Approach for Dynamic Data Schema CityGML 3.0

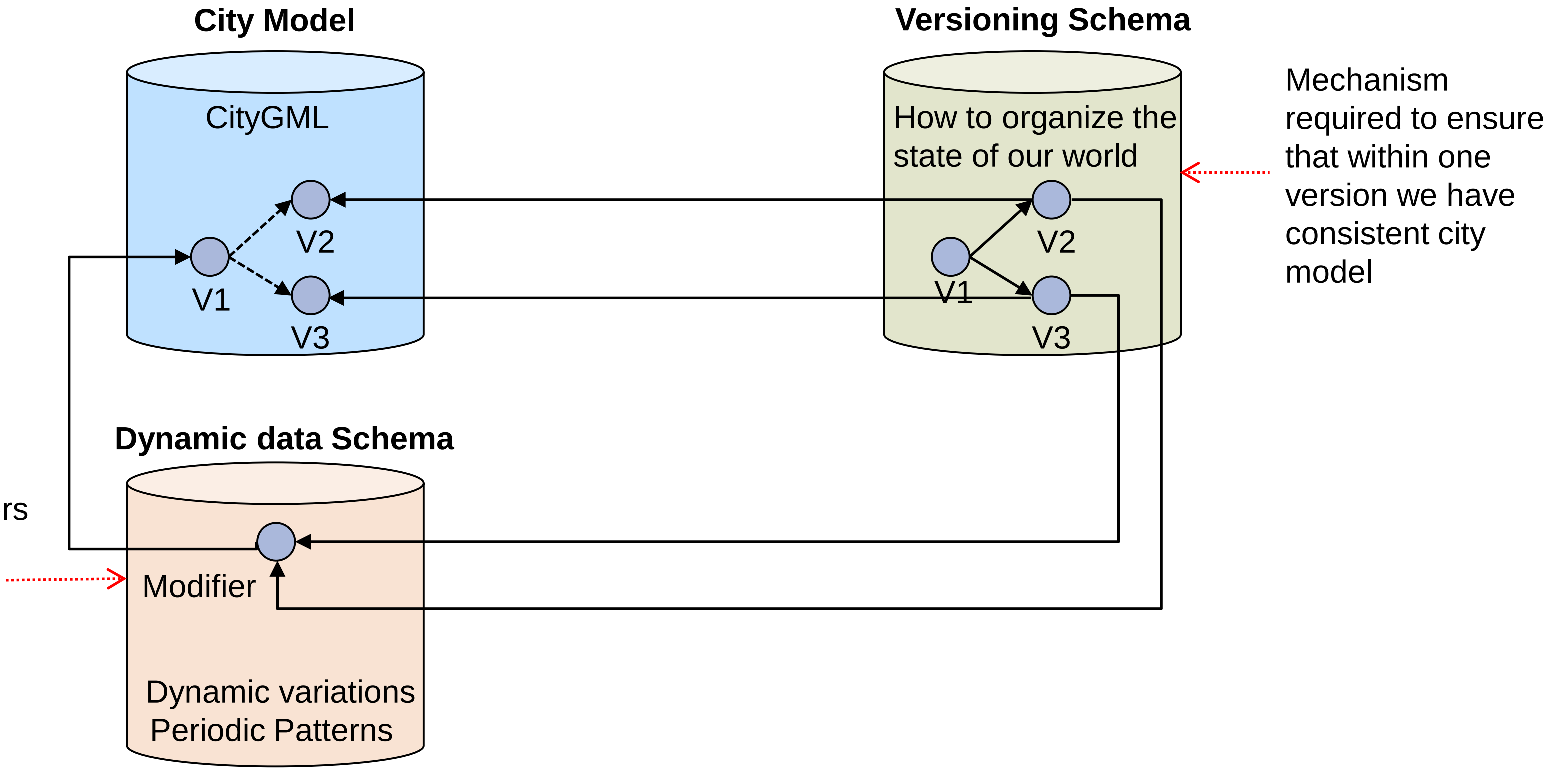
Kanishk Chaturvedi

Chair of Geoinformatics
Technische Universität München

kanishk.chaturvedi@tum.de

Conceptual Model

- To develop
  - A versioning schema, which represents the evolution of the city (model) in the form of different versions.
  - A dynamic data schema, where dynamic variations can be stored in special types of features, which would be interpreted as modifiers to the static CityGML model.

