Integration of CityGML and Air Quality Spatio-Temporal Data Series via OGC SOS

Wanji Zhu
Karlsruhe Institute of Technology European Institute for Energy Research
Email: Wanji.Zhu@eifer.uni-karlsruhe.de

Sven Wursthorn
Karlsruhe Institute of Technology
Email: Sven.Wursthorn@kit.edu

Alexander Simons
European Institute for Energy Research
Email: Alexander.Simons@eifer.org

Alexandru Nichersu
European Institute for Energy
Email: Alexandru.Nichersu@eifer.org Research

***Abstract*—Cities have always been considered a horsepower of industrial growth over centuries. However, urbanization often has environmental effects. In this paper we will focus on one of them, namely air pollution, and how we can integrate 3D city models and air quality sensor data. In most cases, air quality data is acquired from static sensor systems like weather stations, which lack the coverage required for urban simulation. Applying mobile data for air quality models is more reasonable due to the spatial coverage of the urban model. Yet academic research has been scarce on the integration of city models and spatio-temporal varying sensor data. To close the gap, this paper proposes a novel approach to integrate city model data and spatio-temporal data in an interoperable and standardized way. The geometry aspects of a city are described by those defined in CityGML and the air spatio-temporal varying data is retrieved by OGC Sensor Observation Service. The approach uses Cesium JS, a JavaScript library for visualizing the semantical city model stored in CityGML and the OGC SOS observation service from 52°North for the air quality data transmission.**

***Keywords*—CityGML; SOS; Spatio-Temporal; Varying Data; Air Quality**

I. Introduction

Cities have to navigate around different concepts for urban planning to solve problems they are facing today and tomorrow. The smart city vision involves the integration of solutions for information and communication technologies to help in providing municipal services, such as. sanitation, water supply, street network, public libraries, schools, food inspection, fire department, police, ambulance, other health department issues and transportation. The integration of classical municipal services and Web services is a big challenge for many players worldwide, e.g. IT services providers, urban systems operators, energy providers. Open interoperable standards and the use of geo-spatial information provides an environment filled with opportunities for cities as they empower them to use their socioeconomical data for different applications in the domains of analysis and modeling, visualization, semantics and metadata. An important component of a city's success is the perceived quality of life. Citizens and companies expect to have better access to services, health facilities and consumption goods. These are often associated to the notion of common wealth. The environmental effects are often associated with urbanization and affect the quality of life. Air pollution is one of the most important side effects of urbanization. In Europe, 3D city models have become readily available and it is only the next logical step to link them to air quality sensor data.

A. Serving CityGML via WFS

1) Concept of CityGML: The City Geography Markup Language (CityGML) is a comprehensive concept for the modeling and exchange of 3D city and landscape models. As an open data model, CityGML defines full-blown classes and relations describing 3D objects regarding their geometry, topology, semantics and appearance properties. This allows the employment of virtual 3D city models for complicated analytical tasks in different domains such as simulations, urban data mining and facility management [1].

CityGML defines five consecutive Levels of Detail (LOD), from terrain alone to architectural models (outside and interior). With increasing LOD, objects become more detailed both at the geometric and thematic level [2]. In addition to their attributes, objects in CityGML can have external references to corresponding objects in other databases or data sets. CityGML also provides an extension mechanism with the Application Domain Extension (ADE) in order to enrich data for specific domain areas. That is where different kinds of models, such as environmental noise models, energy models and air quality models, can come into play. The most important development for using CityGML and energy related data together has been built through the ADE mechanism and is called the EnergyADE [3].

The objective of the EnergyADE is the storing and managing of data needed for simulating complex urban energy models, which consider existing international building and energy data specifications (INSPIRE, BEDES). An important feature of this ADE is the possibility to handle different data qualities regarding level of details and complexity of energy models (E.g. monthly energy balance of ISO 13790,

sub-hourly dynamic simulation of software such as CitySim or EnergyPlus) [4].

