

CityGML Energy ADE V. 1.0 Specification

Joachim Benner, KIT
Version 1.0
März 2018

Table of Content

1	Introduction	3
2	Energy ADE thematic modules	3
2.1	Energy ADE Core module	4
2.2	Building physics module	7
2.2.1	<i>ThermalZone</i>	7
2.2.2	<i>ThermalBoundary</i> and <i>ThermalOpening</i>	7
2.3	Occupants Behaviour module	11
2.3.1	<i>UsageZone</i>	13
2.3.2	<i>BuildingUnit</i>	13
2.3.3	<i>Occupants</i> and <i>Household</i>	13
2.3.4	Facilities	14
2.4	Material and Construction module	14
2.5	Energy systems module	15
2.5.1	Energy System base classes	16
2.5.2	Energy conversion systems	18
2.5.3	Energy distribution and storage system	18
2.6	Support classes	19
2.6.1	Time series	20
2.6.2	Schedules	21
2.6.3	Weather data	22
3	Unit of Measurement identification	22
4	Energy ADE Codelists	25
5	Further information	26

1 Introduction

The CityGML Energy Application Domain Extension (Energy ADE) aims at extending the CityGML 2.0 standard with energy-related entities and attributes necessary to perform energy analyses at building and urban scale, such as energy demand diagnostics, solar potential study, simulation of low-carbon energy strategies etc...

In accordance with the philosophy of CityGML, the Energy ADE aims to be flexible in terms of compatibility with different data qualities and levels of details. Its central goal is to provide a unique and standard-based data model to overcome, on one side, data interoperability issues and, on the other side, to allow for both detailed single-building energy simulations and city-wide bottom-up energy assessment. The Energy ADE focuses primarily on the building, its physical properties and the systems installed in it. It is not meant to cover urban centralised energy infrastructures, like district heating system or gas networks, as they are instead the focus of another application domain extension, namely the Utility Network ADE (Kutzner and Kolbe, 2016). Some parts of the Energy ADE can however be used beyond the building scale to characterise, for example, the energy demand of other urban objects like street lamps, or the energy production of power plants.

2 Energy ADE thematic modules

The actual version of the Energy ADE uses and extends only two modules of the CityGML base standard: The *CityGML_Core* module and the *Building* module. The data model is structured into 6 functional parts, which mutual dependencies are illustrated in Figure 1. The main purpose of this modularization is to eliminate direct dependencies between the 4 main functional modules: *Building Physics*, *Occupant Behaviour*, *Material and Construction* and *Energy Systems*.

The different modules of the Energy ADE provide the following functionality, which is described in more detail in the following sections:

- The *Energy ADE Core* module (see chapter 2.1) provides a number of ADE properties extending CityGML classes and defines abstract base classes of the four central thematic modules as well as a number of general data types, enumerations and codelists.
- The *Building Physics* (see chapter 2.2) module supports parameters for single- or multi-zone building energy simulations.
- The *Occupant Behaviour* module (see chapter 2.3) enables to model the energy relevant behaviour of the building's occupants.
- The *Material and Construction* module (see chapter 2.4) provides physical parameters of building materials.
- The *Energy Systems* module (see chapter 2.5) enables to represent the energy conversion, distribution, storage and emission devices of a building and the energy flow between them.
- The module *Supporting Classes* (see chapter 2.6) contains various classes to represent time series, schedules and weather data.

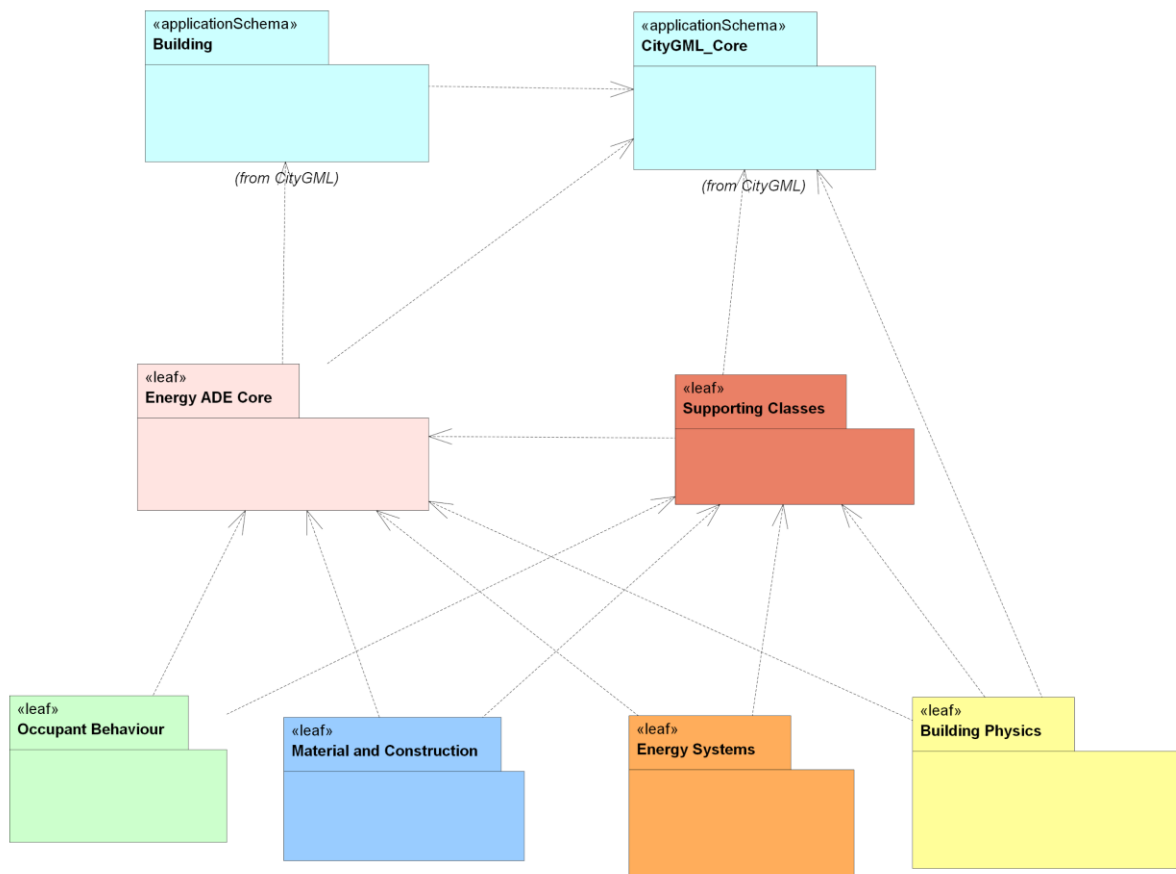


Figure 1: Modular structure of the Energy ADE

2.1 Energy ADE Core module

The Energy ADE Core module (see Figure 2) has three main functions: It extends classes of the CityGML base standard with energy relevant properties, it provides abstract base classes of the four central functional modules, and it defines a number of data types, enumerations and codelists which are used in more than one functional module. More information on these can be found in the feature catalogue.

In the actual version of the Energy ADE data model, only two classes of the CityGML base standard are directly extended with application specific properties: *_CityObject* (CityGML module *CityGML_Core*) and *_AbstractBuilding* (CityGML module *Building*). The extension of the base class *_CityObject* by the relation *demands*, pointing to the Energy ADE feature type *EnergyDemand*, enables to relate any city object with energy demand information. Here, energy demand is defined as the useful energy required satisfying a specific end-use, such as heating, cooling or provision of domestic hot water.