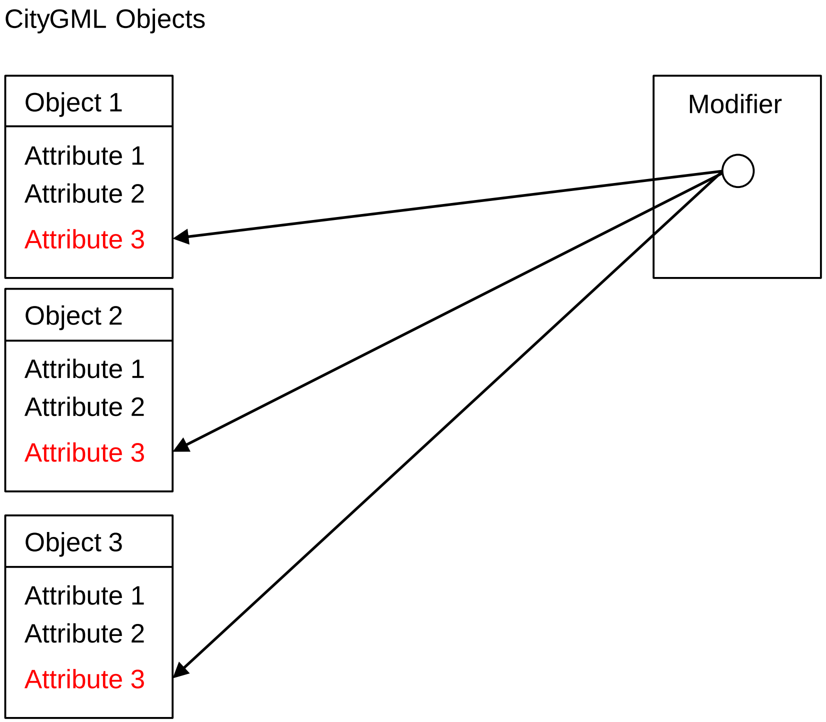


Mechanism required to ensure that within one version we have consistent city model

Provides replacers / overriders, replacing the attributes in the static CityGML model

Modifier features

- Modifier features can refer to a specific property of a CityGML feature which value can be then overridden or replaced by the dynamic value specified in the modifier feature

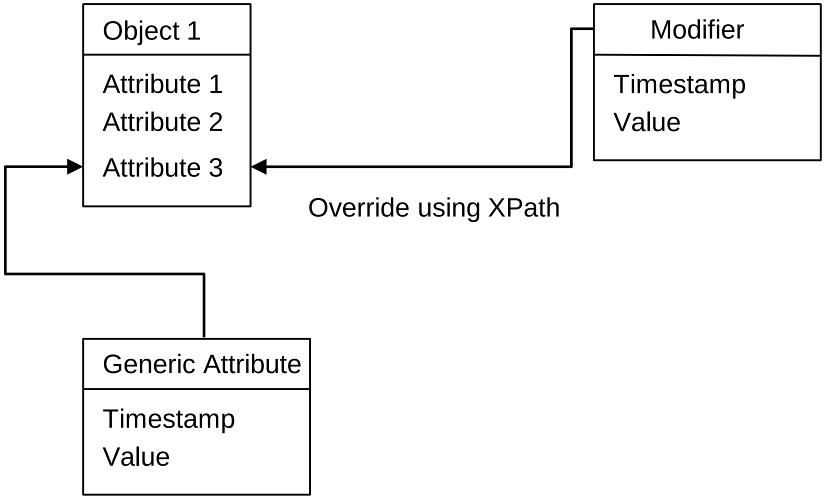


Required Attributes of Modifier feature

- Timestamp
  - Time positions
  - Time Instants
  - Intervals
  - Time periods (begin and end time)
  - Patterns or cycles
- Associated value
  - Value array or table
  - Data block (e.g., CSV)
  - File (e.g., texture images)
  - Sensor Data Web Services (e.g, URI to a web service)
- Mapping function (if required)
  - Modifier should specify how the value for a specific time is being determined

