Dynamic Data Schema - CityGML 3.0

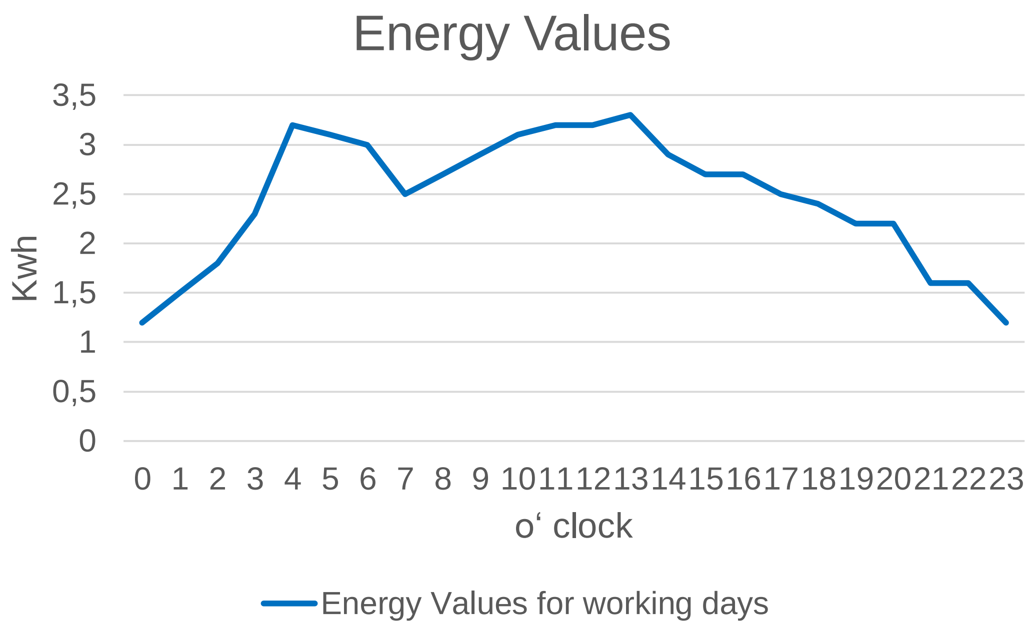
Kanishk Chaturvedi

Chair of Geoinformatics
Technische Universität München

kanishk.chaturvedi@tum.de

Need for supporting patterns

- As presented in 10th WP6 meeting, the Timeseries allows supporting absolute start and end points
  - Within which the attribute values can be mapped
  - Can be represented as tabulation of measured data
- For example, mapping of energy values of a building for every hour in a day

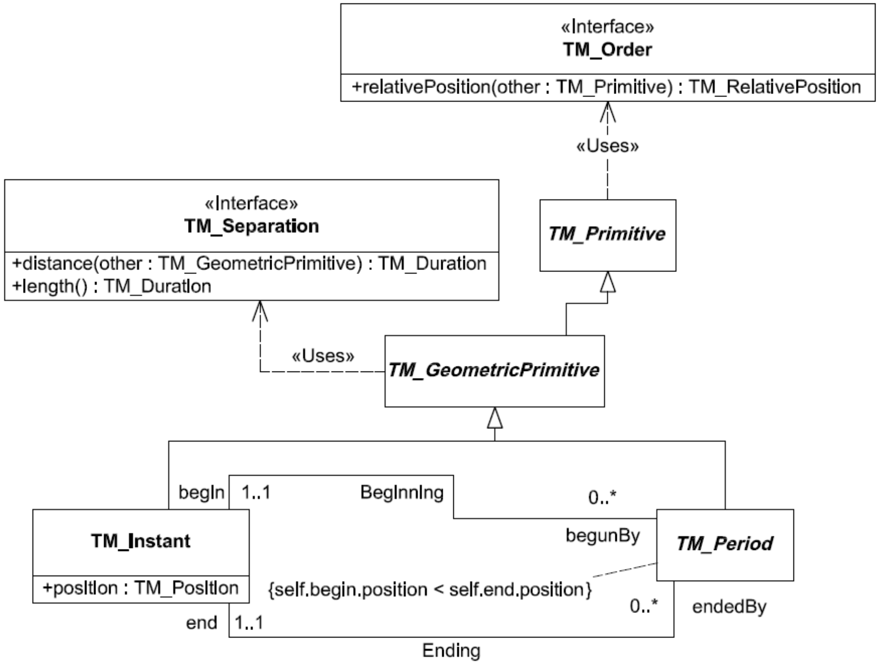


Need for supporting patterns

- However, in many applications, it is not sufficient just to provide a means for the tabulation of absolute time-value pairs.
  - They may require patterns to represent dynamic variations of properties based on statistics and general rules.
  - For example, during energy demand estimations, the energy values reflect specific patterns for individual weekdays and weekends.
- Such patterns should also be supported by the proposed dynamic data schema.
  - This requires to use relative time points.

