

SOLAR POTENTIAL ANALYSIS AND INTEGRATION OF THE TIME-DEPENDENT SIMULATION RESULTS FOR SEMANTIC 3D CITY MODELS USING DYNAMIZERS

K. Chaturvedi*, B. Willenborg, M. Sindram, T. H. Kolbe

Technische Universität München, Chair of Geoinformatics, 80333 Munich, Germany
(kanishk.chaturvedi, b.willenborg, maximilian.sindram, thomas.kolbe)@tum.de

KEY WORDS: Semantic 3D City Models, CityGML, Solar potential simulation, Dynamizers

ABSTRACT:

Semantic 3D city models play an important role in solving complex real-world problems and are being adopted by many cities around the world. A wide range of application and simulation scenarios directly benefit from the adoption of international standards such as CityGML. However, most of the simulations involve properties, whose values vary with respect to time, and the current generation semantic 3D city models do not support time-dependent properties explicitly. In this paper, the details of solar potential simulations are provided operating on the CityGML standard, assessing and estimating solar energy production for the roofs and facades of the 3D building objects in different ways. Furthermore, the paper demonstrates how the time-dependent simulation results are better-represented inline within 3D city models utilizing the so-called Dynamizer concept. This concept not only allows representing the simulation results in standardized ways, but also delivers a method to enhance static city models by such dynamic property values making the city models truly dynamic. The dynamizer concept has been implemented as an Application Domain Extension of the CityGML standard within the OGC Future City Pilot Phase 1. The results are given in this paper.

1. INTRODUCTION

1.1 Semantic 3D City Models

Semantic 3D city models describe spatial, graphical and thematic aspects of the cityscapes by decomposing and classifying the occupied physical space according to a semantic data model. The relevant real world entities are represented by the ontological structure including thematic classes, attributes and their interrelationships (Kolbe, 2009). The international standard CityGML (Gröger et al., 2012) issued by the Open Geospatial Consortium (OGC) is a popular standard for modelling and exchanging semantic 3D city models. This standard facilitates the integration of heterogeneous data from multiple sources and allows for the representation of the geometrical and semantic attributes of the city level objects along with their interrelationship to other objects. Other examples of semantic information models are INSPIRE Data Themes (INSPIRE, 2014) and AAA (AdV, 2014). INSPIRE (Infrastructure for Spatial Information in the European Community) define Europe-wide consistent conceptual schemas for various data themes, while the AAA (AFIS-ALKIS-ATKIS) Reference Model defines conceptual schemas for the German geospatial base data of geo-topography and real estate cadastre. Industry Foundation Classes (IFC, 2017) is another example of a semantic information model, which has been developed by buildingSMART International, and provides the open standard for sharing Building Information Modelling (BIM) data among different software applications. The major advantage of semantic information models in comparison to visualization models such as Google Earth and Apple Maps is that they make it possible for machines/algorithms to distinguish urban objects like buildings and use their rich thematic and geometric information for queries, statistical computation, simulation, and visualization. For this reason, today, more and more cities worldwide such as Berlin, Singapore, Paris, Zurich, Vienna, London, New York, Vancouver, Montreal, and Helsinki are

developing and maintaining semantic 3D city models. Most of these models are based on the CityGML standard. These semantic 3D city models are directly being used in many applications and simulations for solving complex and real-world problems.

1.2 Role of Semantic 3D city models in simulations

A wide range of application domains and use cases benefit from 3D city models. (Biljecki et al., 2015) provide an extensive review of different applications of 3D city models. The authors classified the use cases in two broad categories, (i) *non-visualization* use cases, which do not require the visualization of the 3D city model as well as the results of the operations that the use case comprises, and (ii) *visualization-based* use cases, where visualization of the city model and the results play an important role. For example, solar potential analysis (c.f. section 3) is an instance of *non-visualization* use case in which the simulation results can be visualized but this is not essential to achieve the purpose of the use case. The results can be stored in a database, which can be queried without the need of being visualized. On the other hand, the applications related to navigation, gaming, and also urban planning fall into the category of *visualization-based* use cases. In these cases, the visualization of the objects is very important, and the use cases would not make much sense without it. In total, the authors identified more than 29 use cases including more than 100 applications, which are arranged into these categories. (Willenborg et al., 2017) further elaborate on three specific use cases to illustrate how the usage of semantic 3D city models can help in better understanding of the built environment. The first application is a simulation tool for assessing and estimating solar energy production for the roofs and facades of the 3D building objects in different ways. The simulation tool operates on the semantic 3D city models defined according to the CityGML standard. By combining a transition model, sun position calculation, and an approximation of the sky dome, the

* Corresponding author

solar power from direct, diffuse, and global sunlight irradiations are estimated for individual months and years. The second application demonstrates the estimation of the energy demand of buildings based on official statistical data and the simulation of refurbishment measures. The third application is on the simulation of detonations in urban space where a semantic 3D city model is used as data exchange platform for the simulation application. Most of the simulations involve time-dependent attributes, for example, the monthly values of solar irradiations or energy demand estimations for a building. These time-dependent result values are stored as static generic attributes because the city models do not support the explicit representation of time-dependent attribute values like time-series data.

