
DEXPI Specification

Release 2.0.0

DEXPI e.V.

Oct 10, 2025

CONTENTS

1	Introduction	3
1.1	Motivation for DEXPI	3
1.2	Objectives	3
1.3	Expectations	3
2	Basics	5
2.1	DEXPI XML	5
2.2	Builtin	16
3	Models	19
3.1	Core	19
3.2	Plant	183
3.3	Process	759
4	Appendix	1065
4.1	Glossary	1065
4.2	Reference P&ID	1065

License

This specification is provided by the DEXPI Initiative under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0). The full Legal Code of the license is available at <https://creativecommons.org/licenses/by/4.0/legalcode>. The following informative text is reproduced from <https://creativecommons.org/licenses/by/4.0>:

You are free to:

- **Share** – copy and redistribute the material in any medium or format
- **Adapt** – remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

- **Attribution** – You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- **No additional restrictions** – You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation.

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

INTRODUCTION

DEXPI e.V. (Data EXchange in the Process Industry) is an association under German law.

1.1 Motivation for DEXPI

Due to the lack of interoperability between CAE¹ (and other) systems, companies today face high efforts in data exchange while working together to execute projects for planning, construction and operation of process plants. Parties typically exchanging data in such projects are e.g. EP/EPCs², owner-operators, and vendors, but also site services and authorities. One of the main reasons for this high effort is the lack of an agreed understanding across the different systems, e.g. by means of a commonly used standard for data exchange in the process industry. To become more efficient during planning, construction and operation of plants, a data exchange model based on the ISO 15926 standard was established.

1.2 Objectives

The objective is to develop and promote a general method for data exchange, data interoperability and data integration for the process industry covering all phases of the lifecycle of a (petro-)chemical plant, ranging from specification of functional requirements to assets in operation. This method shall cover formats and content to address various problems seen today:

- Avoid format conversions (and thereby data loss) when passing engineering data and documents across CAE system boundaries.
- Make handover of engineering data during and at the end of a project easy and cost-effective.
- Reduce data exchange barriers between different CAE systems or different customizations of the same CAE systems.
- Support long-term storage of plant data in a CAE system independent format. Today's commonly used standard formats like the non data-centric PDF don't support value added improvements (or at best insufficiently).
- Simplify co-existence of different CAE systems within a company, e.g. due to mergers/acquisitions or different priorities in different business units.

1.3 Expectations

EP/EPCs, suppliers, and owner operators want to minimize the cost for handling engineering data during planning, construction and operation of process plants between different CAE systems and they want to create opportunities for new value-added functions based on the available engineering data. Therefore the CAE vendors will implement a valid global standard for data exchange into their CAE systems.

¹ CAE: Computer Aided Engineering

² EPC: Engineering-Procurement-Construction

The involved parties from DEXPI e.V. have defined a common data model which is based on the ISO 15926 standard. The resulting data model, as reflected in this specification, is aligned with other projects in the global ISO 15926 community. The CAE vendors will implement this common data model as the basis for data exchange and will deliver it as part of their default system configuration.

With the information model of the DEXPI 2.0 specification the information included in block flow diagrams (BFD, process model), process flow diagrams (PFD, process model), and piping and instrumentation diagrams (P&IDs³, plant model) can be represented. Objective of the first phase of the initiative was the transfer of a P&ID from one P&ID system to another P&ID system. The data transfer must include graphics, symbols, topology, all engineering attributes, enumerations, select lists etc. to enable seamless continuation of work on the P&ID in the destination system. With DEXPI 2.0, the process model is included and combined in the same specification as the plant model, now also enabling the transfer of process related information (as represented on BFDs and PFDs), including simulation results (like stream tables).

³ P&ID: Piping and Instrumentation Diagram

2.1 DEXPI XML

2.1.1 Overview

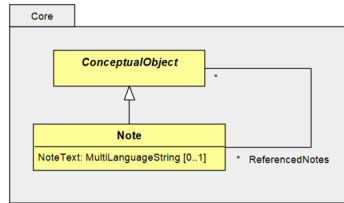
New Data Format

Principles

- XML (decision DEXPI group, approved by board)
- working title “DEXPI XML”
- key requirements
 - simple (based on very few basic rules)
 - easily readable for humans
 - future-proof
- suitable to transport both
 - the **information models** (“the specifications”, simple alternative to XMI)
 - the **actual data** (e.g., an actual P&ID or an actual PFD; simple replacement for Proteus in case of P&ID)

Example DEXPI XML: Information Model

Model with two Classes



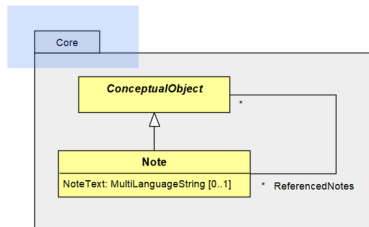
UML Diagram

```

<Model
  name = "Core"
  uri = "http://www.dexpi.org/specification/2.0/Core">
  <AbstractClass name = "ConceptualObject">
    <ReferenceProperty
      name = "ReferencedNotes"
      isOrdered = "false"
      isUnique = "false"
      lower = "0"
      oppositeLower = "0"
      type = "/Note"/>
    </AbstractClass>
  <ConcreteClass
    name = "Note"
    superTypes = "/ConceptualObject">
    <DataProperty
      name = "NoteText"
      isOrdered = "false"
      isUnique = "false"
      lower = "0"
      type = "/DataTypes.MultiLanguageString"
      upper = "1"/>
    </ConcreteClass>
  ...
  
```

Example DEXPI XML: Information Model

Model with two Classes



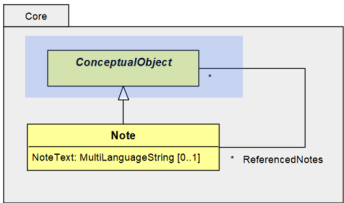
UML Diagram

```

<Model
  name = "Core"
  uri = "http://www.dexpi.org/specification/2.0/Core">
  <AbstractClass name = "ConceptualObject">
    <ReferenceProperty
      name = "ReferencedNotes"
      isOrdered = "false"
      isUnique = "false"
      lower = "0"
      oppositeLower = "0"
      type = "/Note"/>
    </AbstractClass>
  <ConcreteClass
    name = "Note"
    superTypes = "/ConceptualObject">
    <DataProperty
      name = "NoteText"
      isOrdered = "false"
      isUnique = "false"
      lower = "0"
      type = "/DataTypes.MultiLanguageString"
      upper = "1"/>
    </ConcreteClass>
  ...
  
```

Example DEXPI XML: Information Model

Model with two Classes

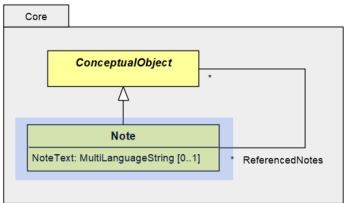


UML Diagram

```
<Model
  name = "Core"
  uri = "http://www.dexpi.org/specification/2.0/Core">
  <AbstractClass name = "ConceptualObject">
    <ReferenceProperty
      name = "ReferencedNotes"
      isOrdered = "false"
      isUnique = "false"
      lower = "0"
      oppositeLower = "0"
      type = "/Note"/>
    </AbstractClass>
  <ConcreteClass
    name = "Note"
    superTypes = "/ConceptualObject">
    <DataProperty
      name = "NoteText"
      isOrdered = "false"
      isUnique = "false"
      lower = "0"
      type = "/DataTypes.MultiLanguageString"
      upper = "1"/>
    </ConcreteClass>
  ...
```

Example DEXPI XML: Information Model

Model with two Classes



UML Diagram

```
<Model
  name = "Core"
  uri = "http://www.dexpi.org/specification/2.0/Core">
  <AbstractClass name = "ConceptualObject">
    <ReferenceProperty
      name = "ReferencedNotes"
      isOrdered = "false"
      isUnique = "false"
      lower = "0"
      oppositeLower = "0"
      type = "/Note"/>
    </AbstractClass>
  <ConcreteClass
    name = "Note"
    superTypes = "/ConceptualObject">
    <DataProperty
      name = "NoteText"
      isOrdered = "false"
      isUnique = "false"
      lower = "0"
      type = "/DataTypes.MultiLanguageString"
      upper = "1"/>
    </ConcreteClass>
  ...
```

Example DEXPI XML: Data

Object in a P&ID



CentrifugalPump1 : CentrifugalPump
DesignVolumeFlowRate = VolumeFlowRate(Value=200.0, Unit=VolumeFlowRateUnit.MetreCubedPerHour)

UML Diagram (details removed!)

```
<Model
  name = "mini_pid"
  uri = "http://www.example.org/mini_pid">
  <Import
    source = "http://www.dexpi.org/specification/2.0/Core"
    prefix = "Core"/>
  <Import
    source = "http://www.dexpi.org/specification/2.0/Plant"
    prefix = "Plant"/>
  <Object
    id = "CentrifugalPump1"
    type = "Plant/Equipment.CentrifugalPump">
    <Data property = "DesignVolumeFlowRate">
      <AggregatedDataValue type = "Core/PhysicalQuantities.VolumeFlowRate">
        <Data property = "Unit">
          <DataReference data = "Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour"/>
        </Data>
        <Data property = "Value">
          <Double>200.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
  ...
```

Example DEXPI XML: Data

Object in a P&ID



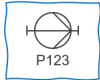
CentrifugalPump1 : CentrifugalPump
DesignVolumeFlowRate = VolumeFlowRate(Value=200.0, Unit=VolumeFlowRateUnit.MetreCubedPerHour)

UML Diagram (details removed!)

```
<Model
  name = "mini_pid"
  uri = "http://www.example.org/mini_pid">
  <Import
    source = "http://www.dexpi.org/specification/2.0/Core"
    prefix = "Core"/>
  <Import
    source = "http://www.dexpi.org/specification/2.0/Plant"
    prefix = "Plant"/>
  <Object
    id = "CentrifugalPump1"
    type = "Plant/Equipment.CentrifugalPump">
    <Data property = "DesignVolumeFlowRate">
      <AggregatedDataValue type = "Core/PhysicalQuantities.VolumeFlowRate">
        <Data property = "Unit">
          <DataReference data = "Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour"/>
        </Data>
        <Data property = "Value">
          <Double>200.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
  ...
```

Example DEXPI XML: Data

Object in a P&ID



CentrifugalPump1 : CentrifugalPump
DesignVolumeFlowRate = VolumeFlowRate(Value=200.0, Unit=VolumeFlowRateUnit.MetreCubedPerHour)

UML Diagram (details removed!)

```
<Model
  name = "mini_pid"
  uri = "http://www.example.org/mini_pid">
  <Import
    source = "http://www.dexpi.org/specification/2.0/Core"
    prefix = "Core"/>
  <Import
    source = "http://www.dexpi.org/specification/2.0/Plant"
    prefix = "Plant"/>
  <Object
    id = "CentrifugalPump1"
    type = "Plant/Equipment.CentrifugalPump">
    <Data property = "DesignVolumeFlowRate">
      <AggregatedDataValue type = "Core/PhysicalQuantities.VolumeFlowRate">
        <Data property = "Unit">
          <DataReference data = "Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour"/>
        </Data>
        <Data property = "Value">
          <Double>200.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
  ...
```

DEXPI XML

2.1.2 DEXPI XML Schema

Download DEXPI_XML_Schema_RC_1.xsd.

There is also a generated XML Schema documentation.

Several DEXPI XML files are provided. See

- *Model Builtin*,
- *Model Core*,
- *Model Plant*,
- *Model Process*,
- *DEXPI Reference P&ID*.

The root element of a DEXPI XML file has the name `Model`. Its complex type is `Model`.

Simple Types

ID

An ID is an `xsd:string` with the same restrictions as a `name`. In addition, an ID must be unique in a DEXPI XML file.