In CityGML the class *_AbstractBuilding* is used to represent either a complete building (class *Building*) or a building part (class *BuildingPart*). The extension of these feature types in the Energy ADE follows different goals. In order to support rough assessments of a building's energy demand, a number of general energy-related parameters are defined: a classification of the general building usage (*buildingType*), a rough classification of the building

construction structure (*constructionWeight*), important geometrical (*volume*, *floorArea*) and locational (*referencePoint*, *heightAboveGround*) parameters, and average material parameters (see section 2.4) of the building's exterior shell (*aggregatedBuildingConstruction*). If a building has energy performance certificates, the corresponding information (*energyPerformanceCertification*) can also be specified.

All properties mentioned so far only support rough energy assessments. In order to also support detailed energy simulations at building level taking into account time-variant weather conditions and occupants' behaviour, more sophisticated models for the building physics and the building usage are needed. The Energy ADE therefore supports the partition of a building into different thermal zones (see section 2.2) and usage zones (see section 2.3). The abstract base classes of the corresponding modules (*AbstractThermalZone* and *AbstractUsageZone*) are defined in the *Energy ADE Core* module. Furthermore, this module defines two additional abstract base classes: *AbstractConstruction* to represent energy relevant physical and optical properties of building elements like walls, roofs or windows (see section 2.4), and *AbstractEnergySystem* to represent the energy conversion, distribution, storage and emission systems of a building (see section 2.5).



Figure 2: Energy ADE Core module

2.2 Building physics module

The Building Physics module is the core of the Energy ADE. It defines a number of new "thermal" entities (*ThermalZone*, *ThermalBoundary*, *ThermalOpening*) and relates them with "physical" entities (*Building:Room*, *Building:_BoundarySurface*, *Building:_Opening*) of the CityGML data model.

2.2.1 *ThermalZone*

The *ThermalZone* is a new object introduced in the Energy ADE to realize building heating and cooling demand calculations. A *ThermalZone* is a part of a building which serving as the smallest spatial region for building heating and cooling demand calculation. A CityGML Building object may reference several *ThermalZone* objects, for instance in the case of a mixed-usage building, or to distinguish rooms or zones with different orientations (i.e. solar gains) and/or thermal behaviour. A *ThermalZone* is generally a "thermal homogeneous" space considered as isothermal, but may also refer to several building rooms with different usage boundary conditions for simplified building energy modelling.

A *ThermalZone* contains a series of energy-related attributes characterizing its geometry (*floorArea*, *volume*), its conditioning status (*isCooled*, *isHeated*, *indirectlyHeatedAreaRatio*) and overall building physics properties (*additionalThermalBridgeUValue*, *infiltrationRate*, *effectiveThermalCapacity*). All these attributes are optional. Furthermore, a *ThermalZone* may optionally contain an explicit volume geometry (specified by *volumeGeometry*), useful in particular for visualisation purposes, but not necessary for heating and cooling demand calculations. The *ThermalZone* may also be related to one or more rooms (*Building:Room*). The actual surface boundaries of a *ThermalZone* are defined by means of *ThermalBoundary* objects (see later).

If occupied, a *ThermalZone* must be related to at least one *UsageZone*, which contains the usage boundary conditions for the heating and cooling demand calculation (see chapter 2.3). In order to preserve the modularity of the Energy ADE, the corresponding relation is provided by the base classes *AbstractThermalZone* and *AbstractUsageZone* (see chapter 2.1). A *ThermalZone* may even be related to several *UsageZone* objects for simplified modelling of mixed-usage space, in which case the usage boundary conditions of the *UsageZone* must be aggregated and weighted according with their *floorArea* attribute.

2.2.2 *ThermalBoundary* and *ThermalOpening*

A *ThermalBoundary* represent the physical relationship between two *ThermalZone* objects, or one *ThermalZone* and the building environment. Its geometrical representation is a planar or nearly planar surface. Each *ThermalZone* is geometrically closed by its whole set of bounding *ThermalBoundary* objects, specified in the relationship *boundedBy*.

Via the relation *construction* pointing to an *AbstractConstruction*, a *ThermalBoundary* is related with a *Construction* object (see chapter 2.4), representing overall optical properties and an ordered set of material layers of the corresponding building element (e.g. wall, roof, base or interior slab). The related *Construction* physically represents the "opaque" part of the building element. Any "openings" (doors or windows) with differing optical and physical

properties are modelled as *ThermalOpening* objects (see below), being related with the *ThermalBoundary* via the relation *contains*. A number of properties specify the type (e.g. *interiorWall* or *roof*) of a *ThermalBoundary* (*thermalBoundaryType*), its size (*area*), orientation (*azimuth*, *inclination*) and geometry (*surfaceGeometry*). All these properties refer to the complete building element, including all openings.

A *ThermalBoundary* object must refer to its one or two corresponding *ThermalZone* objects via the ordered relation *delimits*. In the case where the *ThermalBoundary* delimits one *ThermalZone* from the building environment, corresponding then to the external boundary of a building, its geometrical representation normally coincides with the external surfaces of the related outer wall, roof or basement floor. In this case, the *ThermalBoundary* may be linked to corresponding CityGML *_BoundarySurface* objects (e.g. a *WallSurface*, *RoofSurface* or *GroundSurface* objects in LoD2), through the relationship *relatesTo*. In the case where a *ThermalBoundary* separates two adjacent *ThermalZone*, corresponding then to an intermediate floor, ceiling, or a shared wall, its geometrical representation coincides with the plane laying at the middle of this construction thickness (see Figure 3)

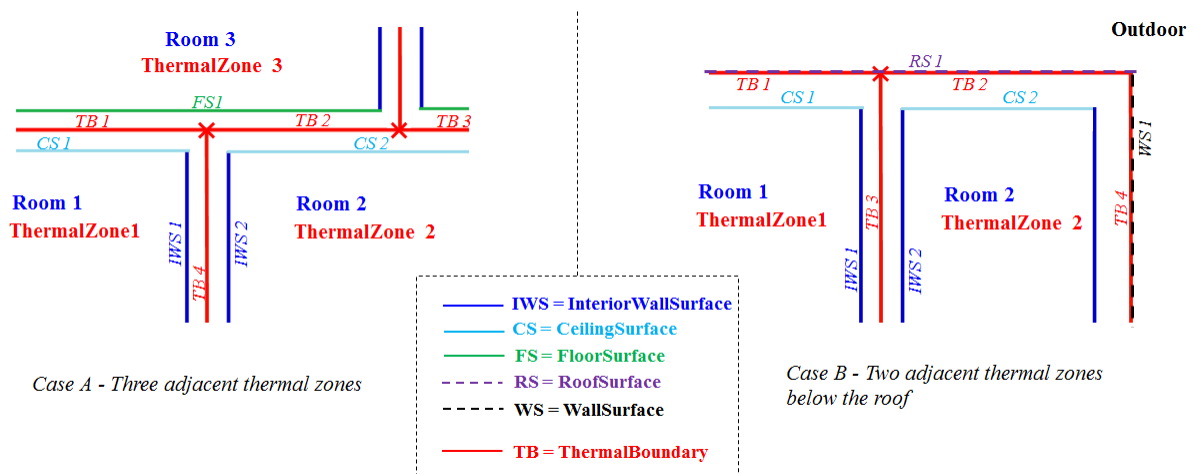


Figure 3: Localization of interior and exterior *ThermalBoundary* objects

The order of the two related *ThermalZone* objects is significant. For simulating the energy transfer between two *ThermalZone* objects, or between a *ThermalZone* and the environment, it is essential to know which *ThermalZone* is in contact with which layer (class *Layer*, see chapter 2.4) of the related *Construction*. This information is indicated by the order of the *Layer* objects in *Construction* and the order of the *ThermalZone* objects related with a *ThermalBoundary* (relation *delimits*), which must observe the following rules:

- For exterior *ThermalBoundary* objects, the first *Layer* is facing the exterior environment, and the last *Layer* the building interior. The normal of the *surfaceGeometry* geometrical property is pointing out of the building.
- For *ThermalBoundary* objects of type *intermediaryFloor* or *basementCeiling*, the first *Layer* is facing the lower *ThermalZone* and the last *Layer* the upper *ThermalZone*. Then, the first relation *delimitsBy* points to the upper *ThermalZone*, and the last relation *delimitsBy* points to the lower *ThermalZone*. The normal of the *surfaceGeometry* geometrical property is pointing downwards.