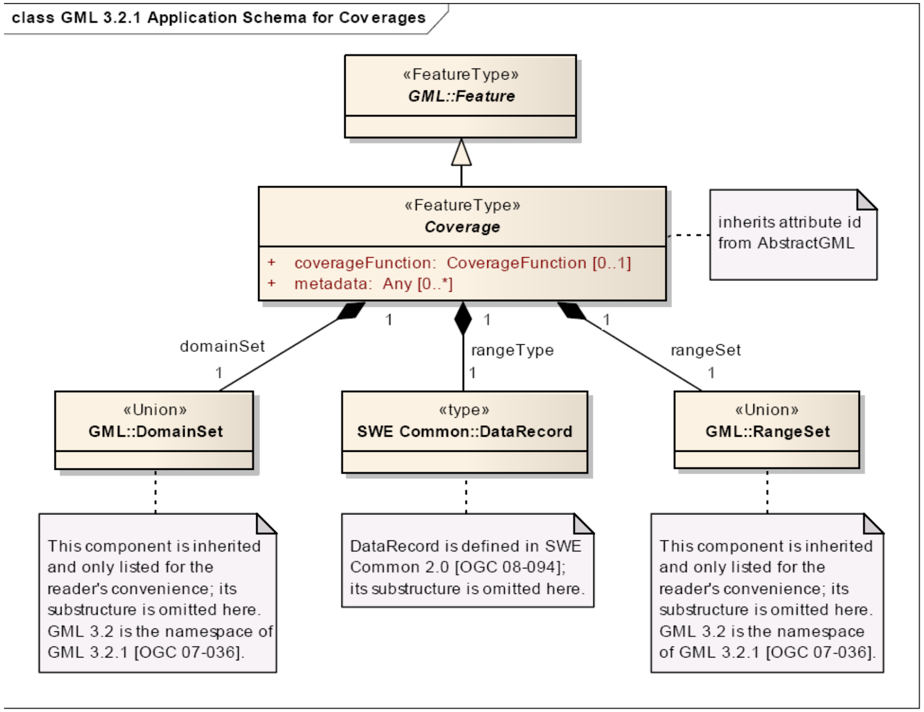
Example scenario

- Dynamic attribute (such as heat demand) within CityGML object can be added as generic attribute which can be replaced using modifiers according to the time range.



GML Coverage

- The proposed modifier approach has close relation with GML Coverage, where spatial coordinates are mapped to a specific value according to a mapping function

