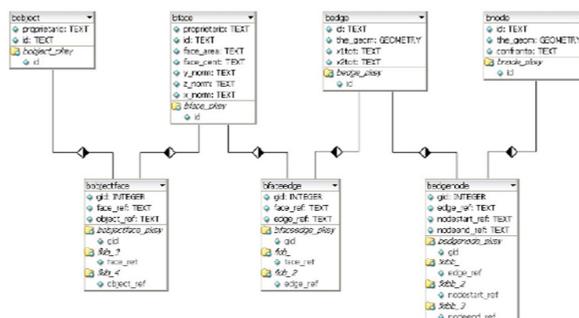


Figura 3 - Schema che utilizza elementi quad

- il primo nel quale le superfici (il modello GIANT è fortemente basato sulla costruzione di solidi tramite elementi a livello di complessità decrescente) hanno conformazione triangolare;
 - il secondo nel quale le superfici (ad esempio elementi parete di edifici) sono costituiti da elementi geometrici a conformazione quadrilatera (quad).
- Il modello a triangoli dopo la sua sperimentazione è stato abbandonato perché sostituibile con il successivo modello.

Il modello con elementi quadrilateri è un po' più complesso poiché permette di descrivere un solido come insieme di facce quadrilatere a loro volta costituiti da edge le quali sono ognuna definite tramite nodi (terminazioni degli edge). Tutto ciò viene posto in essere tramite quattro tabelle di relazione. E' questo un modello geometrico topologico aderente al modello GIANT. Le funzioni anche in questo caso non sono definite sotto l'aspetto dell'interfaccia, ma hanno assolto bene il loro compito ai fini della sperimentazione.

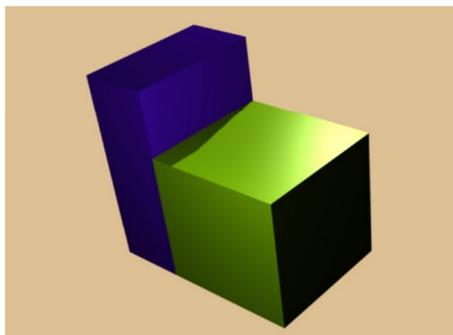


Figura 4 - Due edifici contigui

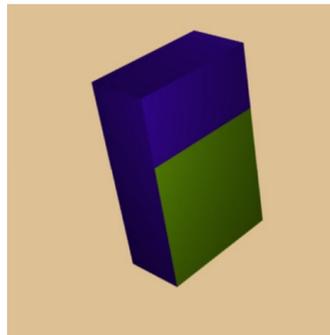


Figura 5 - L'edificio più alto con il muro in comune

Le funzioni sviluppate permettono sia di trasferire il modello dati da Blender verso il database geografico, che la procedura inversa cioè il caricamento in Blender del modello grafico archiviato nel database ai fini di una sua modifica. A tale scopo ogni oggetto archiviato è caratterizzato da un identificatore unico (UUID).

Il risultato di questa fase permette di affermare che:

- è possibile disegnare ambienti urbani 3D con software di modellazione solida come Blender;
- è possibile trasferire le informazioni geometriche e topologiche nonché le informazioni semantiche che caratterizzano oggetti geometrici come oggetti geografici in tabelle gestite tramite un database server relazionale dotato di estensione per la gestione di database geografici (cosiddette estensioni spaziali).

Realizzazione di un visualizzatore 3D in grado di visualizzare informazioni geografiche di un predefinito modello

Questa fase della ricerca aveva come scopo la creazione di un'applicazione client-server in grado di supportare richieste basate su un Web Feature Service 3D WFS (cioè simili al WFS OGC ma valide per un modello tridimensionale) e di un visualizzatore operante come applet funzionante su Internet browser, supportata dal WFS3D fornito dal server, in grado di permettere la consultazione delle informazioni strutturate secondo il modello GIS 3D GIANT ed archiviate nel database geografico.

Il modello *GIANT* supportato da uno *schema file* prevede sei classi o entità astratte che sono necessarie per ricostruire un ambiente 3D del mondo reale. Esse sono:

1. _Costruction.
2. _TransportationInfrastructure.
3. _Vegetation.
4. _Hydrographicentity.
5. _Reliefmap.
6. _NetworkService.

Come esempio è stata implementata solo la classe Costruction definita all'interno di GIANT.