Semantic 3D city models also play an important role in smart city applications. Smart Cities are an emerging field, which focuses on better utilizing a city's assets and resources utilizing modern world technologies. There are several smart cities initiatives, such as (IBM, 2016) and (Microsoft, 2016) advertising their smart grid, smart homes, smart traffic and the Internet of Things (IoT) solutions, which try to improve living in rapidly growing megacities. (Moshrefzadeh et al., 2017) propose a new concept called Smart District Data Infrastructure (SDDI), which highlights the importance of semantic 3D city models in the context of smart cities and allows linking them with time-dependent properties (such as real-time sensor observations) using open and interoperable standards. Similarly, an initiative from OGC called Future City Pilot Phase 1 (OGC FCP1, 2016) focuses on how the use of the international standards such as CityGML and IFC together can provide stakeholders with information, knowledge and insight which enhances financial, environmental, and social outcomes for citizens living in cities. One of the objectives of this pilot is to demonstrate how dynamic city models can provide better services to the citizens as well as can help performing better analyses.

However, current generation semantic 3D models are static in nature. The explicit support of time-dependent properties will open doors to many new possibilities and research areas. It will not only help simulations by including the dynamic simulation results directly within the city models, but also help numerous smart city applications by embedding and linking real-time observations from sensors and IoT devices.

2. MAKING 3D CITY MODELS DYNAMIC

In order to model the time-dependent properties with the city models, it is important to distinguish between different types of changes that take place in cities over time. In general, these changes can be categorized according to their geometry, topology, appearance, and semantics. They may occur one after another or at the same time. This norm further considers whether a model supports spatio-temporal real world objects that change continuously or just objects that are subject to discrete changes. An additional emerging criterion that further categorizes existing approaches that follow the continuous paradigm is whether the latter can deal with the movement of spatial objects over time. Furthermore, in semantic 3D city models, the relevant real world entities are represented by objects with thematic and spatial attributes and interrelationships to other objects. Hence, the changes may also affect the interrelationships. In addition, city objects may be decomposed into parts based on deeply nested structures that can be observed in the real world. For example, a building may be decomposed into different (main) building parts like walls,

stairs, etc. and these may again consist of parts like windows or doors. The changes may also be related to such semantic decompositions, which are currently not supported by any standard.

In addition to the mentioned approaches, the changes in cities can also be categorized based on their frequencies. Some of these changes may be slower in nature, e.g. (i) the history or evolution of cities such as construction or demolition of buildings, and (ii) managing multiple versions of the city models. (Chaturvedi et al., 2017) propose a new conceptual model in order to model such changes with 3D city models. The other types of changes may represent high frequent or dynamic variations of the object properties, e.g. variations of (i) thematic attributes such as changes of physical quantities (energy demands, temperature, solar irradiation levels), (ii) spatial properties such as change of a feature's geometry, with respect to shape and location (moving objects), and (iii) real-time sensor observations. In this case, only some of the properties of otherwise static objects need to represent such time-varying values. Dynamizer (Chaturvedi and Kolbe, 2016) is a new concept which allows modelling and integrating such highly dynamic properties with the static city models making them truly dynamic. Both of the mentioned data models for slower and highly dynamic changes have been developed based on the CityGML standard, however, they can also be directly applied to other GML-based application schemas including the European INSPIRE data themes and national standards for topography and cadastres like the British Ordnance Survey Mastermap or the German cadastre standard ALKIS.

2.1 Dynamizer

Dynamizer is a data structure to represent dynamic values in different and generic ways. The source of dynamic data may vary for different applications. The values may be obtained from (i) external files (e.g., CSV files) or data from external files included inline, (ii) external databases (e.g. tabulated values of simulation specific data), or (iii) real-time sensor observations (e.g. air quality sensors and smart meters). The dynamizers provide an explicit way to model such dynamic variations in the form of timeseries. The timeseries are expressed using the well-defined standards such as (OGC TimeseriesML, 2015) and OGC Observations & Measurements (OGC O&M, 2010). The timeseries may be defined for a single data source as a tabulation of the measured data using atomic timeseries. One common example illustrating such scenario is mapping of the electricity consumption values of a building for every hour in a day or the solar irradiation values of a building surface for every month in a year. However, in many applications, it is not sufficient just to provide a means for the tabulation of time-value pairs. They may require patterns to represent dynamic variations of properties based on statistics and general rules. For example, January monthly summaries for the electricity consumption of a building might be described as "all-Januaries 2001-2010". Similarly, the electricity consumption values may reflect generic patterns for individual weekdays/weekends in a week or a month. Another example scenario may also be determining patterns for specific seasons (such as spring, summer, autumn and winter) over ten years. In order to support such patterns, dynamizers include the concept of composite timeseries.

Dynamizers not only allow representing the dynamic data in generic ways, but also deliver a method to enhance static city models by such dynamic property values. It references a specific property (e.g. spatial, thematic or appearance

properties) of an object within a 3D city model providing dynamic values overriding the static value of the referenced object attribute. In this way, dynamizers can be used to inject dynamic variations of city object properties into an otherwise static representation. The advantage in using such approach is that it allows only selected properties of city models to be made dynamic. If an application does not support dynamic data, it simply does not allow/include these special types of features. In addition, dynamizer is well capable of representing the real-time sensor observations either inline as atomic timeseries or by linking to external sensor based services such as Sensor Observation Service (OGC SOS, 2012) and SensorThings API (Liang et al., 2016).

As a part of the OGC Future City Pilot Phase 1, dynamizers have been now successfully implemented as an Application Domain Extension (ADE) for the CityGML standard. The ADE mechanism allows for the systematic extension of each CityGML object type by additional attributes as well as the introduction of new object types. This implementation allows dynamizers to be used with the current version of the CityGML standard (version 2.0). However, this concept is already being discussed within the CityGML Standard Working Group and is intended to be included as a new module in the next major release (version 3.0) of the CityGML standard. Within the pilot project, the Dynamizer ADE addresses two use cases, (i) integrating real-time sensor observations with the 3D city model of Greenwich, London, and (ii) integrating time-dependent solar irradiation analysis results with semantic 3D city model of Rennes, France. In the following sections, the details of the solar potential simulation and representation of the results using dynamizers are provided.