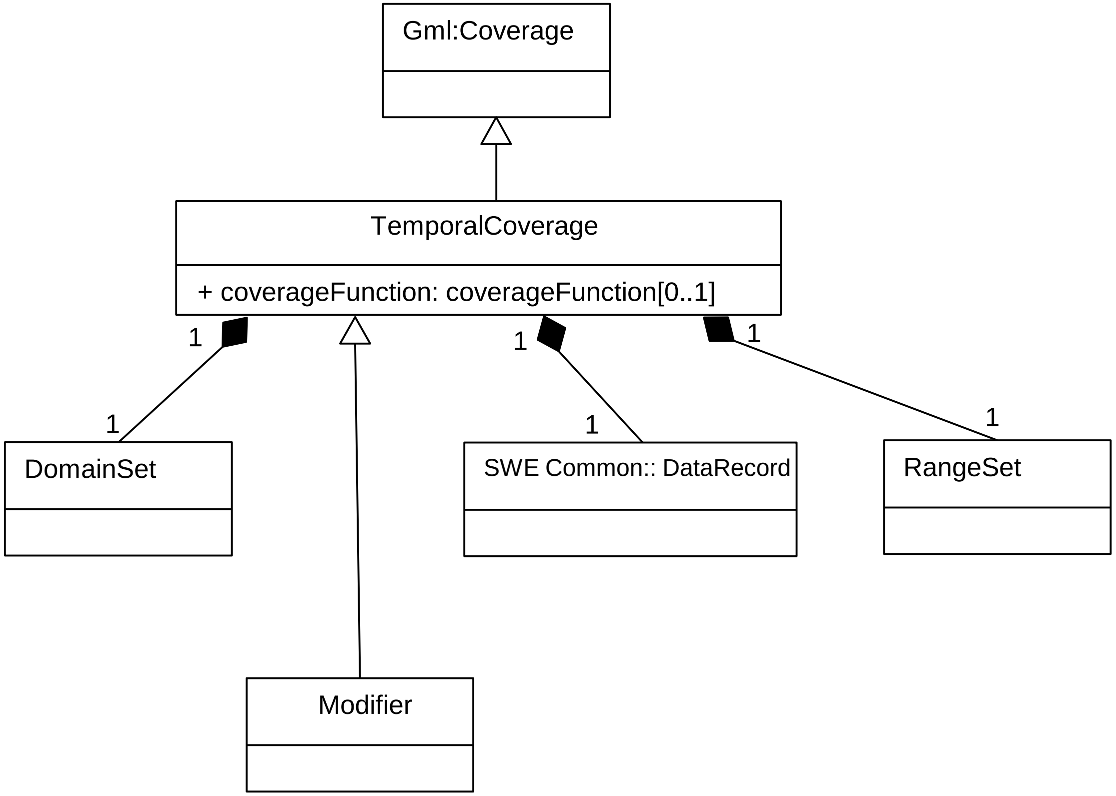

class GML 3.2.1 Application Schema for Coverages
«FeatureType»
GML::Feature
«FeatureType»
Coverage
+ coverageFunction: CoverageFunction [0..1]
+ metadata: Any [0..*]
inherits attribute id from AbstractGML
1
1
1
domainSet
rangeType
rangeSet
1
1
1
«Union»
GML::DomainSet
«type»
SWE Common::DataRecord
«Union»
GML::RangeSet
This component is inherited and only listed for the reader's convenience; its substructure is omitted here. GML 3.2 is the namespace of GML 3.2.1 [OGC 07-036].
DataRecord is defined in SWE Common 2.0 [OGC 08-094]; its substructure is omitted here.
This component is inherited and only listed for the reader's convenience; its substructure is omitted here. GML 3.2 is the namespace of GML 3.2.1 [OGC 07-036].