Key Points (Absolute time)

- ISO 19108 defines Absolute time points in two ways (TM_Instant and TM_Period)
  - TM_Instant is a primitive that represents position in time. The position can be associated with a single temporal reference system (e.g., Gregorian Calendar).
  - However, it may allow to calculate a relative time point with respect to TM_Instant
    - e.g., 1 Day w.r.t. TM_Instant (and not w.r.t. Gregorian Calendar)



Source : [ISO 19108 Temporal Schema]

Key Points (Absolute time)

- Possible representations of Absolute Time
  - Time_Instant (defines time position)
    - E.g., 2015-05-22T13:00:00 (Timestamp)
    - 2015-05-22 (yyyy-mm-dd)
    - 2015-05 (yyyy-mm)
    - 2015 (yyyy)
    - Monday?
  - Time_Period (having begin and end time positions)
    - 2015-05-01 to 2015-05-31
    - Possible to determine the length of the period or the temporal distance between begin and end points

Key Points (Absolute time)

- Relating Absolute Time Points
  - The operation TM_Order in ISO 19108 is used to determine position of a time relative to another time position.
  - These relative positions are based on the 13 temporal relationships identified by Allen[1]. Hence, it allows to perform comparitive operations on time periods.
  - However, metric or arithmetic operations can be very beneficial in defining the patterns in our schema.
    - E.g. 1-July-2015 + 1 Month = 1-August-2015
    - Or, 1-July-2015 07:00:00 + 1 Hour = 1-July-2015 08:00:00
  - Such features are already available in Databases (such as Oracle), but not defined in ISO 19108 or GML. However, there is a mention about adding such features within the scope of Temporal DWG[2].
  - ISO 19108 or GML can be extended to include such arithmetic operations.

Source[1] : ALLEN, J. F., Maintaining Knowledge about Temporal Intervals, Communications of the ACM, 1983, vol. 26 pp. 832-843
Source[2] : http://external.opengeospatial.org/twiki_public/TemporalDWG/WebHome

Key Points (Relative time)

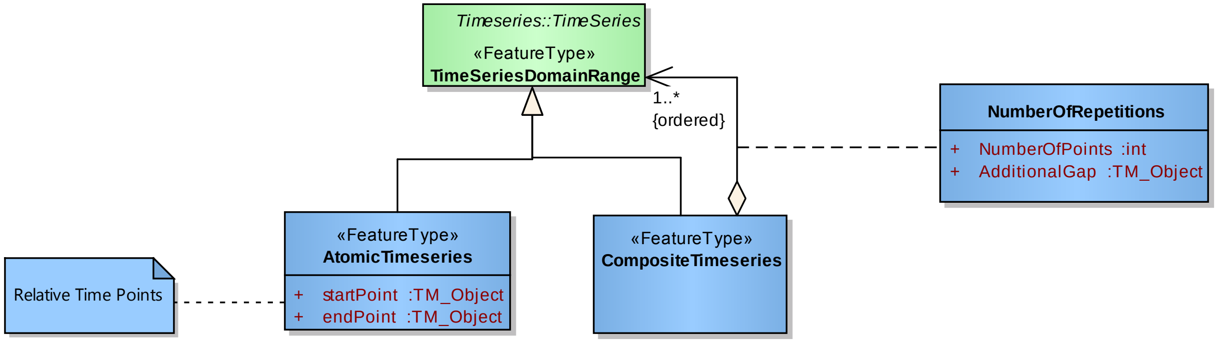
- Relative Time Points
  - ISO 19108 has no direct model support for relative time points.
  - However, we may be able to define ‘local‘/‘relative‘ time referenence system.
    - If we specify a time of 1 hour with respect to such a local system, we can then model
      - which time values should have absolute time points and
      - which time attributes are referring to relative time points.

Starting point of reference

- Repeating patterns within Outlook or Google Calendars
- Allows to schedule meeting repetitions as
  - Daily (Specific point of time every day)
  - Weekly (Specific point of time on specific day(s) every week)
  - Monthly
    - Specific day of the month
    - Specific day of the week
  - Yearly
- Repetiton frequency (E.g. every 1 day or every 2 months etc.)
- Beginning of the repetitions
  - Absolute time point
- Termination of the repetitions, defining
  - Number of occurences (after which the repetition would stop)
  - Specific Date (after which the repetition would stop)

Patterns

- Patterns can be defined by extending TimeSeriesDomain as
  - Atomic Timeseries, having
    - startPoint (Relative Time Point)
    - endPoint (Relative Time Point)
  - Composite Timeseries
    - Allows a TimeSeries to consist of multiple ordered Atomic Timeseries and also composite Timeseries
    - Allows to define specific patterns
    - Association Class ‘NumberOfRepetitions‘ defines
      - the total number of repetitions
      - Also defines a gap allowing to join multiple time series



Patterns

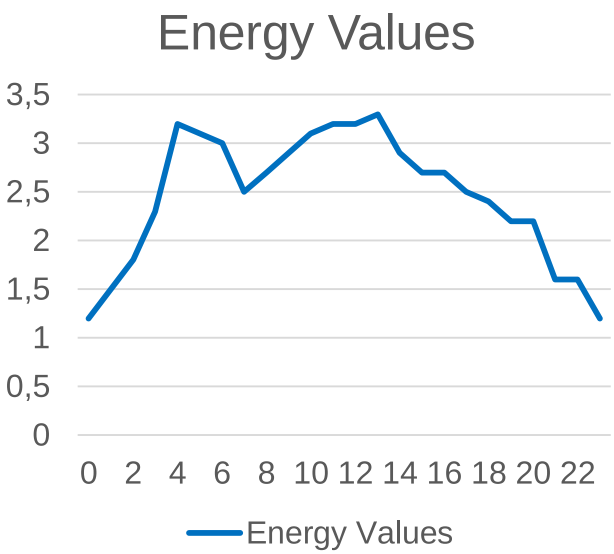
- The composite timeseries may reflect specific patterns
  - E.g., the daily energy values can be defined using an atomic Timeseries.
  - Further, a composite timeseries of energy values for all days of week can represent a weekly pattern

- Using the coverage approach, these patterns can be defined in the domain range and their values can be defined in the range set.

