3DCityDB extension for the CityGML Energy ADE 0.8

(PostgreSQL version)

Giorgio Agugiaro

23 May 2017

7th CityGML Energy ADE Workshop, Grenoble, France

giorgio.agugiaro@ait.ac.at
Smart Cities and Regions Research Field
Center for Energy
AIT - Austrian Institute of Technology
Vienna, Austria

Motivation & goals

- The CityGML Energy ADE has reached an acceptable level of maturity

- Growing demand for an ADE-aware DB solution (not only Energy ADE!)
  - Work in progress by the 3DCityDB development team
    - Automatic mapping from OO to ER model
    - Automatic generation of DB schema
    - Extension of the Importer/Exporter
    - …
    - All these fantastic goodies for *any* ADE!
    - BUT: it will take time till it is ready

- So far, lack of existing DB implementations for the Energy ADE:
  - partial AND/OR
  - non-open AND/OR
  - poorly or not documented at all

Motivation & goals

- Implementation of the Energy ADE for PostgreSQL
  - (Manual) mapping from OO to ER
  - Complete implementation (for PostgreSQL) of v. 0.8
  - Particular care of documentation
  - Released soon under to Apache 2.0 license
- Gather experience on how to extend to 3DCityDB
- Offer a shared starting point to collect feedback, experiences, suggestions…
- Contribute to foster adoption and further development of the Energy ADE

Design criteria (excerpt)

- Build upon the existing objects of the 3DCityDB… but keep the original ("vanilla") untouched (for the sake of the Importer/Exporter)
- Define a non-concurrent way of extending the 3DCityDB with *any* ADEs (e.g. Energy ADE + UtilityNetwork ADE)
- Stay close to the original “style” of the 3DCityDB when it comes to tables, constraints, naming conventions, data types, etc.
- Possibly keep the number of new tables in check
- Implementation for PostgreSQL, but avoid potential technology lock-ins for future conversions to other DBs (as far as possible)

Implementation steps

- Define and agree upon rules to make the 3DCityDB "ADE-compatible"
  - Enable to "register" *any* ADE
    - Add a metadata module
    - Add functions to help installing/removing an ADE

Implementation steps

- Define and agree upon rules to make the 3DCityDB "ADE-compatible"
  - Enable to "register" *any* ADE
    - Add a metadata module
    - Add functions to help installing/removing an ADE
  - Define rules how to map ADE-classes to new/existing tables
    - Adopt naming convention for new DB entities

Implementation steps

- Define and agree upon rules to make the 3DCityDB "ADE-compatible"
  - Enable to "register" *any* ADE
    - Add a metadata module
    - Add functions to help installing/removing an ADE
  - Define rules how to map ADE-classes to new/existing tables
    - Adopt naming convention for new DB entities
  - Add an ADE-hook mechanism to certain existing functions. E.g.:
    - delete_building() → must work also with ADE-AbstractBuilding
    - delete_cityobject() → must work also with new CityObjects
    - delete_cityobjectgroup() → must work also with new CityObjects
    - get_envelope_cityobject() → same as above

Implementation steps

- Define and agree upon rules to make the 3DCityDB "ADE-compatible"
  - Enable to "register" *any* ADE
    - Add a metadata module
    - Add functions to help installing/removing an ADE
  - Define rules how to map ADE-classes to new/existing tables
    - Adopt naming convention for new DB entities
  - Add an ADE-hook mechanism to certain existing functions. E.g.:
    - delete_building() → must work also with ADE-AbstractBuilding
    - delete_cityobject() → must work also with new CityObjects
    - delete_cityobjectgroup() → must work also with new CityObjects
    - get_envelope_cityobject() → same as above
  - (Enable/extend existing tools to be ADE-compatible: citygml4j, Importer/Exporter, etc.)

- All rules are agreed upon within the 3DCityDB development team and are being implemented for the next 3DCityDB release

Implementation steps

- Define and agree upon rules to make the 3DCityDB "ADE-compatible"
  - Enable to "register" *any* ADE
    - Add a metadata module
    - Add functions to help installing/removing an ADE
  - Define rules how to map ADE-classes to new/existing tables
    - Adopt naming convention for new DB entities
  - Add an ADE-hook mechanism to certain existing functions. E.g.:
    - delete_building() → must work also with ADE-AbstractBuilding
    - delete_cityobject() → must work also with new CityObjects
    - delete_cityobjectgroup() → must work also with new CityObjects
    - get_envelope_cityobject() → same as above
  - (Enable/extend existing tools to be ADE-compatible: citygml4j, Importer/Exporter, etc.)

- All rules are agreed upon within the 3DCityDB development team and are being implemented for the next 3DCityDB release

Metadata module

(as of 22 May 2017)