A partire dall'entità Costruction si è creata una struttura PostGis per contenere la geometria relativa ad un'entità tenendo in considerazione le relazioni topologiche fornite da GML 3.1 .

Dal lato server si ha un DataBase software server basato su Postgresql+Postgis implementato secondo lo schema GIANT3D considerando tutti i vincoli e le relazioni fra le entità geografiche che compongono il modello stesso.

Inoltre vi è un application server che fornisce le operazioni indicate dalle specifiche WFS OGC cioè implementa il web service geografico 3D, estrae i dati dal database in base alle richieste ricevute e presenta i dati secondo lo schema GIANT3D.

Lato Client (browser internet) c'è un Applet Web che costituisce il visualizzatore - basato su X3D - in grado di connettersi ad un servizio WFS3D che restituisce i dati in GIANT3D. Di conseguenza i dati vengono trasformati in X3D e visualizzati nell'interfaccia grafica. Il WFS3D rende disponibili tre funzioni fra quelle previste dal WFS OGC classico, che sono:

1. GetCapabilities.
2. DescribeFeatureType.
3. GetObject.

GetCapabilities è la funzione che descrive le funzionalità del servizio. Indica le operazioni che possono essere eseguite e elenca i tipi di dati che sono supportati.

DescribeFeatureType è la funzione che descrive i tipi supportati dal servizio. Nel caso del WFS3D descrive i tipi previsti nel modello GIANT3D.

GetObject è equivalente a GetFeature del WFS classico con la differenza che in WFS3D vengono restituiti oggetti GIANT3D.

WFS3D è basato su tecnologia J2EE ed è implementato su Apache TOMCAT 6.0 su piattaforma J2EE_5. Il punto di accesso del servizio è una servlet, scritta in Java, che smista le varie richieste al componente che fornisce l'operazione o funzione richiesta.

Il servizio gestisce delle richieste HTTP, provenienti dal browser internet, secondo le specifiche OGC. Le richieste sono tipo GET e POST. Le richieste tipo GET adottano la modalità KVP (Key Value Pair) con la seguente sintassi name=value&.

L'implementazione è stata eseguita con Eclipse Galileo 3.5 Enterprise Edition. L'architettura del servizio è composta da cinque oggetti.

1. L'oggetto Service.
2. L'oggetto GetCapabilities.
3. L'oggetto DescribeFeatureType.
4. L'oggetto GetObject.
5. L'oggetto RequestUtil.

L'oggetto Service è la servlet che fornisce il servizio. Riceve tutte le richieste che gli arrivano dalla rete. Attraverso il RequestUtil individua l'operazione richiesta effettuando il parsing della richiesta e la inoltra all'oggetto corrispondente. Una volta ottenuta la risposta, la inoltra sulla connessione HTTP stabilita dalla richiesta.

Come indicato nei paragrafi precedenti il servizio lavora attualmente con 3 classi di oggetti geografici:

1. Building.
2. BuildingGroup.
3. Artefact.

GetObject costruisce la query ed attraverso driver JDBC si connette al Database PostGis. La prima query individua gli oggetti GIANT3D. Dopo aver individuato gli oggetti che soddisfano la richiesta si

esegono le query per la ricostruzione geometrica. In particolare si individuano i bounded dell'oggetto che sono:

1. Wall.
2. Roof.

Oltre ai Wall e Roof si individuano le parti orizzontali che sono i piani che costituiscono l'oggetto e vengono chiamati *horizontalPart* nel modello GIANT.

Per ogni oggetto vengono eseguite diverse query, che restituiscono i dati relativi a Wall, Roof e *HorizontalPart* del Building. Queste entità corrispondono alle facce di un solido e ognuna di essa fa riferimento a una riga *Bface*. Ogni riga *Bface* è una relazione topologica face delle specifiche GML. Come da specifiche GML ogni face è composta da Edge e ogni Edge è composta da node che sono costituiti da Point.

Per mezzo di query a cascata si ricavano quindi i punti che appartengono a un solido e tutte le relazioni topologiche che permettono di individuare gli edge e le face che compongono il solido. In questa fase viene ricostruito l'oggetto, verificando anche l'ordine di connessione dei vari edge assegnando loro un verso, in modo da avere i nodi correttamente connessi. Una volta individuati gli edge di una 'face' bisogna stabilire l'orientamento della 'face' per stabilire la parte interna ed esterna della faccia. Per ricavare il verso si applica la regola della mano destra considerando la normale alla faccia che viene estratta dal database PostGis.