Source: OGC 09-146r1 GML Application Schema - Coverages

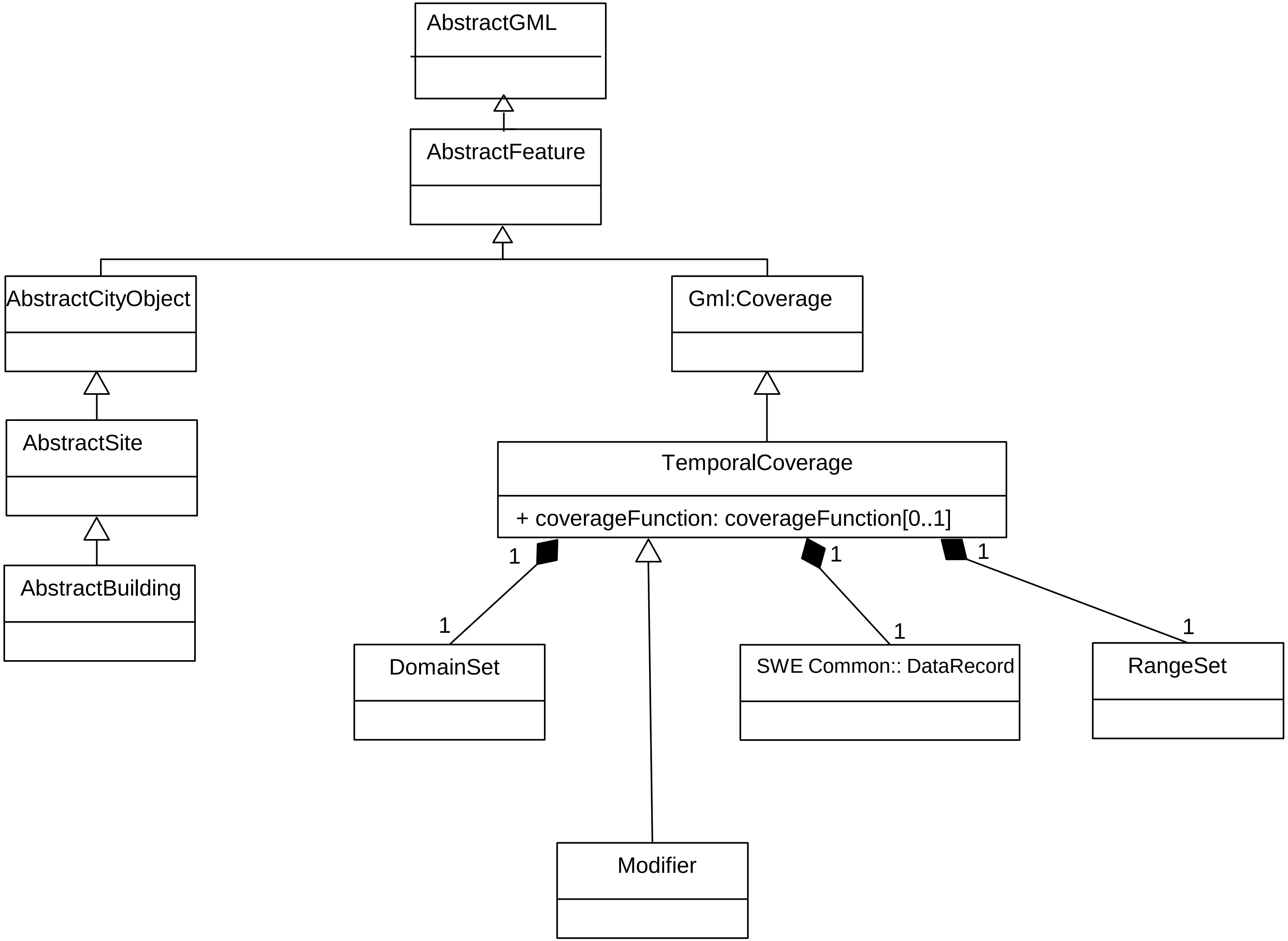
Temporal Coverage

- GML coverage can be extended to create temporal coverage where the new temporal domain set will support time objects and the values in the range set can be mapped to the time values in the domain set.
  - Domain Set
    - Can support gml:AbstractTimeObject such as gml:TimeInstant or gml:TimePeriod
    - GML 3.2 does not support periodic/cyclic patterns. However, AIXM supports schedulers, which might be helpful here.
  - Range Set
    - Value array
    - Scalar value lists
    - Data blocks (in the form of Tuple Lists such as CSV)
    - Files (Absolute file, File references, or MIME Types)
  - Coverage function
    - Mapping rule (table etc)
    - String or URI
    - Linear (if not defined)

Temporal Coverage



Temporal Coverage



Example Instance Data

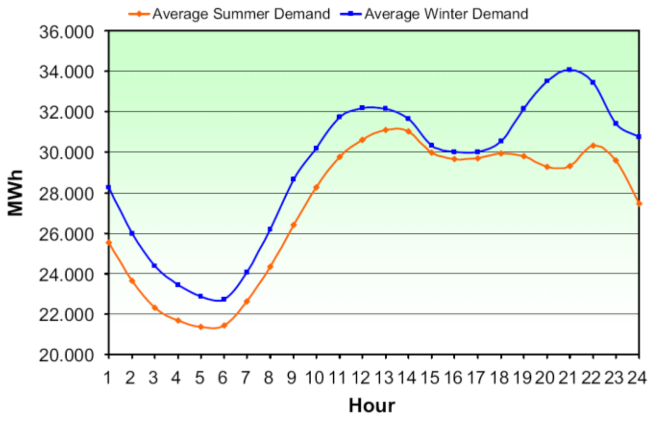
```
<cityObjectMember>
  <Building gml:id = "building1">
        <gen:dateAttribute name = "TimeInstants">
              <gen:value xlink:href = "//identifier[text()='b1_TimeInstant']"/>
        </gen:dateAttribute>
        <gen:stringAttribute name = "HeatDemands">
              <gen:value xlink:href = "//identifier[text()='b1_HeatDemand']"/>
        </gen:stringAttribute>
  </Building>
</cityObjectMember>
<TemporalCoverage>
  <TimeDomain>
        <MultiTimeInstants>
              <TimeMembers>
                    <gml:TimeInstant gml:id = "building1_T1">
                          <gml:identifier>b1_TimeInstant</gml:identifier>
                          <gml:timePosition>2013-01-01</gml:timePosition>
                    </gml:TimeInstant>
                    <gml:TimeInstant gml:id = "building1_T2">
                          <gml:identifier>b1_TimeInstant</gml:identifier>
                          <gml:timePosition>2013-02-01</gml:timePosition>
                    </gml:TimeInstant>
              </TimeMembers>
        </MultiTimeInstants>
  </TimeDomain>
  <gml:rangeSet>
        <gml:valueArray>
              <gml:valueComponents>
                    <b1_HeatDemand>135</b1_HeatDemand>
                    <b1_HeatDemand>144</b1_HeatDemand>
              </gml:valueComponents>
        </gml:valueArray>
  </gml:rangeSet>
</TemporalCoverage>
```