CityGML is released and implemented as a GML application schema for the Geography Markup Language version 3.1.1, issued by the Open Geospatial Consortium (OGC) and the ISO TC211 [5]. This allows CityGML to be used with the whole family of GML compatible OGC Web Services for data accessing, processing and cataloging like Web Feature Service (WFS), Web Processing Service (WPS) and the Catalog Service [6].

2) Advantages of a WFS: Since CityGML is quickly being adopted on an international level, serving and retrieving CityGML data over the web becomes more appealing than any time before. This is also a focus of this paper. Using traditional communication techniques, such as File Transfer Protocol (FTP), to share CityGML data led to some underlying problems. The major one is that CityGML data size can become rather large even for a small geographical region, when the CityGML model is at a high level of detail. This causes issues for users or clients uploading and downloading data. To solve this problem, the Web Feature Service (WFS) Standard, first published with version 1.0.0 in 2002, comes into play. The key characteristic of the WFS is to allow users to share geographic information at fine-grained feature and feature property level rather than at the file level. That completely changes the way in which geographic information is created, modified and exchanged via the Internet [7]. There are multiple advantages to serving CityGML via a WFS. One of the most important advantages is that the WFS is a standardized interface. Rather than sharing the whole dataset of CityGML data, which is stored on a relational geodatabase using the 3DCityDb structure [8], data can be exchanged based on specific feature types, such as buildings, transportation or others defined by the CityGML public schema. This allows CityGML data to be queried by using various filters depending on implementation of conformance of the WFS. For example, a user could request all buildings based on a given Bounding Box by using a WFS instance, which supports the spatial filter query.

3) Obstacles of Implementation: As depicted above, serving CityGML via a WFS improves the performance of exchanging data. However, a number of technical challenges arise from the characteristics of the CityGML model. In the first place, the CityGML model makes extensive use of complex data types for properties and nesting of features within feature collections. This, in consequence, causes CityGML to have a deep nested data structure. In addition, the geometry types supported in relational databases are often more limited than those used in CityGML, which could be a restriction for those WFS implemented on top of a relational database [9].

Despite the technical challenge mentioned before, at least two commercial solutions have appeared to cover the existing market demand. The Project 3D City DB, which provides a 3D geodatabase based on CityGML, released an implementation for the 2.0 version of the OGC Web Feature Service. However, this satisfies only the simple WFS conformance class and the advanced version including more WFS operations is commercially available from the company virtualcity SYSTEMS GmbH Berlin [8]. Another example is the WFS integrated with CityGML offered by Snowflake. It supports both the OGC WFS 1.0 and 1.1 specifications [1]. These products satisfied the need of the market to some extent, but open source projects and free products with full functionality for users in research and development are currently limited.

One of the goals of this paper is to search for an open source solution to serve CityGML via a WFS with advanced functionality. Thus, to achieve this objective, all software used should be open source, and, as such, an open mechanism to provide CityGML data is needed. One of those solutions tested for this paper was GeoServer in combination with its “Application Schema” extension. The main reason for choosing GeoServer to serve CityGML is that it has implemented the OGC WFS standard. This is considered a reference implementation of the WFS and provides full-fledged WFS functionality including discovery, query, locking, transaction and stored query operations [10]. In addition, its “Application Schema” extension allows users to define complex feature types by “Feature Chaining” [11], in which feature types are configured independently, with relationships specified in the mapping file [12]. This implies that using GeoServer with the Application Schema satisfies not only the complexity of feature types defined in the CityGML model, but also provides WFS operations, as well as various query filters.

The GeoServer Application Schema is made up of two concepts called “Feature Mapping” and “Feature Chaining”. “Feature Mapping” defines the relationships between a source feature type (in most cases tables in a relational database) and a target feature type, which is declared in a public GML application schema. “Feature Chaining” helps to construct a complex feature type, in which simple features can be structured in a hierarchical way. These two concepts allow the Application Schema to map complex feature types defined in CityGML. However, there are some restrictions to how this works. The most important one is that all public GML application schemas used for mapping with the GeoServer Application Schema must satisfy the GML encoding rule, known as the “Striping” rule. The “Striping” rule requires that a complex type is never the direct property of another complex type. The complex types are always contained in a property type to ensure that their specific type is encoded in a surrounding element or container element. In other words, the GeoServer Application Schema supports those schemas, which enforce all the GML encoding rules. In terms of CityGML, a couple of schemas are involved to describe feature types of city objects. However, not all schemas obey the GML encoding rules. One of them is the generic.xsd, which defines the generic attributes of city objects. It is pivotal to storing additional information about all kinds of models, such as energy models and heat models. This is the main reason why GeoServer with Application Schema can’t completely satisfy the need.