A questo punto si sono ricavati tutti gli elementi per ricostruire una scena GIANT3D. Il GIANT come è stato detto in precedenza è un application schema basato su GML. Il GML 3.1 fa uso di XLink il quale fornisce gli strumenti necessari per linkare gli elementi di un documento XML evitando ripetizioni di dati già presenti.

Per concludere, la piattaforma del servizio può essere vista come composta dai seguenti componenti:

1. Una servlet su Apache Tomcat 6.0 che fornisce l'accesso http.
2. 4 moduli java che eseguono operazioni del servizio.
3. Driver JDBC a supporto della connessione PostGis che contiene le istanze GIANT3D.
4. API JDK GeoTool per l'analisi delle richieste.

Applet Viewer WFS3D

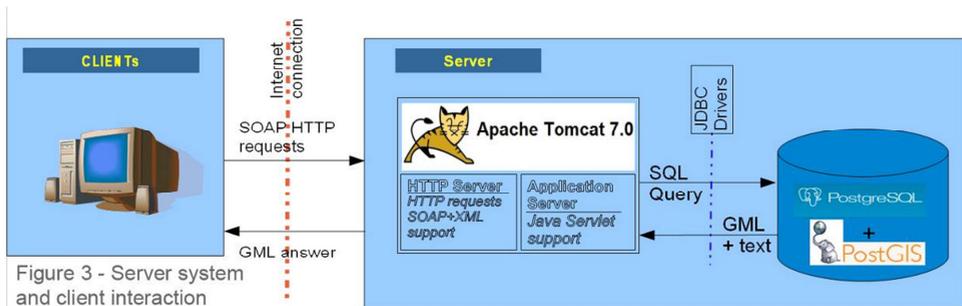


Figure 6 – Sistema di interazione client -Server

Utilizzando le librerie Java Xj3d è stata realizzata un'applet in grado di connettersi ad un WFS3D e visualizzare la scena 3D GIANT.

L'applet, con l'utilizzo del componente Swing Java JTabbedPane, è diviso in 3 schermate accessibili tramite delle linguette dell'interfaccia:

- Browser.
- GetObject WFS3D.
- Other Operations.

Nella schermata browser vengono visualizzati sia il modello 3D del mondo reale che i dati relativi (struttura del modello e attributi).

Costruito il filtro WHERE desiderato la risposta viene inviata. La risposta XML ottenuta, nel linguaggio GML3, è compatibile con GIANT. Affinché venga visualizzata dal browser X3D l'applet esegue una operazione di trasformazione da XML a X3D.

Il visualizzatore è dotato di tutti gli strumenti di navigazione di una scena 3D attivati dai pulsanti in basso nel visualizzatore. Le funzioni sono: zoom, panning, rotazione e ecc...

Nella schermata Browser ci sono tre aree. L'area di visualizzazione è posizionata nella parte destra. Nella parte sinistra ci sono due aree di testo e delle check box per l'attivazione di opzioni di visualizzazione della scena con le LineSet X3D (Wire Frame) o di Oggetti in trasparenza.

L'area di testo in alto a sinistra contiene un albero JTree che permette di analizzare la struttura della scena con i suoi relativi oggetti.

L'area di testo in basso permette di visualizzare gli attributi di ognuno degli oggetti geografici selezionati nell'area di testo superiore.

Il visualizzatore permette di analizzare gli oggetti verificando la strutturazione delle facce che lo compongono ed identificare parti interne all'oggetto.

La terza schermata Other Operations è utile per verificare il funzionamento del servizio WFS3D. In essa è possibile connettersi ad un servizio WFS3D per l'esecuzione delle operazioni GetCapabilities e DescribeFeatureType.