3. SOLAR POTENTIAL SIMULATION

Solar irradiation is an abundantly available, clean, silent and secure energy source. In the future, photovoltaics (PV) and solar thermal (ST) collectors are going to play an important role in decentralized electricity and heat production, especially in cities, where a significant portion of the energy is consumed. According to the EU Directive 2010/31/EU new buildings after the year 2020 have to cover for their energy demand with their local energy production. To meet the requirements of the EU Zero Energy Building concept, in the future, much larger areas for solar energy production will be required. Hence, building facades increasingly gain interest for PV generation, as the facade area of modern cities is much larger than the roof area, offers better maintenance conditions (lower dust accumulation) and is mostly devoid of buildings installations (e.g. chimneys, dormers) (Redweik et al., 2013). Additionally, the combination of energy production with other building functions like heat insulation, cladding or window illumination with semi-transparent PV panels may provide other interesting benefits (Catita et al., 2014).

For the successful deployment of PV and ST systems in urban areas, the solar energy potential for roofs and facades needs to be investigated. Thereby, the influence of shadowing effects of the surrounding topographic features and local meteorological and climate conditions need to be taken into account. As discussed in (Freitas et al., 2015), the modelling of solar potential has been a topic of research for years. However, many approaches rely on digital surface models (DSM) as data source, where urban features cannot be distinguished computationally, lack support for calculations on vertical/tiled 3D surfaces or provide limited analytic and visualization capabilities.

The approach for estimating the solar irradiation for building roofs and facades on city scale described in this section is based on semantic 3D city models, which offer a detailed 3D representation of the cityscape combined with well established tools for data management, analysis and visualization and high data availability. The inclusion of the time-dependent simulation results in the data model and possible ways of visualization are discussed as well.

3.1 Large scale solar potential estimation based on semantic 3D city models according to the CityGML standard

The introduced model estimates the direct, diffuse and global solar irradiation and the Sky View Factor (SVF) for building facades and roofs. Its implementation is based on the 3DCityDB (3DCityDB, 2017), which is an open source database solution for 3D city models according to the international standard CityGML. The model considers the shadowing effects of buildings, vegetation objects, and optionally a digital terrain model (DTM) or DSM. Reflected radiation is neglected. The only input data required is a 3D city model according to the CityGML standard in Level of Detail 2 (LoD2), having roof and wall surfaces represented as thematic surfaces.

For modelling the direct solar radiation, a combination of the transition model according to (Fu and Rich, 1999) and an algorithm for computing the position of the sun according to (Grena, 2012) is used. First, the positions of the sun are calculated for an entire year. The sun positions are stored in 3DCityDB as point features lying on a 100,000 km radius around an observation point of free choice, which is usually the centre of the city model. The temporal resolution of the sun positions can be configured according to accuracy requirements. Typically, a one-hour interval is selected for balanced quality and performance settings. Second, the radiation power of each sun point is estimated using the transition model and attached to each point as attribute.

For robustness against regional atmospheric differences the transition model is calibrated using freely available data from the NASA Atmospheric Science Data Centre. In an iterative process the model is adjusted to 22-year mean radiation parameters from the NASA surface meteorology and Solar Energy (SSE) mission (NASA, 2017), which can be queried by LAT/LON coordinates.

The estimation of the diffuse radiation and the SVF is performed using an approximation of the sky half dome with points, where each point represents a spherical segment. The radiation power of each point is computed using the Standard Overcast Model according to (Fu and Rich, 1999). As for the sun points, these so called hemisphere points are stored in the 3DCityDB around the observation point with a 100,000 km radius.

Further, a calculation basis on facades and roofs of the buildings of the city model is required. Therefore, a point grid is created on roof and wall surfaces, where each point represents the same fraction of the area of the surface it is attached to. These points are used as reference points for the irradiated solar energy. The density of the point grid can be adapted considering quality and performance aspects. Details on the overall concept and the steps until here are given in (Zahn, 2015).

The shadows cast by the surrounding topographic features are considered by applying a ray tracing approach. For each point

of the aforementioned point grid on the buildings, rays to all sun and hemisphere points are created and tested for intersection with surrounding buildings and the DTM/DSM using the ray / triangle intersection test according to (Möller and Trumbore, 2005). The required triangulation of the building geometries is performed with the Java3D library (Oracle, 2016) in advance. To decrease the number of expensive intersection tests, the triangles are stored in a bounding volume octree index.

For the processing of city models of arbitrary size, covering e.g. whole cities, a tiling strategy has been realized. First, the simulation domain is split into a regular grid of cells with a freely configurable cell size. During the simulation run, each cell is calculated individually including its eight neighbour cells for the visibility analysis to decrease main memory consumption.

The approach above has been implemented in a Java application and successfully evaluated for the London Borough of Barking and Dagenham (LBBD), the city of Helsinki and the city of Rennes in France.

3.2 Storage and visualization of simulation results

The simulation tool estimates the direct, diffuse and global irradiation in different spatial and temporal resolutions: (a) The highest temporal resolution available depends on the configured temporal resolution of the sun points. Typically, temporal aggregation is performed per hour, month and year. (b) The highest spatial resolution is defined by the resolution of the point grid on the buildings. The results are spatially aggregated per facade, roof surface, and building.