B. Serving Spatio-Temporal Air Quality Data via SOS

1) Concept of SOS: The Sensor Observation Service (SOS) standard provides a standardized interface for managing and retrieving metadata and observations from heterogeneous sensor systems, which include a variety of sensor types, such as in-situ and remote sensors, mobile sensor platforms or networks of static sensors. In particular, this standard specifies how the observation, sensor description, as well as computational representation of observed features are accessed in an interoperable way. It further defines means to register new sensors and delete old ones. This standard also provides a mechanism to insert new observations and remove existing ones [13]. SOS 2.0 relies on the Sensor Model Language (SensorML) standard to encode meta description of sensor and the Observation and Measurement standard (O&M Model) to encode data gathered by sensors. The SOS is an OGC standard and ultimately only defines a service interface but not an implementation. There are currently several Open Source implementations of this service, such as 52°North SOS, deegree SOS and istSOS. To achieve the goal of air quality data integration with CityGML, 52°North SOS was chosen for our SOS deployment. 52°North SOS is standardized, interoperable and has a consistent implementation [14].

2) Description of Air Quality Data: The air quality data used in this paper is supplied by the "AERO-TRAM" project, sponsored by environmental ministry of Baden-Württemberg and realized by the Karlsruhe Institute of Technology. The latter's aim is an extensively automated, long-term study of concentration, spatial distribution of parameters such as O3, NO, NOx, CO, CO_2, H_2O(g), particle size distribution and total particle count in the area of Karlsruhe. In this project, the spatial distribution of air particles was continuously measured by an inlet sensor system, which is carried by the S-Bahn's (the local train system) two trams, the S1 and S2 in Karlsruhe. The selection of the actual running line depends on the internal scheduling procedure of the municipal transport service. Line S1 characterizes a specific height profile and Line S2 has a complete flat profile [15]. Each observation of air quality took place in a short time interval when the trams moved along their paths. In other words, the air quality sensor data varied both in spatial and temporal domains.