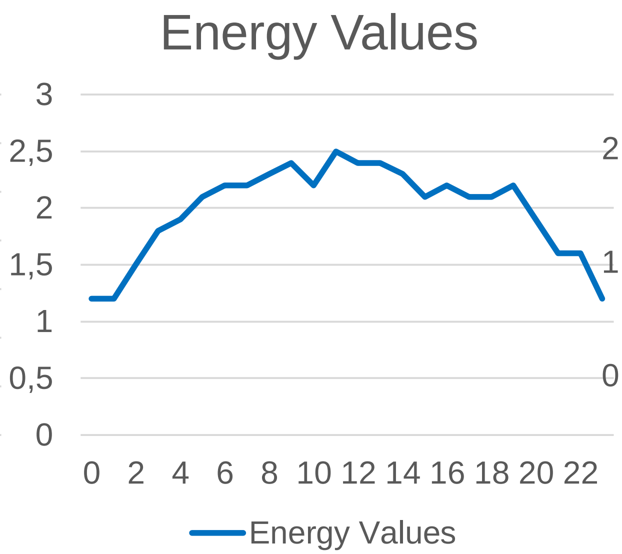
- Advantages:
  - Patterns can have sub-patterns of arbitrary depths
  - Using additional gap, multiple time series can be joined reflecting patterns, e.g.
    - A pattern of energy values for only weekdays/weekends, or
    - Comparison/pattern of energy demands for summers and winters over a period of 5 years

Examples – Atomic Timeseries

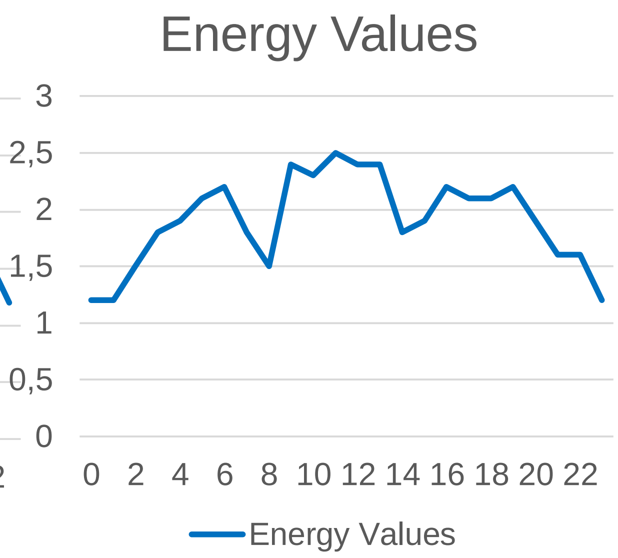
- Atomic Timeseries can be defined once for specific relative time points/series.
- E.g, energy values for a weekday, a Saturday and a Sunday can be defined once as atomic timeseries.



(Working day)



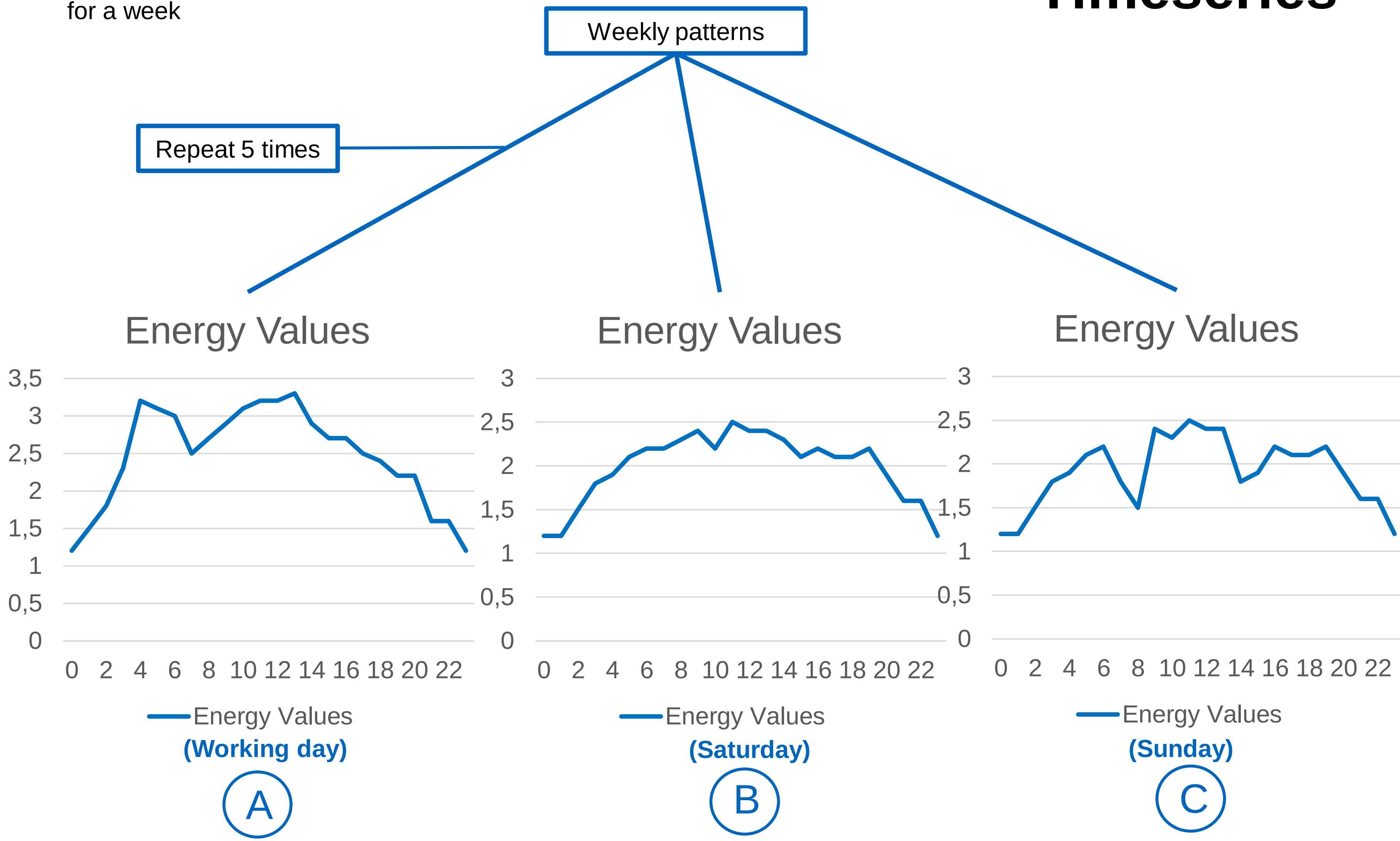
(Saturday)