IDREF

An IDREF is a reference to an ID in the same file. It starts with a number sign (“hash”, #) that is immediately followed by the referenced ID.

name

A `name` is an `xsd:string` that

- starts with an underscore (`_`) or an upper (`A-Z`) or lower case (`a-z`) letter (ASCII only).
- The start character may be followed by an arbitrary number of underscores, upper or lower case letters, or digits (`0-9`) in an arbitrary order.

nameOrIdReference

A `nameOrIdReference` is either a `nameReference` or an `IDREF`.

nameOrIdReferences

A `nameOrIdReferences` contains one or more `nameOrIdReference` parts, separated by a space character.

nameReference

A `nameReference` is a reference to a `name` in an arbitrary DEXPI XML file. It is composed of

- a reference to a DEXPI XML file. This is either the `prefix` of an imported file (see complex type `Import`) or the empty string if it refers to the current file (i.e., the file that contains the `nameReference`);
- a slash character (`/`);
- a `name`. This `name` refers to the child element of the XML file's `Model` element with that `name`.

Optionally, the `nameReference` may contain further names, each separated by a dot (`.`), in order to refer to further, nested elements.

Complex Types

AbstractDataType

An abstract data type. An abstract data type is the only admissible supertype for any data type.

Attributes:

- `name`: the name of the abstract data type
- `superTypes`: the supertypes of the abstract data type; only other abstract data types are allowed

AggregatedDataType

An aggregated data type.

Attributes:

- `name`: the name of the aggregated data type
- `superTypes`: the supertypes of the aggregated data type; only abstract data types are allowed

Children:

- `DataProperty*`: the data properties declared for the aggregated data type

AggregatedDataValue

A value (“instance”) of an `AggregatedDataType`.

Attributes:

- `type`: the type of the `AggregatedDataValue`, i.e., an `AggregatedDataType`

Children:

- `Data*`: the instance’s values for the data properties declared for the `AggregatedDataType`

BooleanType

A primitive type, corresponding to `xsd:boolean`.

Attributes:

- `name`: the name of the type
- `superTypes`: the supertypes of the type; only abstract data types are allowed

AbstractClass

An abstract class.

Attributes:

- `name`: the name of the class

Children:

- `(CompositionProperty | DataProperty | ReferenceProperty)*`: the properties declared for the class

ClassExtension

An extension of a class that can be used to “inject” additional properties.

Attributes:

- `baseType`: a reference to the class that is extended
- `name`: the name of the class extension

Children:

- `(CompositionProperty | DataProperty | ReferenceProperty)*`: the properties declared for the class extension

Components

The values for a composition property for an object.

- `property`: the name of the property

Children:

- `(Object | ObjectReference)+`: the values for the property; if the property is ordered, the order of the values is relevant

CompositionProperty

A composition property, i.e., a property whose type must be a class and which indicates a composition (as opposed to a simple reference, cf. ReferenceProperty).

Attributes:

- isOrdered: whether the property is ordered, i.e., whether the order of values is relevant
- lower: lower multiplicity bound
- name: the name of the property
- type: the type of the property; must be a class
- upper: upper multiplicity bound

ConcreteClass

An concrete (non-abstract) class.

Attributes:

- name: the name of the class

Children:

- (CompositionProperty | DataProperty | ReferenceProperty)*: the properties declared for the class

Data

The values for a data property for an object or aggregated data type.

- property: the name of the property

Children:

- (AggregatedDataValue | DataReference | Boolean | DateTime | Double | Integer | String)+: the values for the property; if the property is ordered, the order of the values is relevant

DataProperty

A data property, i.e., a property whose type must be a data type.

Attributes:

- isOrdered: whether the property is ordered, i.e., whether the order of values is relevant
- isUnique: whether the property is unique, i.e., whether a value can appear multiple times (only relevant is upper is greater than 1)
- lower: lower multiplicity bound
- name: the name of the property
- type: the type of the property; must be a class
- upper: upper multiplicity bound

DataReference

A reference to a data value defined elsewhere.

This is used to refer to a SingletonValue or an EnumerationLiteral.

Attributes:

- data: the referenced data value

DateTimeType

A primitive type, corresponding to xsd:dateTime.

Attributes:

- name: the name of the type
- superTypes: the supertypes of the type; only abstract data types are allowed

DoubleType

A primitive type, corresponding to xsd:double.

Attributes:

- name: the name of the type
- superTypes: the supertypes of the type; only abstract data types are allowed

Enumeration

An enumeration.

Attributes:

- name: the name of the enumeration
- superTypes: the supertypes of the enumeration; only abstract data types are allowed

Children:

- EnumerationLiteral*: the literals of the enumeration

EnumerationLiteral

An enumeration literal.

Attributes:

- name: the name of the enumeration literal

Import

An import of a model by another model.

Attributes:

- prefix: a simple string to be used to refer to the imported model (in order to avoid the potentially complex uri)
- source: the uri of the imported model

IntegerType

A primitive type, corresponding to xsd:integer.

Attributes:

- name: the name of the type
- superTypes: the supertypes of the type; only abstract data types are allowed

Model

A model. A model is a top-level package that corresponds to a DEXPI XML file.

Attributes:

- name: the name of the model
- uri: a unique identifier for the model

Children:

- Import*: imports of other models that are referenced by the model
- (Package | AbstractClass | ConcreteClass | AbstractDataType | AggregatedDataType | Enumeration | SingletonType | SingletonValue | BooleanType | DateTimeType | DoubleType | IntegerType | StringType | Object)*: the content of the package

Object

A value (“instance”) of a Class.

Attributes:

- id: optional; an identifier for the object to be used for references within the same DEXPI XML file
- name: optional: a name for the object to be used for references, also from other files
- type: the type of the object, i.e., a class

Children:

- (Components | Data | References)*: the instance’s values for the properties declared for the Class

ObjectReference

A reference to an object.

Attributes:

- object: the actual reference

Package

A package.

Attributes:

- name: the name of the package

Children:

- (Package | AbstractClass | ConcreteClass | AbstractDataType | AggregatedDataType | Enumeration | SingletonType | SingletonValue | BooleanType | DateTimeType | DoubleType | IntegerType | StringType | Object)*: the content of the package

ReferenceProperty

A reference property, i.e., a property whose type must be a class and which indicates a simple reference (as opposed to a composition, cf. CompositionProperty).

Attributes:

- isOrdered: whether the property is ordered, i.e., whether the order of values is relevant
- isUnique: whether the property is unique, i.e., whether a value can appear multiple times (only relevant if upper is greater than 1)
- lower: lower multiplicity bound
- name: the name of the property
- oppositeLower: the opposite lower multiplicity bound
- oppositeUpper: the opposite upper multiplicity bound
- type: the type of the property; must be a class
- upper: upper multiplicity bound

References

The values for a reference property for an object.

- property: the name of the property
- objects: references to the values

SingletonType

A singleton data type, i.e., a type that has a single value (“instance”).

Attributes:

- name: the name of the type
- superTypes: the supertypes of the type; only abstract data types are allowed

SingletonValue

The value of a singleton type.

Attributes:

- name: the name of the value
- type: the singleton type

StringType

A primitive type, corresponding to xsd:string.

Attributes:

- name: the name of the type
- superTypes: the supertypes of the type; only abstract data types are allowed

2.2 Builtin

2.2.1 AnyURI

Builtin::AnyURI

Overview

StringType

<<primitive>>
AnyURI

2.2.2 Boolean

Builtin::Boolean

Overview

BooleanType

<<primitive>>
Boolean

2.2.3 DateTime

Builtin::DateTime

Overview

DateTimeType

<<primitive>>
DateTime

2.2.4 Double

Builtin::Double

Overview

DoubleType

<<primitive>>
Double

2.2.5 Integer

Builtin::Integer

Overview

IntegerType

<<primitive>>
Integer

2.2.6 String

Builtin::String

Overview

StringType

<<primitive>>
String

2.2.7 Undefined

Builtin::Undefined

Overview

UndefinedType

<<primitive>>
Undefined

2.2.8 UnsignedByte

Builtin::UnsignedByte

Overview

IntegerType

<<primitive>>
UnsignedByte

Builtin

2.2.9 Overview

Model

URI: <https://data.dexpi.org/meta/1.0.0/Builtin>

Builtin.xml

2.2.10 Members

Data Types

Name	Type
<i>AnyURI</i>	StringType
<i>Boolean</i>	BooleanType
<i>DateTime</i>	DateTimeType
<i>Double</i>	DoubleType
<i>Integer</i>	IntegerType
<i>String</i>	StringType
<i>Undefined</i>	UndefinedType
<i>UnsignedByte</i>	IntegerType

3.1 Core

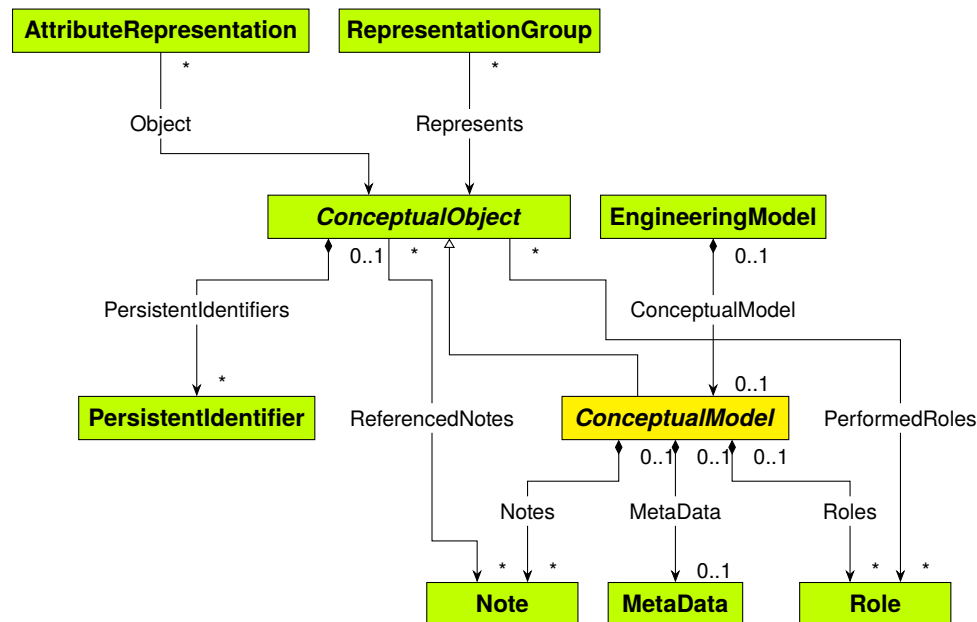
3.1.1 ConceptualModel

Core::ConceptualModel

Overview

Abstract Class

The conceptual content of an *EngineeringModel*, i.e., engineering information independent from its graphical representation.



Supertypes

- *ConceptualObject*

Subtypes

- *PlantModel*
- *ProcessModel*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>MetaData</i>	0..1	<i>MetaData</i>	no
<i>Notes</i>	*	<i>Note</i>	no
<i>Roles</i>	*	<i>Role</i>	no

MetaData

Composition Property

The *MetaData* of the *ConceptualModel*.

Type: *MetaData*

Multiplicity: 0..1

Ordered: no

Notes

Composition Property

The *Notes* of the *ConceptualModel*.

Type: *Note*

Multiplicity: *

Ordered: no

Roles

Composition Property

The *Roles* of the *ConceptualModel*.

Type: *Role*

Multiplicity: *

Ordered: no

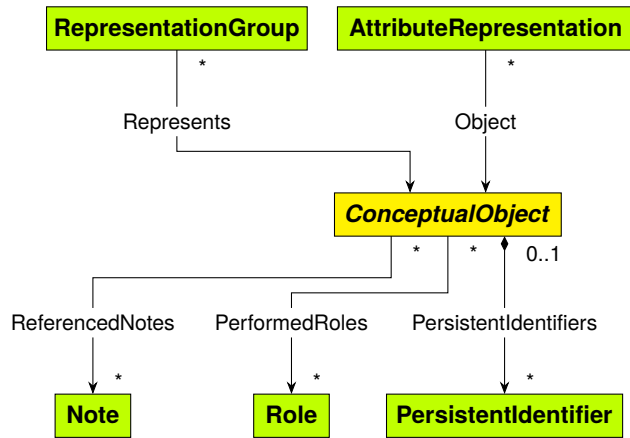
3.1.2 ConceptualObject

Core::ConceptualObject

Overview

Abstract Class

The abstract base class of all classes used in a *ConceptualModel*.