- For all other interior *ThermalBoundary* objects, the first relation *delimitsBy* points to the *ThermalZone* facing the last *Layer*, and the last relation *delimitsBy* points to the *ThermalZone* facing the first *Layer*. The normal of the *surfaceGeometry* geometrical property is pointing out of the first related *ThermalZone*.

A *ThermalOpening* thermally represents the "holes" in building elements like doors or windows. Thus, a *ThermalOpening* has the orientation of the *ThermalBoundary* to which it belongs, but normally a differing size (property *area*) and geometry (property *surfaceGeometry*). Additional information concerning shading devices and natural ventilation are specified via the properties *indoorShading*, *ourdoorShading* and *openableRatio*.

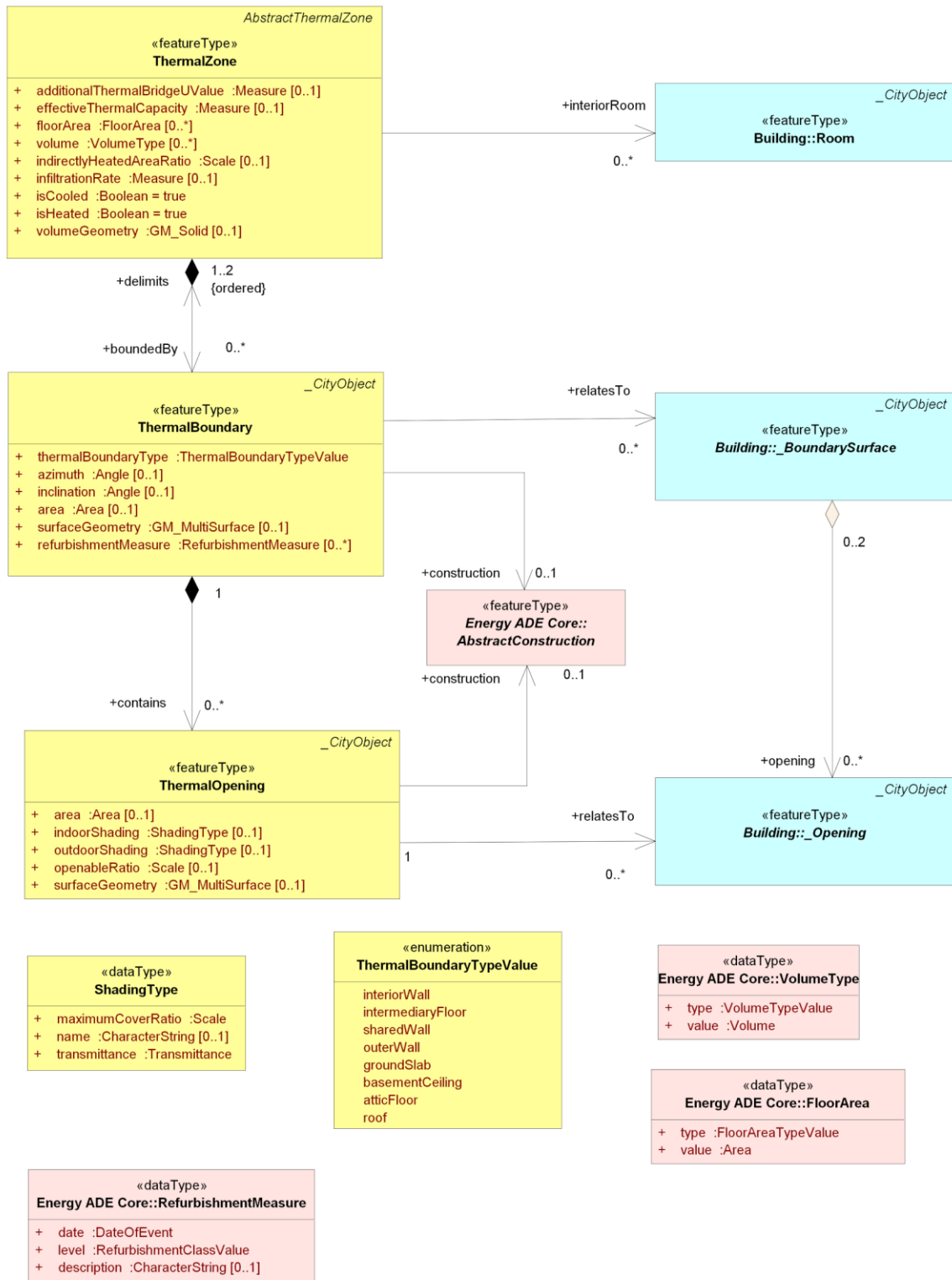


Figure 4: Building Physics module

2.3 Occupants Behaviour module

The Occupants Behaviour module (see Figure 5) contains the detailed characterization of the building's usage, emphasizing people and thermally relevant facilities. It is related to the rest of the Energy ADE and CityGML model through the class *UsageZone*. One building may be related with several *UsageZone* objects. Due to the type of information it allows to store, the Occupants Behaviour module may be used also for multi-field analysis (e.g. socio-economics or demographics).

Each *UsageZone* object may be related with several *BuildingUnit* objects, representing the premises (dwellings, offices etc.) located inside the defined *UsageZone*. Especially for estimating internal energy gains, the occupants of a *UsageZone* or *BuildingUnit* are represented by the classes *Occupants* and *Household*, and the thermal energy producing facilities (lighting, domestic hot water, electrical appliances) by the base class *Facilities* and the derived classes *DHWFacilities*, *LightingFacilities* and *ElectricalAppliances*.