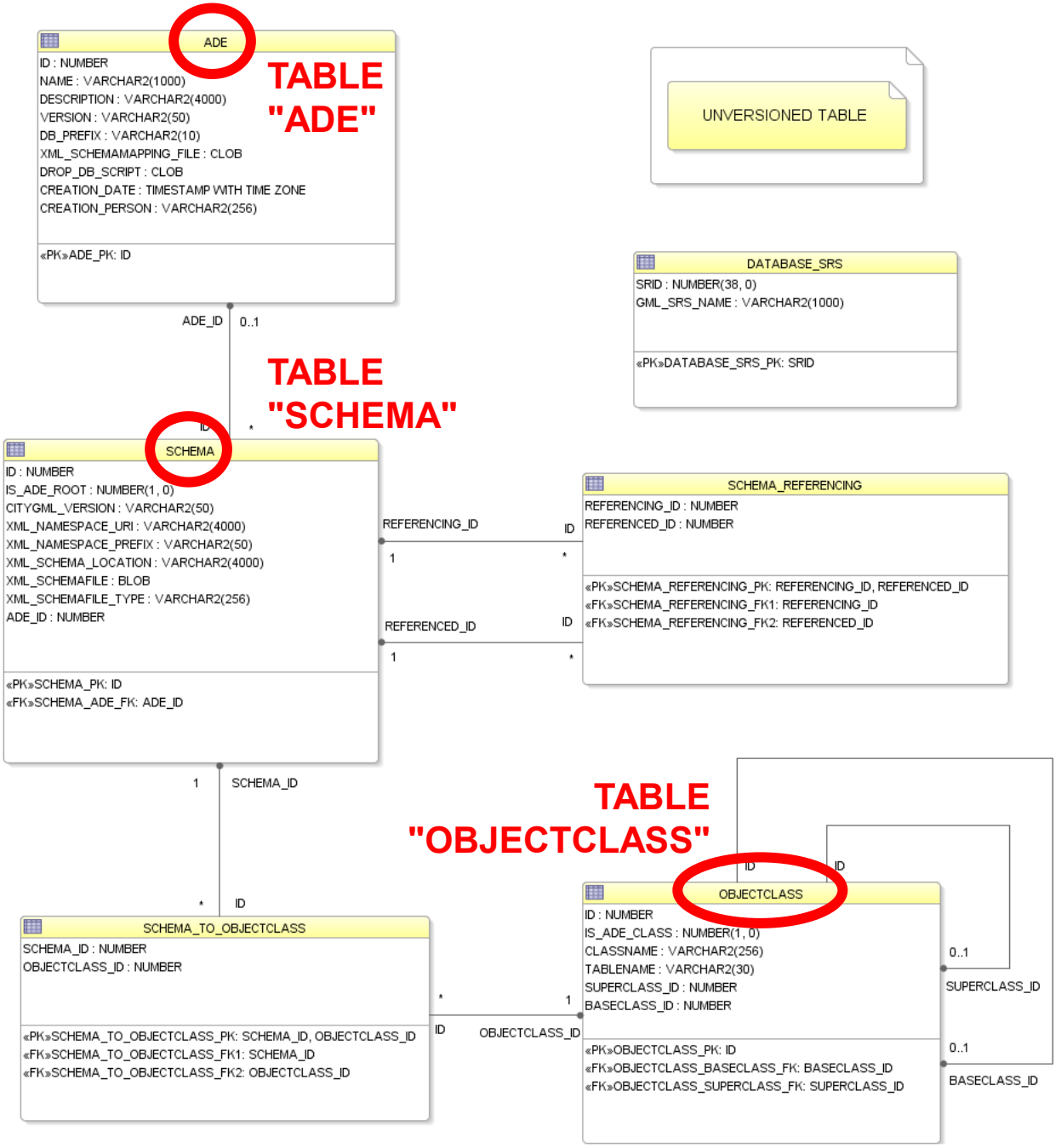


Image source: Courtesy of Z. Yao, TU München

Edit Data - PostgreSQL 9.3 (localhost:5432) - energy_db2 - ade

File Edit View Tools Help

No limit

	id [PK] serial	name character varying(1000)	version character varying(50)	description character varying(4000)	db_prefix character varying(10)
1	1	Energy ADE	0.8	Energy Application Domain Extension v. 0.8	nrg8a
2	2	Energy ADE	0.9	Energy Application Domain Extension v. 0.9	nrg9
3	3	UtilityNetworks ADE	0.9	Utility Network Application Domain Extension v.	utn
4	4	Dynamizer ADE	1.0	Dynamizer Application Domain Extension v. 1.0	dyn
*					

TABLE "ADE"

TABLE "SCHEMA"

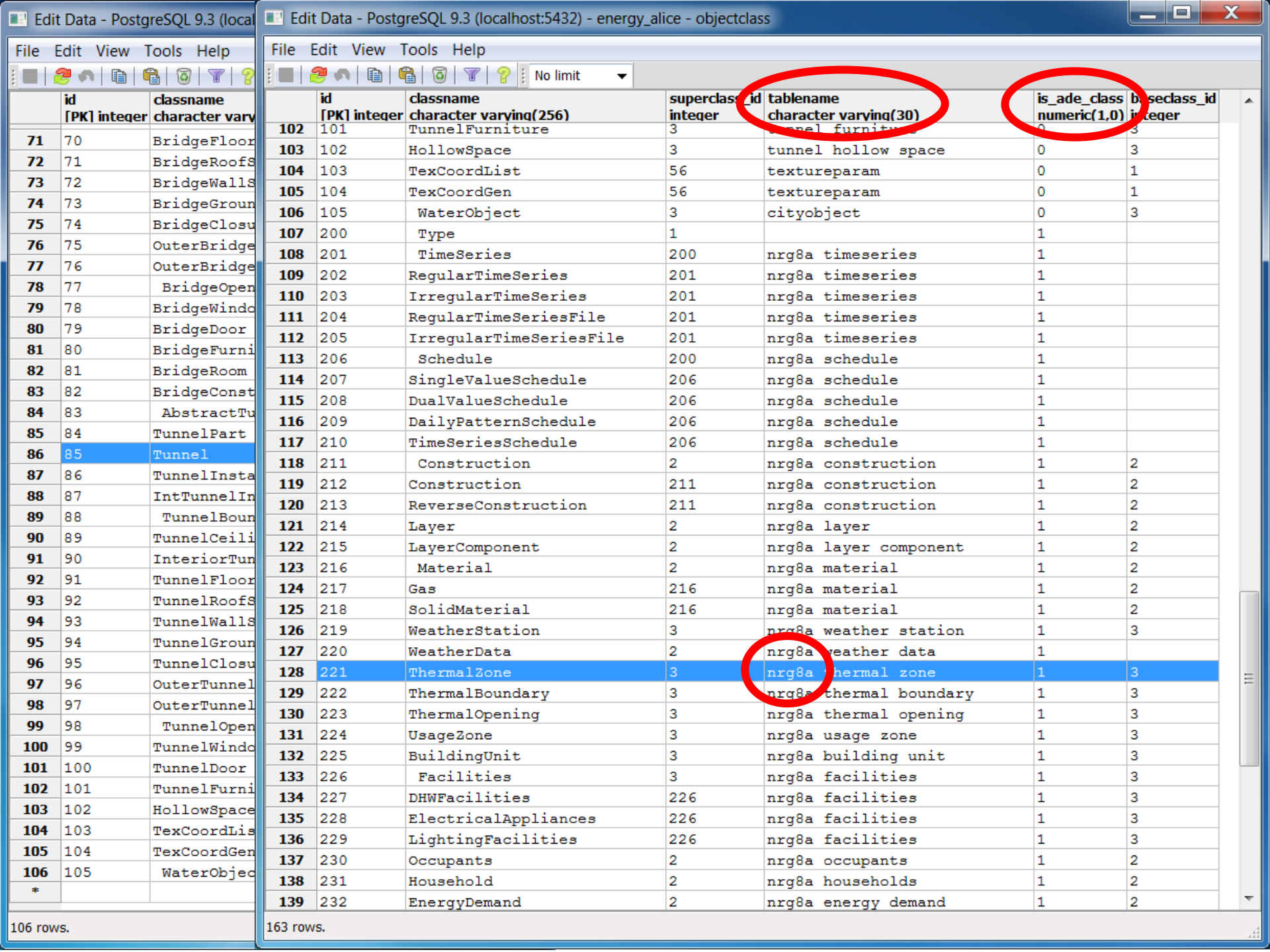
Edit Data - PostgreSQL 9.3 (localhost:5432) - energy_db2 - schema