Subtypes

- *ActuatingElectricalFunction*
- *ActuatingElectricalSystem*
- *ActuatingFunction*
- *ActuatingSystem*
- *AgitatorRotor*
- *BriquettingRoller*
- *Chamber*
- *ColumnInternalsArrangement*
- *ColumnSection*
- *Composition*
- *ConceptualModel*
- *ControlledActuator*
- *CoolingTowerRotor*
- *CrusherElement*
- *Displacer*
- *DryingChamber*
- *ElectronicFrequencyConverter*
- *FilterUnit*
- *FilteringCentrifugeDrum*
- *GearBox*
- *GrindingElement*
- *HeatExchangerRotor*
- *Impeller*
- *InformationVariant*
- *InstrumentationActivity*

- *InstrumentationLoopFunction*
- *InstrumentationSystemActivity*
- *ListOfMaterialComponents*
- *MaterialComponent*
- *MaterialState*
- *MaterialStateType*
- *MaterialTemplate*
- *MeasuringElement*
- *MeasuringSystem*
- *MetaData*
- *MixingElementAssembly*
- *MotorAsComponent*
- *Mount*
- *Note*
- *Nozzle*
- *OperatedValveReference*
- *PelletizerDisc*
- *Pipe*
- *PipeOffPageConnector*
- *PipeOffPageConnectorReference*
- *PipingComponent*
- *PipingNetworkSegment*
- *PipingNetworkSystem*
- *PipingNode*
- *PlantStructureItem*
- *Port*
- *Positioner*
- *ProcessConnection*
- *ProcessInstrumentationFunction*
- *ProcessSignalGeneratingFunction*
- *ProcessStep*
- *ProcessStepDetail*
- *PropertyBreak*
- *QualifiedValue*
- *Screw*
- *SedimentalCentrifugeDrum*

- *SensorwellReference*
- *SieveElement*
- *SignalConveyingFunction*
- *SignalOffPageConnector*
- *SignalOffPageConnectorReference*
- *SprayNozzle*
- *TaggedPlantItem*
- *TransmissionSystem*
- *Transmitter*
- *TubeBundle*
- *Vent*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>PersistentIdentifiers</i>	*	<i>PersistentIdentifier</i>	no

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>PerformedRoles</i>	*	<i>Role</i>	no	no	*
<i>ReferencedNotes</i>	*	<i>Note</i>	no	no	*

PerformedRoles

Reference Property

The roles performed by the *ConceptualObject*.

Type: *Role*

Multiplicity: *

Ordered: no

Unique: no

Opposite Multiplicity: *

PersistentIdentifiers

Composition Property

The PersistentIdentifiers of the *ConceptualObject*.

Type: *PersistentIdentifier*

Multiplicity: *

Ordered: no

ReferencedNotes

Reference Property

The *Notes* about the *ConceptualObject*.

Type: *Note*

Multiplicity: *

Ordered: no

Unique: no

Opposite Multiplicity: *

3.1.3 DataTypes

MultiLanguageString

Core::DataTypes::MultiLanguageString

Overview

Aggregated Data Type

A container for one or more *SingleLanguageString*'s. *MultiLanguageString* is used as the type of data attributes which have language-dependent string values.

<<datatype>> MultiLanguageString
SingleLanguageStrings : SingleLanguageString [1..*]

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>SingleLanguageStrings</i>	1..*	<i>SingleLanguageString</i>	no	no

SingleLanguageStrings

Data Property

Language-dependent values of the *MultiLanguageString*.

Type: *SingleLanguageString*

Multiplicity: 1..*

Ordered: no

Unique: no

Example

```
[
  SingleLanguageString(
    Language = "en",
    Value = "This is some sample text.")
]
```

QuantityProvenance

Core::DataTypes::QuantityProvenance

Overview

Enumeration

<<enumeration>> QuantityProvenance
Calculated
Estimated
Observed
Set
Specified

Members

Literals

Name
Calculated
Estimated
Observed
Set
Specified

QuantityRange

Core::DataTypes::QuantityRange

Overview

Enumeration

<<enumeration>> QuantityRange
Actual Average LowerLimit Nominal Normal UpperLimit

Members

Literals

Name
Actual
Average
LowerLimit
Nominal
Normal
UpperLimit

Scope

Core::DataTypes::Scope

Overview

Enumeration

<<enumeration>> Scope
Alarm Allowable Design Expected Incidental Operating Protection Rated Test Warning

Members

Literals

Name	
Alarm	The value raises an alarm.
Allowable	The value that can be tolerated for an extended period. It has a specific meaning in safety specifications.
Design	The value is that used for design. It has a specific meaning in safety specifications.
Expected	A value that is estimated and subject to uncertainty.
Incidental	The value that can be tolerated for a short period. It has a specific meaning in safety specifications.
Operating	A value specified for operation.
Protection	The value initiates a protective action.
Rated	A value assigned by the manufacturer representing a specified operating condition.
Test	The value that will be set to test a system or piece of equipment.
Warning	The value raises a warning.

SingleLanguageString

Core::DataTypes::SingleLanguageString

Overview

Aggregated Data Type

A SingleLanguageString contains a *Value* and a *Language* tag. SingleLanguageString is only used within *MultiLanguageString*.

<<datatype>> SingleLanguageString		
Language :	Undefined	String
Value :	Undefined	String

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Language</i>	1	<i>Undefined</i> <i>String</i>	no	no
<i>Value</i>	1	<i>Undefined</i> <i>String</i>	no	no

Language

Data Property

The language of the *Value* of the *SingleLanguageString*. If not *Undefined*, the language must be given as a language tag standardized by IETF BCP 47 (Best Current Practice 47 by the Internet Engineering Task Force). IETF BCP 47 is based on ISO 639.

Type: *Undefined* | *String*

Multiplicity: 1

Ordered: no

Unique: no

Language tags

In DEXPI, only those language tags are allowed

- that are listed in the normative registry file at <https://www.iana.org/assignments/language-subtag-registry/language-subtag-registry> from September 29, 2020,
- that consist of exactly two letters (in particular extended tags such as *en-US* or *de-CH-1996* are not allowed),
- that refer to an actual language (the *type* defined by IETF BCP 47 must be *language*),
- and that are not deprecated.

The table below contains all permitted language tags.

Tag	Language
aa	Afar
ab	Abkhazian
ae	Avestan
af	Afrikaans
ak	Akan
am	Amharic
an	Aragonese
ar	Arabic
as	Assamese
av	Avaric
ay	Aymara
az	Azerbaijani
ba	Bashkir
be	Belarusian
bg	Bulgarian
bh	Bihari languages
bi	Bislama
bm	Bambara
bn	Bengali/Bangla
bo	Tibetan
br	Breton
bs	Bosnian
ca	Catalan/Valencian
ce	Chechen
ch	Chamorro
co	Corsican
cr	Cree
cs	Czech
cu	Church Slavic/Church Slavonic/Old Bulgarian/Old Church Slavonic/Old Slavonic
cv	Chuvash
cy	Welsh
da	Danish

continues on next page

Table 1 – continued from previous page

Tag	Language
de	German
dv	Dhivehi/Divehi/Maldivian
dz	Dzongkha
ee	Ewe
el	Modern Greek (1453-)
en	English
eo	Esperanto
es	Spanish/Castilian
et	Estonian
eu	Basque
fa	Persian
ff	Fulah
fi	Finnish
fj	Fijian
fo	Faroese
fr	French
fy	Western Frisian
ga	Irish
gd	Scottish Gaelic/Gaelic
gl	Galician
gn	Guarani
gu	Gujarati
gv	Manx
ha	Hausa
he	Hebrew
hi	Hindi
ho	Hiri Motu
hr	Croatian
ht	Haitian/Haitian Creole
hu	Hungarian
hy	Armenian
hz	Herero
ia	Interlingua (International Auxiliary Language Association)
id	Indonesian
ie	Interlingue/Occidental
ig	Igbo
ii	Sichuan Yi/Nuosu
ik	Inupiaq
io	Ido
is	Icelandic
it	Italian
iu	Inuktitut
ja	Japanese
jv	Javanese
ka	Georgian
kg	Kongo
ki	Kikuyu/Gikuyu
kj	Kuanyama/Kwanyama
kk	Kazakh
kl	Kalaallisut/Greenlandic

continues on next page

Table 1 – continued from previous page

Tag	Language
km	Khmer/Central Khmer
kn	Kannada
ko	Korean
kr	Kanuri
ks	Kashmiri
ku	Kurdish
kv	Komi
kw	Cornish
ky	Kirghiz/Kyrgyz
la	Latin
lb	Luxembourgish/Letzeburgesch
lg	Ganda/Luganda
li	Limburgan/Limburger/Limburgish
ln	Lingala
lo	Lao
lt	Lithuanian
lu	Luba-Katanga
lv	Latvian
mg	Malagasy
mh	Marshallese
mi	Maori
mk	Macedonian
ml	Malayalam
mn	Mongolian
mr	Marathi
ms	Malay (macrolanguage)
mt	Maltese
my	Burmese
na	Nauru
nb	Norwegian Bokmål
nd	North Ndebele
ne	Nepali (macrolanguage)
ng	Ndonga
nl	Dutch/Flemish
nn	Norwegian Nynorsk
no	Norwegian
nr	South Ndebele
nv	Navajo/Navaho
ny	Nyanja/Chewa/Chichewa
oc	Occitan (post 1500)
oj	Ojibwa
om	Oromo
or	Oriya (macrolanguage)/Odia (macrolanguage)
os	Ossetian/Ossetic
pa	Panjabi/Punjabi
pi	Pali
pl	Polish
ps	Pushto/Pashto
pt	Portuguese
qu	Quechua

continues on next page

Table 1 – continued from previous page

Tag	Language
rm	Romansh
rn	Rundi
ro	Romanian/Moldavian/Moldovan
ru	Russian
rw	Kinyarwanda
sa	Sanskrit
sc	Sardinian
sd	Sindhi
se	Northern Sami
sg	Sango
sh	Serbo-Croatian
si	Sinhala/Sinhalese
sk	Slovak
sl	Slovenian
sm	Samoan
sn	Shona
so	Somali
sq	Albanian
sr	Serbian
ss	Swati
st	Southern Sotho
su	Sundanese
sv	Swedish
sw	Swahili (macrolanguage)
ta	Tamil
te	Telugu
tg	Tajik
th	Thai
ti	Tigrinya
tk	Turkmen
tl	Tagalog
tn	Tswana
to	Tonga (Tonga Islands)
tr	Turkish
ts	Tsonga
tt	Tatar
tw	Twi
ty	Tahitian
ug	Uighur/Uyghur
uk	Ukrainian
ur	Urdu
uz	Uzbek
ve	Venda
vi	Vietnamese
vo	Volapük
wa	Walloon
wo	Wolof
xh	Xhosa
yi	Yiddish
yo	Yoruba

continues on next page

Table 1 – continued from previous page

Tag	Language
za	Zhuang/Chuang
zh	Chinese
zu	Zulu

Example

"en"

Value**Data Property**

The actual value of the *SingleLanguageString*.

Type: *Undefined* | *String*

Multiplicity: 1

Ordered: no

Unique: no

Example

"This is some sample text."

Core::DataTypes

Overview**Package**

Basic data types such as strings, numbers, and enumerations.