3) Encoding mobile Sensor Data: In order to provide the air quality data via SOS, two key processes should be undertaken, which include (1) encoding data in a standardized format and (2) populating an O&M database with data behind the SOS instance. Encoding data in a standard format is, more specifically, the encoding of the description of sensors and the sensor measurements based on the SensorML Model and O&M Model specifications respectively. The encoded file format should be compliant to the binding mechanisms supported by the SOS instance, such as XML, JSON and SOAP. Since the inlet sensor system is carried by the tram vehicle and the observation is measured during its movement, our sensor system falls into the category of mobile sensor platforms. Thus, the air quality sensor data should be considered as time series sampling. The part of encoding sensor metadata according to the SensorML 2.0 includes the description of our physical sensor systems. The following information has to be defined: sensor identifier, definition of sensor input and output, sensors' constraints and their connections within the sensor system . On the other hand, the part of encoding sensor data is to actually convert the sensor measurements into the concept "Observation". To define an observation, six elements should be involved according to the O&M Model. These are: "procedure", "phenomenon time", "result time", "observed property", "feature of interest" and "result". Listing 1 demonstrates how these six elements of one single observation are defined in our case. The "procedure" should be linked to a specific sensor identifier defined in the sensor metadata part and the "phenomenon time" is considered to be equal to "result time". This is represented by the sampling timestamps, with a temporal resolution of recording varying every second. The measurement of each timestamp is represented by a centered one-minute mean over a window of 60 data points. "Feature of interest" represents the geometric information of the observation, which for a static sensor system like weather stations can be depicted by the location of the relevant sensor. In this case, the "feature of interest" is considered a sampling feature, analyzing the "bubble of air". As for a mobile sensor platform, the geolocation of each observation varies all the time during the movement of the sensors. Thus, "feature of interest" in this case should be used as a domain feature (known as sampled feature). The street on which our tram passed along and the geometry point of each observation is described as a sampling feature. The second process is the import of the encoded data into the database. To that end a data feeder is needed. The SOS instance used in this paper, the 52°North SOS, has released a tool called "SOS Importer", which supports the import of data into the database suitable both for mobile and static sensor platforms.

```
1 <om:OM_Observation gml:id="o_01">
2   <gml:description>NO</gml:description>
3   <gml:identifier codeSpace="">s1-no1-observation
        -0</gml:identifier>
4   <om:type xlink:href="http://www.opengis.net/def/
        observationType/OGC-OM/2.0/OM_Measurement"/>
5   <om:phenomenonTime>
6      <gml:TimeInstant gml:id="phenomenonTime_0">
7        <gml:timePosition>2015-12-13T05:59:40</gml
             :timePosition>
8      </gml:TimeInstant>
9   </om:phenomenonTime>
10  <om:resultTime xlink:href="#phenomenonTime_0"/>
11  <om:procedure xlink:href="cld_66"/>
12  <om:parameter>
13     <om:NamedValue>
14        <om:name xlink:href="http://www.opengis.
              net/def/param-name/OGC-OM/2.0/
              samplingGeometry"/>
15        <om:value xsi:type="gml:
               GeometryPropertyType">
```

```
16          <gml:Point gml:id="sp_01">
17              <gml:pos srsName="http://www.opengis
                    .net/def/crs/EPSG/0/4326">
                    49.052822 8.388629</gml:pos>
18          </gml:Point>
19         </om:value>
20      </om:NamedValue>
21   </om:parameter>
22   <om:observedProperty xlink:href="http://sweet.
         jpl.nasa.gov/2.0/chemCompound.owl#NO"/>
23   <om:featureOfInterest xlink:href="http://
         localhost/geoserver/wfs?request=GetFeature&
         typeNames=tram:tram&featureID=s1"/>
24   <om:result xsi:type="gml:MeasureType" uom="ppb"
         >3.08</om:result>
25 </om:OM_Observation>
```

Listing 1. Sample XML excerpt of a single observation

4) Serving Mobile Sensor Data with the 52°North SOS's Sensor Web RESTful API 2.0: The Representational State Transfer (RESTful) Web services are considered much more lightweight and often better integrated with HTTP than SOAP-based Web services, because RESTful Web services do not require XML messages or WSDL service descriptions [16]. Thus, all the data used for this paper are designed to be accessed by RESTful services and then are used on the client side. The 52°North SOS supports the RESTful API, whose current stable version is 1.0 and it supports HTTP GET as well as POST methods. This RESTful API provides easy access to time series information from the SOS instances. However, there are limitations to serving air quality data via this API. It currently only supports time series data from static sensors and not yet from mobile sensor platforms. One of the core query parameters named "station" is designed for orientation of resources and only considers the "feature of interest" of observation as a sampling feature. This means the stations can only represent the location of a static sensor. In our case, the "feature of interest" is designed as a domain feature, which is not a simple geometry point. The geometry information of each observation, which is encoded as a sampling feature, will not be accessible using the current API. In order to supply sensor data from heterogeneous sensor systems, the 52°North SOS has released a new beta version of the Sensor Web REST API 2.0, which aims to support mobile sensor systems. With the extension of this API, the mobile air quality data can be accessed and later integrated with CityGML data on the client side.