The point grid is stored in a PostGIS database schema with the output parameters attached in different temporal resolutions as attributes for the points. For visualization of the simulation results textures are generated based on the point grid data. For this purpose, texture coordinates and the mapping of the points to the pixel position of the later image are stored during the creation of the point grid. In order to increase the resolution of the texture images, the point values are computed using bilinear interpolation. Subsequently, a freely selectable colour scheme is used for visualization.

Figure 1 shows the textures of the global irradiation values for the months: February, March, April and May. The colour gradient ranges from dark green (no irradiation) over yellow to red (maximum monthly irradiation of 153 [kWh / m²]). This type of visualization allows for a quick identification of suitable areas for PV. However, in order to be able to perform profound analyses the values are stored as attributes in the city model in addition to the purely visual representation of the solar irradiation values as textures. Based on the point grid results, the different temporal resolutions for direct, diffuse and global irradiation are computed and stored as generic attributes for each spatial aggregation level (wall, roof surfaces and building) in the city model. Currently, the temporal classification of a simulation result parameter is encoded in the attribute name as a suffix. For instance, an attribute named *globalRadMonth_01* denotes the aggregate global irradiation estimate on a specific feature for January. Similarly, individual generic attributes can be defined for storing the irradiation values for individual months. In the same way, the attributes can be defined for the aggregate global irradiation estimate on a specific feature for the entire year such as *globalRadYear*. The code representation shows an excerpt of a WallSurface of a building in a CityGML instance file for the corresponding attributes.

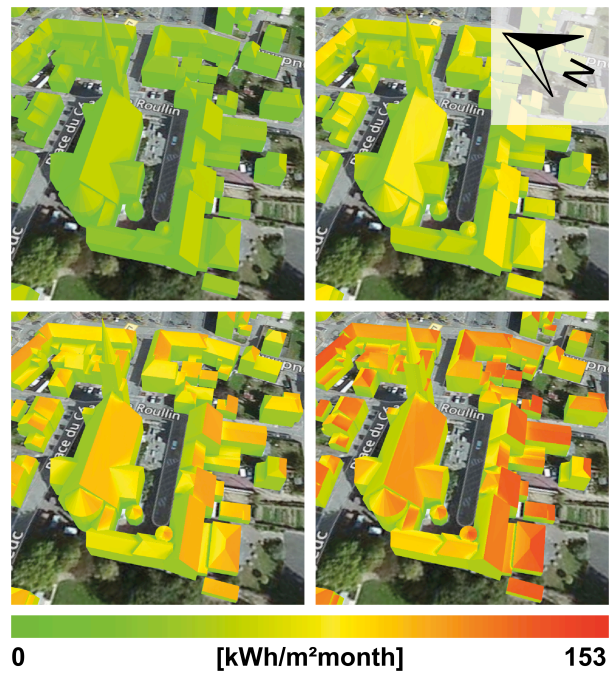


Figure 1. Textures of the global irradiation values for the months (left to right): February, March, April, and May

As depicted in figure 2, the monthly estimates for the three irradiation types direct, diffuse, and global for a facade surface are visualized in a web client for 3D city models as an attribute table. This data visualization has some limitations, as it makes it hard for the viewer to quickly grasp relevant information from the data. For instance, the table representation does not allow to directly observe the full range of the data, as it will usually not be possible to view the monthly attributes for direct, diffuse and global irradiation at the same time, because a table of that size would not fit on the screen. More difficulties arise, when trying to compare the three variables to each other or recognize monthly deviations. The usability of such analyses will drastically be improved, if city model standards allow time-dependent variations of such result values to a common generic attribute as illustrated by the dynamizer concept in section 4.

```
<bldg:WallSurface gml:id="UUID_01_WS_1">
  <gen:doubleAttribute name="globalRadYear">
    <gen:value>77004.913</gen:value>
  </gen:doubleAttribute>
  <gen:doubleAttribute name="globalRadMonth_01">
    <gen:value>4293.446</gen:value>
  </gen:doubleAttribute>
  <gen:doubleAttribute name="globalRadMonth_02">
    <gen:value>5563.502</gen:value>
  </gen:doubleAttribute>
  <gen:doubleAttribute name="globalRadMonth_03">
    <gen:value>7010.33</gen:value>
  </gen:doubleAttribute>
  <gen:doubleAttribute name="globalRadMonth_04">
    <gen:value>7180.839</gen:value>
  </gen:doubleAttribute>
  <!-- all months in a year -->
  <!-- ..... -->
  <gen:doubleAttribute name="globalRadMonth_12">
    <gen:value>4010.239</gen:value>
  </gen:doubleAttribute>
  <!-- diffuse radiation -->
  <!-- direct radiation --->
</bldg:WallSurface>
```



Figure 2. The tabular representation of monthly irradiation values of a building. Screenshot taken from (3DCityDB-Web-Map, 2017)

4. REPRESENTING SOLAR ANALYSIS RESULTS USING DYNAMIZERS

Dynamizers, on the one hand, allow representing the time-dependent simulation results in standardized ways, and on the other hand, allow referencing to the static generic attributes and override their values accordingly. Figure 3 shows an instance diagram for integrating the time-dependent simulation results with the CityGML document using the Dynamizers. In this illustration, the generic attributes of the yearly values of direct, diffuse, and global irradiations of the buildings have been generated using the solar potential simulation tool as mentioned in the previous section. These are named as *directRadYear*, *diffuseRadYear*, and *globalRadYear* respectively. These three generic attributes are static and do not change with respect to time. However, in order to represent the monthly values of direct, diffuse, and global irradiations, there is no need to define multiple generic attributes for each month. Only one generic attribute can be defined for each of the property named as *directRadMonth*, *diffuseRadMonth*, and *globalRadMonth* respectively. These generic attributes are dynamic in nature and will be changed every month. However, the dynamic values of solar analysis results can be represented using a well-defined data structure. The new dynamizer object can be created for each building surface allowing for the representation of dynamic values and overriding the specific attribute values of the surfaces according to the defined dynamic values.