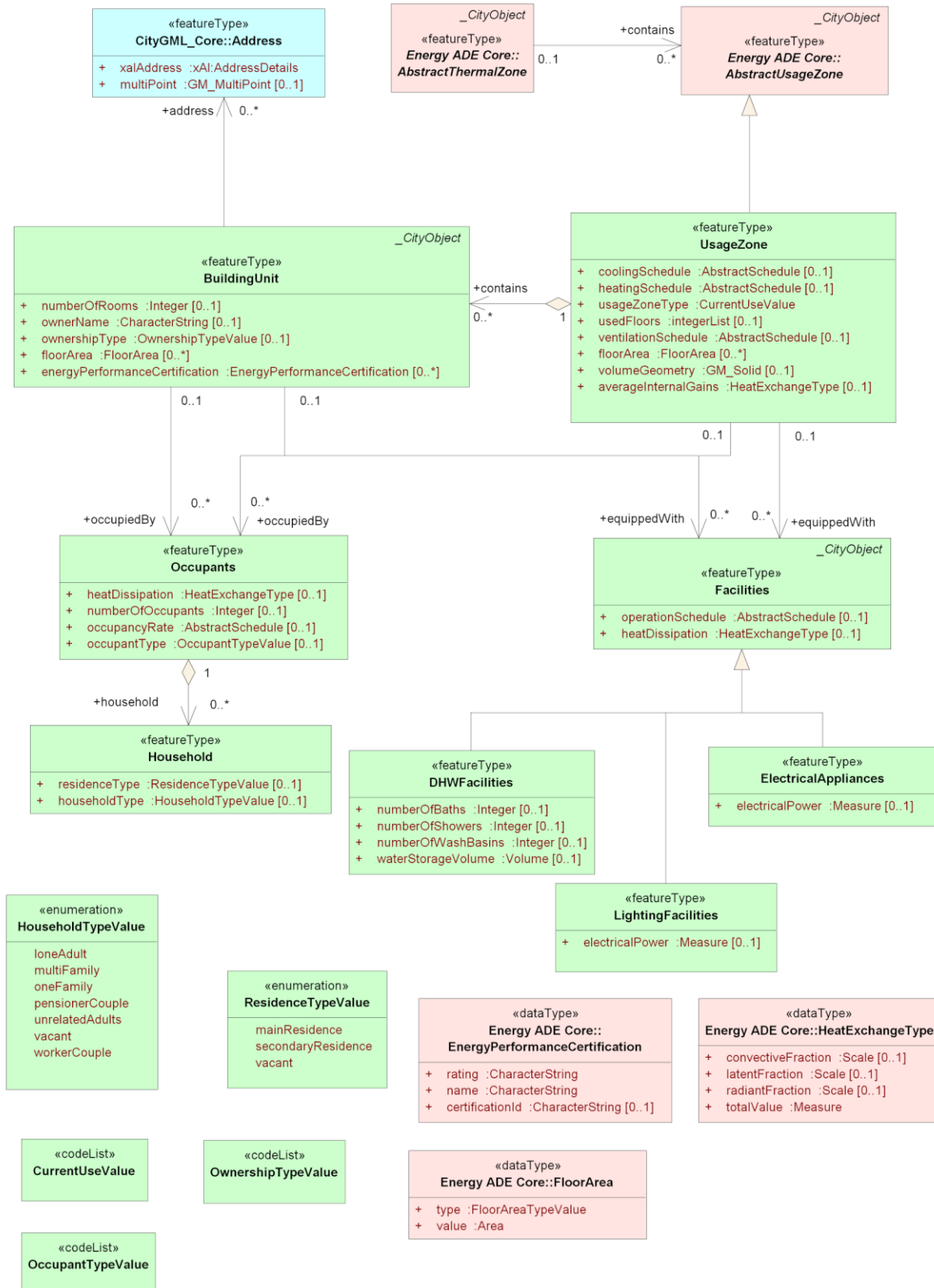


Figure 5: Occupants Behaviour module

2.3.1 UsageZone

The *UsageZone* is a new object introduced in the Energy ADE to realize building usage analysis, and to calculate the energy demand related to occupant-depending end-uses such as domestic hot water, electrical appliances, cooking etc. When related to the *ThermalZone*, it also allows providing the zone usage conditions (e.g. internal gains, HVAC schedules) for space heating and the cooling demand calculations. An *UsageZone* defines the zone of a building with homogeneous usage conditions and indoor climate control settings.

Minimally defined by the mandatory attribute *usageZoneType* of type *CurrentUseValue*, it is a semantic object with an optional geometry (*volumeGeometry*). The HVAC schedules are characterized by the optional attributes *heatingSchedule*, *coolingSchedule* and *ventilationSchedule* (respectively for the heating and cooling set-point temperature schedules, and the air ventilation schedules). Beside the *volumeGeometry* attribute, the building levels occupied by this *UsageZone* may be also indicated by means of the attribute *usedFloors* (0 corresponding to the ground floor). As for *_AbstractBuilding* and *ThermalZone*, *UsageZone* may be characterized by several *floorArea* attributes of different types (e.g. net floor area, gross floor area).

The optional *averageInternalGains* attribute corresponds to the sum of the energy dissipated from the occupants and the facilities inside the zone. Its type *HeatExchangeType* (Energy ADE Core module) allows to detail the proportion of the different heat exchanges (convective, radiant and latent fraction).

2.3.2 BuildingUnit

BuildingUnit is a class introduced in the Energy ADE mainly to hold ownership information of a single usage zone for building usage analyses or for energy demand calculations. A *BuildingUnit* is a part of a single *UsageZone* which can be defined as a subdivision of a Building with its own lockable access from the outside or from a common area (i.e. not from another *BuildingUnit*), which is atomic, functionally independent, and may be separately sold, rented out, inherited, etc. The class holds owner information (attributes *ownerName* and *ownershipType*) as well as optional attributes for the number of rooms in the unit (*numberOfRooms*), its floor area (*floorArea*) and existing energy performance certifications (*energyPerformanceCertification*). The *BuildingUnit* may also be related to one or more addresses (property *address*).

2.3.3 Occupants and Household

An *Occupants* object identifies a homogeneous group of occupants of a *UsageZone* or *BuildingUnit*, categorized in one *occupancyType* (e.g. residents, workers, visitors etc.). *Occupants* is furthermore characterized by a given number of persons (*numberOfOccupants*) which occupied the corresponding zone following a certain time schedule (*occupancyRate*). Moreover, for the thermal modelling of the zone, the heat dissipation (*heatDissipation*) of a single occupant must also be specified.

For detailed analysis of occupant behaviour, the affiliation of occupants to specific households may be represented by relating an *Occupants* object with one or many *Household* objects.

One *Household* object identifies a group of persons living in the same dwelling, in the case where occupants are residents. They are defined by a type (*householdType*: one family, an adult, etc.) and the type of residence (*residenceType*: main residence, secondary residence or vacant).

2.3.4 Facilities

Facilities objects represent devices, except HVAC systems (which are modelled in the Energy Use and System module), consuming and dissipating thermal energy. Each *UsageZone* or *BuildingUnit* object may relate to one or more *Facilities* objects. There are three types of facilities: Facilities for domestic hot water (*DHWFacilities*), electrical appliances (*ElectricalAppliances*) and lighting (*LightingFacilities*). Each of them is characterized by a period of use (*operationSchedule*) and an emitted heat (*heatDissipation*), as well as some specific attributes depending on the facility type. In this case, the attribute *totalValue* of the data type *HeatExchangeType* holds the emitted thermal energy per square metre floor area of the usage zone.

2.4 Material and Construction module

The different classes of the *Material and Construction* module physically characterise building elements like walls, roofs, slabs or windows, detailing their structure and specifying their thermal and optical properties. The central feature type of the module is *Construction*, which may either be used directly or as *ReverseConstruction*, modelling a *baseConstruction* with inverted order of layers. In Figure 6 the corresponding UML diagram is depicted.

Each *Construction* object may be characterised by optical and/or physical properties. The *OpticalProperties* type specifies the *emissivity* (ratio of the long-wave radiation emitted by an object), the reflectance (fraction of incident radiation which is reflected by an object), the *transmittance* (fraction of incident radiation which passes through an object) and the *glazingRatio* (proportion of the construction surface which is transparent). There is no property for radiation absorptance, because according to the Kirchoff and Lambert law, the absorptance and the emittance are equal for a given wavelength range for a diffuse grey body.

The thermal properties of a *Construction* may be characterised with two possible "levels of detail": Either with the heat transmission coefficient *uValue* for steady-state thermal modelling, or by an ordered list of *Layer* objects, detailing different layers of materials and their thermal behaviour. Each *Layer* is composed of one or more *LayerComponent* objects representing a homogeneous part of a *Layer* (composed of a unique material) covering a given fraction (*areaFraction*) of it and having a specific *thickness*. Each *LayerComponent* is related with exactly one material object, either a *Gas* or a *SolidMaterial*. Both classes are derived from the abstract super class *AbstractMaterial*, which can optionally be related with an *ImageTexture* for visualization purposes.

The class *Gas* is foreseen to represent material layers with negligible thermal capacity like air gaps. It is characterized by a R-Value or thermal resistance (property *rValue*), and the

information whether the layer *isVentilated*. The class *SolidMaterial* aggregates a number of physical properties of construction materials like thermal *conductivity*, *density* and the *specificHeat* coefficient. Furthermore, information about the total amount of void space in the material (*porosity*), the water vapour transmission rate (*permeance*), and the amount of carbon (*embodiedCarbon*) and heat (*embodiedHeat*) embodied in the material can be represented.