(Sunday)

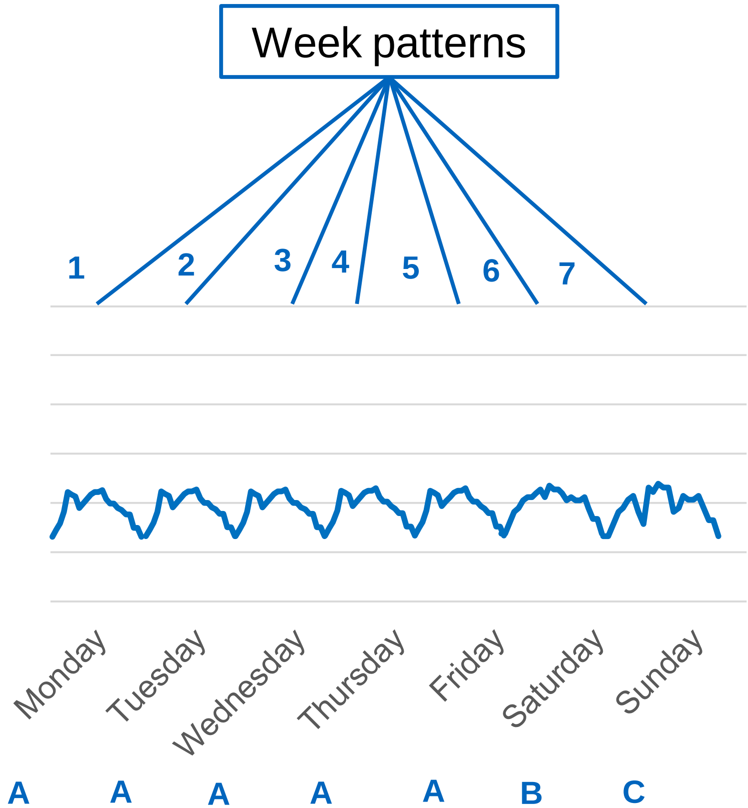
Composite Timeseries

- However, composite timeseries allow ordered repetitions of atomic timeseries for a number of times.
- E.g., the atomic timeseries of a weekday can have 5 repetitions to obtain the patterns for a week



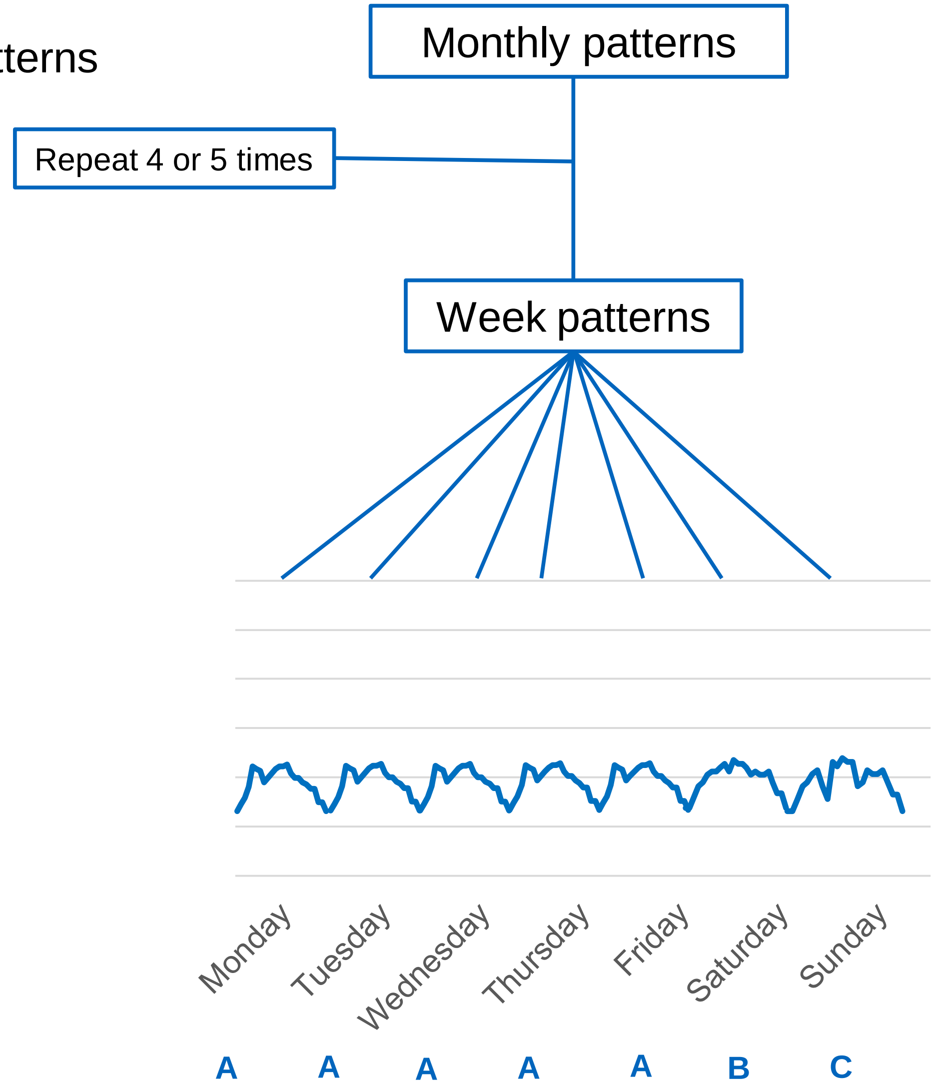
Composite Timeseries (Complex patterns)