File Edit View Tools Help

No limit

	id [PK] serial	ade_id integer	is_ade_root numeric(1,0)	citygml_version character varying(50)	xml_namespace_uri character varying(4000)	xml_names character v	xml_schema_location character varying(4000)	xml_sche bytea
1	2				http://www.opengis.net/gml	gml	http://schemas.opengis.net/gml/3.1.1/base/gml.xsd	<binary
2	3				urn:oasis:names:tc:ciq:xsdschema:xAL:2.0	xAL	http://docs.oasis-open.org/election/external/xAL.xsd	<binary
3	4				http://www.ascc.net/xml/schematron	sch		<binary
4	5			1.0	http://schemas.opengis.net/citygml/1.0	core	http://schemas.opengis.net/citygml/1.0/cityGMLBase.xsd	<binary
5	6			1.0	http://schemas.opengis.net/citygml/appearance/1.0	app	http://schemas.opengis.net/citygml/appearance/1.0/appearan	<binary
6	7			1.0	http://schemas.opengis.net/citygml/building/1.0	bldg	http://schemas.opengis.net/citygml/building/1.0/building.x	<binary
7	8			1.0	http://schemas.opengis.net/citygml/cityfurniture/1.0	frn	http://schemas.opengis.net/citygml/cityfurniture/1.0/cityF	<binary
8	9			1.0	http://schemas.opengis.net/citygml/cityobjectgroup/1.0	grp	http://schemas.opengis.net/citygml/cityobjectgroup/1.0/cit	<binary
9	10			1.0	http://schemas.opengis.net/citygml/generics/1.0	gen	http://schemas.opengis.net/citygml/generics/1.0/generics.x	<binary
10	11			1.0	http://schemas.opengis.net/citygml/landuse/1.0	luse	http://schemas.opengis.net/citygml/landuse/1.0/landUse.xsd	<binary
11	12			1.0	http://schemas.opengis.net/citygml/relief/1.0	dem	http://schemas.opengis.net/citygml/relief/1.0/relief.xsd	<binary
12	13			1.0	http://schemas.opengis.net/citygml/transportation/1.0	tran	http://schemas.opengis.net/citygml/transportation/1.0/tran	<binary
13	14			1.0	http://schemas.opengis.net/citygml/vegetation/1.0	veg	http://schemas.opengis.net/citygml/vegetation/1.0/vegetati	<binary
14	15			1.0	http://schemas.opengis.net/citygml/waterbody/1.0	wtr	http://schemas.opengis.net/citygml/waterbody/1.0/waterBody	<binary
15	16			1.0	http://schemas.opengis.net/citygml/texturedsurface/1.0	tex	http://schemas.opengis.net/citygml/texturedsurface/1.0/tex	<binary
16	17			2.0	http://schemas.opengis.net/citygml/2.0	core	http://schemas.opengis.net/citygml/2.0/cityGMLBase.xsd	<binary
17	18			2.0	http://schemas.opengis.net/citygml/appearance/2.0	app	http://schemas.opengis.net/citygml/appearance/2.0/appearan	<binary
18	19			2.0	http://schemas.opengis.net/citygml/bridge/2.0	brid	http://schemas.opengis.net/citygml/bridge/2.0/bridge.xsd	<binary
19	20			2.0	http://schemas.opengis.net/citygml/building/2.0	bldg	http://schemas.opengis.net/citygml/building/2.0/building.x	<binary
20	21			2.0	http://schemas.opengis.net/citygml/cityfurniture/2.0	frn	http://schemas.opengis.net/citygml/cityfurniture/2.0/cityF	<binary
21	22			2.0	http://schemas.opengis.net/citygml/cityobjectgroup/2.0	grp	http://schemas.opengis.net/citygml/cityobjectgroup/2.0/cit	<binary
22	23			2.0	http://schemas.opengis.net/citygml/generics/2.0	gen	http://schemas.opengis.net/citygml/generics/2.0/generics.x	<binary
23	24			2.0	http://schemas.opengis.net/citygml/landuse/2.0	luse	http://schemas.opengis.net/citygml/landuse/2.0/landUse.xsd	<binary
24	25			2.0	http://schemas.opengis.net/citygml/relief/2.0	dem	http://schemas.opengis.net/citygml/relief/2.0/relief.xsd	<binary
25	26			2.0	http://schemas.opengis.net/citygml/transportation/2.0	tran	http://schemas.opengis.net/citygml/transportation/2.0/tran	<binary
26	27			2.0	http://schemas.opengis.net/citygml/tunnel/2.0	tun	http://schemas.opengis.net/citygml/tunnel/2.0/tunnel.xsd	<binary
27	28			2.0	http://schemas.opengis.net/citygml/vegetation/2.0	veg	http://schemas.opengis.net/citygml/vegetation/2.0/vegetati	<binary
28	29			2.0	http://schemas.opengis.net/citygml/waterbody/2.0	wtr	http://schemas.opengis.net/citygml/waterbody/2.0/waterBody	<binary
29	30			2.0	http://schemas.opengis.net/citygml/texturedsurface/2.0	tex	http://schemas.opengis.net/citygml/texturedsurface/1.0/tex	<binary
30	32	1	1	2.0	http://www.sig3d.org/citygml/2.0/energy/0.8.0	energy	http://www.sig3d.org/citygml/2.0/energy/0.8.0/EnergyADE 0	<binary
31	34	2	1	2.0	http://www.sig3d.org/citygml/2.0/energy/0.9.0	energy	http://www.sig3d.org/citygml/2.0/energy/0.9.0/EnergyADE 0	<binary
32	35	3	1	2.0	namepsace uri to utility network	util	url to schema location utility network	<binary
33	36	4	1	2.0	namepsace uri to dynamizer	dynam	url to schema location dynamizer	<binary
*								

33 rows.