The *Dynamizer* class consists of three attributes: (i) *attributeRef*, (ii) *startTime*, and (iii) *endTime*. *attributeRef* refers to the generic attributes generated for monthly direct, diffuse, and global irradiations using XPath expressions. XPath is a W3C recommendation used to navigate through the elements and attributes within an XML document. In this way, the dynamizers will only refer to attributes which are dynamic in nature and do not impact the other static attributes such as yearly irradiation values. Furthermore, *startTime* and *endTime* are absolute time points denoting the time span for which the dynamizer provides dynamic values. Each dynamizer object can contain the *dynamicData* in the form of timeseries. There are two possible ways to model the timeseries: (i) *AtomicTimeseries*, for representing atomic/independent timeseries, (ii) *CompositeTimeseries* for modelling nested timeseries in order to represent specific patterns.

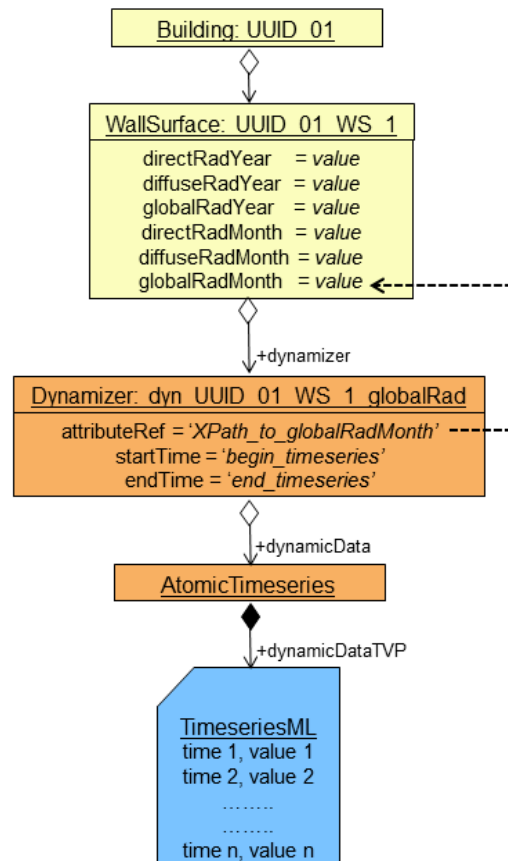


Figure 3. An instance diagram of representing the time-dependent solar potential simulation results using dynamizers.

In this scenario, only *AtomicTimeseries* were generated to represent monthly solar irradiation values. The irradiation values were represented as interleaved time-value pairs according to the TimeseriesML standard. However, it is also possible to represent the result values according to different other standards such as OGC Observations and Measurements (O&M). O&M is one of the core standards for the response models of the sensor-based standards such as Sensor Observation Service and SensorThings API. In this way, the sensor observations can also be represented as a timeseries within the *AtomicTimeseries* of dynamizers.

The code representation on the next page shows how the solar irradiation result values of the same building (c.f. section 3.2) can be represented using dynamizers. In the representation, it can be observed that the wall surface has generic attributes defined for storing the irradiation results. *globalRadYear* denotes the aggregate global irradiation estimate of the surface for the entire year. However, instead of defining the separate generic attributes for each month, there is only one generic attribute for the monthly irradiation values. The dynamizer is defined for representing the monthly global irradiation values of the wall surface. It contains the attribute reference, which refers to the value of the attribute *globalRadMonth* of the wall surface using an XPath expression. It also contains the attributes start and end time, which define the total time span for the solar irradiation result values.

```

<bldg:WallSurface gml:id="UUID_01_WS_1">
  <gen:doubleAttribute name="globalRadYear">
    <gen:value>77004.913</gen:value>
  </gen:doubleAttribute>
  <!------- yearly values ----->
  <!-- direct and diffuse radiations -->
  <gen:doubleAttribute name="globalRadMonth">
    <gen:value>4293.446</gen:value>
  </gen:doubleAttribute>
</bldg:WallSurface>
<dyn:Dynamizer gml:id="dyn_UUID_01_WS_1_globalRad ">
  <dyn:attributeRef>
    <!-- Single line XPath Expression -->
    //bldg:WallSurface[@gml:id='UUID_01_WS_1']
    /doubleAttribute[@name='globalRadMonth']
    /gen:value
  </dyn:attributeRef>
  <dyn:startTime frame="#ISO-8601">
    2015-01-01T00:00:00Z
  </dyn:startTime>
  <dyn:endTime frame="#ISO-8601">
    2016-01-01T00:00:00Z
  </dyn:endTime>
  <dyn:dynamicData>
    <dyn:AtomicTimeseries>
      <dyn:dynamicDataTVP>
        <tsml:TimeseriesTVP>
          <tsml:point>
            <tsml:MeasurementTVP>
              <tsml:time>2015-01</tsml:time>
              <tsml:value>4293.446</tsml:value>
            </tsml:MeasurementTVP>
          </tsml:point>
          <tsml:point>
            <tsml:MeasurementTVP>
              <tsml:time>2015-02</tsml:time>
              <tsml:value>5563.502</tsml:value>
            </tsml:MeasurementTVP>
          </tsml:point>
          <tsml:point>
            <tsml:MeasurementTVP>
              <tsml:time>2015-03</tsml:time>
              <tsml:value>7010.33</tsml:value>
            </tsml:MeasurementTVP>
          </tsml:point>
          <!-- all months in a year -->
          <!-- ..... -->
          <tsml:point>
            <tsml:MeasurementTVP>
              <tsml:time>2015-12</tsml:time>
              <tsml:value>4010.239</tsml:value>
            </tsml:MeasurementTVP>
          </tsml:point>
        </tsml:TimeseriesTVP>
      </dyn:dynamicDataTVP>
    </dyn:AtomicTimeseries>
  </dyn:dynamicData>
</dyn:Dynamizer>
<dyn:Dynamizer>
  <!------- Dynamizers for ----- -->
  <!-- direct and diffuse radiations -->
  <!-- ..... -->
</dyn:Dynamizer>
    
```