Members**Data Types**

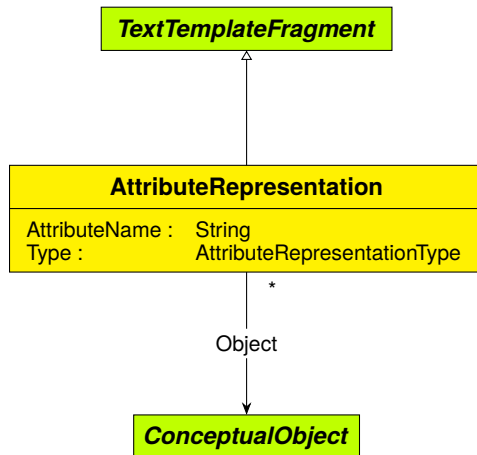
Name	Type
<i>MultiLanguageString</i>	Aggregated Data Type
<i>QuantityProvenance</i>	Enumeration
<i>QuantityRange</i>	Enumeration
<i>Scope</i>	Enumeration
<i>SingleLanguageString</i>	Aggregated Data Type

3.1.4 Diagram**AttributeRepresentation**

Core::Diagram::AttributeRepresentation

Overview

Concrete Class



Supertypes

- *TextTemplateFragment*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Object</i>	1	<i>ConceptualObject</i>	no	no	*

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>AttributeName</i>	1	<i>String</i>	no	no
<i>Type</i>	1	<i>AttributeRepresentationType</i>	no	no

AttributeName

Data Property

Type: *String*

Multiplicity: 1

Ordered: no

Unique: no

Example

"SubTagName"

Object

Reference Property

Type: *ConceptualObject*

Multiplicity: 1

Ordered: no

Unique: no

Opposite Multiplicity: *

Type

Data Property

Type: *AttributeRepresentationType*

Multiplicity: 1

Ordered: no

Unique: no

Example

Value

Example in DEXPI XML

```
<Object type="Core/Diagram.AttributeRepresentation">
  <Data property="AttributeName">
    <String>SubTagName</String>
  </Data>
  <Data property="Type">
    <DataReference data="Core/Diagram.AttributeRepresentationType.Value"/>
  </Data>
</Object>
```

AttributeRepresentationType

Core::Diagram::AttributeRepresentationType

Overview

Enumeration

<<enumeration>> AttributeRepresentationType
Units Value ValueAndUnits

Members

Literals

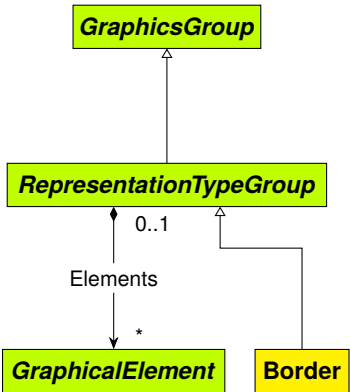
Name
Units
Value
ValueAndUnits

Border

Core::Diagram::Border

Overview

Concrete Class



Supertypes

- *RepresentationTypeGroup*

Example in DEXPI XML

```
<Object type="Core/Diagram.Border"/>
```

Color

Core::Diagram::Color

Overview

Aggregated Data Type

A color.

Color is based on the RGB color model (e.g., see [Wikipedia](#) for an overview). For each channel *red*, *green*, and *blue*, Color has an attribute *R*, *G*, and *B* that gives the intensity of the channel. Intensities are given as values of `Core.DataTypes.UnsignedByte`, i.e., integers in the range [0, 255].

<<datatype>> Color
B : UnsignedByte G : UnsignedByte R : UnsignedByte

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>B</i>	1	<i>UnsignedByte</i>	no	no
<i>G</i>	1	<i>UnsignedByte</i>	no	no
<i>R</i>	1	<i>UnsignedByte</i>	no	no

Simplified Literals

Within this specification, we use the hexadecimal SVG representation of a *Color* (without the string delimiters) as a convenient literal to refer to that *Color*.

For example, `#0000ff` refers to the `<Core.Diagram.Color>` *blue* with `R=0`, `G=0`, and `B=255`.

Mapping to SVG

There are several syntactical variants to map a *Color* to SVG. Here, we use the [RGB Hexadecimal Notation](#) with 6 digits as defined by [standard-css-color-4](#). It has the form `'#RRGGBB'`, where `RR`, `GG`, and `BB` are 2-digit hexadecimal representations of the integer values of the attributes *R*, *G*, and *B*. Conventional hexadecimal digits are used: `0`, ..., `9`, `a`, ..., `f`. Capitalized versions of the latter are also allowed: `A`, ..., `F`. To ensure 2 digits for each integer, a leading zero is added if required.

example

Consider the *Color* with `R=0`, `G=10`, and `B=255`.

The hexadecimal representations of the attributes are `00`, `0a`, and `ff`.

Hence, the SVG representation of the color is `'#000aff'`.

B

Data Property

The intensity of the blue channel of the *Color*.

Type: *UnsignedByte*

Multiplicity: 1

Ordered: no

Unique: no

Example

255

G

Data Property

The intensity of the green channel of the *Color*.

Type: *UnsignedByte*

Multiplicity: 1

Ordered: no

Unique: no

Example

0

R

Data Property

The intensity of the red channel of the *Color*.

Type: *UnsignedByte*

Multiplicity: 1

Ordered: no

Unique: no

Example

127

ConfidentialityClassification

*Core::Diagram::*ConfidentialityClassification

Overview

Enumeration

<<enumeration>>
ConfidentialityClassification
ConfidentialInformation
NonConfidentialInformation

Members

Literals

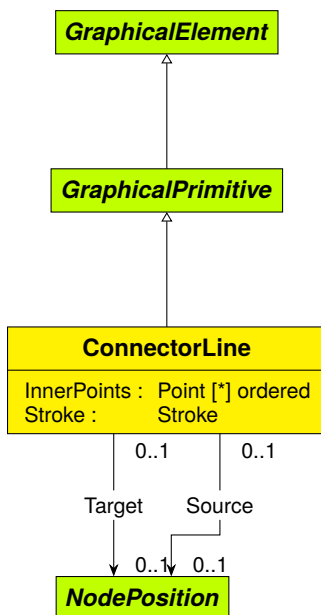
Name	Symbol	RDL Label	RDL URI
ConfidentialInformation	confidential	CONFIDENTIAL INFORMATION	http://data.posccaesar.org/rdl/RDS4316590816
NonConfidentialInformation	not confidential	NON CONFIDENTIAL INFORMATION	http://sandbox.dexpi.org/rdl/NonConfidentialInformation

ConnectorLine

Core::Diagram::ConnectorLine

Overview

Concrete Class



Supertypes

- *GraphicalPrimitive*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Source</i>	0..1	<i>NodePosition</i>	no	no	0..1
<i>Target</i>	0..1	<i>NodePosition</i>	no	no	0..1

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>InnerPoints</i>	*	<i>Point</i>	yes	no
<i>Stroke</i>	1	<i>Stroke</i>	no	no

InnerPoints

Data Property

The inner points of the *ConnectorLine*.

Type: *Point*

Multiplicity: *

Ordered: yes

Unique: no

Example

```
[
  Point(
    X = 9.7,
    Y = 16.64)
]
```

Source

Reference Property

Type: *NodePosition*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

Stroke

Data Property

The stroke of the *ConnectorLine*.

Type: *Stroke*

Multiplicity: 1

Ordered: no

Unique: no

Example

```
Stroke(  
  Color = Color(  
    B = 0,  
    G = 0,  
    R = 128),  
  DashStyle = Solid,  
  Width = 0.3)
```

Target

Reference Property

Type: *NodePosition*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

Example in DEXPI XML

```
<Object type="Core/Diagram.ConnectorLine">  
  <Data property="InnerPoints">  
    <AggregatedDataValue type="Core/Diagram.Point">  
      <Data property="X">  
        <Double>9.7</Double>  
      </Data>  
      <Data property="Y">  
        <Double>16.64</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
  <Data property="Stroke">  
    <AggregatedDataValue type="Core/Diagram.Stroke">  
      <Data property="Color">  
        <AggregatedDataValue type="Core/Diagram.Color">  
          <Data property="B">  
            <Integer>0</Integer>  
          </Data>  
          <Data property="G">  
            <Integer>0</Integer>  
          </Data>  
          <Data property="R">  
            <Integer>128</Integer>  
          </Data>  
        </AggregatedDataValue>  
      </Data>  
    </AggregatedDataValue>  
  </Data>
```

(continues on next page)

(continued from previous page)

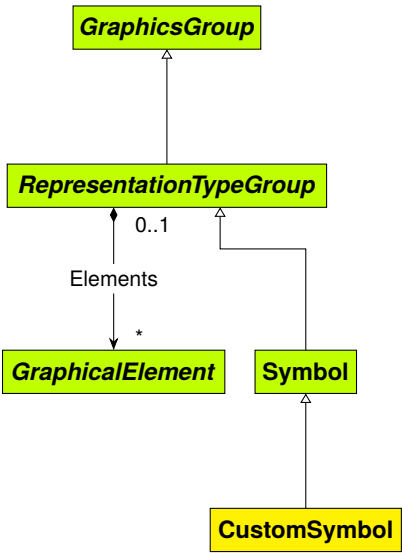
```
<Data property="DashStyle">
  <DataReference data="Core/Diagram.DashStyle.Solid"/>
</Data>
<Data property="Width">
  <Double>0.3</Double>
</Data>
</AggregatedDataValue>
</Data>
</Object>
```

CustomSymbol

Core::Diagram::CustomSymbol

Overview

Concrete Class



Supertypes

- *Symbol*

Example in DEXPI XML

```
<Object type="Core/Diagram.CustomSymbol"/>
```

DashStyle

Core::Diagram::DashStyle

Overview

Enumeration

A `DashStyle` describes a pattern for drawing a stroke. The pattern gives the lengths of those parts of a stroke that are actually drawn and of those parts that are not drawn.

<code><<enumeration>></code> DashStyle
Dash DashShortDash Dot LongDash LongDashShortDash LongDashShortDashShortDash ShortDash Solid

Members

Literals

Name
Dash
DashShortDash
Dot
LongDash
LongDashShortDash
LongDashShortDashShortDash
ShortDash
Solid

Appearance

Each `DashStyle` is characterized by a sequence of an even number of double values. The first of these values gives the length of a visible part of a stroke, the second one the length of an invisible part. Optional further pairs of double values in the pattern are applied in the same way. The entire pattern is applied repeatedly until the end of a stroke.

The lengths in a `DashStyle`'s pattern are given in terms of the *Stroke's Width* in mm, i.e., the dimensionless double values must be multiplied with the *Width* in order to get the actual lengths.

The following table gives heuristically proven patterns for the different *DashStyles*.