Edit Data - PostgreSQL 9.3 (localhost:5432) - energy_alice - objectclass
File Edit View Tools Help
No limit
id [PK] integer
classname character varying(256)
superclass_id integer
tablename character varying(30)
is_ade_class numeric(1,0)
baseclass_id integer
102 101 TunnelFurniture 3 0 3
103 102 HollowSpace 3 tunnel hollow space 0 3
104 103 TexCoordList 56 textureparam 0 1
105 104 TexCoordGen 56 textureparam 0 1
106 105 WaterObject 3 cityobject 0 3
107 200 Type 1 1
108 201 TimeSeries 200 nrg8a timeseries 1
109 202 RegularTimeSeries 201 nrg8a timeseries 1
110 203 IrregularTimeSeries 201 nrg8a timeseries 1
111 204 RegularTimeSeriesFile 201 nrg8a timeseries 1
112 205 IrregularTimeSeriesFile 201 nrg8a timeseries 1
113 206 Schedule 200 nrg8a schedule 1
114 207 SingleValueSchedule 206 nrg8a schedule 1
115 208 DualValueSchedule 206 nrg8a schedule 1
116 209 DailyPatternSchedule 206 nrg8a schedule 1
117 210 TimeSeriesSchedule 206 nrg8a schedule 1
118 211 Construction 2 nrg8a construction 1 2
119 212 Construction 211 nrg8a construction 1 2
120 213 ReverseConstruction 211 nrg8a construction 1 2
121 214 Layer 2 nrg8a layer 1 2
122 215 LayerComponent 2 nrg8a layer component 1 2
123 216 Material 2 nrg8a material 1 2
124 217 Gas 216 nrg8a material 1 2
125 218 SolidMaterial 216 nrg8a material 1 2
126 219 WeatherStation 3 nrg8a weather station 1 3
127 220 WeatherData 2 nrg8a weather data 1
128 221 ThermalZone 3 nrg8a thermal zone 1 3
129 222 ThermalBoundary 3 nrg8a thermal boundary 1 3
130 223 ThermalOpening 3 nrg8a thermal opening 1 3
131 224 UsageZone 3 nrg8a usage zone 1 3
132 225 BuildingUnit 3 nrg8a building unit 1 3
133 226 Facilities 3 nrg8a facilities 1 3
134 227 DHWFacilities 226 nrg8a facilities 1 3
135 228 ElectricalAppliances 226 nrg8a facilities 1 3
136 229 LightingFacilities 226 nrg8a facilities 1 3
137 230 Occupants 2 nrg8a occupants 1 2
138 231 Household 2 nrg8a households 1 2
139 232 EnergyDemand 2 nrg8a energy demand 1 2
163 rows.
Edit Data - PostgreSQL 9.3 (local
File Edit View Tools Help
id [PK] integer
classname character vary
71 70 BridgeFloor
72 71 BridgeRoofS
73 72 BridgeWallS
74 73 BridgeGroun
75 74 BridgeClosu
76 75 OuterBridge
77 76 OuterBridge
78 77 BridgeOpen
79 78 BridgeWindo
80 79 BridgeDoor
81 80 BridgeFurni
82 81 BridgeRoom
83 82 BridgeConst
84 83 AbstractTu
85 84 TunnelPart
86 85 Tunnel
87 86 TunnelInsta
88 87 IntTunnelIn
89 88 TunnelBoun
90 89 TunnelCeili
91 90 InteriorTun
92 91 TunnelFloor
93 92 TunnelRoofS
94 93 TunnelWallS
95 94 TunnelGroun
96 95 TunnelClosu
97 96 OuterTunnel
98 97 OuterTunnel
99 98 TunnelOpen
100 99 TunnelWindo
101 100 TunnelDoor
102 101 TunnelFurni
103 102 HollowSpace
104 103 TexCoordLis
105 104 TexCoordGen
106 105 WaterObjec
106 rows.

Implementation steps

- Define and agree upon rules to make the 3DCityDB "ADE-compatible"
  - Enable to "register" *any* ADE
    - Add a metadata module
    - Add functions to help installing/removing an ADE
  - Define rules how to map ADE-classes to new/existing tables
    - Adopt naming convention for new DB entities
  - Add an ADE-hook mechanism to certain existing functions. E.g.:
    - delete_building() → must work also with ADE-AbstractBuilding
    - delete_cityobject() → must work also with new CityObjects
    - delete_cityobjectgroup() → must work also with new CityObjects
    - get_envelope_cityobject() → same as above
  - (Enable/extend existing tools to be ADE-compatible: citygml4j, Importer/Exporter, etc.)

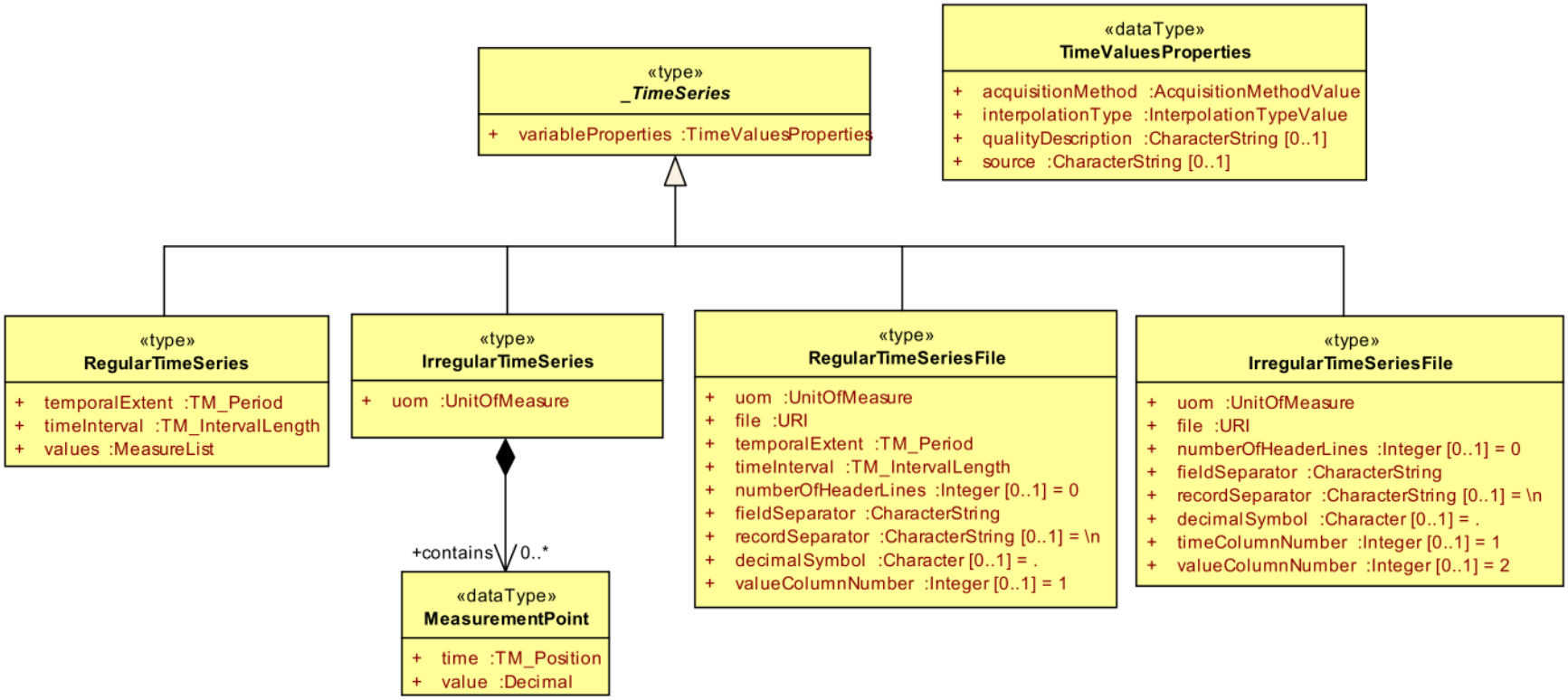
- All rules are agreed upon within the 3DCityDB development team and are being implemented for the next 3DCityDB release