- Weekly patterns, consisting of
  - Five patterns for weekday
  - Two patterns for weekend (Saturday and Sunday)



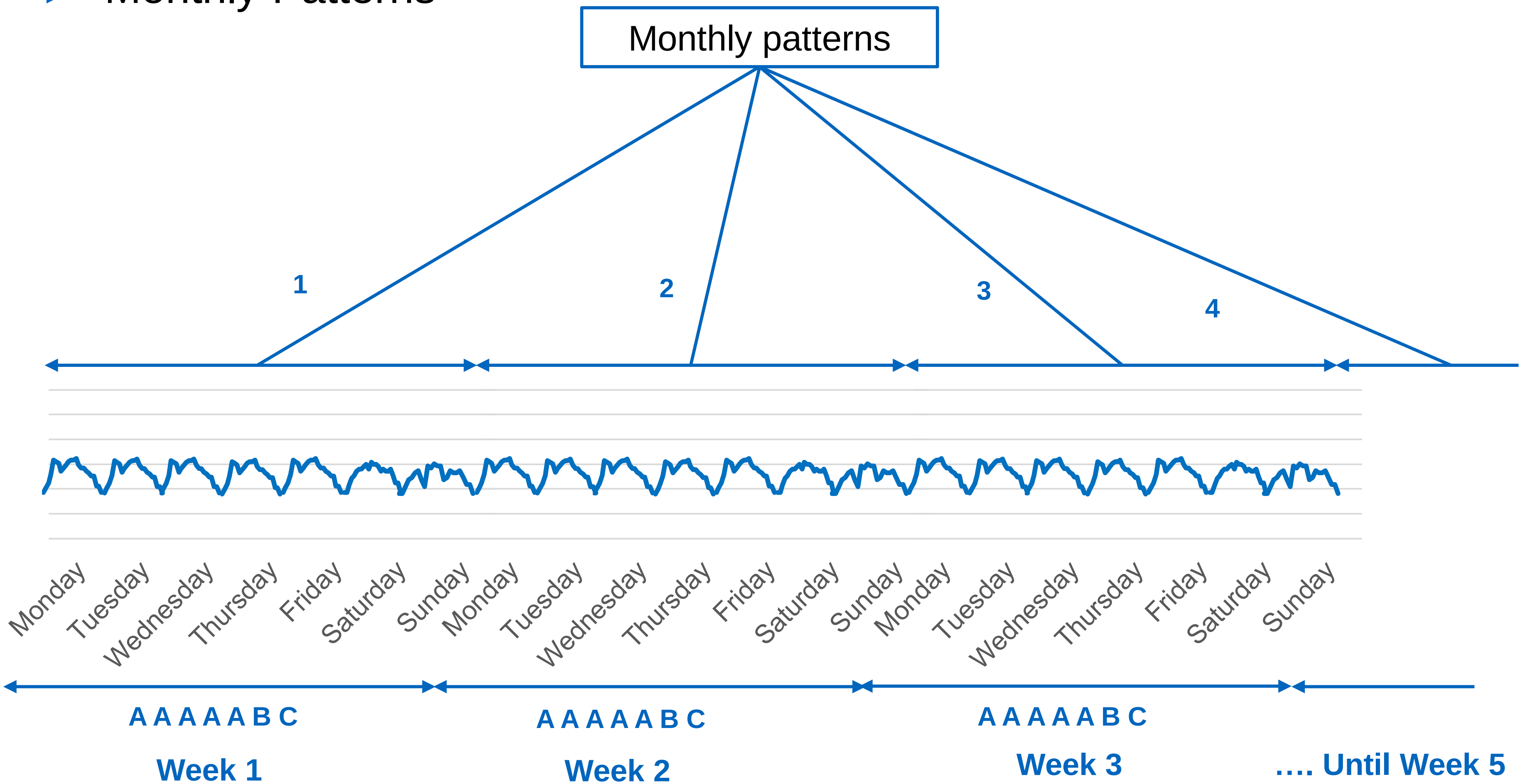
Composite Timeseries (Complex patterns)