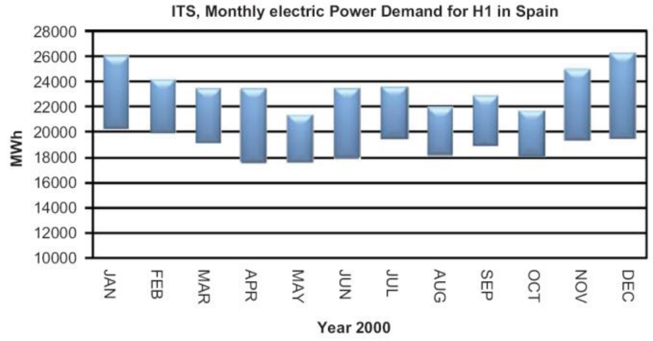
Reference to dynamic attributes using XPath

Mapping of time domain and range values using temporal coverage

Use case 1–Energy demand using time intervals

- As described in [1], the demand forecasting can be done depending on the time horizon selected
  - Short term from 1 hour to 1 week
  - Medium term from a week to a year



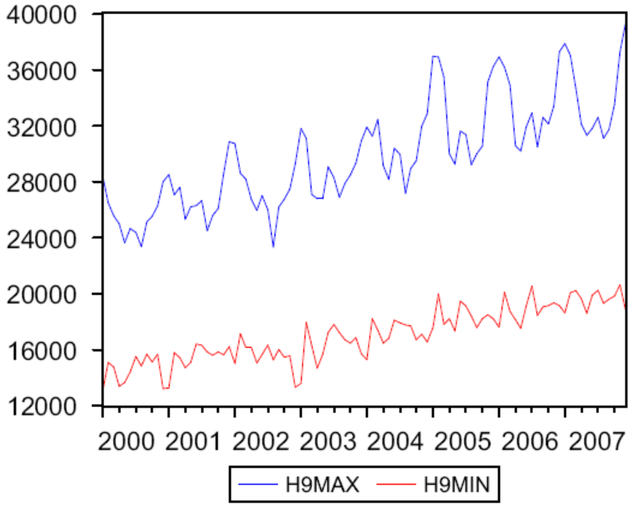


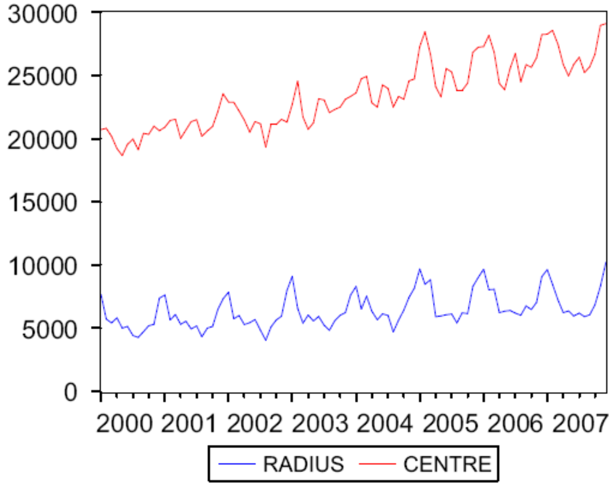
Upper Bound	26091	24219	23513	23491	21373	23445	23609	22008	22842	21768	24956	26302
Lower Bound	20328	19920	19137	17679	17582	18000	19494	18198	18909	18141	19342	19523
	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC

Source:[1]
C. García-Ascanio and C. Maté, “Electric power demand forecasting using interval time series: A comparison between VAR and iMLP,”*Energy Policy*, vol. 38, no. 2, pp. 715–725, Feb. 2010

Use case 1–Energy demand using time intervals

- Long term for more than a year



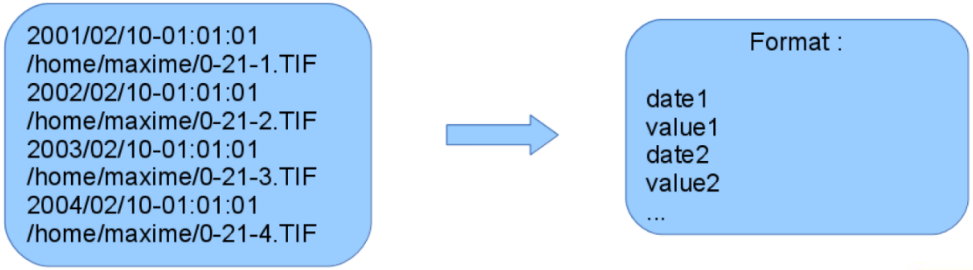


- ▸ All the three scenarios can be mapped within temporal coverage using gml:TimeInstant or gml:TimePeriods