C. Integration of CityGML and Air Quality Data

In this paper, the JavaScript library "Cesium" was chosen for visualization on the client side. Cesium is a WebGL virtual globe and map engine, which is capable of displaying KML and COLLADA/gITF files built on CityGML data [17]. There are several approaches considered for the integration of CityGML and air quality data via SOS. The first one is to display the urban model described by CityGML on Cesium and visualize the observation data as sampling points overlaying the city objects. The workflow of this approach is depicted by Figure 1. The second approach is to couple SOS by referencingthe WFS

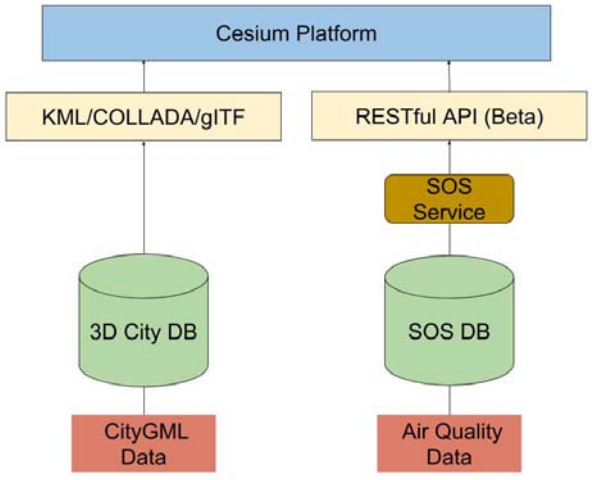

Fig. 1. Workflow of Integration between CityGML and Air Quality Data

request of CityGML to the "feature of interest" of observations. This approach works because the 52°North SOS can handle the features both in an internal and external way. The third approach is the use of Dynamizers [18]. Dynamizers are a mechanism that allows the storing of dynamic values separately from the original attributes in CityGML. This approach is contrary to the second one. The new CityGML data type "SensorConnection" defined in Dynamizers can be linked by various SOS requests for sensor metadata and the observations. For our work we used the first approach. Figure 2 and Figure 3 illustrate a Web demo based on the current infrastructure. As Figure 2 depicts, tram vehicle S1's observations, whose observed property is NO, are visualized as a track of color-coded dots (see legend in the second floating window on the right of Figure 2). The observations are previously classified into different groups based on their values. A semantic 3D model of north-west Karlsruhe is also loaded to the platform. The track information, such as its platform, procedure and observed property etc. is displayed on the info box of the Web demo. Figure 3 shows a time slot interface, which allows the user to observe how the results of the track varies in a specific interval.

II. CONCLUSION

The workflow of this paper can be summarized into two stages, namely data acquisition and data fusion. In terms of data acquisition, the OGC WFS and SOS standards are applied to retrieve CityGML and spatio-temporal varying air quality data respectively. It has been proved that GeoServer and its extension Application Schema is not a complete solution to serve CityGML via WFS, due to the issues with CityGML and the encoding rules of GML. A solution would be a more advanced CityGML WFS with RESTful API. As for the air quality data, the 52°North SOS SNAPSHOT 4.4.0 release modified the database schema to support mobile sensor platforms and provides a beta 2.0 RESTful API to extend query for "tracks". This has proven to be suitable to store observations whose

Fig. 2. Web Demo of Integration between CityGML and Air Quality Data on Cesium. The dotted line shows color coded NO values along the tram line.

Fig. 3. Web Demo of Integration between CityGML and Air Quality Data on Cesium. The graph shows the variation of NO values within a specific interval.

"feature of interest" is a domain feature. This can satisfy the demand of mobile sensor data provision. The second stage, data fusion or data integration, is achieved by accessing heterogeneous data with the RESTful API. The approach used in this paper is appropriate when only visualization is needed. The downside of this approach is the lack of relationships between city objects in the model and observation data.

ACKNOWLEDGEMENT

The authors of this study would like to thank the researchers and colleagues of the "AERO-TRAM" project for the data provision, their colleagues at EIFER -European Institute For Energy Research- as well as KIT-IPF Karlsruhe Institute of Technology Institute of Photogrammetry and Remote Sensing for their continued support and assistance. A special thanks goes to Christian Malewski for advice on setting up the project. The methodological basis presented here was developed in the context of a research and development project called the EDF City Simulation Platform Project. It has been running from 2012 to 2016 in EIFER and other institutes of EDF (Electricite de France).

REFERENCES

[1] K.-H. Häfele, "Citygml wiki." [Online]. Available: http://www.citygmlwiki.org/index.php?title=Citygml Wiki (visited on 13/05/2016).