Solid

linecap: butt	dash pattern: []	
linecap: round	dash pattern: []	

Dot

linecap: butt	dash pattern: [1, 1]	
linecap: round	dash pattern: [0, 2]	

Dash

linecap: butt	dash pattern: [6, 6]	
linecap: round	dash pattern: [5, 7]	

LongDash

linecap: butt	dash pattern: [12, 12]	
linecap: round	dash pattern: [11, 13]	

LongDashShortDash

linecap: butt	dash pattern: [12, 12, 3, 12]	
linecap: round	dash pattern: [11, 13, 2, 13]	

ShortDash

linecap: butt	dash pattern: [4, 0.5]	
linecap: round	dash pattern: [3, 1.5]	

LongDashShortDashShortDash

linecap: butt	dash pattern: [12, 12, 3, 5, 3, 12]	
linecap: round	dash pattern: [11, 13, 2, 6, 2, 13]	

DashShortDash

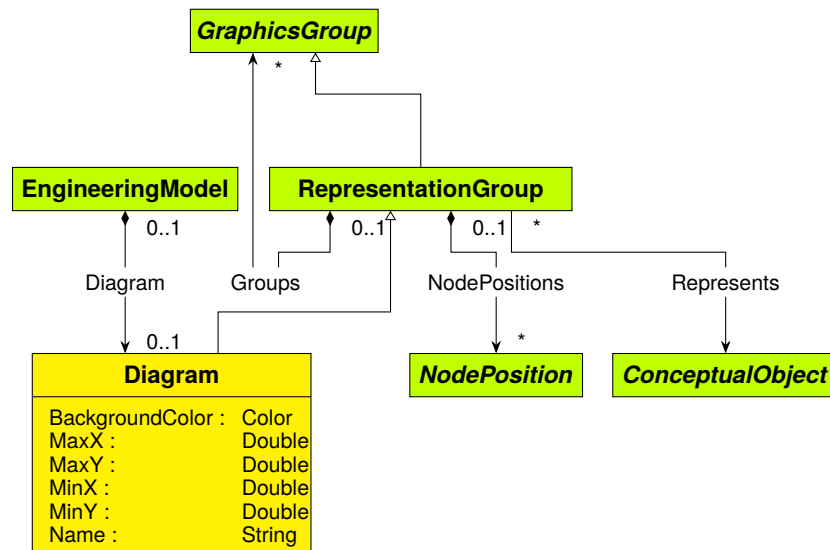
linecap: butt	dash pattern: [6, 6, 3, 6]	
linecap: round	dash pattern: [5, 7, 2, 7]	

Diagram

Core::Diagram::Diagram

Overview

Concrete Class



Supertypes

- *RepresentationGroup*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>BackgroundColor</i>	1	<i>Color</i>	no	no
<i>MaxX</i>	1	<i>Double</i>	no	no
<i>MaxY</i>	1	<i>Double</i>	no	no
<i>MinX</i>	1	<i>Double</i>	no	no
<i>MinY</i>	1	<i>Double</i>	no	no
<i>Name</i>	1	<i>String</i>	no	no

BackgroundColor

Data Property

Type: *Color*

Multiplicity: 1

Ordered: no

Unique: no

Example

```
Color(  
    B = 255,  
    G = 255,  
    R = 255)
```

MaxX**Data Property**

Type: *Double*

Multiplicity: 1

Ordered: no

Unique: no

Example

```
100.0
```

MaxY**Data Property**

Type: *Double*

Multiplicity: 1

Ordered: no

Unique: no

Example

```
40.0
```

MinX**Data Property**

Type: *Double*

Multiplicity: 1

Ordered: no

Unique: no

Example

```
0.0
```

MinY**Data Property****Type:** *Double***Multiplicity:** 1**Ordered:** no**Unique:** no**Example**

0.0

Name**Data Property****Type:** *String***Multiplicity:** 1**Ordered:** no**Unique:** no**Example**

"Example P&ID"

Example in DEXPI XML

```
<Object type="Core/Diagram.Diagram">
  <Data property="BackgroundColor">
    <AggregatedDataValue type="Core/Diagram.Color">
      <Data property="B">
        <Integer>255</Integer>
      </Data>
      <Data property="G">
        <Integer>255</Integer>
      </Data>
      <Data property="R">
        <Integer>255</Integer>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="MaxX">
    <Double>100.0</Double>
  </Data>
  <Data property="MaxY">
    <Double>40.0</Double>
  </Data>
  <Data property="MinX">
    <Double>0.0</Double>
  </Data>
```

(continues on next page)

(continued from previous page)

```

<Data property="MinY">
  <Double>0.0</Double>
</Data>
<Data property="Name">
  <String>Example P&ID</String>
</Data>
</Object>

```

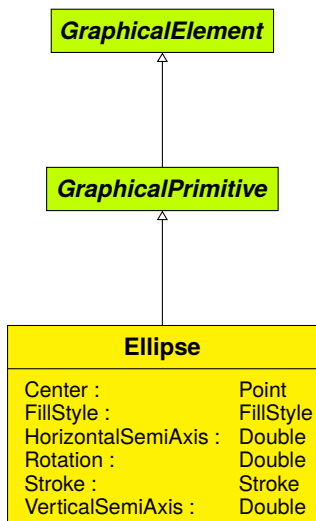
Ellipse

Core::Diagram::Ellipse

Overview

Concrete Class

An ellipse.



Supertypes

- *GraphicalPrimitive*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Center</i>	1	<i>Point</i>	no	no
<i>FillStyle</i>	1	<i>FillStyle</i>	no	no
<i>HorizontalSemiAxis</i>	1	<i>Double</i>	no	no
<i>Rotation</i>	1	<i>Double</i>	no	no
<i>Stroke</i>	1	<i>Stroke</i>	no	no
<i>VerticalSemiAxis</i>	1	<i>Double</i>	no	no

Geometry

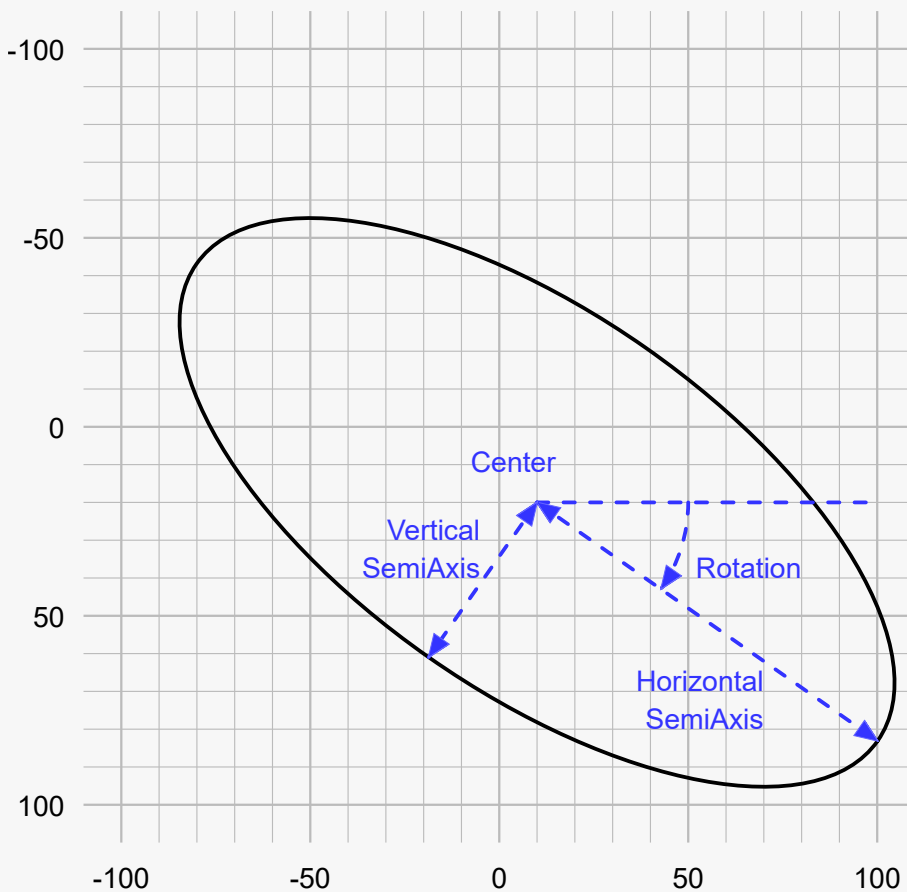
The geometry of an Ellipse is characterized by its *Center*, its two axes (*HorizontalSemiAxis* and *VerticalSemiAxis*), and the *Rotation* of the *HorizontalSemiAxis* w.r.t. the x axis of the owner. The *Rotation* is measured clockwise and in degrees.

example

We consider an Ellipse with

- *Center* (10, 20),
- *HorizontalSemiAxis* 110,
- *VerticalSemiAxis* 50, and
- *Rotation* 35.

The Ellipse will look like this:



Visual Center

The visual center of an Ellipse is its actual *Center*.

Mapping to SVG

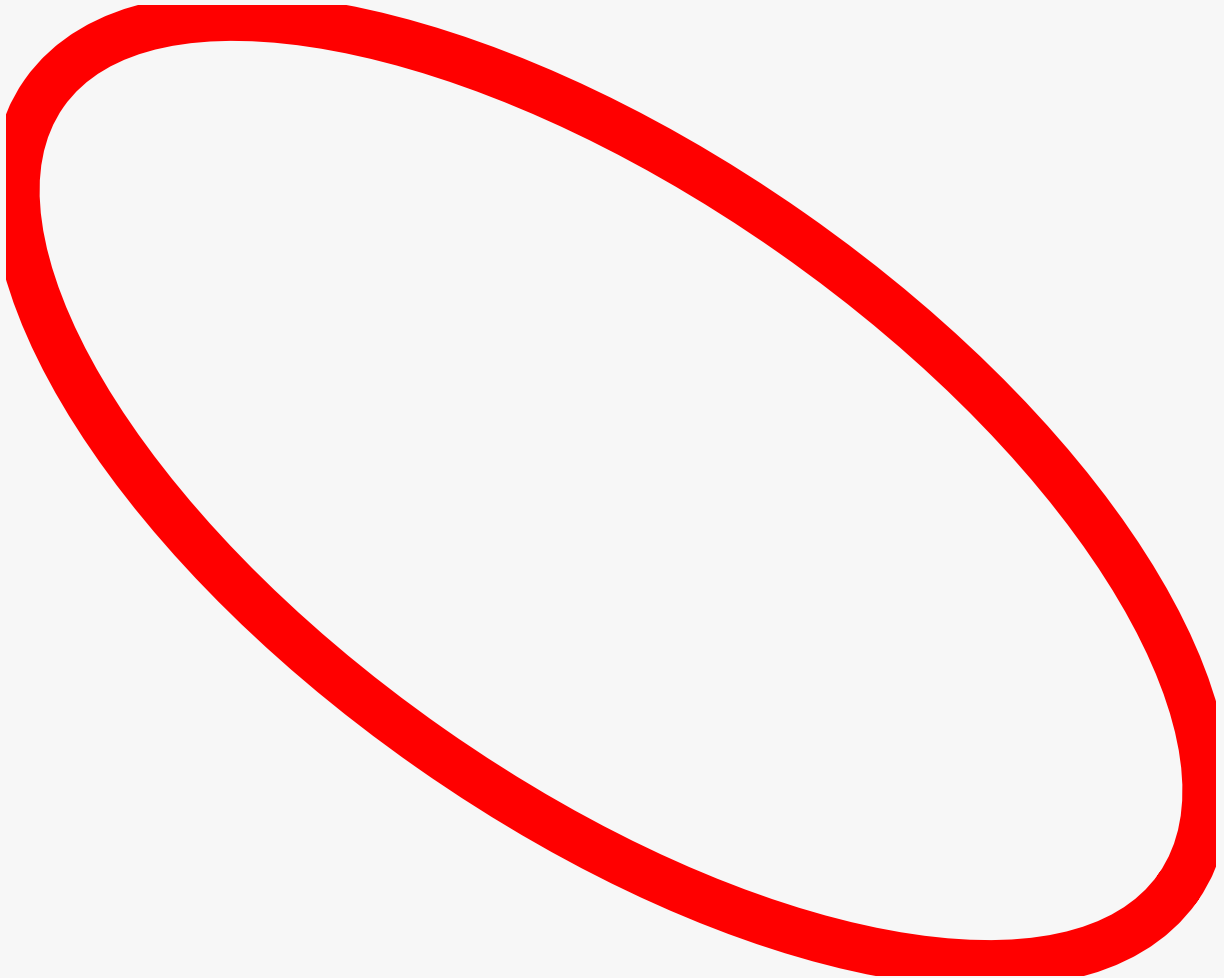
An *Ellipse* corresponds to an `svg:ellipse` with these attributes:

- `transform` describes both the *Center* (c_x, c_y) and the *Rotation* r of the Ellipse.
`transform = "translate( + str( $c_x$ ) + , + str( $c_y$ ) + ) rotate( + str( $r$ ) + )"`
- `rx` and `ry` are string representations of *HorizontalSemiAxis* and *VerticalSemiAxis*.
- `stroke`, `stroke-dasharray`, `stroke-dashoffset` and `stroke-width` are set according to the SVG mapping rules for the *Stroke* of the *Ellipse* (see *Stroke*).
- `stroke-linecap = "round"` and `stroke-linejoin = "round"` reflect the heuristic for line caps and line joins.
- `vector-effect = "non-scaling-stroke"` reflects the heuristic for scaled symbols.
- `fill` is set according to the SVG mapping rules for the *FillStyle* of the *Ellipse* (see *FillStyle*).

example

We assume that the *Stroke* of the *Ellipse* above is 2mm #ff0000 Solid and that its *FillStyle* is *Transparent*.