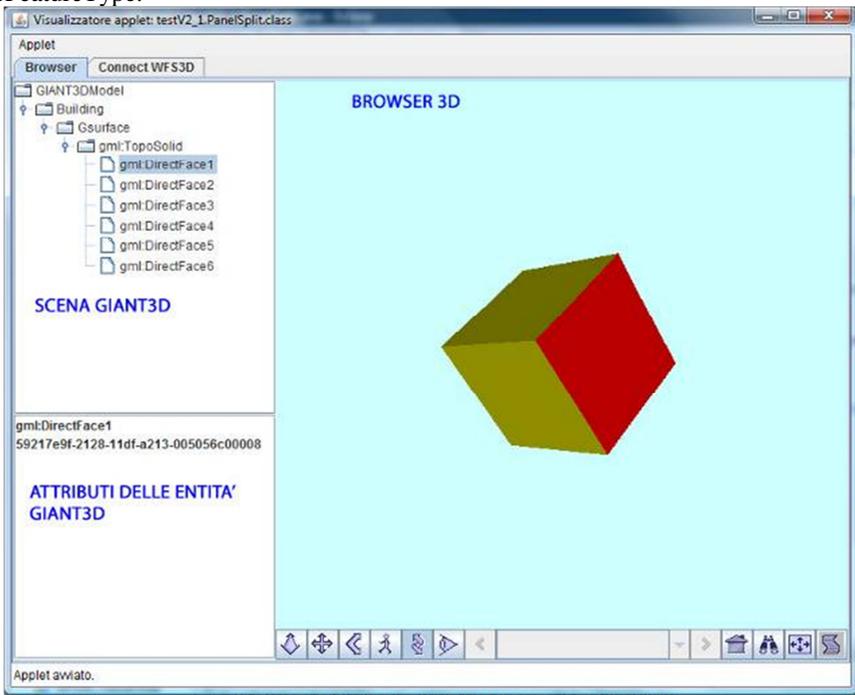


Figure 7 - Visualizzazione di un modello 3D semplificato

Conclusioni

Il progetto di ricerca illustrato dimostra che è possibile realizzare sistemi di visualizzazione GIS 3D basati su modelli cartografici, implementati secondo uno schema WSD compatibile con GML3 OGC, i quali possono anche interrogare la banca dati geografica implementata in accordo al modello cartografico e contenente anche delle relazioni topologiche sia fra gli oggetti geometrici che fra gli

oggetti geografici (scomposizione a diversi livelli e componenti).

Sono stati utilizzati, in accordo a quanto proposto nel progetto di ricerca, software liberi ed open source.

Ottimi risultati sono stati ottenuti utilizzando software libero ed *open source* per modellare e memorizzare i dati geografici. L'uso di Blender per costruire oggetti tridimensionali permette di sfruttare appieno questo potente software. D'altra parte, il database PostGIS ha molte caratteristiche ed è in grado di allocare e gestire correttamente grandi quantità di dati come quelli geografici.

Gli script di Python si comportano come connettore tra questi software, che permette una buona interazione bidirezionale.

Ulteriore passo previsto è lo sviluppo di una nuova parte di sistema tale da permettere l'archiviazione di dati provenienti da sensori posizionali relativi ad utenti presenti all'interno degli edifici e la loro visualizzazione dinamica all'interno del modello.

Riferimenti bibliografici

Arens, C., J. Stoter and P. van Oosterom, (2005). "Modelling 3-D spatial objects in a geo-DBMS using a 3D primitive". *Computer & Geosciences*, Volume 31,2, pp. 165-177.

Foley, J., van Dam, A., Feiner, S. & Hughes, J., (1995): *Computer Graphics: Principles and Practice*. Addison Wesley, 2nd Ed.

Khuan, C.T. and A. Abdul-Rahman, (2006). "0D feature in 3D planar polygon testing for 3D spatial analysis". *Geoinformation Science Journal*, Vol. 6, No. 1, pp 884-92.

Khuan, C.T., A. Abdul-Rahman and S. Zlatanova, (2008). "3D Solids and Their Management In DBMS". *Advances in 3D Geoinformation Systems*, Springer Berlin Heidelberg, pp. 279-311.

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

Scianna A., Ammoscato A., Corsale R. (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);

Scianna A., Ammoscato A. (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.

Scianna A. (2010). "Experimental studies for the definition of 3D geospatial web services". In: proceedings of webmgs 2010 - 1st International workshop on pervasive web mapping, geoprocessing and services su DVD rom. Como - Italy, 26-27 agosto 2010

Yanbing, W., W. Lixin, S. Wenzhong and L. Xiaomeng, (2006). "3D Integral Modeling for City Surface & Subsurface". *Innovations in 3D Geo Information Systems*, Springer Berlin Heidelberg, pp. 95-105.

Zlatanova, S., (2006). "3D geometries in spatial DBMS". *Innovations in 3D Geo Information Systems*, Springer Berlin Heidelberg, pp. 1-14.