[2] C. Mtral, G. Falquet, and K. Karatzas, "Ontologies for the integration of air quality models and 3d city models," *In Conceptual Models for Practitioners*, Jan 2012.

[3] "Citygml energy ade." [Online]. Available: http://www.citygmlwiki.org/index.php?title=CityGML Energy ADE (visited on 30/06/2016).

[4] R. Nouvel, R. Kaden, J.-M. Bahu, J. Kaempf, P. Cipriano, M. Lauster, J. Benner, E. Munoz, O. Tournaire, and E. Casper, "Genesis of the citygml energyade," in *Proceedings of International Conference CISBAT 2015 Future Buildings and Districts Sustainability from Nano to Urban Scale*, Scartezzini and Jean-Louis, Eds., sep 2015. [Online]. Available: https://infoscience.epfl.ch/record/213436

[5] G. Gröger, T. H.Kolbe, C. Nagel, and K.-H. Häfele, "OGC City Geography Markup Language (CityGML) Encoding Standard," OGC 12-019, Version 2.0.0, apr 2012. [Online]. Available: http://www.opengis.net/spec/citygml/2.0 (visited on 15/06/2016).

[6] "Citygml homepage." [Online]. Available: http://www.citygml.org (visited on 13/05/2016).

[7] P. P. A. Vretanos, "OGC Web Feature Service 2.0 Interface Standard With Corrigendum," OGC 09-025r1 and ISO/DIS 19142, Version 2.0.0, nov 2010. [Online]. Available: http://www.opengeospatial.org/standards/wfs (visited on 13/06/2016).

[8] "3dcitydb documentation v3," pp. 217–218, 2015. [Online]. Available: http://www.3dcitydb.org/3dcitydb/documentation/ (visited on 18/06/2016).

[9] E. Curtis, *Advances in 3D Geoinformation Systems*. Springer Berlin Heidelberg, dec 2008, ch. Serving CityGML via Web Feature Services in the OGC Web Services -Phase 4 Testbed, pp. 331–340.

[10] "Wfs basics." [Online]. Available: http://docs.geoserver.org/2.2.1/user/services/wfs/basics.html (visited on 12/05/2016).

[11] "Feature chaining." [Online]. Available: http://docs.geoserver.org/stable/en/user/data/app-schema/feature-chaining.html (visited on 14/05/2016).

[12] B. Caradoc-Davies and R. Angreani, "Geoserver application schema support: Complex web feature service for geoscience interoperability," Manuscript -CSIRO Research Publications Repository, nov 2010.

[13] A. Bröring, C. Stasch, and J. Echterhoff, "OGC Sensor Observation Service Interface Standard," OGC 12-006, Version 2.0.0, apr 2012. [Online]. Available: http://www.opengis.net/doc/IS/SOS/2.0

[14] “Sensor web community.” [Online]. Available: http://52north.org/communities/sensorweb/ (visited on 30/06/2016).

[15] R. Hagemann, U. Corsmeier, C. Kottmeier, R. Rinke, A. Wieser, and B. Vogel, “Spatial variability of particle number concentrations and {NOx} in the karlsruhe (germany) area obtained with the mobile laboratory aero-tram,” *Atmospheric Environment*, vol. 94, pp. 341 – 352, 2014. [Online]. Available: http://www.sciencedirect.com/science/ article/pii/S1352231014003987

[16] E. Jendrock, R. C.-N. I. Evans, D. G. K. Haase, and W. M. C. Srivathsa, “Introduction to web services.” [Online]. Available: http://docs.oracle.com/javaee/6/tutorial/doc/gijti.html (visited on 28/06/2016).

[17] “Demos 3dcitydb.” [Online]. Available: http://cesiumjs.org/demos/3dcitydb.html (visited on 28/06/2016).

[18] K. Chaturvedi and T. H. Kolbe, “Dynamizers -Modeling and Implementing Dynamic Properties for Semantic 3D City Models,” in *Eurographics Workshop on Urban Data Modelling and Visualisation*, F. Biljecki and V. Tourre, Eds. The Eurographics Association, 2015.

.....