Mapping rules (excerpt)

- New tables: **db_prefix + table_name**
  - E.g Thermal_Zone -> NRG_THERMAL_ZONE

Mapping rules (excerpt)

- New tables: **db_prefix + table_name**
- With classes != CityObject, create own tables
  - Example 1: Timeseries, Schedule:
    - NRG_TIME_SERIES, NRG_SCHEDULE
  - Example 2: Features (e.g. EnergyDemand, Occupants, Construction)
    - NRG_ENERGY_DEMAND, NRG_OCCUPANTS, NRG_CONSTRUCTION


«type»
_TimeSeries
+ variableProperties :TimeValuesProperties
«dataType»
TimeValuesProperties
+ acquisitionMethod :AcquisitionMethodValue
+ interpolationType :InterpolationTypeValue
+ qualityDescription :CharacterString [0..1]
+ source :CharacterString [0..1]
«type»
RegularTimeSeries
+ temporalExtent :TM_Period
+ timeInterval :TM_IntervalLength
+ values :MeasureList
«type»
IrregularTimeSeries
+ uom :UnitOfMeasure
+contains
0..*
«dataType»
MeasurementPoint
+ time :TM_Position
+ value :Decimal
«type»
RegularTimeSeriesFile
+ uom :UnitOfMeasure
+ file :URI
+ temporalExtent :TM_Period
+ timeInterval :TM_IntervalLength
+ numberOfHeaderLines :Integer [0..1] = 0
+ fieldSeparator :CharacterString
+ recordSeparator :CharacterString [0..1] = \n
+ decimalSymbol :Character [0..1] = .
+ valueColumnNumber :Integer [0..1] = 1
«type»
IrregularTimeSeriesFile
+ uom :UnitOfMeasure
+ file :URI
+ numberOfHeaderLines :Integer [0..1] = 0
+ fieldSeparator :CharacterString
+ recordSeparator :CharacterString [0..1] = \n
+ decimalSymbol :Character [0..1] = .
+ timeColumnNumber :Integer [0..1] = 1
+ valueColumnNumber :Integer [0..1] = 2

