

## SERVICES ORIENTED SMART CITY PLATFORM BASED ON 3D CITY MODEL VISUALIZATION.

F.Prandi, M.Soave, F. Devigili, M. Andreolli, R. De Amicis

Fondazione Graphitech, via alla Cascata, 56/C, 38123 Trento - Italy  
(federico.prandi, marco.soave, federico.devigili, michele.andreolli, raffaele.de.amicis)@graphitech.it

Commission IV/5

**KEY WORDS:** Geospatial Data Infrastructure, Web based, Online 3D/4D Visualization, Interoperability, Rendering, City

### ABSTRACT:

The rapid technological evolution, which is characterizing all the disciplines involved within the wide concept of smart cities, is becoming a key factor to trigger true user-driven innovation. However to fully develop the Smart City concept to a wide geographical target, it is required an infrastructure that allows the integration of heterogeneous geographical information and sensor networks into a common technological ground. In this context 3D city models will play an increasingly important role in our daily lives and become an essential part of the modern city information infrastructure (Spatial Data Infrastructure).

The work presented in this paper describes an innovative Services Oriented Architecture software platform aimed at providing smart-cities services on top of 3D urban models. 3D city models are the basis of many applications and can become the platform for integrating city information within the Smart-Cities context.

In particular the paper will investigate how the efficient visualisation of 3D city models using different levels of detail (LODs) is one of the pivotal technological challenge to support Smart-Cities applications. The goal is to provide to the final user realistic and abstract 3D representations of the urban environment and the possibility to interact with a massive amounts of semantic information contained into the geospatial 3D city model.

The proposed solution, using OCG standards and a custom service to provide 3D city models, lets the users to consume the services and interact with the 3D model via Web in a more effective way.

## 1. INTRODUCTION

### 1.1 Motivation

Recent studies have demonstrated as in the next future half of world's population will live in cities. Is therefore essential to develop efficient techniques to assist the management of modern cities.

Cities are 'systems of systems' and this could stand as the simplest definition for the term. The management of these complex objects requires a platform able to integrate city information from different resources. However, most of the current city and urban developments are based on vertical ICT solutions leading to an unsustainable sea of systems and market islands. The emerging trend is going towards a unified urban-scale ICT platform transforming a City into an open innovation platform called Smart-City.

3D city models are the basis of many applications and can become the platform for integrating city information within the Smart-Cities context.

3D city models have been widely used in various applications such as urban planning, traffic control, disaster management etc. and make to people easier to understand the spatial properties of urban objects, especially if compared with traditional 2D maps. In fact, these have a great capability of abstraction and synthesis but are hard to understand when the vertical information content increases.

The real world is in 3D and it is natural for the human brain to interpret 3D scenes; as a consequence 3D city models is becoming a hotter topic in both academia and industry.

The goal of this paper is to assess how 3D city models can be used as base for the deployment of smart-cities services acting as a platform for the integration, analysis and visualization of

information from different sources. In particular the work will investigate as the efficient visualisation of 3D city models in different levels of detail (LODs) is one of the pivotal technological challenge in the development of these applications.

Nowadays the fast network connection available on mobile devices makes it possible to access 3D city models anywhere (Coors and Zipf 2007). However, considering the data volume and complexity of 3D city models, in order to improve the visualisation efficiency, an optimisation of 3D city models is required.

### 1.2 Smart cities and digital earth

Geographic information technologies are of key importance for the deployment and implementation of ICTs in the Smart City because they give quick access to different layers of information that may be combined and integrated to facilitate analysis of a situation and make the best decisions (Daniel and Doran 2013). This applies in particular to digital cities where networks of data and computers represent networks of people and buildings.

EU project 'European Smart Cities'<sup>1</sup> provided one of the most well-known definition of Smart Cities where six dimension were identified (Smart Economy, Smart Mobility, Smart Governance, Smart Living, Smart People and Smart Environment).