- Monthly Patterns



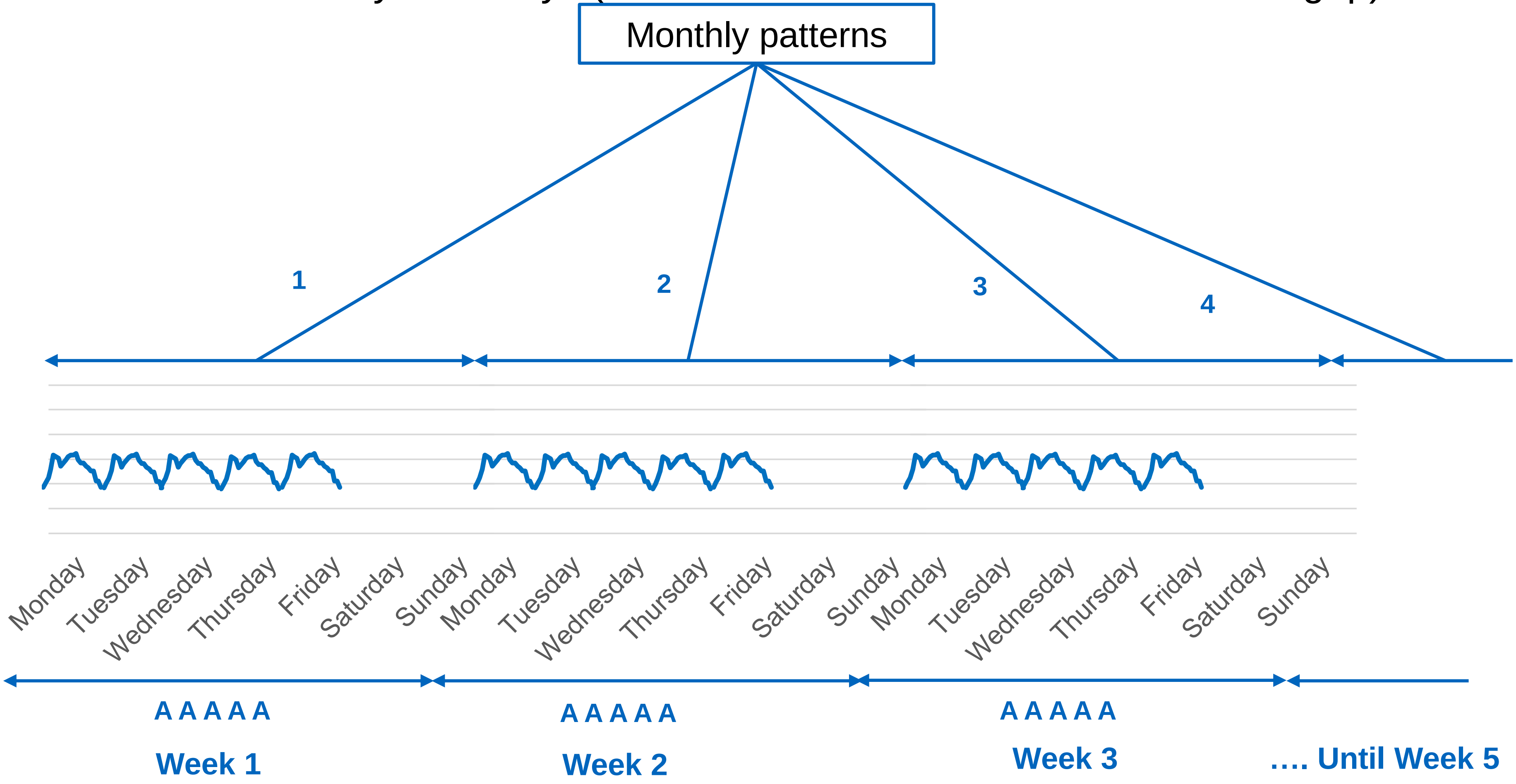
Composite Timeseries (Complex patterns)