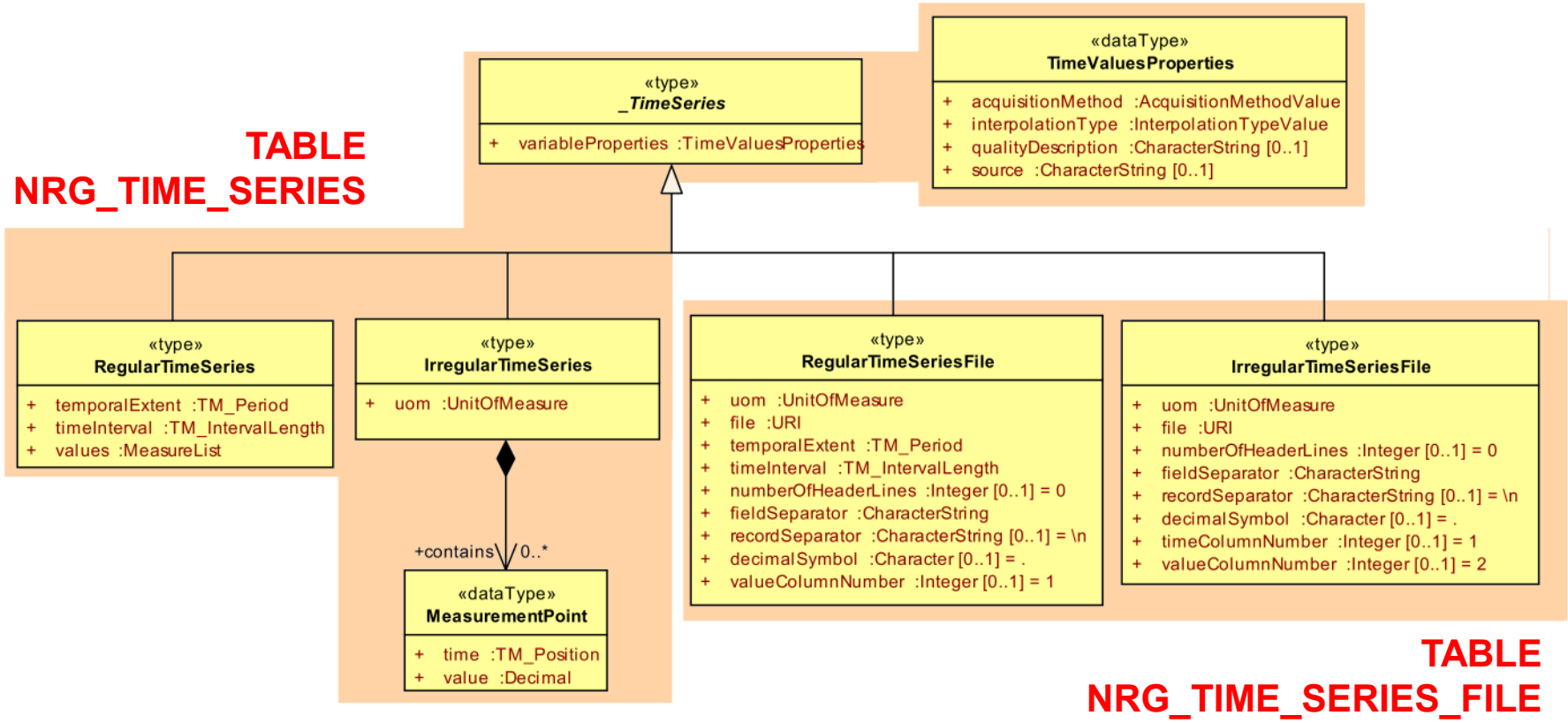


TABLE
NRG_TIME_SERIES
«type»
_TimeSeries
+ variableProperties :TimeValuesProperties
«dataType»
TimeValuesProperties
+ acquisitionMethod :AcquisitionMethodValue
+ interpolationType :InterpolationTypeValue
+ qualityDescription :CharacterString [0..1]
+ source :CharacterString [0..1]
«type»
RegularTimeSeries
+ temporalExtent :TM_Period
+ timeInterval :TM_IntervalLength
+ values :MeasureList
«type»
IrregularTimeSeries
+ uom :UnitOfMeasure
+contains
0..*
«dataType»
MeasurementPoint
+ time :TM_Position
+ value :Decimal
«type»
RegularTimeSeriesFile
+ uom :UnitOfMeasure
+ file :URI
+ temporalExtent :TM_Period
+ timeInterval :TM_IntervalLength
+ numberOfHeaderLines :Integer [0..1] = 0
+ fieldSeparator :CharacterString
+ recordSeparator :CharacterString [0..1] = \n
+ decimalSymbol :Character [0..1] = .
+ valueColumnNumber :Integer [0..1] = 1
«type»
IrregularTimeSeriesFile
+ uom :UnitOfMeasure
+ file :URI
+ numberOfHeaderLines :Integer [0..1] = 0
+ fieldSeparator :CharacterString
+ recordSeparator :CharacterString [0..1] = \n
+ decimalSymbol :Character [0..1] = .
+ timeColumnNumber :Integer [0..1] = 1
+ valueColumnNumber :Integer [0..1] = 2
TABLE
NRG_TIME_SERIES_FILE

TABLE TIME_SERIES

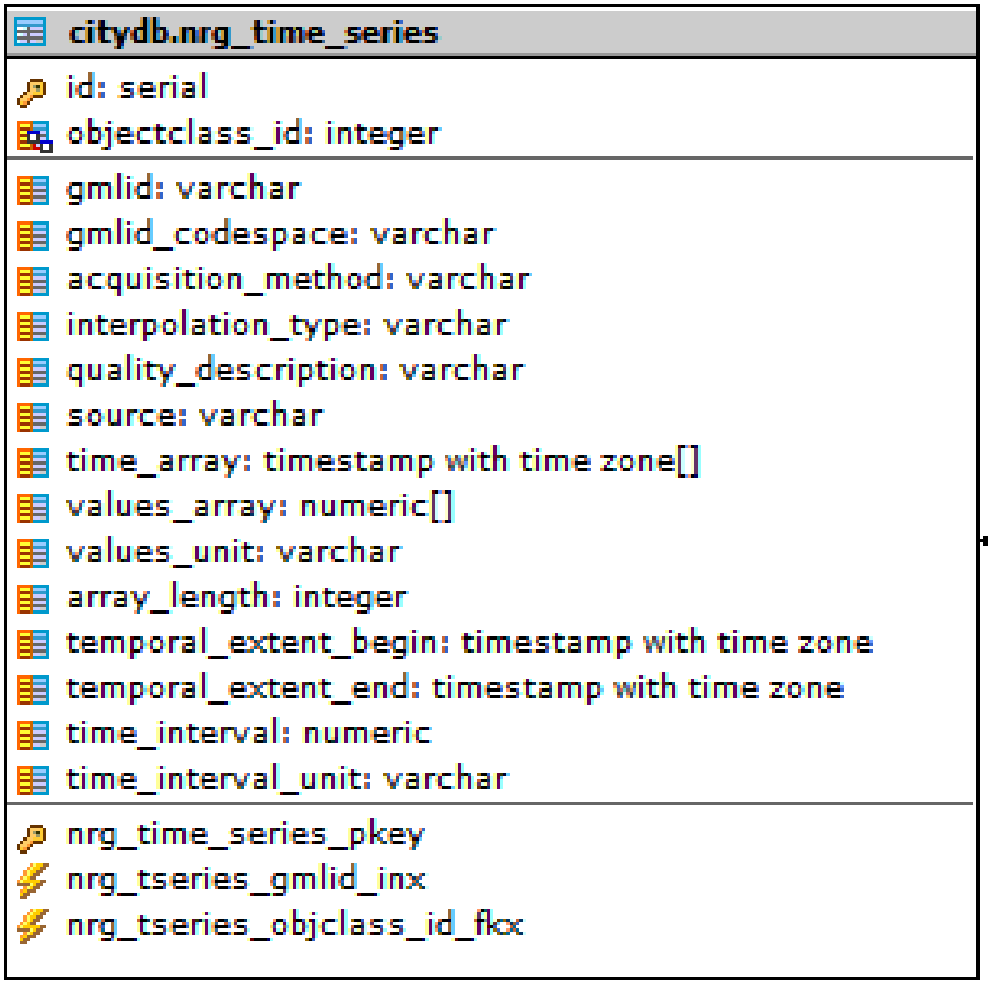
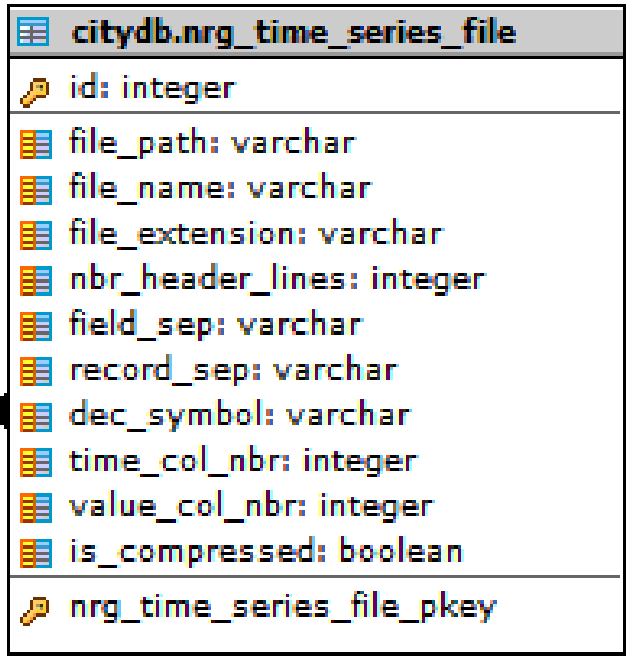


TABLE TIME_SERIES_FILE



Mapping rules (excerpt)

- New tables: **db_prefix + table_name**
- With classes != CityObject, create own tables
- With "simple" ADE CityObjects, start from table CITYOBJECT and store the remaining attributes in another linked table
  - E.g. EnergyConversionSystem → **CITYOBJECT + NRG_CONV_SYSTEM**
  - Same approach as in the vanilla 3DCityDB