To fully develop this Smart City concept at a wide geographical scope an infrastructure that allows the integration of

---

<sup>1</sup> Smart Cities, Ranking of European medium-sized cities,  
<http://www.smart-cities.eu/model.html>  
<http://www.smart-cities.eu/model.html>

heterogeneous and geographically disperse information and sensor networks into a common technological ground is required.

The digital earth paradigm is one requisite for establishing a Smart City, given the pervasiveness of geospatial information, in the management tools and services provided to the citizens and decision makers.

Goodchild (2008) explored the use case for Digital Earth concluding that this has become a very successful case because Geobrowsers, with their emphasis on visualisation and ease of use appeal to human's spatial thinking, 'is one of the fundamental forms of intelligence needed to function in modern society', including Smart Cities. To be more effective, at the urban scale level, Geobrowsers should evolve into platforms where physical components such as buildings, roads and open spaces are accurately described at each scale from single rooms or apartments, to rural villages, towns or mega cities. The widespread use of Lidar and digital photogrammetry has enabled the acquisition of a global layer in 3D. However, to transform this huge amount of data into useful information and support smart-cities development, it is needed to structure 3d data into a well-defined data model and to provide tools and services to exploit the potentiality of 3D visualization.

The main requirements for smart-cities 3D city geo-visualization systems are:

- *Interoperability*: increase the effectiveness and efficiency of the integration on different levels. Interoperability can be improved by applying standards.
- Provide *effective and high quality visual representations*: the effectiveness of a geo-visualization system can be facilitated by advanced, complex, innovative visualization algorithms (Döllner 2005).

The work presented in this paper tries to satisfy these two requirements developing a 3D web-based visualization system based on OGC standards and effective visualization concepts. The architectural framework, based on Services Oriented Architecture (SOA) concepts, provides the 3D city data to a web client designed to support the view process in a very effective way.

The goal is to provide to the final user 3D, realistic and abstract, representations of the urban environment together with the capabilities to easily interact with the massive amounts of semantic information contained into 3D geospatial city model.

### 1.3 Related works

Efficient visualisation of 3D city models is essential to support applications in mobile and desktop environments. In recent years many frameworks have been proposed for this purpose (Bo 2011). These frameworks are traditionally subdivided into three categories characterised by the power required from the client hardware: thick client-server model, thin client-server model and Peer to Peer (P2P) model.

Using VRML format Zhou et al. (2006) created a backend server to generate 3D models, and sent them back to the users. Lerma and Garcia (2004), create VRML plugin to visualise 3D models of the historical centres.

These formats has been used even into the geographical domain to visualize 3D geographical information including city Models. Shinozaki (2008) uses KML for virtual 3D urban design in Tokyo and Fukuoka. Google Earth KML is very popular and, for this reason, it has become OGC standard for expressing geographic annotation and visualisation. However, Google Earth is commercial software and developments based on it are

not possible for several purposes (e.g. it is impossible to manage formats different from KML).

Rodriguez et al. (2013) describes the implementation of a Web3D GIS based on the conversion of CityGML into X3D models and its visualization through the web. Following a very similar approach and implementing the W3DS standard (Reitz et. al 2009) and Prieto and Izkara developed a visualization system for 3D city models. These solutions are very interesting but only focused on visualization, lacking so of advanced analysis and processing features. Furthermore the visualization performances are quite limited.

NASA World Wind (NWW) is an open source virtual globe. It has been used as thick client to deliver 3D content and to provide interoperable access to geographical information together with geospatial processing services (Simões et. al 2011) and, thanks to the flexibility and extensibility of the code architecture, it can be extended to support the visualization of 3D city models (Soave et. al 2013). The main drawbacks of these approaches are related to the great power amount required by the client for parsing, rendering and visualise the model. Another interesting development based on NWW is Gaea+ project (2014) even if at the moment the 3D visualization is limited to the extruded objects.