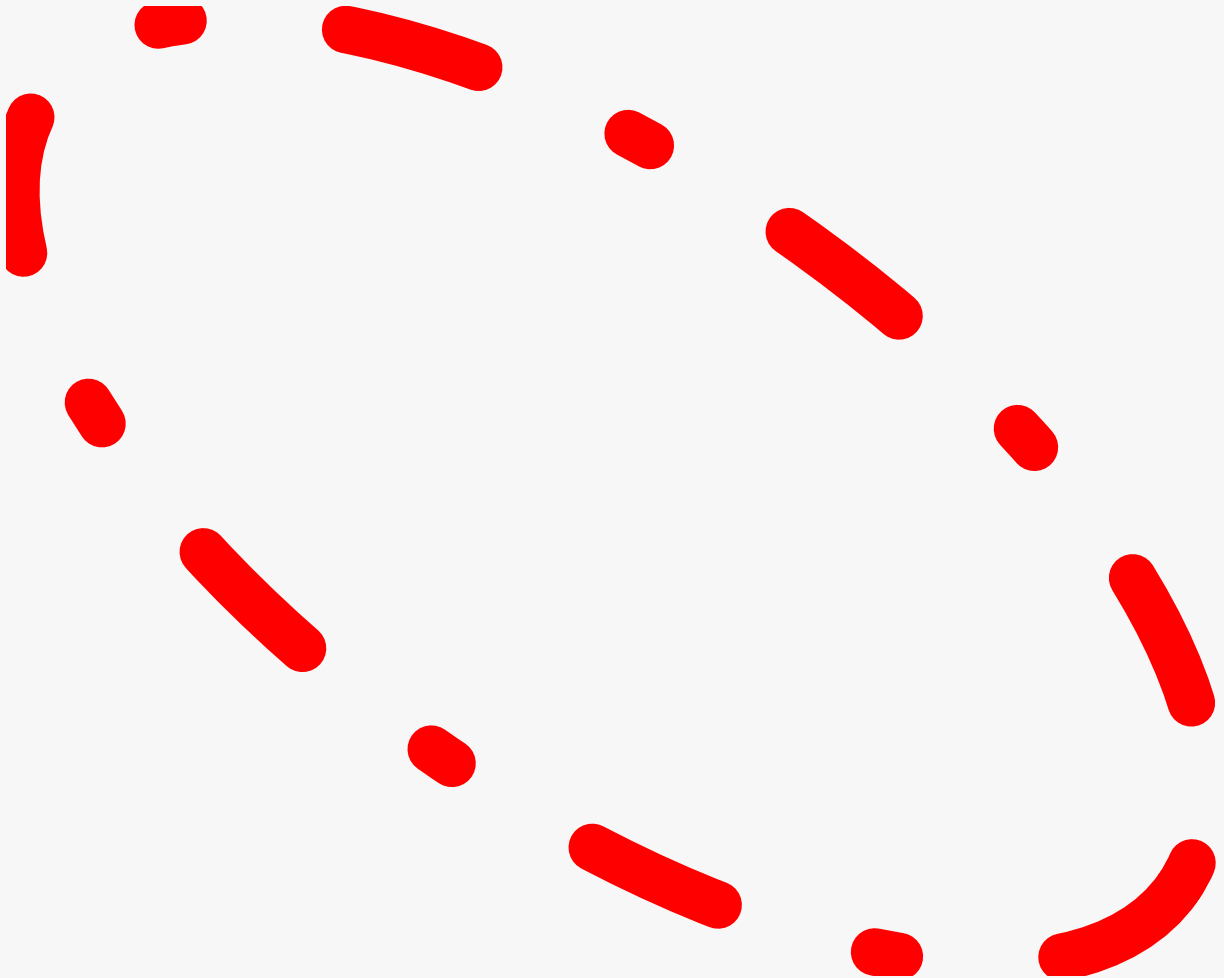
```
{EllipseTransparentSvgCode}
```



example

We consider the same *Ellipse* as in the previous example, except that the *Stroke* is 2mm #ff0000 LongDashShortDash.

```
{EllipseLongDashShortDashSvgCode}
```

**example**

Finally, a variant with *Stroke* 2mm #ff0000 Solid and *FillStyle* Hatch.

```
{EllipseHatchSvgCode}
```

**Center****Data Property**

The center position of the *Ellipse*.

Type: *Point*

Multiplicity: 1

Ordered: no

Unique: no

Example

```
Point(  
  X = 0.0,  
  Y = 0.0)
```

FillStyle

Data Property

The fill style of the *Ellipse*.

Type: *FillStyle*

Multiplicity: 1

Ordered: no

Unique: no

Example

Transparent

HorizontalSemiAxis

Data Property

The length of the horizontal semi-axis of the *Ellipse* in mm.

Type: *Double*

Multiplicity: 1

Ordered: no

Unique: no

Example

7.5

Rotation

Data Property

The rotation of the *Ellipse* around its center, measured clockwise and in degrees. The value must be in the interval [0; 360).

Type: *Double*

Multiplicity: 1

Ordered: no

Unique: no

Example

270.0

Stroke

Data Property

The stroke of the *Ellipse*.

Type: *Stroke*

Multiplicity: 1**Ordered:** no**Unique:** no**Example**

```
Stroke(
  Color = Color(
    B = 0,
    G = 0,
    R = 128),
  DashStyle = Solid,
  Width = 0.3)
```

VerticalSemiAxis**Data Property**

The length of the vertical semi-axis of the *Ellipse* in mm.

Type: *Double***Multiplicity:** 1**Ordered:** no**Unique:** no**Example**

7.5

Example in DEXPI XML

```
<Object type="Core/Diagram.Ellipse">
  <Data property="Center">
    <AggregatedDataValue type="Core/Diagram.Point">
      <Data property="X">
        <Double>0.0</Double>
      </Data>
      <Data property="Y">
        <Double>0.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="FillStyle">
    <DataReference data="Core/Diagram.FillStyle.Transparent"/>
  </Data>
  <Data property="HorizontalSemiAxis">
    <Double>7.5</Double>
  </Data>
```

(continues on next page)

(continued from previous page)

```

<Data property="Rotation">
  <Double>270.0</Double>
</Data>
<Data property="Stroke">
  <AggregatedDataValue type="Core/Diagram.Stroke">
    <Data property="Color">
      <AggregatedDataValue type="Core/Diagram.Color">
        <Data property="B">
          <Integer>0</Integer>
        </Data>
        <Data property="G">
          <Integer>0</Integer>
        </Data>
        <Data property="R">
          <Integer>128</Integer>
        </Data>
      </AggregatedDataValue>
    </Data>
    <Data property="DashStyle">
      <DataReference data="Core/Diagram.DashStyle.Solid"/>
    </Data>
    <Data property="Width">
      <Double>0.3</Double>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="VerticalSemiAxis">
  <Double>7.5</Double>
</Data>
</Object>

```

EllipseArc

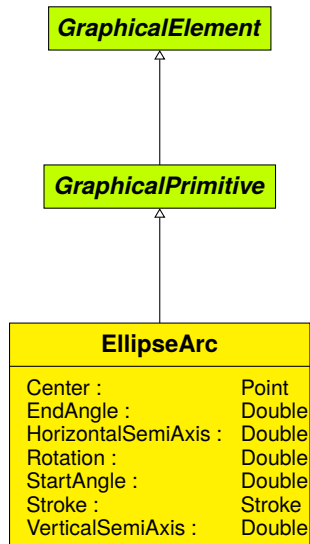
Core::Diagram::EllipseArc

Overview

Concrete Class

‘

A segment of an ellipse.



Supertypes

- *GraphicalPrimitive*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Center</i>	1	<i>Point</i>	no	no
<i>EndAngle</i>	1	<i>Double</i>	no	no
<i>HorizontalSemiAxis</i>	1	<i>Double</i>	no	no
<i>Rotation</i>	1	<i>Double</i>	no	no
<i>StartAngle</i>	1	<i>Double</i>	no	no
<i>Stroke</i>	1	<i>Stroke</i>	no	no
<i>VerticalSemiAxis</i>	1	<i>Double</i>	no	no

Geometry

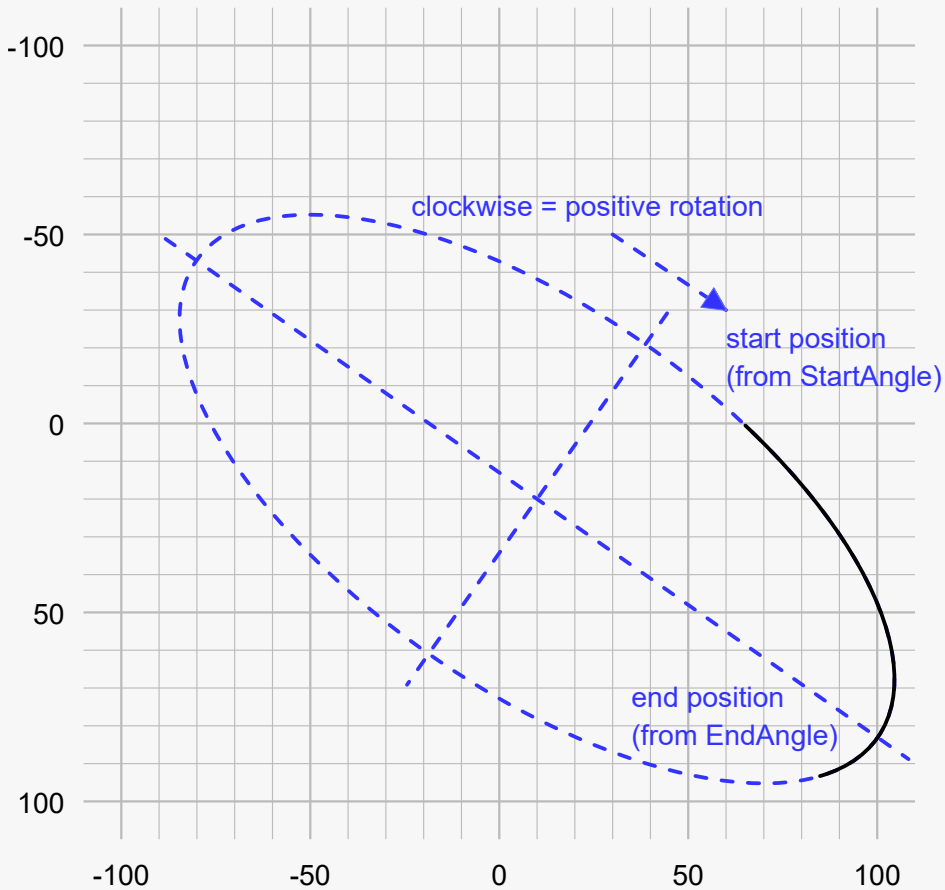
An *EllipseArc* is a segment of an ellipse. The geometry of the underlying ellipse is described in the same way as the geometry of the actual *Ellipse* class (*Center*, *HorizontalSemiAxis*, *VerticalSemiAxis*, *Rotation*). In addition, an *EllipseArc* has two attributes *StartAngle* and *EndAngle* that determine the start and end position of the arc.

example

We consider an *EllipseArc* with

- *Center* (10, 20),
- *HorizontalSemiAxis* 110,
- *VerticalSemiAxis* 50, and
- *Rotation* 35.

These attributes describe the blue dashed ellipse in this figure (cf. the *Ellipse* example which has the same attribute values).