Mapping rules (excerpt)

- New tables: **db_prefix + table_name**
- With classes != CityObject, create own tables
- With "simple" ADE CityObjects, start from table CITYOBJECT and store the remaining attributes in another linked table
- With "complex" ADE CityObjects, same as before + FK to referenced objects
  - E.g. UsageZone -> **CITYOBJECT + USAGE_ZONE**
    + FK to SCHEDULE
    + FK to SURF_GEOMETRY

Mapping rules (excerpt)

- New tables: **db_prefix + table_name**
- With classes != CityObject, create own tables
- With "simple" ADE CityObjects, start from table CITYOBJECT and store the remaining attributes in another linked table
- With "complex" ADE CityObjects, same as before + FK to referenced objects
- With ADE-extended CityObjects, same as before + FK to referenced objects
  - E.g. ADE AbstractBuilding -> **CITYOBJECT + BUILDING + NRG_BUILDING**
    + FK to THERMAL_ZONE
    + FK to ….
    + FK to SURF_GEOMETRY

Mapping rules (excerpt)

- Example: ADE-AbstractBuilding

CITYOBJECT
<- EXT_REF
<- GEN_ATTR
<- GENERALIZATION
<- ENERGY_CONV_SYSTEM
<- WEATHER_DATA
<- TIMESERIES

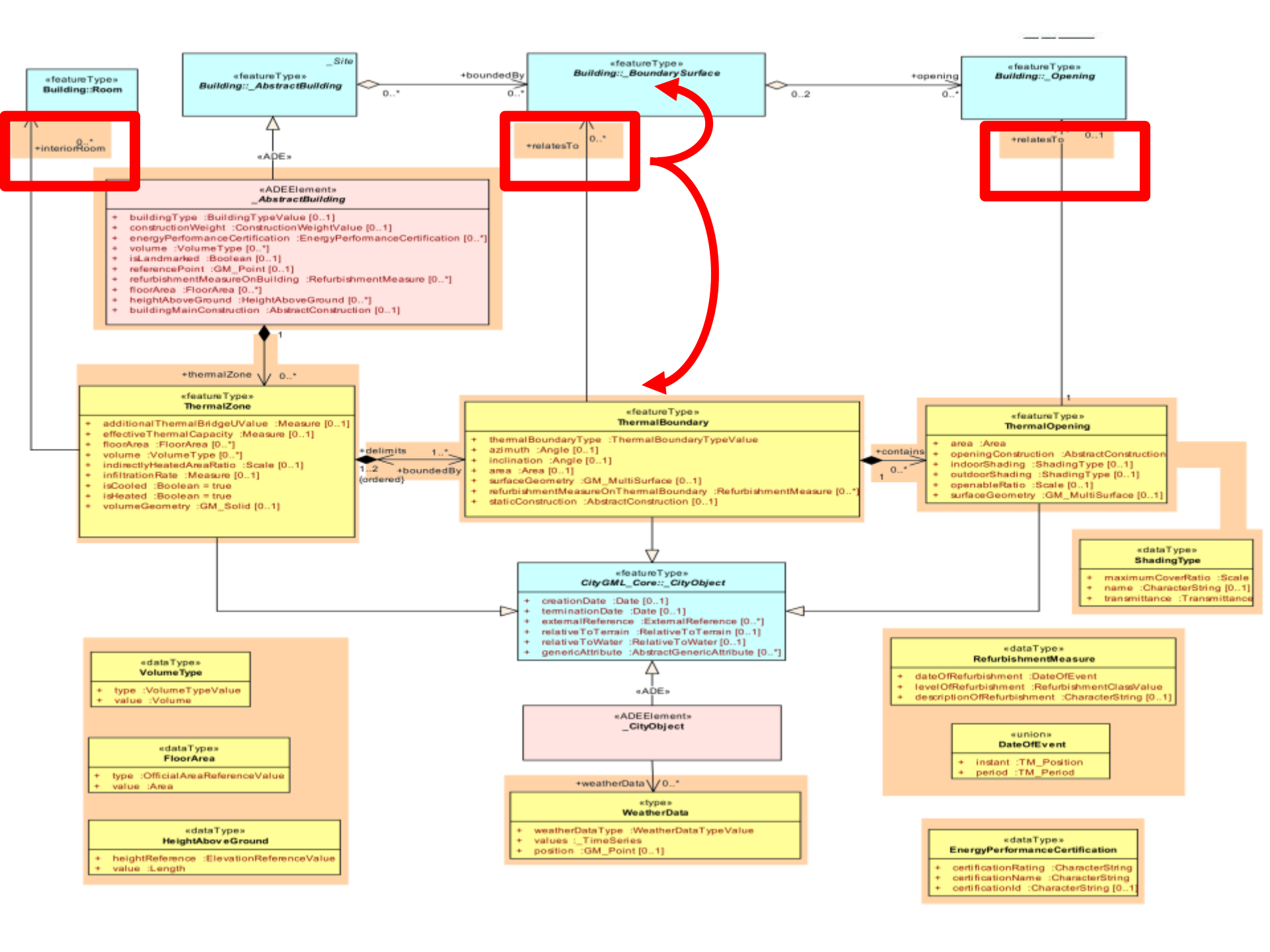
+ BUILDING
<- ROOM
…

+ NRG_BUILDING
<- THERMAL_ZONE
<- USAGE_ZONE
...

Mapping rules (excerpt)

- In case of **1:n and m:n relations** between ADE and vanilla CityGML objects, use an **association table** instead of a new column in a vanilla table
  - Example 1: ThermalOpening to _Opening
  - Example 2: ThermalBoudary to _BoundarySurface

In other words: KEEP THE VANILLA TABLES UNCHANGED!