This paper presents a new SOA approach able to render the 3D City Model using a very efficient 3D data format and performance-oriented rendering techniques. The goal of proposed approach is to improve the rendering performances of the NWW based web-client.

The second pillar of this work is to ensure interoperability and easy management of 3D city Models including a seamless access to semantic information of the model itself. In recent years many applications are suitable for visualization purposes over the web (e.g. Apple/C3, GoogleMaps) but these systems, impressive in terms of visualization, are based on the generation of reliable meshes that are a decimated version of the original point cloud (Kuschik, 2013). This approach make impossible to consider city objects as single independent features and is not fully suitable to provide semantic information and additional services. In our work semantic representation of a virtual 3D city model is ensured by the transformation of raw data into a model compliant with the CityGML standard (Stadler et al. 2008).

## 2. 3D SMART CITY PLATFORM

### 2.1 City Model and Smart-city services

3D city models will become an essential part of the modern city information infrastructure (Spatial Data Infrastructure) and, like 2D cartographic maps, they can be used to integrate various data from different sources for public accessible visualisation and many other applications.

The implementation of 3D city models is main part of our research: it is clear as many applications such as urban planning, facility management and personal navigation require semantic information about the city objects besides the geometry models.

CityGML is an open data model and XML-based standard for the representation of 3D urban objects. It is an OGC standard realised for the storage and exchange of virtual 3D city models and covers broad thematic fields of city objects: geometrical and topological aspects can be accurately described and linked with their semantic part.

CityGML supports five Levels of Details (LOD) to reflect different data collection processes supporting different application requirements and facilitating efficient visualisation

and data analysis (Figure 1). LOD0 is a 2.5D Digital Terrain Model. LOD1 is the prismatic model of buildings with flat roofs. LOD2 contains differentiated roof structures and surfaces. LOD3 contains models with detailed wall and roof structures. LOD4 adds to LOD3 model the interior structures. Therefore, objects, which are not yet explicitly modelled in the current version of CityGML, can be represented using the concept of generic objects and attributes. In addition, extensions of the CityGML data model within a specific application domain can be implemented using the Application Domain Extensions (ADE). The final purpose was to deliver an open platform, based on interoperable 3D CityGML UIMs, on top of which is possible to deploy various ‘smart city’ services.

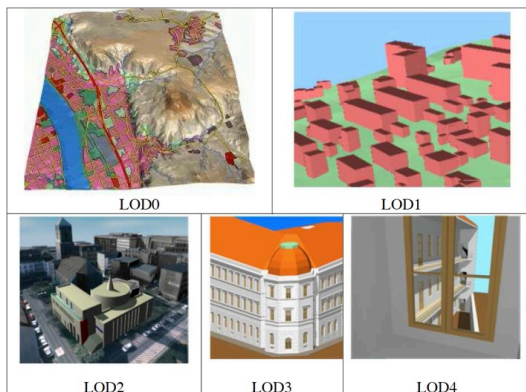


Figure 1 CityGML LODs as defined in OGC specifications.

Computer technologies create the underlying platform to support 3D smart cities applications.

The study on 3D city models focus on these following main aspects.

**2.1.1 3D City Model Generation:** although there are a lot of tools to support the process, they still not satisfy the needs to fully automatically model large city area. Is a fact that most of current 3D city models are created manually (Moreira et al. 2013).

The first level of our platform is a server-side wrapping software that generates 3D buildings models in LoD1 and LoD2, starting from Digital Terrain Model (DTM), Digital Surface Model (DSM) and building footprint (2D). The DSM data must have a resolution of at least 1 m but using 2 m resolute data we can still generate buildings model at LoD1. A good detection of the roofs composition is achieved having a minimum of 6 dots per roof area. As a consequence to recognize a gambled roof a minimum of 12 is necessary meanwhile for a hipped roof a minimum of 24 points are needed.