Further, the dynamizer includes the dynamic data in the form of timeseries data. By using the atomic timeseries, the monthly irradiation values can be represented in a structured way. As shown above, the timeseries has been defined as interleaved time-value pairs according to the OGC TimeseriesML standard (OGC TimeseriesML, 2015). This standard allows defining individual time points for each month with their corresponding values.

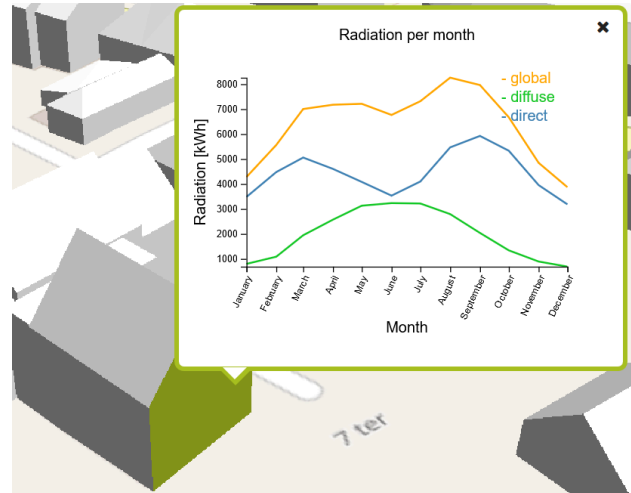


Figure 4. The interactive timeseries representation of monthly irradiation values of a building. Screenshot taken from (virtualcityMAP, 2017).

The major advantage with such representations in comparison to previous approaches is that it allows comparing and analysing different properties in a more interactive manner. With the timeseries representation, the full range of data can be observed in easier ways. Figure 4 is a screenshot from virtualcityMAP application (virtualcityMAP, 2017) showing the values of solar irradiation values in the form of a timeseries graph. This application was developed within the OGC Future City Pilot phase 1. Such visualizations help users to quickly grasp relevant information from the data. Furthermore, the standards such as TimeseriesML support interpolation and aggregation properties, which are helpful in mapping missing values or multiple values to the specific time points. Using composite timeseries, it is also possible to represent patterns such as how the solar irradiation values for the specific buildings vary in different seasons over a period of ten years.

5. MANAGEMENT OF THE HIGHLY DYNAMIC SIMULATION RESULTS

In order to store and manage semantic 3D city models, there are already stable databases available. For example, 3DCityDB stores, represents, and manages large CityGML datasets on top of a standard spatial relational database management system such as Oracle Spatial and PostGIS. It provides a Java front-end application named '3DCityDB Importer/Exporter', which allows for high performance importing and exporting CityGML datasets with arbitrary file sizes. The contents can be exported in the form of different visualization formats such as KML, COLLADA, and glTF, allowing the 3D objects to be viewed and interactively explored in web applications. 3DCityDB also allows extending its functionalities in a modular way by the installation of plugins, which add specific abilities to interact with the 3D city database. For instance, by using the Spreadsheet Generator Plugin, arbitrary subsets of city model data such as generic attributes can be exported in tabular form having the selected attributes from a 3D city database instance whether as a CSV file or directly be uploaded as a Google Spreadsheet Document or Google Fusion Table. There is an ongoing research (Yao and Kolbe, 2017), which provides an automatic way for dynamically extending the 3DCityDB to support storage and management of CityGML models with ADEs. This approach allows managing atomic and composite

timeseries of the Dynamizer ADE within the 3DCityDB. The time points within timeseries can further be queried and used using standard SQL operations.

In many scenarios (especially in smart city applications), the dynamic data may belong to different stakeholders (e.g. owners, operators, solution providers, citizens, and visitors), and different data sources (e.g. databases, web services, and external files). In order to manage such complex distributed systems, Spatial Data Infrastructures (SDIs) such as the SDDI (Moshrefzadeh et al., 2017) are set up. The SDIs involve standardized information models (such as CityGML, O&M, and TimeseriesML) and web services (such as Web Feature Service, Catalog Service, and Sensor Observation Service) allowing for the exchange of data in efficient and flexible ways. With the help of such infrastructures, it is possible to retrieve the information from remote and distributed resources (such as databases or files) using web services. This approach avoids data to be stored in centralized databases. Following the same approach, highly dynamic simulation results can be retrieved by using a lightweight web service called Mini Sensor Observation Service.

