

Integrating Dynamic Data and Sensors with Semantic 3D City Models

Kanishk Chaturvedi, Thomas H. Kolbe

Chair of Geoinformatics
Technische Universität München
Germany

Geospatial Sensor Web Conference
Münster, Germany
30 August, 2016

Semantic 3D City Models

▶ Semantic 3D City Models

- Relevant objects of the urban space are classified and their spatial and thematic properties will be described
- Are key for **Urban Information Modelling**



▶ CityGML (OGC international standard since 2008)

- **Data model** (UML) + Exchange format (based on GML3)
- Different thematic areas + Levels of Detail concept (LOD0 - LOD4)
- 3D geometry, 3D topology, semantics, and appearance

▶ CityGML is very useful in environmental & energy simulations, disaster management, training simulators

- In most simulations, time plays an important role, i.e. dynamic and time-varying properties

▶ **Time-varying properties are not yet supported in CityGML**

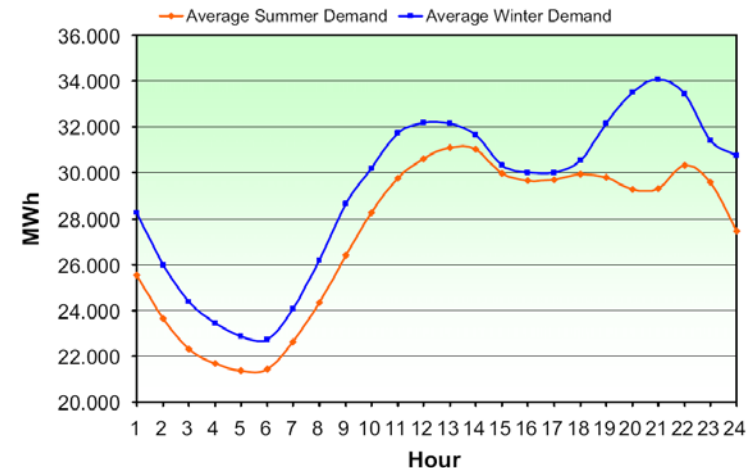
Time-varying properties

▶ Slower Changes

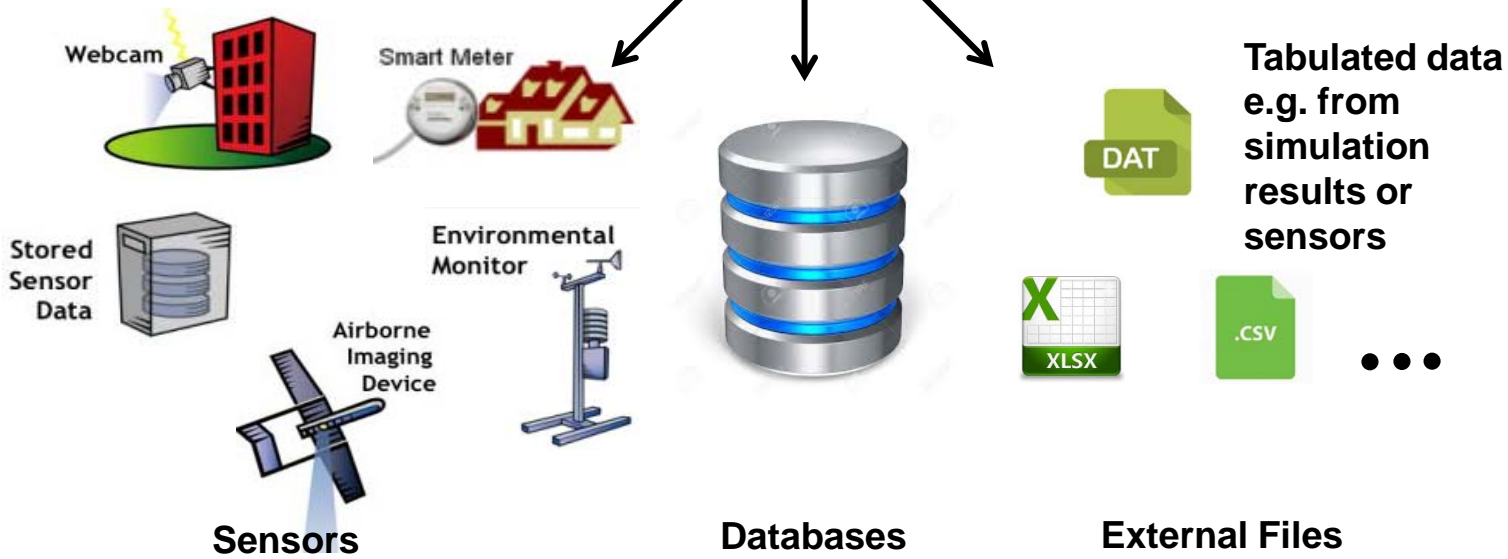
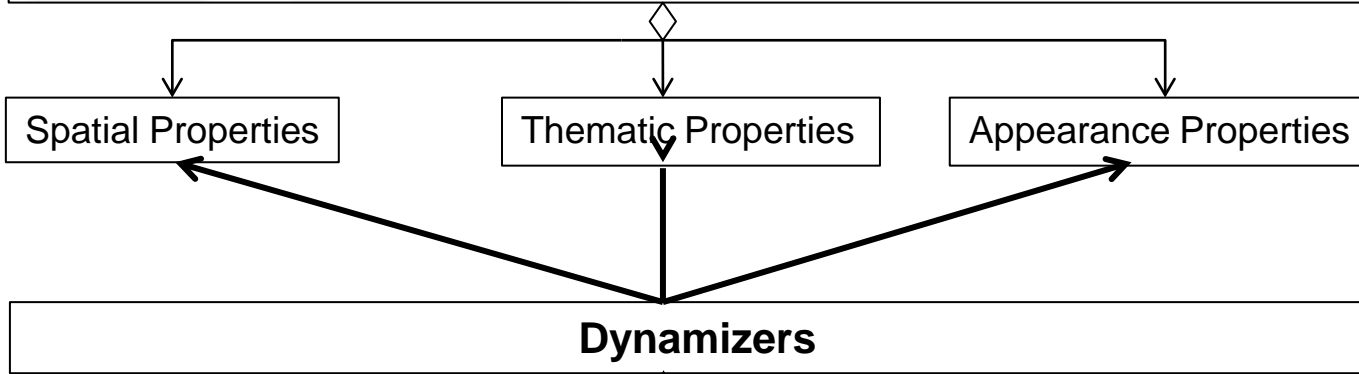
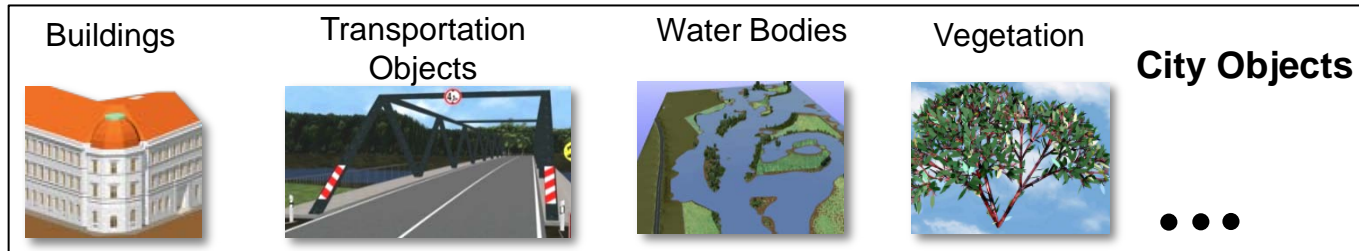
- History or evolution of cities/city models → versioning concept

▶ Highly dynamic changes

- **Variations of spatial properties:** change of a feature's geometry, both in respect to shape and to location (e.g. moving objects)
- **Variations of thematic attributes:** changes of physical quantities like energy demands, mean temperature, solar irradiation; air quality in streets and buildings
- **Variations with respect to sensor or real-time data**

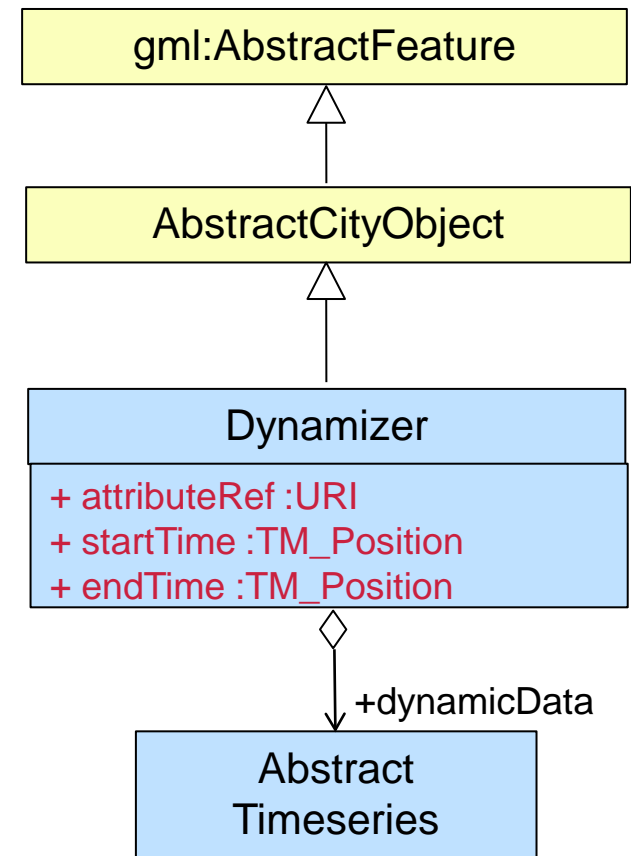


Source: C. García-Ascanio and C. Maté, "Electric power demand forecasting using interval time series: A comparison between VAR and iMLP," *Energy Policy*