The service is asynchronous and, once finished the generation, the user can validate the generated model both downloading and visualizing it or through a report of the generated model quality. The report is based on the number of the different LoD reconstructed buildings, and highlights the number of LoD1 building identified in this way due to missing information.

Results are strongly influenced by topographic maps and DSM quality and resolution. The scale of the topographic/cadastral datasets and the number of represented details can influence the quality of the 3D reconstruction.

**2.1.2 3D City Model Management and extension:** the City Model generated using our web service is composed of buildings solid and shell, according to standard LoD specifications.

To support additional smart city services these models have to move a step towards: the information model must contain

semantic information about city objects besides the geometry models.

In order to achieve this, two different challenges has been faced off. The first is technological: the system requirements include the capability of the platform to store and populate the semantic information related to the city objects. Oracle and PostgreSQL databases have been used to store and manage the data.

Furthermore, with the aim to enrich the cityGML model with additional information (i.e. the averaged solar irradiation value for each roof surface), a set of back-office functionalities has been defined. These functionalities, which can have a WPS interface, take as input the output of other process or values contained in external tables populating the cityGML tables with enriched information.

The definition of these cityGML tables is the second challenge, which is mainly methodological. Indeed, it is know that extensions of the cityGML data model, applying to specific application fields, can be implemented using the ADEs or using generic objects/attributes.

The first result of our projects was to provide a significant contribution to the standards in the domain of smart city services, thus having a wider adoption of CityGML as open standards for 3D smart city services (Prandi et al. 2013).

**2.1.3 Smart City Services:** with this term we will identify the specific application services based on information contained on the enriched 3D City Model.

Spatial Data Infrastructures (SDI) are integrated, multi-levelled hierarchy of interconnected repository of geographical data. Those are based on partnerships at corporate, local, state, provincial, national or multi-National and global levels.

In this context SDI and spatial technologies are now used routinely in decision making to support city planning and forecasting. At the same time they are useful also to address some of the world’s most pressing societal problems. Furthermore, in the new era of Location Base Services (LBS), localized information are often described by Points of Interest (POI).

However, in the context of cities, the lack in an efficient and effective three-dimensional viewer solution limits the ability exploit the full potential of 3D city models.

For these reasons our goal is to develop a platform where, thanks to an efficient visualization of 3D city context, it will be possible to consume spatial and not-spatial services in more effective way. For instance citizens could take advantage not only from existing SDI but also from 3D city model while creating and sharing spatial knowledge, as sensors in their own right. Furthermore, mobile 3D smart city platform with augmented reality visualization will constitute a powerful and immersive tool to gain a better understanding and knowledge of surrounding environment on site and, thus, make more informed decisions.

## 2.2 System architecture

In this section we present the overall architecture of the 3D smart city platform, which is fundamental a SOA composed by three levels: i) the data level, ii) the middleware and iii) the application level. The architectural framework takes in account the SOA concepts, the 3D Geovisualization requirements and OGC standards compliance and it is shown in figure 2.

The data layer stores the geographical information that have to be processed or visualized by the platform. The Raw Data and the 3D City Model mainly compose them.

With the term “Raw Data” we will indicate both the raw information used as input for the cityGML model generation, both the whole of the geographical and not geographical information needed to provide the smart services functionalities.

For instance Digital Terrain Model (DTM) is a key feature for the cityGML generation but is also a dataset provided as WMS services for the terrain visualization. In the same way Digital Surface Model (DSM) is used in the city model generation but is even a fundamental input for the solar irradiation map calculation.

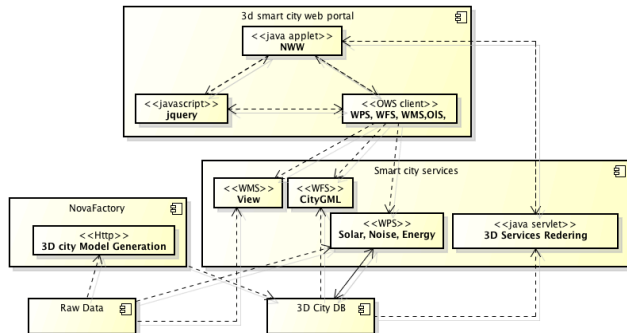


Figure 2 Service-oriented architecture of the smart city platform

All these “Raw Data” are ingested on the smart city platform Databases to be provided to the final users as services or to be used at the middleware level in order to enrich the 3D city models (figure 3).