- Monthly Patterns



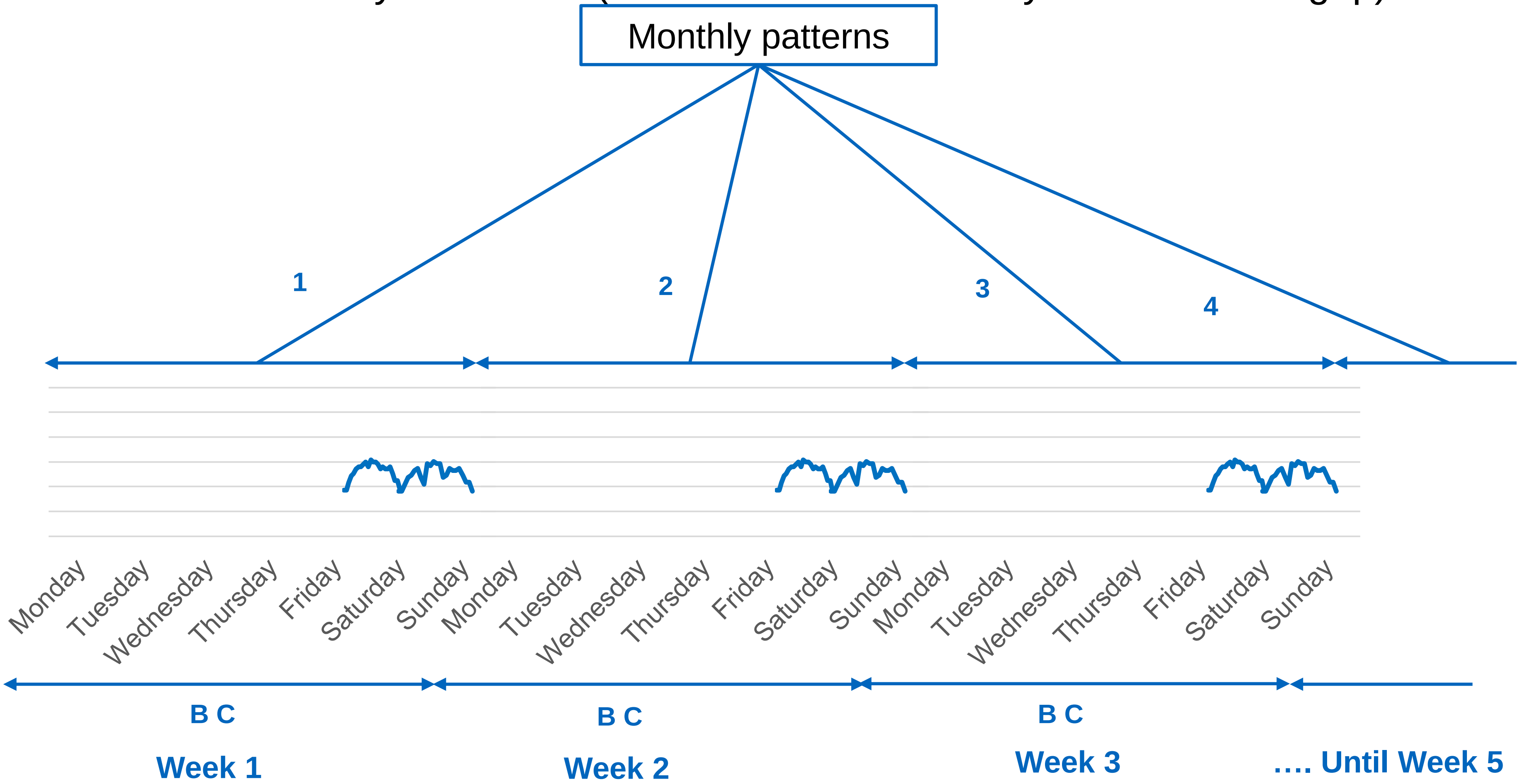
Composite Timeseries (Complex patterns)

- Patterns for only weekdays (blank values of weekends as additional gap)



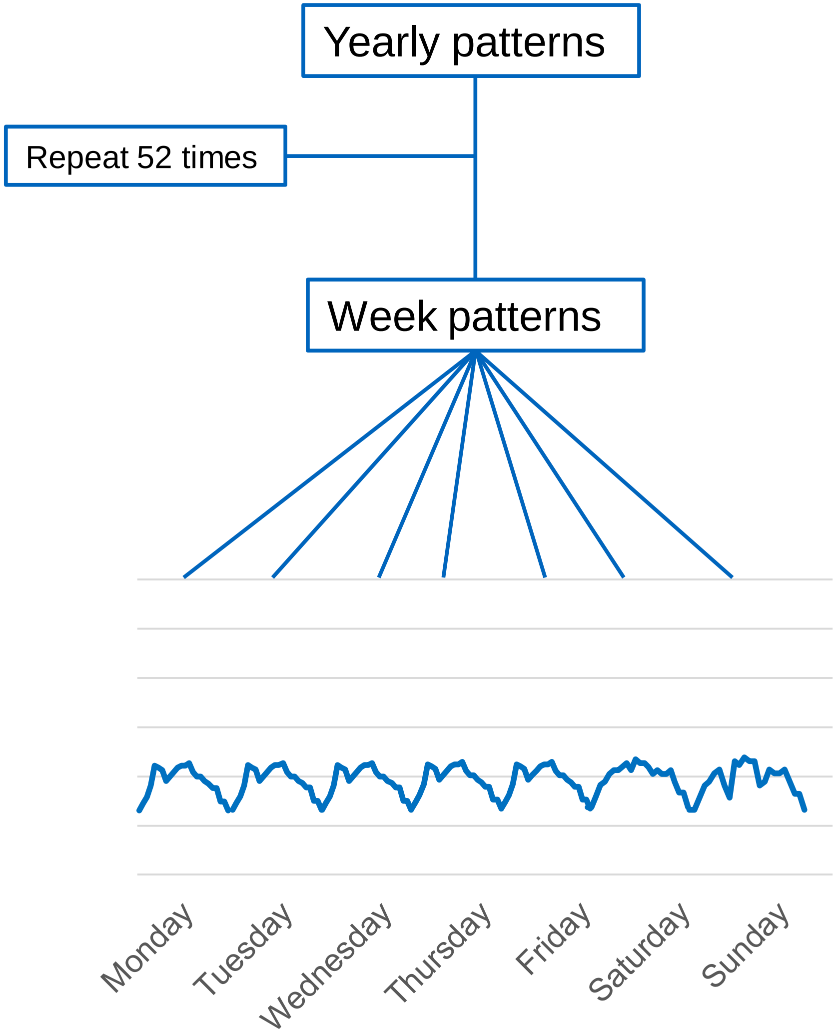
Composite Timeseries (Complex patterns)

- Patterns for only weekends (blank values of weekdays as additional gap)



Composite Timeseries (Complex patterns)