5.1 Mini Sensor Observation Service

The “Mini Sensor Observation Service” (Mini SOS) is a very basic and lightweight web service, which allows accessing timeseries from distributed data sources and representing them using standardized interfaces such as Sensor Observation Service and SensorThings API. These interfaces allow retrieving the highly dynamic sensor observations according to well-defined response models such as OGC O&M. Furthermore, the RESTful API from 52° North (52°North SWE, 2017) is a lightweight API which provides easy access to timeseries information from the SOS instances. It does not provide full access to all information available via the SOS interface but does provide the timeseries information and observation data in a quite condensed way. 52° North offers a sophisticated SOS web client as Open Source software that uses their proprietary API. The web client, thus, can also be used with the Mini SOS.

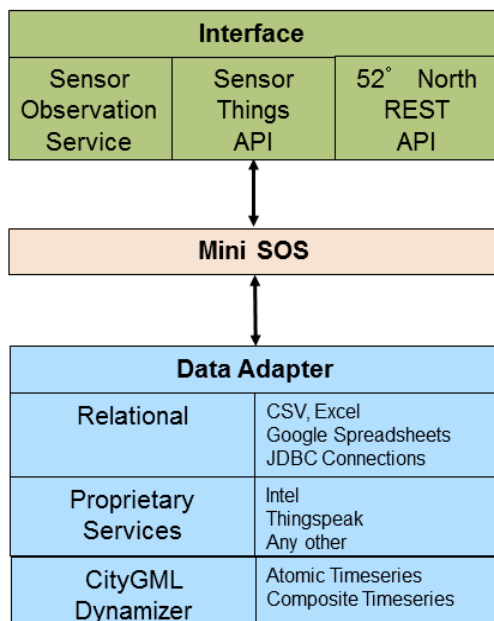


Figure 5. The conceptual illustration of Mini Sensor Observation Service

As shown in figure 5, the Mini SOS allows accessing the timeseries data according to Sensor Observation Service, SensorThings API, and RESTful API developed by 52° North. The Mini SOS also supports data adapters allowing to access timeseries from arbitrary data sources. The Dynamizer Adapter accesses CityGML documents with dynamizers such as the monthly solar irradiation values of a building surface and represents them according to the web interfaces. It helps reading, querying and visualizing the dynamizer timeseries without having the need to store them in a database. The service also supports relational adapters, which allow accessing the external files such as CSV or Excel sheets, cloud based services such as Google Spreadsheet and Google Fusion Tables, and also external databases. Furthermore, there are adapters available for accessing timeseries from the external web services such as SOS, SensorThings API, and also proprietary IoT services such as Intel and ThingSpeak. In this way, the Mini SOS supports accessing timeseries from different data sources and provides well-defined standards without having the need of their storage. This service is currently in the implementation process.

6. CONCLUSIONS AND FUTURE WORK

In this paper, the details of solar potential simulation are provided for assessing and estimating solar energy production for the roofs and facades of 3D building objects in different ways. The simulation tool operates on 3D models structured according to the CityGML standard and generates the monthly and yearly estimates of direct, diffuse, and global irradiation values for the building surfaces. The paper further demonstrates how the highly dynamic simulation results can be better represented using the CityGML Dynamizer ADE. The Dynamizer approach not only allows representing the simulation results in standardized ways, but also delivers a method to enhance static city models by such dynamic property values making the city models truly dynamic. It also helps in analysing the simulation results in more interactive ways leading to better decision-making. The Dynamizer ADE has been successfully implemented as a part of OGC Future City Pilot phase 1. In addition, the dynamizer is well capable of representing the real-time sensor observations either inline or by linking to external sensor based services. Furthermore, the paper introduces the Mini Sensor Observation Service, which allows accessing timeseries from CityGML Dynamizers and other distributed data sources and representing them using standardized interfaces without the need of data replication.

The implementations mentioned in the paper include integrating the time-dependent thematic properties with the city models. However, the dynamizers are also capable of supporting spatial and appearance properties of the city models, which will be implemented in near future. Solutions will also be developed to support dynamizer features in the databases as well as in visualization clients. The Dynamizer concept is already being discussed within the CityGML Standard Working Group and is intended to be included as a new module in the next major release (version 3.0) of the CityGML standard.

ACKNOWLEDGEMENTS

Part of this work has been carried out within the pilot project OGC Future City Pilot Phase 1 (FCP1). We thank FCP1 sponsors and participants for supporting this work. We also acknowledge our partners in the project Smart Sustainable Districts funded by the Climate-KIC of the European Institute of Innovation and Technology (EIT).