The 3D city models are stored and managed using cityGML standard into databases such as Oracle and PostgreSQL. The database schema used is the 3DCityDB, a free and Open Source 3D geo database to store, represent, and manage virtual 3D city models developed by Institute for Geodesy and Geoinformation Science of the Berlin University of Technology.

The middleware is the logical layer where the information are processed, combined and exposed to the application layer. It is mainly composed by 4 components, which can interact together:

- A custom Java based component (in i-SCOPE project we have deployed a version of NovaFactory<sup>2</sup> software) dedicated to the database management and the back-end process triggering.
- A map server, providing WMS, WFS (Geoserver) and WPS services (Deegree).
- A Java custom servlet able to retrieve the city model information based on the client request and provide the optimized rendered objects.
- Several external software components needed to perform the required smart services functionalities. These parts are not present in figure 2 because they can change according to the specific purpose.

The custom application, is mainly based on the citygml4j libraries to reading, parsing and exporting of CityGML files and it is, from the user point a view, a black box application. The aim of this component is to import and export data from the 3DCityDB and run the processes needed to elaborate and enrich the data.

The map server contains services that manage and grant access to Geodata such as the Web Map Service (WMS) and Web Feature Service (WFS). The WFS will ensure the access to the semantic information contained on the 3D city Model. Web Processing Services (WPS) can encapsulate additional functionalities that are required for a specific service process in a generic way. The last middleware component, which will be

described in the next section, is devoted to provide the 3D portrayal service needed for the effective visualization of the 3D model in the web client. It is based on a Java servlet and is able to retrieve on the database the portion of City Model requested by the client, sending it in a specific optimized format.

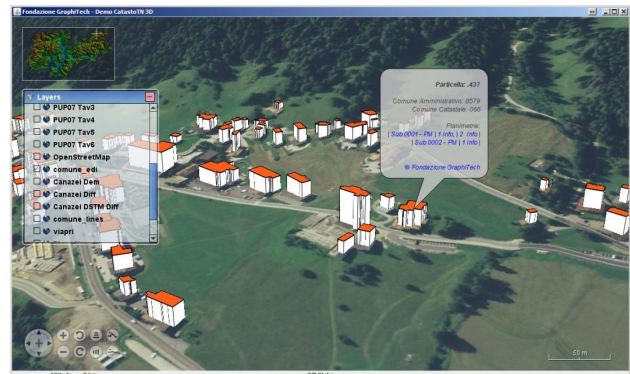


Figure 3. 3D city model enriched with the cadastral information, clicking on the specific building it is possible to retrieve all cadastral information of the contained flats.

Finally, at the application level, there is the web client. It is based on Java web applet and JavaScript technology. It allows to consume the OGC standard web services and to visualize the city models object in a efficient way.

An example of the web client application is shown in figure 4

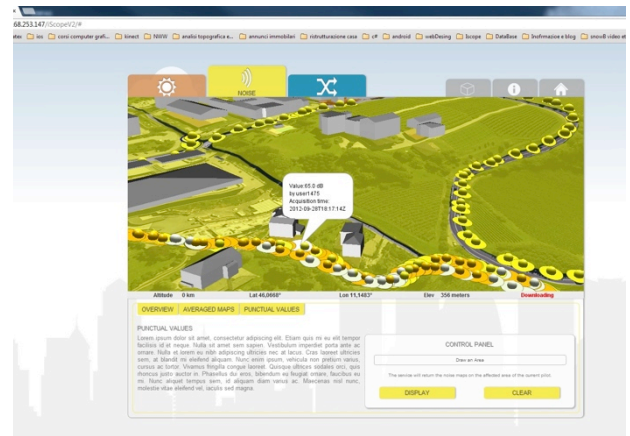


Figure 4: The web client application. The 3D buildings are shown together with a service providing the noise measured by the citizen through mobile phone. The panel on the web page is used to perform the requests and visualise the retrieved informations.