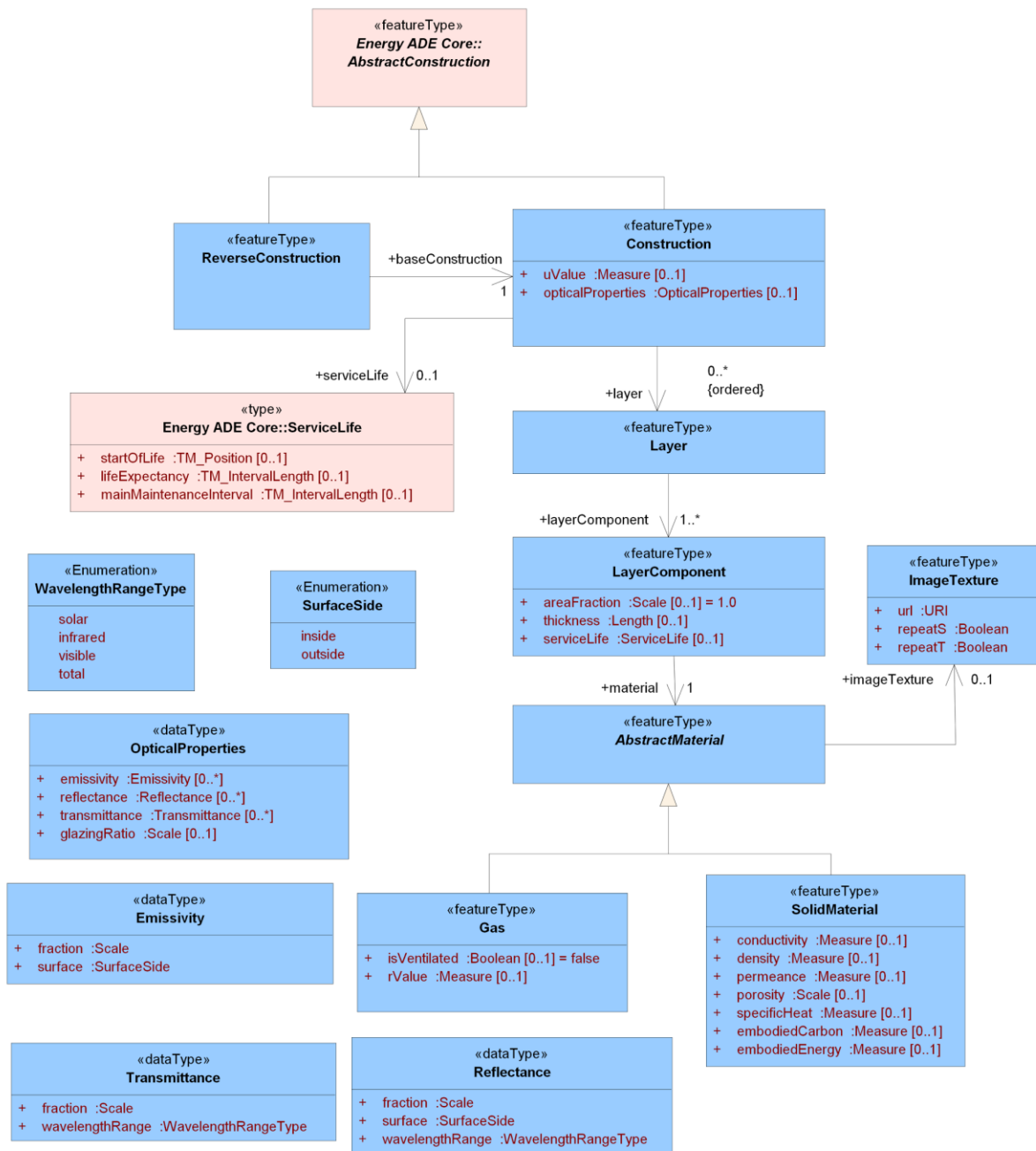


Figure 6: Energy ADE *Material and Construction* module

2.5 Energy systems module

The Energy System Module describes the whole energy transformation chain, from the energy sources to the energy demand of different city objects. This comprises the mapping

of different energy forms (energy demand, energy flow and energy sources) as well as energy supply systems consisting of conversion, distribution, storage and emitter systems. The Energy Systems Module is related to the *CityGML_Core* through the *EnergyDemand* and *AbstractEnergySystem* both from the Energy ADE core. Both elements extend the base class *_CityObject* of the *CityGML_Core*.

With the class *EnergyDemand*, detailed information regarding heating, cooling or electricity demands of a *_CityObject* (typically *_AbstractBuilding*, *ThermalZone* or *UsageZone*) can be modeled. This comprises the amount of energy modeled as time series as well as general demand specifications like the type of end use, the maximum load or the type of energy carrier. The separation of the class *EnergyDemand* from the rest of the Energy System Module (*AbstractEnergySystems* and subclasses) enables to specify energy demands of city objects without mapping a specific energy supply system. Especially for simulations with a low level of details, basic information about the energy demands of city objects are often sufficient.

2.5.1 Energy System base classes

A description of the energy supply system of city objects is supported by the class *AbstractEnergySystem* and its subclasses (see Figure 7). The complete energy supply system is further divided into four subsystems: Energy conversion system (*AbstractEnergyConversionSystem* and derived classes, see chapter 2.5.2 Figure 8), energy distribution and storage system (*AbstractEnergyDistributionSystem* / *AbstractStorageSystem* and derived classes, see chapter 2.5.3 Figure 9) and emitter system (*EmitterSystem*). All four subsystems directly inherit from *AbstractEnergySystem*.

The class *Emitter* describes the end unit that emits the useful energy (e.g. heat) in a zone where it is installed. It is characterized the emitter type (e.g. radiator), the installed power and the thermal exchange type, which provides the proportions of different thermal exchange mechanisms (e.g. convective or radiative).

In addition to the description of physical appliances, the flow of energy through the energy system is mapped and represented in the corresponding class *EnergyFlow*. Within this class, specific information regarding the amount of energy could be mapped as a time series. The information regarding the flow of energy are relevant for every subsystem (conversion, distribution, storage and emitter system) and therefore the class *EnergyFlow* is used in every of these subsystems. The attributes describing the flow of energy, representing an external energy source, are depicted by the *EnergySource*. Therefore, this enables to assign certain environmental factors of the flow of energy, such as primary energy factor, CO₂ emission factor and also the energy density.