REFERENCES

- 3DCityDB, 2017. 3D City Database (3DCityDB) Homepage, <http://www.3dcitydb.org/3dcitydb/3dcitydbhomepage>
- 3DCityDB-Web-Map, 2017. 3DCityDB-Web-Map: Cesium-based 3D viewer and JavaScript API for the 3D City Database, <https://github.com/3dcitydb/3dcitydb-web-map>
- 52°North SWE, 2017. 52N Sensor Web Community - Sensor Observation Service, <http://52north.org/communities/sensorweb/sos>
- AdV, 2014. Dokumentation zur Modellierung der Geoinformationen des amtlichen Vermessungswesens (GeoInfoDok), Version 7.0.1. Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland (AdV)
- Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., Çöltekin, A., 2015. Applications of 3D City Models: State of the Art Review. *ISPRS International Journal of Geo-Information* 4 (4), 2842–2889
- Catita, C., Redweik, P., Pereira, J., Brito, M.C., 2014. Extending solar potential analysis in buildings to vertical facades. *Computers & Geosciences* 66, 1–12
- Chaturvedi, K., Kolbe, T.H., 2016. Integrating Dynamic data and Sensors with Semantic 3D City Models in the context of Smart Cities. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*. IV-2/W1, pp. 31–38
- Chaturvedi, K., Smyth, C.S., Gesquière, G., Kutzner, T., Kolbe, T.H., 2017. Managing Versions and History Within Semantic 3D City Models for the Next Generation of CityGML. In: Abdul-Rahman, A. (Ed.), *Advances in 3D Geoinformation*. Springer International Publishing, Cham, pp. 191–206
- Freitas, S., Catita, C., Redweik, P., Brito, M.C., 2015. Modelling solar potential in the urban environment: State-of-the-art review. *Renewable and Sustainable Energy Reviews* 41, 915–931
- Fu, P., Rich, P.M., 1999. Design and implementation of the Solar Analyst: an ArcView extension for modeling solar radiation at landscape scales. *Proceedings of the Nineteenth Annual ESRI User Conference*, pp. 1–31
- Grena, R., 2012. Five new algorithms for the computation of sun position from 2010 to 2110. *Solar Energy* 86 (5), 1323–1337
- Gröger, G., Kolbe, T.H., Nagel, C., Häfele, K.-H., 2012. OGC City Geography Markup Language (CityGML) Encoding Standard - Version 2.0.0. OGC Doc. No. 12-019. Open Geospatial Consortium Inc, OGC, <http://www.opengeospatial.org/standards/citygml>
- IBM, 2016. IBM Smarter Cities - Future cities - United States, http://www.ibm.com/smarterplanet/us/en/smarter_cities/overview/. (Accessed 12 May, 2017)
- IFC, 2017. Industry Foundation Classes (IFC) Overview Summary, <http://www.buildingsmart-tech.org/specifications/ifc-overview/ifc-overview-summary> (Accessed 26 January, 2017)
- INSPIRE, 2014. D2.5: Generic Conceptual Model, Version 3.4. Generic Conceptual Model of the INSPIRE data specifications, http://inspire.ec.europa.eu/documents/Data_Specifications/D2.5_v3.4.pdf. (Accessed 10 April, 2017)
- Kolbe, T.H., 2009. Representing and exchanging 3D city models with CityGML. In: *3D geo-information sciences*. Springer, pp. 15–31
- Liang, S., Huang, C.-Y., Khalafbeigi, T., 2016. OGC SensorThings API Part 1: Sensing, Version 1.0, Open Geospatial Consortium. OGC. Doc. No. 15-078r6, <http://www.opengeospatial.org/standards/sensorthings>
- Microsoft, 2016. Microsoft CityNext, <https://partner.microsoft.com/en-US/Solutions/CityNext>. (Accessed 12 May, 2017)
- Möller, T., Trumbore, B., 2005. Fast, minimum storage ray/triangle intersection. *ACM SIGGRAPH 2005 Courses*, ACM, pp. 1–7
- Moshrefzadeh, M., Chaturvedi, K., Hijazi, I., Donaubaue, A., Kolbe, T.H., 2017. Integrating and Managing the Information for Smart Sustainable Districts - The Smart District Data Infrastructure (SDDI). In: Kolbe, T.H., Bill, R., Donaubaue, A. (Eds.), *Geoinformationssysteme 2017. Beiträge zur 4. Münchner GI-Runde*, Wichmann Verlag, Berlin
- NASA, L.R.C., 2017. Surface meteorology and Solar Energy (SSE) Data and Information website, https://eosweb.larc.nasa.gov/project/sse/sse_table
- OGC FCP1, 2016. Future City Pilot Phase 1, <http://www.opengeospatial.org/projects/initiatives/fcp1>. (Accessed 1 June, 2017)
- OGC O&M, 2010. OGC Observations and Measurements, Version 2.0. OGC Doc. No. 10-004r3, <http://www.opengeospatial.org/standards/om>
- OGC SOS, 2012. OGC Sensor Observation Service 2.0 Interface Standard, Open Geospatial Consortium. OGC Doc. No. 12-006, <http://www.opengeospatial.org/standards/sos>
- OGC TimeseriesML, 2015. OGC TimeseriesML 1.0 - Timeseries Profile of Observations and Measurements, Version 1.0. OGC Doc. No. 15-043r3, <http://www.opengeospatial.org/standards/tsml>
- Oracle, 2016. Java 3D Project Website, <https://java3d.java.net/>
- Redweik, P., Catita, C., Brito, M.C., 2013. Solar energy potential on roofs and facades in an urban landscape. *Solar Energy* 97, 332–341
- virtualcityMAP, 2017. virtualcityMAP – the demonstrator application developed under OGC Future City Pilot Phase 1, <http://rennes.virtualcitymap.de/>
- Willenborg, B., Sindram, M., Kolbe, T.H., 2017. Applications of 3D City Models for a better understanding of the Built Environment: (accepted). In: Behnisch, M., Meinel, G. (Eds.), *Trends in Spatial Analysis and Modelling*. Springer, Berlin
- Yao, Z., Kolbe, T.H., 2017. Dynamically Extending Spatial Databases to support CityGML Application Domain Extensions using Graph Transformations. In: Kersten, T.P. (Ed.), *Tagungsband der 37. Wissenschaftlich-Technischen Jahrestagung der DGPF. Deutsche Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V., Würzburg*, pp. 316–331
- Zahn, W., 2015. Sonneneinstrahlungsanalyse auf und Informationsanreicherung von großen 3D-Stadtmodellen im CityGML-Schema. Masterarbeit, Technische Universität München, <https://mediatum.ub.tum.de/node?id=1276236>