Source:[1]
C. García-Ascanio and C. Maté, “Electric power demand forecasting using interval time series: A comparison between VAR and iMLP,”*Energy Policy*, vol. 38, no. 2, pp. 715–725, Feb. 2010

Use case 2:Evolution of building texture during time

- [2] describes the use case of texture temporalization, showing the evolution of tempurature
- The different TIFF images have been used as textures with respect to different time instants

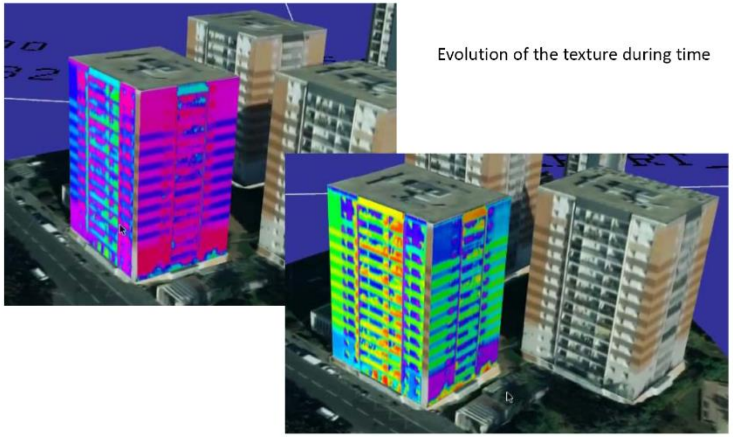


2001/02/10-01:01:01
/home/maxime/0-21-1.TIF
2002/02/10-01:01:01
/home/maxime/0-21-2.TIF
2003/02/10-01:01:01
/home/maxime/0-21-3.TIF
2004/02/10-01:01:01
/home/maxime/0-21-4.TIF

Format :

date1
value1
date2
value2
...

- The result can be shown as



Evolution of the texture during time

Source:[2]
https://github.com/opengeospatial/CityGML-3.0/blob/master/WP%2006%20Resources/Meetings/2nd/2014_09_08_Temporal_CityGML_Gesquiere.pdf

Use case 2:Evolution of building texture during time

- Example – the mapping of files with time values can be used in <<texture>> attribute

```
<TemporalCoverage>
  <TimeDomain>
           <MultiTimeInstants>
                        <TimeMembers>
                                  <gml:TimeInstant gml:id = "building1_T1">
                                            <gml:identifier>b1_TimeInstant</gml:identifier>
                                            <gml:timePosition>2001-02-10 01:01:01</gml:timePosition>
                                  </gml:TimeInstant>
                                  <gml:TimeInstant gml:id = "building1_T2">
                                            <gml:identifier>b1_TimeInstant</gml:identifier>
                                            <gml:timePosition>2002-02-10 01:01:01</gml:timePosition>
                                  </gml:TimeInstant>
                        </TimeMembers>
           </MultiTimeInstants>
  </TimeDomain>
  <gml:rangeSet>
           <gml:File>
                        <gml:fileReference>/home/maxime/0-21-1.TIF </gml:fileReference>
                        <gml:fileReference>/home/maxime/0-21-2.TIF </gml:fileReference>
           </gml:File>
  </gml:rangeSet>
</TemporalCoverage>
```

Open questions

- How can we reference an OGC Web Sensor Observation Service using temporal coverage or any other approach in GML?
  - The aim can be to retrieve sensor data using a URL, for example, http://opendap.co-ops.nos.noaa.gov/ioos-dif-sos/SOS?service=SOS&request=GetObservation&version=1.0.0&observedProperty=air_temperature&offering=urn:ioos:station:NOAA.NOS.CO-OPS:8454000&responseFormat=text%2Fcsv&eventTime=2015-03-12T00:00:00Z/2015-03-12T23:59:00Z
- How can we add periodic/cyclic patterns in coverage domain set?
  - GML 3.2 does not support periodic/cyclic patterns. However, AIXM supports schedulers, which might be helpful here.
- This approach is suitable for dynamic changes in attributes. How can we handle the changes in spatial positions (for example, moving features)?
- Should we use existing GML coverages or create separate coverage features within CityGML?