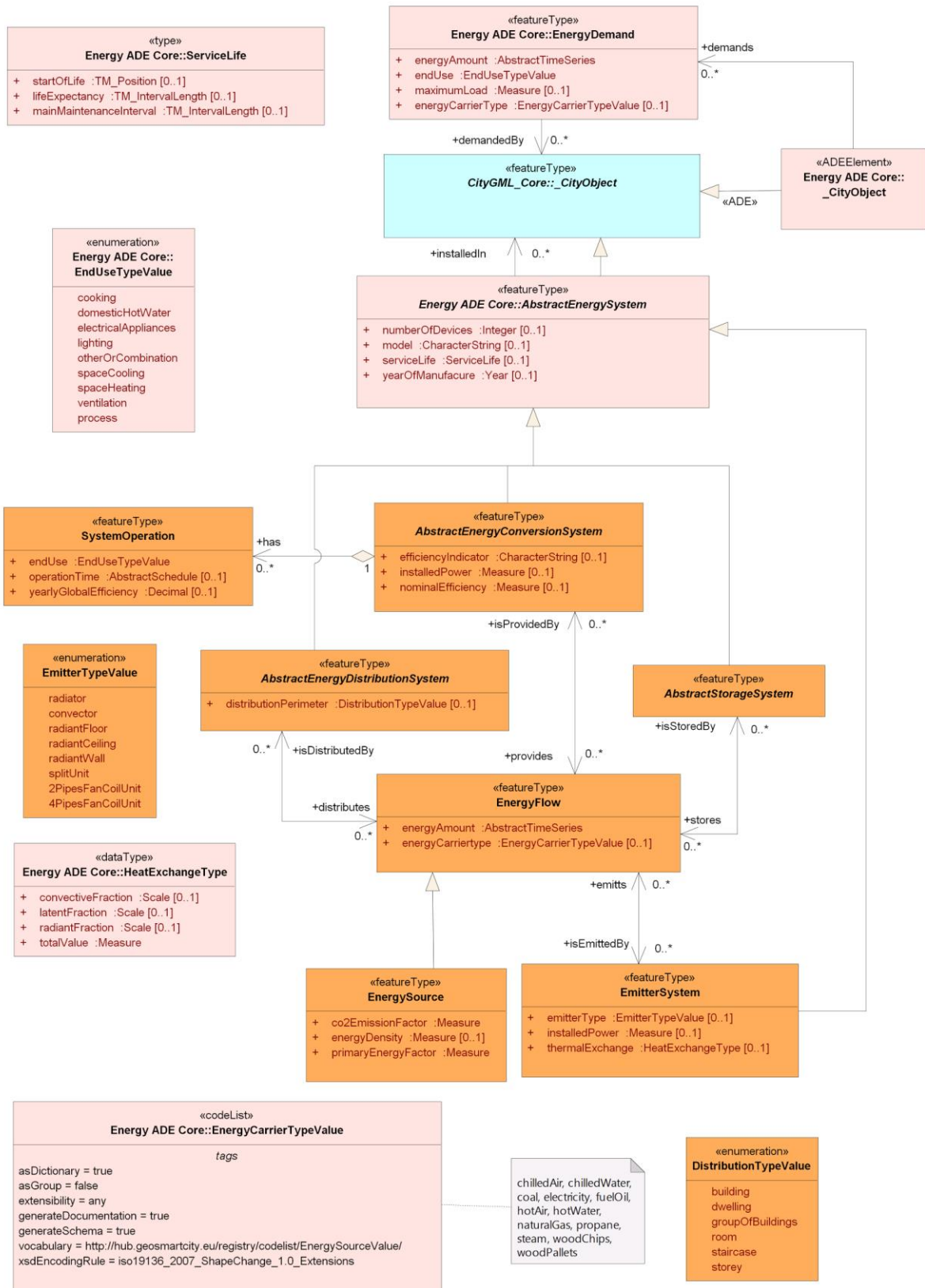


Figure 7: Energy Systems module overview

2.5.2 Energy conversion systems

The energy conversion system (see Figure 8) highlights the technical aspects concerning the conversion of energy from one type to another. The subclasses represent specific conversion systems, such as boilers, chillers, heat pumps, etc. For these conversion systems, fundamental physical properties, focusing on performance indicators (efficiency) or technical specifications are described. For the energy system classes, solar energy systems are further modelled. Moreover, the solar energy systems are differentiated into photovoltaic systems, solar thermal systems or hybrid systems, and are furthermore related to the CityGML classes *_BoundarySurface* and *BuildingInstallation*, which allows to relate a solar system directly to a specific boundary surface of a building installation (e.g. a roof). Furthermore, the *AbstractEnergyConversionSystem* relates to a *SystemOperation* object providing detailed information regarding operation times or performance indicators (e.g. yearly efficiency) for energy conversion systems.



Figure 8: Modelling of energy conversion systems

2.5.3 Energy distribution and storage system

The *energy distribution system* (see Figure 9) is used to portray basic technical specifications of thermal or electrical distribution systems, such as thermal hydraulic networks or electricity grids. However, it does not support complete network topologies for network simulations.

The energy storage system describes thermal and electrical storage systems, such as water tanks or batteries. The data model is designed similarly to the energy distribution system:

The main class *AbstractStorageSystem* is specified by the sub-specifications under thermal systems (*ThermalStorageSystem*) and the electrical systems (*PowerStorageSystem*).

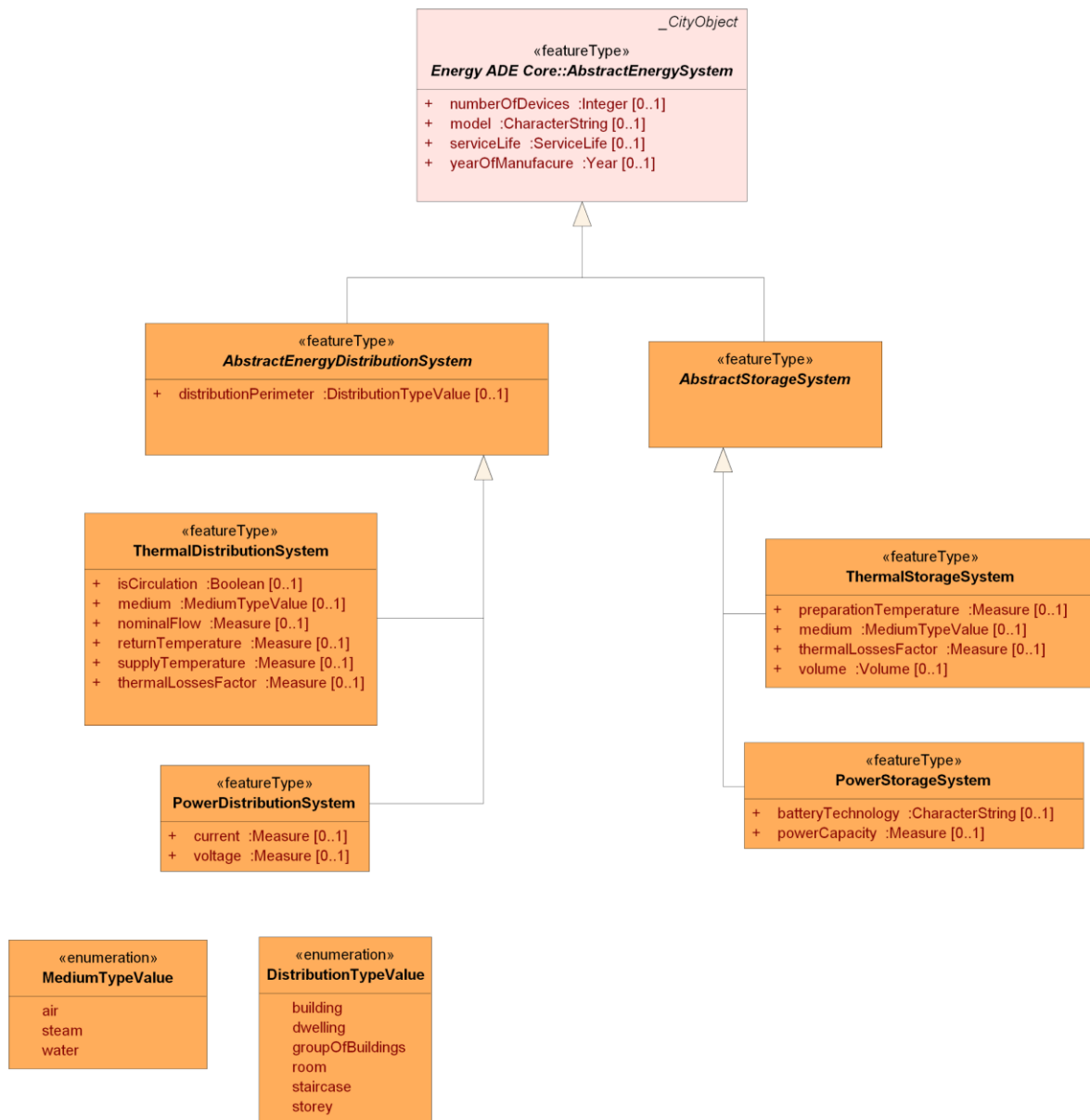


Figure 9: Modelling of energy distribution and storage systems

2.6 Support classes

There are a number of supporting classes and data types being used by the four functional modules. The following functional groups can be identified

- Representation of time series of physical values;
- Representation of different kinds of schedules for, e.g., set-point temperatures, operation of electrical devices or lighting, or occupancy rates;
- Representation of weather or climate data.