The architecture supports two user profiles and the relative tasks: Administrator and sample user.

An Administrator uses the admin section of Web-Client for content management. The admin-platform has the capabilities of importing raw geodata into the system and pre-processing it generating a custom format ready to be rendered. The Content Management Services Process is implemented using the Java-based WPS provided by Deegree and Geoserver.

The “Simple-User” 3D Web-Client can be used to interactively explore the virtual 3D city model, to access the sematic information and to run process on available information (i.e assessment on Photovoltaic power of roofs, routing services etc.).

<sup>2</sup> <http://www.moss.de/english/products/novafactory/modules-overview/>

### 3. 3D VISUALIZATION SYSTEM

In this section will be described an approach for the service-based 3D model provisioning and interaction system. We will focus on a specific part of the architecture presented in Section 2.2: the 3D rendering service and the 3D Web-Client. Streaming three-dimensional geographic data is a relatively new topic and no standard has yet been defined. OCG proposed three different services able to stream three-dimensional data to a client. One classification can be done considering differences on the amount of computational power required by the client and the server (fig. 5).

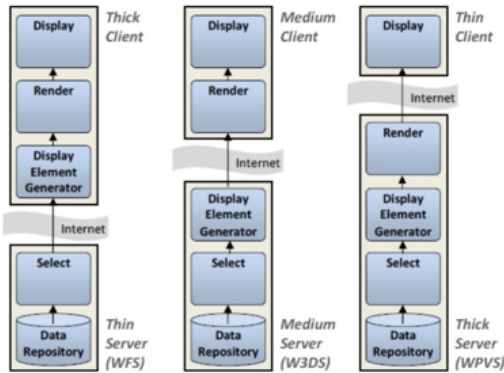


Figure 5 WFS, W3DS and WPVS (deprecated) compared. Source: Draft for Candidate OpenGIS® Web 3D Service Interface Standard. OGC document 09-104r1.

The exposed approach is the subject of our on-going research and we present preliminary results of our prototypical implementation.

#### 3.1 3D Service Rendering

CityGML is designed to represent 3D city models, but not to present or visualise 3D city models directly. In order to overcome this lack OGC proposes two different 3D portrayal services to provide 3D geographic information. One generating 2D images of projected views namely 3DVS, the second W3DS returns tree like structure (scene graph) of nodes, groups, transforms, shapes, materials, and geometries. Our approach follows the W3DS paradigm, however instead to send the request results using the predefined format (W3DS shall support X3D as mandatory format but even KML and COLLADA are allowed) we use a custom graphic optimized data structure. Indeed generally, standard data models and encodings (such as CityGML, X3D, KML or COLLADA) cannot be used as efficient input for specific 3D rendering solutions. Using CityGML files directly has proven to be too heavy on the client. Parsing and tessellation of the geometries, when dataset is complex, can be a very long process even on desktop computers; on mobile this is practically unfeasible. A specific Java servlet must comprise parsing methods capable of retrieve information from CityGML and recode it to our rendering friendly data structure. The format we developed is a tree-like recursive data structure aimed to describe the more complex XML based standards. The format is optimized to minimize redundant information and maximize parsing speed. CityGML geometries are not only converted in the format but tessellated and transformed into GPU-friendly, optimized, triangle based geometries. The syntax is similar to the OBJ format but also supports hierarchies, multiple textures coordinate channels, custom vertex attributes, optimized materials, level of detail, and additional CityGML

metadata defined both for files and objects. Using this optimized data structure is important when bandwidth of the connection to the server is limited and to reduce CPU and memory load when parsing and preparing data for the GPU.