Dynamizer – A New CityGML Feature Type

- ▶ **attributeRef** refers to a specific property of a static CityGML object which value will then be overridden/replaced by the (dynamic) values specified in the ‘Dynamizer’ feature.
- ▶ **startTime** and **endTime** denote time span for which Dynamizer provides dynamic values
- ▶ Dynamizer composes of **AbstractTimeseries**:
 - Allows represent time-variant values in different and generic ways
 - E.g. Timeseries, Sensor observations etc.



Example Scenario

CityGML object

```
<cityObjectMember>
  <Building gml:id = "building1">
    <gen:doubleAttribute name = "HeatDemand">
      <gen:value = xxx />
    </gen:doubleAttribute>
  </Building>
</cityObjectMember>
```

Replacing dynamic attributes using XPath

Source of dynamic data

Estimated (in kwh)	Heat Demand
JAN-15	61578
FEB-15	52148
MAR-15	41011
·	·
·	·
·	·
DEC-15	64984

```
<cityObjectMember>
  <dyn:Dynamizer>
    <dyn:attributeRef> //Building [@gml:id = 'building1']/doubleAttribute[@name = 'HeatDemand']/gen:value</dyn:attributeRef>
    <dyn:startTime> 2015-01-01T00:00:00Z </dyn:startTime>
    <dyn:endTime> 2015-12-31T00:00:00Z </dyn:endTime>
    <dyn:dynamicData>.. </dyn:dynamicData>
  </dyn:Dynamizer>
</cityObjectMember>
```