2.6.1 Time series

All classes to represent time series are derived from an abstract base class *AbstractTimeSeries*, holding a set of time series metadata (property *variableProperties* of type *TimeValuesProperties*) (see Figure 10). The available time series classes can be characterised by two concepts: The usage of a constant or varying time increment between different time series values, and the internal or external storage of the values.

Internal storage, which is realized in the classes *RegularTimeSeries* and *IrregularTimeSeries*, means that the time series values are embedded into the XML document, which may significantly blow up the file size. Alternatively, the time series data may be stored on an external device (classes *RegularTimeSeriesFile* and *IrregularTimeSeriesFile*), in tabular structure. In the latter case, the Energy ADE objects provide meta information to access the device (property *file*) and interpret the content (properties *numberOfHeaderLines*, *fieldSeparator*, *recordSeparator*, *decimalSymbol*, *valueColumnNumber* and *timeColumnNumber*).

The time series representations are additionally characterized by their supported time increments. Regular time series (*RegularTimeSeries*, *RegularTimeSeriesFile*) support the same time increment for all values. Thus, only start and end time instances (*temporalExtent*) and the time increment (*timeInterval*) need to be specified. In irregular time series (*IrregularTimeSeries*, *IrregularTimeSeriesFile*), each value has a specific time stamp, which is either specified with the data type *MeasurementPoint*, or a separate column (*timeColumnNumber*) in the external file.



Figure 10: Energy ADE models for time series of physical data

2.6.2 Schedules

Schedules are used for different purposes in the Energy ADE:

- For specifying set point values for the heating and cooling system,
- For defining the operational schedules of energy systems, ventilation, lighting and electrical appliances, and
- For specifying occupancy rates.

Four different schedule types are available, all being derived from the abstract base class *AbstractSchedule* (see Figure 11):

- The simplest schedule type *ConstantValueSchedule* only specifies one value (property *averageValue*).
- The *DualValueSchedule* defines two different values for a year: One *usageValue* and one *idleValue*. Optionally also the number of days per year (*usageDaysPerYear*) and the number of hours per day (*usageHoursPerDay*), where the *usageValue* is valid, can be specified.
- The most frequently used schedule type is the *DailyPatternSchedule*. It enables to define different time periods within a year (data type *PeriodOfYear*). For each period, one or more *DailySchedule* objects can be defined. The properties of a *DailySchedule* consist of a *dayType* (e.g. *weekDay*, *weekEnd* or a specific day within a week) and a corresponding time series (property *schedule*) for this type of day.
- The most complex and flexible schedule class is the *TimeSeriesSchedule*, supporting an arbitrary time series (property *timeDependingValues*) to define a schedule.

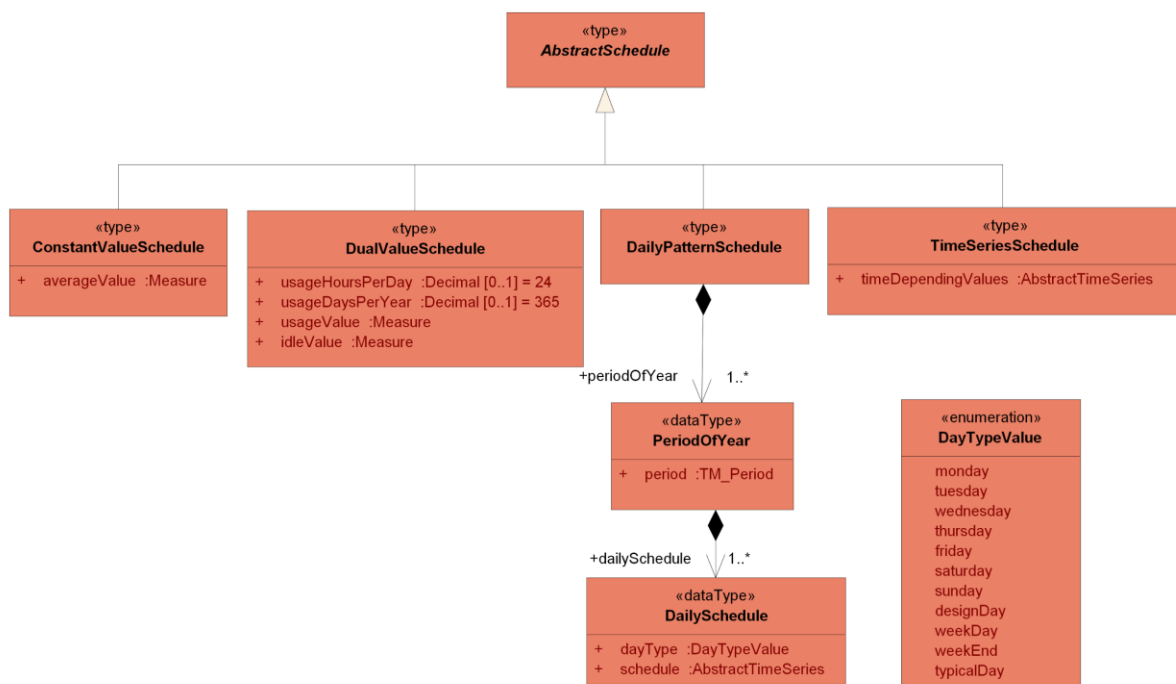


Figure 11: Energy ADE models for schedules

2.6.3 Weather data

Weather and climate information play an important role in energy related simulations. The Energy ADE defined two classes for modelling this kind of data (see Figure 12). The class *WeatherData*, which can be referenced by any CityGML city object via the new relation *weatherData*, supports a time series of meteorological parameters (property *value*), which are characterized by a *weatherDataType*. Optionally, also the geographical location of the measurement (property *position*) can be represented. The feature type *WeatherStation* aggregates different *WeatherData* objects for using this group in a simulation.