The *EllipseArc* has two further geometric attributes

- *StartAngle* 288 and
- *EndAngle* 20.

These two angles do not have a simple geometric interpretation in the figure; we will get back to them in the following sections. However, these angles determine the start position and the end position of the black *EllipseArc* in the figure. The actual *EllipseArc* goes from the start position *in positive direction* (i.e., clockwise) to the end position.

Calculation of start and end position

We use the following notation:

- c_x and c_y : components of the *Center*,
- r_x : *HorizontalSemiAxis*,
- r_y : *VerticalSemiAxis*,
- φ : *Rotation*,
- θ_1 : *StartAngle*,
- θ_2 : *EndAngle*,

- x_1 and y_1 : start position,
- x_2 and y_2 : end position.

technical

The notation is based on that used by the [SVG Arc Implementation Notes](#) which may be useful for implementors.

The start position is

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix} \cdot \begin{pmatrix} r_x \cos \theta_1 \\ r_y \sin \theta_1 \end{pmatrix} + \begin{pmatrix} c_x \\ c_y \end{pmatrix}$$

and the end position is

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix} \cdot \begin{pmatrix} r_x \cos \theta_2 \\ r_y \sin \theta_2 \end{pmatrix} + \begin{pmatrix} c_x \\ c_y \end{pmatrix} .$$

example

In the example, we have

$$\begin{aligned} c_x &= 10 \\ c_y &= 20 \\ r_x &= 110 \\ r_y &= 50 \\ \varphi &= 35 \\ \theta_1 &= 288 \\ \theta_2 &= 20 \end{aligned}$$

With the equations above, we get these coordinates for the start and end positions (cf. the figure above):

$$\begin{aligned} x_1 &= 65.1197 \\ y_1 &= 0.5439 \\ x_2 &= 84.8639 \\ y_2 &= 93.2967 \end{aligned}$$

Interpretation of StartAngle and EndAngle

These angles are not measured in the ellipse on which an *EllipseArc* is based, but in the unit circle:

???

The figure shows the end position w.r.t. the unit circle. Its angle measured from the horizontal semi-axis is $\theta_2 = 20$. The ellipse could be constructed by stretching the unit circle by r_x in direction of the horizontal semi-axis and by r_y in direction of the vertical semi-axis. After this transformation, the end position w.r.t. the unit circle is at the actual end position.

To calculate an actual, ellipse-based angle from an angle w.r.t. the unit circle, use the formula

$$\theta_{actual} = \text{atan2}(r_y \sin \theta, r_x \cos \theta) \bmod 360 .$$

The formulas apply to both the *StartAngle* θ_1 and the *EndAngle* θ_2 . Note that we give angles in degrees, i.e., the actual calculations in a program may require conversions to and from radians.

example

In the example, we have

$$\theta_{actual,1} = \text{atan2}(50 \cdot \sin 288, 110 \cdot \cos 288) \bmod 360 = 305.56$$

and

$$\theta_{actual,2} = \text{atan2}(50 \cdot \sin 20, 110 \cdot \cos 20) \bmod 360 = 9.39 \quad .$$

The inverse formula is

$$\theta = \text{atan2}(r_x \sin \theta_{actual}, r_y \cos \theta_{actual}) \bmod 360 \quad .$$

technical

The rationale for using angles w.r.t. the unit circle is that they are numerically more stable when an *EllipseArc* is scaled with small values (i.e., in a *ShapeUsage*) and that some calculations are simpler (cf. [SVG 2 Arc Implementation Notes](#)).

Mapping to SVG

An *EllipseArc* is mapped to an `svg:path` with these attributes:

- `d` is the `path data` that describes the entire geometry of the *EllipseArc*. The value is the concatenation of the following strings, all of them separated by spaces:
 - `M`
 - `str( $x_1$ )`
 - `str( $y_1$ )`
 - `A`
 - `str( $r_x$ )`
 - `str( $r_y$ )`
 - `str( $\varphi$ )`
 - `str( $f_A$ )`
 - `1`
 - `str( $x_2$ )`
 - `str( $y_2$ )`

f_A is calculated as follows (cf. the definition of the *modulo operator*):

$$\begin{aligned} \Delta\theta &= (\theta_2 - \theta_1) \bmod 360 \quad , \\ f_A &= 0 \quad \text{if } \Delta\theta < 180 \quad , \\ f_A &= 1 \quad \text{if } \Delta\theta \geq 180 \quad . \end{aligned}$$

The other variables have been defined above.

- `stroke`, `stroke-dasharray`, `stroke-dashoffset` and `stroke-width` are set according to the SVG mapping rules for the *Stroke* of the *EllipseArc* (see [Stroke](#)).

- `stroke-linecap = "round"` and `stroke-linejoin = "round"` reflect the heuristic for line caps and line joins.
- `vector-effect = "non-scaling-stroke"` reflects the heuristic for scaled symbols.
- `fill = "none"` is required to avoid that the `svg:path` is filled.

example

We assume that the *Stroke* of the *EllipseArc* above is 2mm #ff0000 solid.

With $\Delta\theta = (20 - 288) \bmod 360 = (-268) \bmod 360 = 92 < 180$ we get $f_A = 0$.

```
{EllipseArcSolidSvgCode}
```

???

example

We consider the same *EllipseArc* as in the previous example, except that the *Stroke* is 2mm #ff0000 Dash.

```
{EllipseArcDashSvgCode}
```



Center

Data Property

The center position of the *EllipseArc*.

Type: *Point*

Multiplicity: 1

Ordered: no

Unique: no

Example

Point(

$X = 2.0,$

$Y = 5.0)$

EndAngle

Data Property

The end angle of the *EllipseArc*, measured clockwise and in degrees. The value must be in the interval [0; 360).

Type: *Double*

Multiplicity: 1

Ordered: no

Unique: no

Example

270.0

HorizontalSemiAxis

Data Property

The length of the horizontal semi-axis of the *EllipseArc* in mm.

Type: *Double*

Multiplicity: 1

Ordered: no

Unique: no

Example

7.5

Rotation

Data Property

The rotation of the *EllipseArc* around its center, measured clockwise and in degrees. The value must be in the interval [0; 360).

Type: *Double*

Multiplicity: 1

Ordered: no

Unique: no

Example

270.0

StartAngle**Data Property**

The start angle of the *EllipseArc*, measured clockwise and in degrees. The value must be in the interval [0; 360).

Type: *Double***Multiplicity:** 1**Ordered:** no**Unique:** no**Example**

90.0

Stroke**Data Property**

The stroke of the *EllipseArc*.

Type: *Stroke***Multiplicity:** 1**Ordered:** no**Unique:** no**Example**

```
Stroke(
    Color = Color(
        B = 0,
        G = 0,
        R = 128),
    DashStyle = Solid,
    Width = 0.3)
```

VerticalSemiAxis**Data Property**

The length of the vertical semi-axis of the *EllipseArc* in mm.

Type: *Double***Multiplicity:** 1**Ordered:** no**Unique:** no

Example

3.5

Example in DEXPI XML

```
<Object type="Core/Diagram.EllipseArc">
  <Data property="Center">
    <AggregatedDataValue type="Core/Diagram.Point">
      <Data property="X">
        <Double>2.0</Double>
      </Data>
      <Data property="Y">
        <Double>5.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="EndAngle">
    <Double>270.0</Double>
  </Data>
  <Data property="HorizontalSemiAxis">
    <Double>7.5</Double>
  </Data>
  <Data property="Rotation">
    <Double>270.0</Double>
  </Data>
  <Data property="StartAngle">
    <Double>90.0</Double>
  </Data>
  <Data property="Stroke">
    <AggregatedDataValue type="Core/Diagram.Stroke">
      <Data property="Color">
        <AggregatedDataValue type="Core/Diagram.Color">
          <Data property="B">
            <Integer>0</Integer>
          </Data>
          <Data property="G">
            <Integer>0</Integer>
          </Data>
          <Data property="R">
            <Integer>128</Integer>
          </Data>
        </AggregatedDataValue>
      </Data>
      <Data property="DashStyle">
        <DataReference data="Core/Diagram.DashStyle.Solid"/>
      </Data>
      <Data property="Width">
        <Double>0.3</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="VerticalSemiAxis">
```

(continues on next page)

(continued from previous page)

```
<Double>3.5</Double>
</Data>
</Object>
```

FillStyle

Core::Diagram::FillStyle

Overview

Enumeration

<<enumeration>> FillStyle
Hatch Solid Transparent

Members

Literals

Name
Hatch
Solid
Transparent

GraphicalElement

Core::Diagram::GraphicalElement

Overview

Abstract Class

GraphicalElement

Subtypes

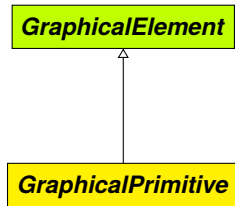
- GraphicalPrimitive
- ShapeUsage

GraphicalPrimitive

Core::Diagram::GraphicalPrimitive

Overview

Abstract Class



Supertypes

- *GraphicalElement*

Subtypes

- *ConnectorLine*
- *Ellipse*
- *EllipseArc*
- *PolyLine*
- *Polygon*
- *Text*

GraphicsGroup

Core::Diagram::GraphicsGroup

Overview

Abstract Class



Subtypes

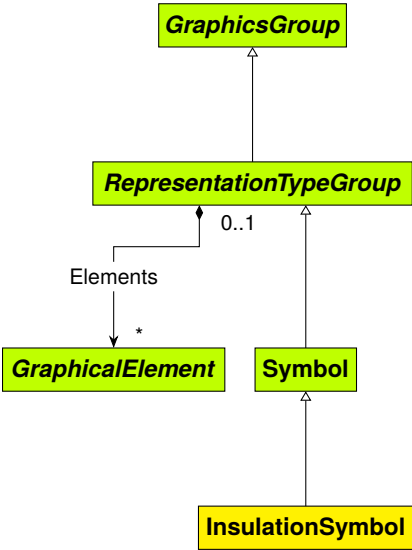
- *RepresentationGroup*
- *RepresentationTypeGroup*

InsulationSymbol

Core::Diagram::InsulationSymbol

Overview

Concrete Class



Supertypes

- *Symbol*

Example in DEXPI XML

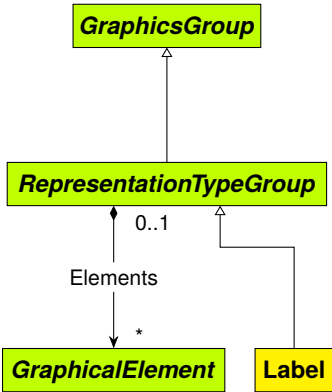
```
<Object type="Core/Diagram.InsulationSymbol"/>
```

Label

Core::Diagram::Label

Overview

Concrete Class



Supertypes

- *RepresentationTypeGroup*

Subtypes

- *ActuatingElectricalSystemNumberLabel*
- *ActuatingSystemNumberLabel*
- *CustomLabel*
- *DeviceInformationLabel*
- *EquipmentBarLabel*
- *EquipmentTagNameLabel*
- *FailActionLabel*
- *FittingLabel*
- *InsulationBreakLabel*
- *InsulationLabel*
- *MPRelevanceLabel*
- *MeasuringSystemNumberLabel*
- *NoteIdentifierLabel*
- *NoteTextLabel*
- *NozzleStandardLabel*
- *OffPageConnectorDescriptionLabel*
- *OffPageConnectorNumberLabel*
- *PipingClassBreakLabel*
- *PipingNetworkSegmentLabel*
- *PipingNetworkSystemLabel*
- *ProcessInstrumentationFunctionLabel*
- *QualityRelevanceLabel*
- *ReducerLabel*
- *ReferencedPIDNumberLabel*
- *SafetyRelevanceLabel*
- *SafetyValveOrFittingLabel*
- *SignalConveyingFunctionLabel*
- *SignalHighHighHighLabel*
- *SignalHighHighLabel*
- *SignalHighLabel*
- *SignalLowLabel*
- *SignalLowLowLabel*
- *SignalLowLowLowLabel*