Parameter	Values
CPU	28GHz Intel i7
GPU	512 MB
RAM	8GB 1333Mhz
Frame Rate	60fps
# of objects	15000
LoD	2

Table 1 Performance of the rendering in functions of Hardware characteristics

The servlet does not only provide re-parsing of CityGML files but also works as a cache server for the prepared tiles. Once a request has been made and a tile is sent to the client the response gets cached for subsequent requests. The servlet provides an OGC-like web service and can be easily modified to access different kind of datasets like 3D geo-database and any other custom 3D data format. These files are defined on a common tile grid well knows both on server and client side and generated on the fly on the server according to the client requests.

The semantic information related to the city objects will be retrieved remotely on-demand by accessing the WFS serving CityGML data.

#### 3.2 Web based 3D client

The application layer of the system is constituted by a web-based application that allows the visualization of the city Model and the interaction with the different smart-services. The geographic visualization section of the client has been developed on top of NWW Java SDK. In particular it is a Java applet embedded on the web page.

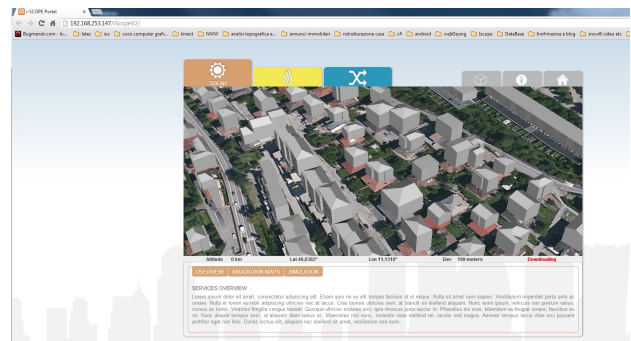


Figure 6 The web client application showing the LoD2 city Model of buildings superimposed to the Digital Terrain model.

Thanks to this approach many useful features can be implemented in parallel: different services like the WMS, WFS and the proposed approach can run concurrently allowing a great flexibility of the entire system.

The city model data are provided by the abovementioned 3D Service Rendering, instead DTM is provided as WMS services in form of height map and it is possible to set-up the resolution level according to the resolution of the original dataset. This module is able to render the 3D City models data in a very efficient way delivering constant high frame rates even with big 3D datasets. The city Model visualization system in the applet is tile based in order to optimize the communication between the server and the client and to improve the caching system.

All the smart service functionalities are provided as WFS interfaces while the query on the semantic part of cityGML is ensured via WFS service. Both these OGC services are deployed using Deegree open source platform. The interaction between the 3D graphical content and the web page functionalities is implemented using JQuery a JavaScript open library.

#### 4. CONCLUSION

In this paper we have presented a framework based on a SOA to provide smart-cities services on top of 3D urban city models.

3D city models became in this way the basis to geographically access to the services and the pillar for integrating city information within the Smart-Cities context.

The proposed solution, using OGC standards and a custom efficient service to provide 3D city models, lets the users to consume the services and interact with the 3D model via Web in a more effective way.

The first tests on the graphical performance have give very good results in terms of rendering performance and number of features represented. Furthermore the framework allows interacting with the data repository through web services providing processing functionalities, which allow creation, export and import of new features.

The software architecture is scalable and interoperable providing different services functionalities provided using WFS interface. Furthermore the information contained on the base CityGML model could be retrieved using the WFS interface to query the CityDB.

The open issues and the future works will be related to improve the accessibility on the semantic part of the city model at the moment limited to the CityGML base information and to increase the interaction with the management part of the model. Furthermore the possibility to insert our custom 3D service rendering on the OGC 3D portrayal initiative in order to fulfil the standard compliance will be investigated. These items have to be solved to obtain an effective and interoperable platform to provide smart-cities services into a 3D city model environment, however the good results obtained demonstrate that combining technologies for 3D visualization and open standards it is possible to deploy a smart-city spatial data infrastructure.