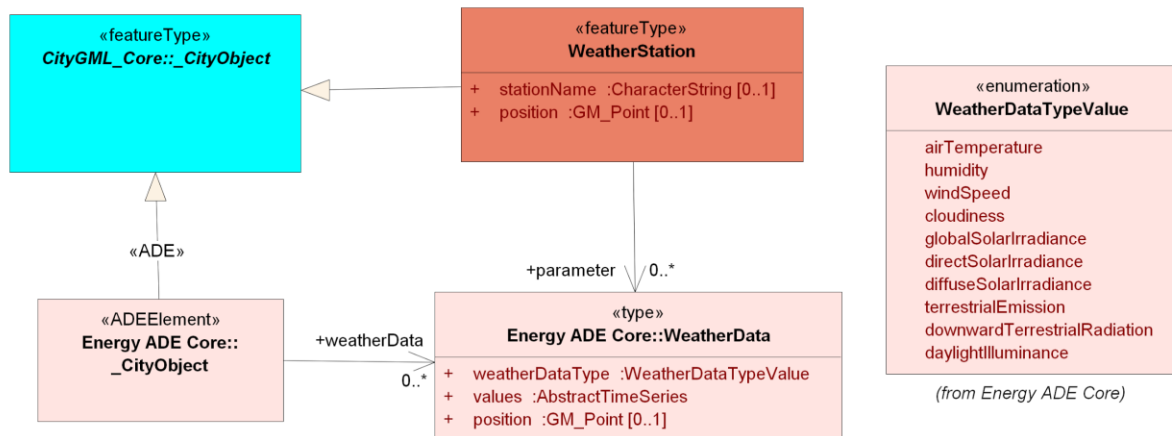


Figure 12: Modelling of weather data

3 Unit of Measurement identification

The GML standard represents physical properties of features with the type *gml:MeasureType*, extending the basic XML data type *xs:double* by a mandatory attribute *uom*. This attribute specifies the corresponding unit of measure, and can either be encoded as *UomSymbol* (arbitrary string, which must not contain colon (":"), space (" "), newline, carriage return or tab characters) or as *UomURI* (arbitrary URI). Because neither a registry for *UomURIs* is available, nor the GML standard itself specifies a list of *UomSymbols*, every GML application schema must specify its own strategy to encode uom identifiers.

Unfortunately, the base standard CityGML totally lacks this kind of information. Because the unambiguous interpretation of physical units of measure is essential for most use cases of the Energy ADE, this specification tries to fill the gaps of the base standard. Therefore, in table 1 a list of uom identifiers (in form of *UomSymbol* strings) is specified for all SI units. In some cases, also alternative uom identifiers (e.g. to represent energy measured in Joule or Kilowatt per hour) are specified.

Physical unit	Definition	uom	Alternative physical unit	uom	Conversion

AMPERE	Electric current	A			
BECQUEREL	Radioactivity	Bq			
CANDELA	Luminous intensity	cd			
CUBIC_METRE	Volume	m ³			
DEGREE_CELSIUS	Thermodynamic temperature	C	KELVIN	K	273,15 K = 0 C
FARAD	Electric capacity	F			
GRAM	Mass	g	KILOGRAM NANOGRAM	kg ng	1 kg = 1.000 g 1 ng = 1 kg*10 ⁻⁹
GRAY	Absorbed radioactive dose	Gy			
HENRY	Inductance	H			
HERTZ	Frequency	Hz			
JOULE	Energy	J	KILOJOULE KILOWATT_HOUR	kJ kWh	1 kJ = 1.000 J 1 kWh = 3.600.000 J
LUMEN	Luminous Flux	lm			
LUX	Illuminance	lx			
METRE	Length	m			
MOLE	Amount of substance	mol			
NEWTON	Force	N			
OHM	Electric resistance	Ohm			
PASCAL	Pressure	Pa	BAR	bar	1 bar = 10.000 Pa
RADIAN	Plane angle	rad	DEGREE	deg	1 deg = (1 rad*180)/π
SECOND	Time	s	HOUR	h	1 h = 3.600 s
SIEMENS	Electric conductance	S			
SIEVERT	Radioactive dose equivalent	Sv			

SQUARE_METRE	Area	m ²			
STERADIAN	Solid angle	sr			
TESLA	Magnetic flux density	T			
VOLT	Electric voltage	V			
WATT	Power	W	KILOWATT	kW	1 kW = 1.000 W
WEBER	Magnetic flux	Wb			

Table 1: uom identifiers for SI-units

Many energy relevant physical properties have complex units of measure, being defined by multiplying and dividing SI units. For generating such kinds of uom identifiers, the following rules are specified:

- A combined uom identifier consists of one numerator and one denominator, which are separated by a "/" character.
- Combined uom identifiers in numerator and denominator are alphabetically ordered and separated by "*" characters.
- Neither in the numerator nor in the denominator, blank characters or brackets are allowed.
- Multiplications of the same uom identifier are expressed with an integer number as suffix, e.g. m*m as m².

Following these rules, table 2 depicts the uom identifiers for physical quantities frequently occurring in the energy area.

Physical unit	Definition	uom
DENSITY	Volumetric mass density: KILOGRAM/CUBIC_METRE	kg/m ³
METRE_PER_SECOND	Velocity: METRE/SECOND	m/s
THERMAL_CONDUCTIVITY	Thermal conductivity: WATT/(METRE*KELVIN)	W/K*m
PERM	Water vapour transmission: NANOGRAM/(SECOND* SQUARE_METRE*PASCAL)	ng/m ² *Pa *s
SPECIFIC_HEAT	Specific heat capacity: KILOJOULE/(KILOGRAM*KELVIN)	kJ/K*kg

WATT_PER_SQUARE_METRE	Irradiance: Radiant flux (power) received by a surface per unit area: WATT/SQUARE_METRE	W/m2
KILOWATT_HOUR_PER_SQUARE_METRE	Energy amount per unit area: KILOWATT_HOUR/m2	kWh/m2
UVALUE	Thermal transmittance: WATT/(KELVIN*SQUARE_METRE)	W/K*m2
RVALUE	Thermal resistance, 1/ UVALUE	K*m2/W
FLOW_RATE	Flow rate of a liquid medium: CUBIC_METRE/SECOND	m3/s
FLOW_RATE_PER_HOUR	Flow rate of a liquid medium: CUBIC_METRE/HOUR	m3/h
SCALE	Ratio of two values, number between 0 and 1	scale
PERCENTAGE	Percentage, decimal number 0 and 100	percent
CLOUDINESS	Meteorological measure for the amount of clouds, integer number between 0 (no clouds) and 8 (fully clouded)	cloud
AIR_CHANGE_RATE_HOUR	Air volume added to or removed from a space in one hour, divided by the volume of the space.	1/h

Table 2: Combined uom identifiers

4 Energy ADE Codelists

For a number of Energy ADE attributes, the feasible values are defined by codelists. Though in principle any Energy ADE instance model may use its own Codelists, standard codelists have been defined by the development group. which are available in the INSPIRE Registry of the GeoSmartCity project (<http://hub.geosmartcity.eu/registry/codelist/>). Table depicts the codelists actually used in the Energy ADE. For accessing these codelists in the registry, the URL of the Codelist registry and the name of the Codelist are concatenated.

Codelist name	Usage in the Energy ADE
<i>RefurbishmentClassValue</i>	Property <i>level</i> of <i>RefurbishmentMeasure</i>
<i>BuildingTypeValue</i>	Property <i>buildingType</i> of <i>_AbstractBuilding</i>
<i>EnergyCarrierValue</i>	Property <i>energyCarrierType</i> of <i>EnergyDemand</i> Property <i>energyCarrierType</i> of <i>EnergyFlow</i>

<i>OccupantTypeValue</i>	Property <i>occupantType</i> of <i>Occupants</i>
<i>OwnershipTypeValue</i>	Property <i>ownershipType</i> of <i>BuildingUnit</i>
<i>CurrentUseValue</i>	Property <i>usageZoneType</i> of <i>UsageZone</i>

Table 3: Codelists used in the Energy ADE

5 Further information

CityGML 2.0 specification	https://portal.opengeospatial.org/files/?artifact_id=47842
CityGML Wiki	http://www.citygmlwiki.org/index.php?title=Citygml_Wiki
Energy ADE Wiki	http://en.wiki.energy.sig3d.org/index.php/Main_Page
Energy ADE GitHub	https://github.com/cstb/citygml-energy
Energy ADE XML Schema	http://www.citygmlwiki.org/images/a/ac/EnergyADE.xsd
Energy ADE UML diagrams (PDF document)	http://www.citygmlwiki.org/images/f/fb/UML-Diagrams_EnergyADE.pdf
Energy ADE Feature Catalogue	http://www.citygmlwiki.org/upload/EnergyADE%201.0/Feature_Catalogue/index.html