- Yearly Patterns



Dynamizers

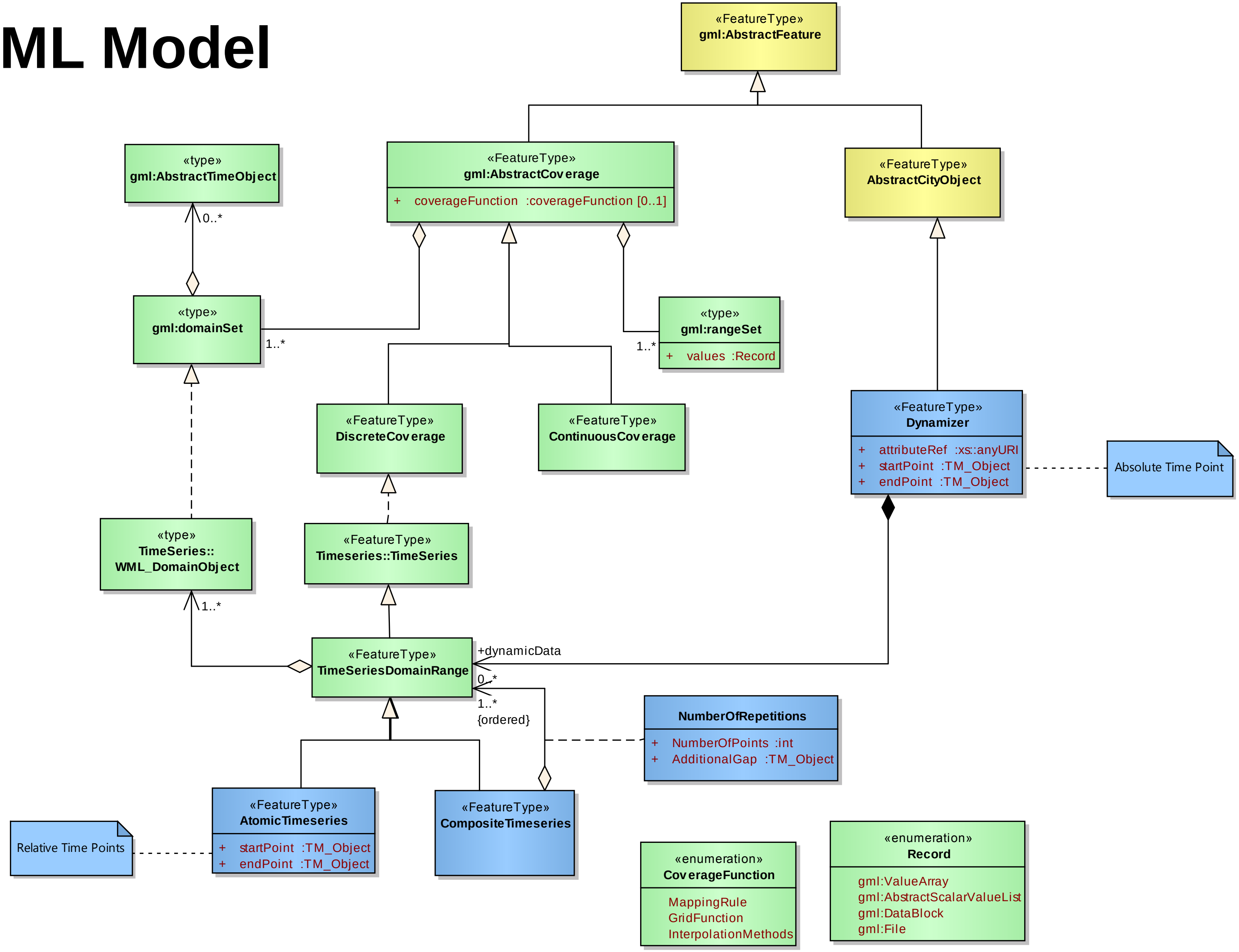
- Dynamizer is a feature which would allow overriding city attributes by time-dynamic attributes
- Dynamizer realizes GML Coverages, consisting
  - Temporal domain set
  - Range set (having attribute values)
    - Can be value arrays, scalar value lists, data blocks or external files
  - Coverage Function, which maps the time values in the domain set to the range values according to a function
    - Can be defined according to mapping rule or grid functions.
    - In case of continuous coverages, interpolation methods can also be defined in coverage functions.
- Using XPath mechanism, dynamizer 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 dynamizer feature

What is missing?

- Specification of absolute start and end point
- Determining relative time points from absolute time points
- Feature and attribute whose value should be overridden

- All of these are modeled as properties of a dynamizer feature

UML Model


«FeatureType»
gml:AbstractFeature
«type»
gml:AbstractTimeObject
«FeatureType»
gml:AbstractCoverage
+ coverageFunction :coverageFunction [0..1]
«FeatureType»
AbstractCityObject
0..*
«type»
gml:domainSet
1..*
«type»
gml:rangeSet
+ values :Record
1..*
«FeatureType»
DiscreteCoverage
«FeatureType»
ContinuousCoverage
«FeatureType»
Dynamizer
+ attributeRef :xs::anyURI
+ startPoint :TM_Object
+ endPoint :TM_Object
Absolute Time Point
«type»
TimeSeries::
WML_DomainObject
«FeatureType»
Timeseries::TimeSeries
1..*
«FeatureType»
TimeSeriesDomainRange
+dynamicData
0..*
1..*
{ordered}
NumberOfRepetitions
+ NumberOfPoints :int
+ AdditionalGap :TM_Object
Relative Time Points
«FeatureType»
AtomicTimeseries
+ startPoint :TM_Object
+ endPoint :TM_Object
«FeatureType»
CompositeTimeseries
«enumeration»
CoverageFunction
MappingRule
GridFunction
InterpolationMethods
«enumeration»
Record
gml:ValueArray
gml:AbstractScalarValueList
gml:DataBlock
gml:File

Future Work

- First approach to link sensors to a city object