«featureType»
Building::Room
+interiorRoom
0..*
_Site
«featureType»
Building::_AbstractBuilding
+boundedBy
0..*
0..*
«featureType»
Building::_BoundarySurface
+opening
0..2
0..*
«featureType»
Building::_Opening
+relatesTo
0..*
+relatesTo
0..1
«ADE»
«ADEElement»
_AbstractBuilding
+ buildingType :BuildingTypeValue [0..1]
+ constructionWeight :ConstructionWeightValue [0..1]
+ energyPerformanceCertification :EnergyPerformanceCertification [0..*]
+ volume :VolumeType [0..*]
+ isLandmarked :Boolean [0..1]
+ referencePoint :GM_Point [0..1]
+ refurbishmentMeasureOnBuilding :RefurbishmentMeasure [0..*]
+ floorArea :FloorArea [0..*]
+ heightAboveGround :HeightAboveGround [0..*]
+ buildingMainConstruction :AbstractConstruction [0..1]
1
+thermalZone
0..*
«featureType»
ThermalZone
+ additionalThermalBridgeUValue :Measure [0..1]
+ effectiveThermalCapacity :Measure [0..1]
+ floorArea :FloorArea [0..*]
+ volume :VolumeType [0..*]
+ indirectlyHeatedAreaRatio :Scale [0..1]
+ infiltrationRate :Measure [0..1]
+ isCooled :Boolean = true
+ isHeated :Boolean = true
+ volumeGeometry :GM_Solid [0..1]
+delimits
1..*
1..2
+boundedBy
{ordered}
«featureType»
ThermalBoundary
+ thermalBoundaryType :ThermalBoundaryTypeValue
+ azimuth :Angle [0..1]
+ inclination :Angle [0..1]
+ area :Area [0..1]
+ surfaceGeometry :GM_MultiSurface [0..1]
+ refurbishmentMeasureOnThermalBoundary :RefurbishmentMeasure [0..*]
+ staticConstruction :AbstractConstruction [0..1]
+contains
1
0..*
1
«featureType»
ThermalOpening
+ area :Area
+ openingConstruction :AbstractConstruction
+ indoorShading :ShadingType [0..1]
+ outdoorShading :ShadingType [0..1]
+ openableRatio :Scale [0..1]
+ surfaceGeometry :GM_MultiSurface [0..1]
«dataType»
ShadingType
+ maximumCoverRatio :Scale
+ name :CharacterString [0..1]
+ transmittance :Transmittance
«featureType»
CityGML_Core::_CityObject
+ creationDate :Date [0..1]
+ terminationDate :Date [0..1]
+ externalReference :ExternalReference [0..*]
+ relativeToTerrain :RelativeToTerrain [0..1]
+ relativeToWater :RelativeToWater [0..1]
+ genericAttribute :AbstractGenericAttribute [0..*]
«ADE»
«ADEElement»
_CityObject
«dataType»
VolumeType
+ type :VolumeTypeValue
+ value :Volume
«dataType»
FloorArea
+ type :OfficialAreaReferenceValue
+ value :Area
«dataType»
HeightAboveGround
+ heightReference :ElevationReferenceValue
+ value :Length
+weatherData
0..*
«type»
WeatherData
+ weatherDataType :WeatherDataTypeValue
+ values :_TimeSeries
+ position :GM_Point [0..1]
«dataType»
RefurbishmentMeasure
+ dateOfRefurbishment :DateOfEvent
+ levelOfRefurbishment :RefurbishmentClassValue
+ descriptionOfRefurbishment :CharacterString [0..1]
«union»
DateOfEvent
+ instant :TM_Position
+ period :TM_Period
«dataType»
EnergyPerformanceCertification
+ certificationRating :CharacterString
+ certificationName :CharacterString
+ certificationId :CharacterString [0..1]

Implementation steps

- Define and agree upon rules to make the 3DCityDB "ADE-compatible"
  - Enable to "register" *any* ADE
    - Add a metadata module
    - Add functions to help installing/removing an ADE
  - Define rules how to map ADE-classes to new/existing tables
    - Adopt naming convention for new DB entities
  - Add an ADE-hook mechanism to some "vanilla" functions. E.g.:
    - delete_building() → must work also with an ADE-AbstractBuilding
    - delete_cityobject() → must work also with new CityObjects
    - delete_cityobjectgroup() → must work also with new CityObjects
    - get_envelope_cityobject() → same as above
  - (Enable/extend existing tools to be ADE-compatible: citygml4j, Importer/Exporter, etc.)

- All rules are agreed upon within the 3DCityDB development team and are being implemented for the next 3DCityDB release

DELETE functions (stored procedures)

- Provide for each "simple" and "complex" stand-alone ADE CityObject a *prefixed* delete function. E.g.
  - **nrg_delete_thermal_zone()**
  - **nrg_delete_usage_zone()**
  - (Conceptually follows the vanilla approach)

- For the ADE-extended CityObjects (i.e. _AbstractBuilding), provide
  - **nrg_delete_building()**, but it takes care only of NRG_BUILDING table

- Add an ADE-hook within vanilla functions
  - **intern_delete_cityobject()** → e.g. contained in delete_building()
  - **delete_cityobject()**

- Same approach for **get_envelope_*()** functions

nrg_delete_cityobject()

- Each ADE has also to provide a general prefixed delete_cityobject() function for the new ADE-CityObjects

FUNCTION nrg_delete_cityobject()

GET classname and db_prefix FROM objectclass

CASE WHEN

classname='**ThermalZone**' THEN **citydb_pkg.nrg_delete_thermal_zone()**

classname='**UsageZone**' THEN **citydb_pkg.nrg_delete_usage_zone()**

classname='**xxx**' THEN citydb_pkg.**nrg_delete_xxx()**

ELSE

END CASE

New ADE-aware delete_cityobject()

- The ADE-hook mechanism basically works like this:

- **FUNCTION delete_cityobject(co_id)**
  - GET **is_ade_class** and **db_prefix** of the objectclass co_id
  - **IF is_ade_class IS TRUE THEN**

    call citydb. || db_prefix ||_delete_cityobject(co_id)
  - **ELSE**

    …

    Same vanilla code of old citydb_pkg.delete_cityobject(co_id)

    …
  - **END IF**

Conclusions

- The implementation of the Energy ADE v. 0.8 for PostgreSQL is nearly complete. Major showstoppers are
  - the completion of the documentation
  - some minor last-minute changes in the metadata module

- Experiences so far have contributed to the definition of
  - the set of rules for the metadata module
  - ADE-hook mechanism for existing functions

- Some (general) questions remain open
  - How to deal with objects which might be repeated many times (e.g. constructions?)
  - Current implementation of time series might be sub-optimal
  - Performance issues have not been taken into account (yet)

Curious to try out?

Willing to participate?

Just contact us!

3D City Database extension

for the

CityGML Energy ADE 0.8

PostgreSQL Version

Version Beta 1

Documentation

Documentation last update: 22 May 2017

AUSTRIAN INSTITUTE OF TECHNOLOGY
TOMORROW TODAY

AIT Austrian Institute of Technology

your ingenious partner

Dr. Giorgio Agugiaro
Center for Energy
Smart Cities and Regions Research Field
AIT - Austrian Institute of Technology
giorgio.agugiaro@ait.ac.at

ACKNOWLEDGEMENTS

Patrick Holcik (AIT)
Zhihang Yao, Thomas Kolbe (TU München)
Claus Nagel (virtualcitySYSTEMS)