Dynamizer

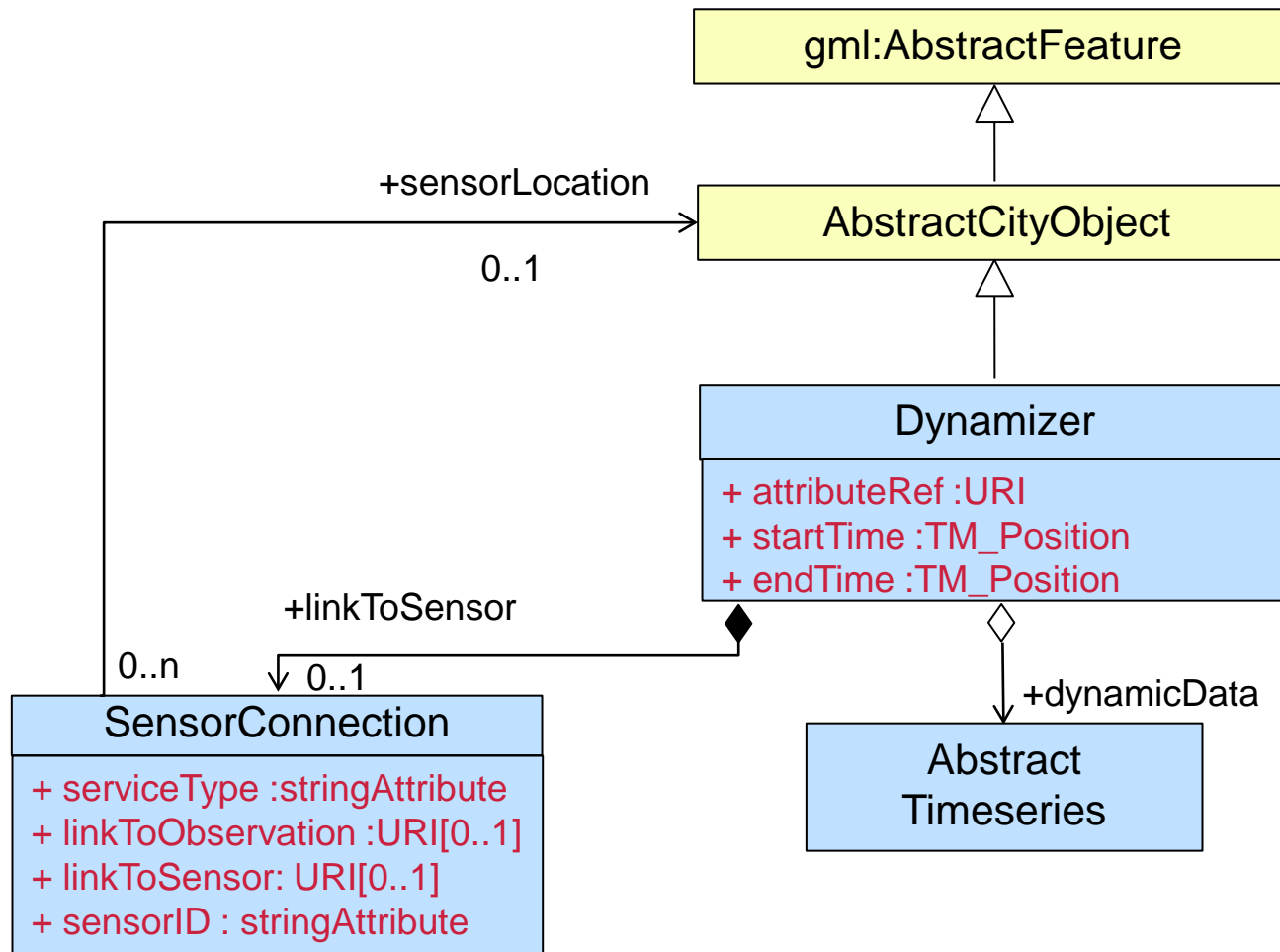
Dynamic Data from Sensors

- ▶ An important source of dynamic data may be sensor services.
- ▶ Two popular standards
 - **OGC Sensor Observation Services (SOS)**
 - Open standard, part of OGC Sensor Web Enablement (SWE)
 - Allows querying real-time sensor data and sensor data timeseries.
 - Observation responses are encoded in O&M standard
 - **OGC SensorThings API**
 - Very lightweight standard to interconnect the Internet of Things devices, data and applications over the web
 - Built on OGC SWE and O&M standards
 - Provides REST services and compact data encodings in JSON format

Source : <http://www.opengeospatial.org/ogc/markets-technologies/swe>

Source : <http://www.sensorup.com/>

Integrating Sensors and Observations



Example for a Sensor Connection



```

<cityObjectMember>
  <dyn:Dynamizer gml:id = "HeatDemandTimeseries" >
    <dyn:attributeRef>//RoofSurface[@gml:id='building1_roofSurface1']
      /doubleAttribute[@name = 'PV_Power_Generation']
      /gen:value </dyn:attributeRef>
    <dyn:startTime>2016-01-01T00:00:00Z</startTime>
    <dyn:endTime>2016-12-01T00:00:00Z</endTime>
    <dyn:linkToSensor>
      <dyn:SensorConnection>
        <dyn:sensorID>. . . </dyn:sensorID> ← Unique Sensor ID
        <dyn:serviceType>. . . </dyn:serviceType> ← SOS or SensorThings API
        <dyn:linkToObservation>. . . </dyn:linkToObservation> ← SOS GetObservation
        <dyn:linkToSensor>. . . </dyn:linkToSensor> ← SOS DescribeSensor
        <dyn:sensorLocation xlink:href="#building1_roofSurface1"/>
          ← Link to CityGML Object
        </dyn:SensorConnection>
      </dyn:linkToSensor>
    </dyn:Dynamizer>
  </cityObjectMember>
  
```

<cityObjectMember> Image source : <http://www.royalgreengas.com/index.php/photovoltaic/residential-buildings>

Summary

- ▶ Dynamizers **enhance static city models** by dynamic property values
 - by referencing a specific attribute (e.g. geometry, thematic or appearance property) of an object
 - overriding the static value of the referenced object attribute by dynamic property values
- ▶ Dynamizers support **multiple dynamic representations**
 - OGC TimeseriesML1.0, OGC O&M
- ▶ Establish **explicit links to sensors**
 - Linking sensor observations with the respective city model objects
 - Providing location of sensors as city objects
- ▶ Support also **nested patterns for values** based on statistics and general rules

Publications

- ▶ Chaturvedi, K. and Kolbe, T. H., 2016. **Integrating Dynamic Data and Sensors with Semantic 3D City Models in the context of Smart Cities.**

Accepted and to be published in

3DGeoInfo 2016 Conference, Athens, Greece.

- ▶ Chaturvedi, K. and Kolbe, T. H., 2015. **Dynamizers – Modeling and Implementing Dynamic Properties for Semantic 3D City Models.**

Published in *3rd Eurographics Workshop on Urban Data Modelling and Visualisation*, TU Delft, The Netherlands.