#### 5. ACKNOWLEDGMENTS

The projects I-SCOPE and SUNSHINE have received funding from the EC, and it has been co-funded by the CIP-Pilot actions as part of the Competitiveness and innovation Framework Programme. The author is solely responsible of this work, which does not represent the opinion of the EC. The EC is not responsible for any use that might be made of information contained in this paper.

##### 5.1 References

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

Bo, M., 2011. Visualisation and Generalisation of 3D City Models. Doctoral Thesis. Royal Institute of Technology (KTH) Department of Urban Planning and Environment Division of Geodesy and Geoinformatics, SE-100 44 Stockholm Sweden. ISBN 978-91-7501-189-9.

Daniel, S., and Doran M., 2013. geoSmartCity: geomatics contribution to the smart city. In Proceedings of the 14th

Annual International Conference on Digital Government Research (dg.o '13). ACM, New York, NY, USA, 65-71.

Döllner J., 2005. Geovisualization and Real-Time 3D Computer Graphics, In International Cartographic Association, edited by Jason Dykes, Alan M. MacEachren and Menno-Jan Kraak, Elsevier, Oxford, 2005, Pages 325-343, Exploring Geovisualization.

Gaea+ Open source.

<http://www.gaeaplus.eu/en/open-source>

Kuschik, G., 2013. Large scale urban reconstruction from remote sensing imagery. In: IAPRS&SIS, Vol.XL-5/W1, 2013.

Lerma J., Vidal J., Portalés C., 2004. Three-Dimensional city model visualisation for real-time guided museum tours, The Photogrammetric Record, Volume 19, Issue 108, pages 360-374.

Macay Moreira J.M., Nex F., Agugiaro G., Remondino F., Lim N.J., 2013. From DSM to 3D building models: a quantitative evaluation. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XL-1/W1, ISPRS Hannover Workshop 2013.

Prandi, F., De Amicis, R., Piffer, S., Soave, M., Cadzow, S., Gonzalez Boix, E., D'Hondt, E., 2013. Using CityGML to deploy Smart-City Services for urban ecosystems. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XL-4/W1, 29th Urban Data Management Symposium, London, United Kingdom

Prieto I., Izkara J. L., (2012). Visualization of 3D city models on mobile devices. In Proceedings of the 17th International Conference on 3D Web Technology (Web3D '12). ACM, New York, NY, USA, 101-104. DOI=10.1145/2338714.2338731 <http://doi.acm.org/10.1145/2338714.2338731>

Reitz T., Kramer M., Thum S., (2009): A processing pipeline for X3D earthbased spatial data view services. Proc. 14th International Conference on 3D Web Technology, Darmstadt, Germany, 16-17 June, pp. 137-145.

Rodrigues J. I., Figueiredo M., Da Costa C., (2013). Web3D GIS for City Models with CityGML and X3D. 17th International Conference on Information Visualisation, 2013 Pages: 384-388, DOI: 10.1109/IV.2013.102

Simões, B., Conti, G., Amicis, R. D., 2011. "User-friendly interfaces for web gis," World Congress in Computer Science, Computer Engineering, and Applied Computing, July 2011.

Soave, M., Devigili, F., Prandi, F., De Amicis, R., 2013. Visualization and analysis of CityGML dataset within a client sever infrastructure. Proceedings of the 18th International Conference on 3D Web Technology. ACM, New York, NY, USA, 215-215. DOI=10.1145/2466533.2466573.

Stadler A., Nagel C., König G., and Kolbe T. H., 2008. Making interoperability persistent: A 3D geo database based on CityGML. In J. Lee and S. Zlatanova, editors, Proceedings of the 3rd Intl. Workshop on 3D Geo-Information, Lecture Notes in Geoinformation & Cartography. Springer, 2008.