- *TypicalInformationLabel*
- *ValveLabel*
- *VendorNameLabel*

Example in DEXPI XML

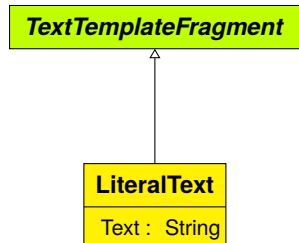
```
<Object type="Core/Diagram.Label"/>
```

LiteralText

Core::Diagram::LiteralText

Overview

Concrete Class



Supertypes

- *TextTemplateFragment*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Text</i>	1	<i>String</i>	no	no

Text

Data Property

Type: *String*

Multiplicity: 1

Ordered: no

Unique: no

Example

"Drawing No.: "

Example in DEXPI XML

```
<Object type="Core/Diagram.LiteralText">
  <Data property="Text">
    <String>Drawing No.: </String>
  </Data>
</Object>
```

MetaData

*Core::Diagram::*MetaData

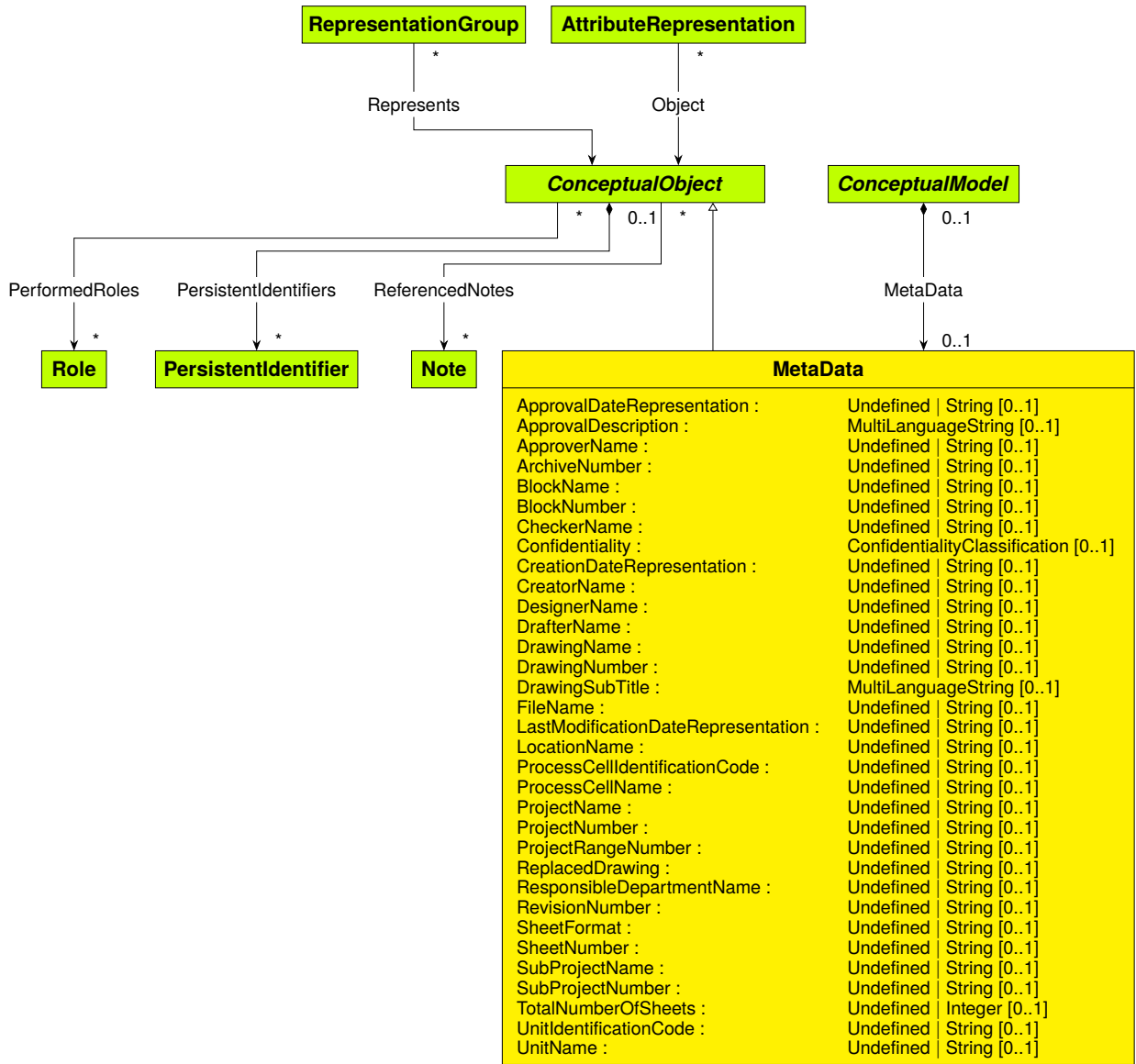
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/MetaData>

META DATA

Meta data about a Core.Container.



Supertypes

- *ConceptualObject*

Subtypes

- *PlantMetadata*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
ApprovalDateRepresentation	0..1	Undefined String	no	no
ApprovalDescription	0..1	MultiLanguageString	no	no

continues on next page

Table 2 – continued from previous page

Name	Multiplicity	Type	Ordered	Unique
<i>ApproverName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>ArchiveNumber</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>BlockName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>BlockNumber</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>CheckerName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>Confidentiality</i>	0..1	<i>ConfidentialityClassification</i>	no	no
<i>CreationDateRepresentation</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>CreatorName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>DesignerName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>DrafterName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>DrawingName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>DrawingNumber</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>DrawingSubTitle</i>	0..1	<i>MultiLanguageString</i>	no	no
<i>FileName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>LastModificationDateRepresentation</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>LocationName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>ProcessCellIdentificationCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>ProcessCellName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>ProjectName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>ProjectNumber</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>ProjectRangeNumber</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>ReplacedDrawing</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>ResponsibleDepartmentName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>RevisionNumber</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>SheetFormat</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>SheetNumber</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>SubProjectName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>SubProjectNumber</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>TotalNumberOfSheets</i>	0..1	<i>Undefined</i> <i>Integer</i>	no	no
<i>UnitIdentificationCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>UnitName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

ApprovalDateRepresentation

Data Property

A representation of the approval date of the drawing. The format of the representation is not prescribed.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ApprovalDateRepresentationAssignmentClass>

APPROVAL DATE REPRESENTATION ASSIGNMENT CLASS

Example

"2016-04-01"

ApprovalDescription**Data Property**

A description of the approval of the drawing.

Type: *MultiLanguageString*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ApprovalDescriptionAssignmentClass>

APPROVAL DESCRIPTION ASSIGNMENT CLASS

Example

```
MultiLanguageString(  
  SingleLanguageStrings = [  
    SingleLanguageString(  
      Language = "en",  
      Value = "approved"),  
    SingleLanguageString(  
      Language = "de",  
      Value = "genehmigt")])
```

ApproverName**Data Property**

The name of the approver of the drawing.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ApproverNameAssignmentClass>

APPROVER NAME ASSIGNMENT CLASS

Example

"A. P. Prover"

ArchiveNumber

Data Property

The archive number of the drawing.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ArchiveNumberAssignmentClass>

ARCHIVE NUMBER ASSIGNMENT CLASS

Example

"XY923-463"

BlockName

Data Property

The name of the related block.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/BlockNameAssignmentClass>

BLOCK NAME ASSIGNMENT CLASS

Example

"a block"

BlockNumber

Data Property

The number of the related block.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/BlockNumberAssignmentClass>

BLOCK NUMBER ASSIGNMENT CLASS

Example

"B987-654"

CheckerName**Data Property**

The name of the checker of the drawing.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/CheckerNameAssignmentClass>

CHECKER NAME ASSIGNMENT CLASS

Example

"C. Hecker"

Confidentiality**Data Property**

The confidentiality classification of the diagram.

Type: *ConfidentialityClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ConfidentialitySpecialization>

CONFIDENTIALITY SPECIALIZATION

Example

ConfidentialInformation

CreationDateRepresentation**Data Property**

A representation of the creation date of the drawing. The format of the representation is not prescribed.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/CreationDateRepresentationAssignmentClass>

CREATION DATE REPRESENTATION ASSIGNMENT CLASS

Example

"2016-04-01"

CreatorName

Data Property

The name of the creator of the drawing.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/CreatorNameAssignmentClass>

CREATOR NAME ASSIGNMENT CLASS

Example

"A. Creator"

DesignerName

Data Property

The name of the designer of the drawing.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignerNameAssignmentClass>

DESIGNER NAME ASSIGNMENT CLASS

Example

"D. E. Signer"

DrafterName

Data Property

The name of the drafter of the drawing.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DrafterNameAssignmentClass>

DRAFTER NAME ASSIGNMENT CLASS

Example

"D. Rafter"

DrawingName**Data Property**

The drawing name.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS2102503531>

DRAWING NAME ASSIGNMENT CLASS

Example

"DEXPI example PID"

DrawingNumber**Data Property**

The drawing number.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DrawingNumberAssignmentClass>

DRAWING NUMBER ASSIGNMENT CLASS

Example

"123/A93"

DrawingSubTitle**Data Property**

The subtitle of the drawing.

Type: *MultiLanguageString*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DrawingSubTitleAssignmentClass>

DRAWING SUB TITLE ASSIGNMENT CLASS

Example

```
MultiLanguageString(  
  SingleLanguageStrings = [  
    SingleLanguageString(  
      Language = "en",  
      Value = "DEXPI Example PID"),  
    SingleLanguageString(  
      Language = "de",  
      Value = "DEXPI Beispiel-R&I"))]
```

FileName

Data Property

The file name of the drawing.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/FileNameAssignmentClass>

FILE NAME ASSIGNMENT CLASS

Example

```
"DEXPI_example_PID.xml"
```

LastModificationDateRepresentation

Data Property

A representation of the last modification date of the drawing. The format of the representation is not prescribed.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LastModificationDateRepresentationAssignmentClass>

LAST MODIFICATION DATE REPRESENTATION ASSIGNMENT CLASS

Example

```
"2016-04-02"
```


LocationName**Data Property**

The location name.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LocationNameAssignmentClass>

LOCATION NAME ASSIGNMENT CLASS

Example

"C1248"

ProcessCellIdentificationCode**Data Property**

The identification code of the related process cell according to ISA-95.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ProcessCellIdentificationCodeAssignmentClass>

PROCESS CELL IDENTIFICATION CODE ASSIGNMENT CLASS

Example

"PC123"

ProcessCellName**Data Property**

The name of the related process cell according to ISA-95.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ProcessCellNameAssignmentClass>

PROCESS CELL NAME ASSIGNMENT CLASS

Example

"a process cell"

ProjectName

Data Property

The name of the related project.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ProjectNameAssignmentClass>

PROJECT NAME ASSIGNMENT CLASS

Example

"a project"

ProjectNumber

Data Property

The number of the related project.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ProjectNumberAssignmentClass>

PROJECT NUMBER ASSIGNMENT CLASS

Example

"P3.1415"

ProjectRangeNumber

Data Property

The range number of the related project.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ProjectRangeNumberAssignmentClass>

PROJECT RANGE NUMBER ASSIGNMENT CLASS

Example

"PR321"

ReplacedDrawing**Data Property**

The drawing replaced by this drawing.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ReplacedDrawingAssignmentClass>

REPLACED DRAWING ASSIGNMENT CLASS

Example

"D321"

ResponsibleDepartmentName**Data Property**

The name of the department responsible for the drawing.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ResponsibleDepartmentNameAssignmentClass>

RESPONSIBLE DEPARTMENT NAME ASSIGNMENT CLASS

Example

"R2-D2"

RevisionNumber**Data Property**

The revision number of the drawing.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/RevisionNumberAssignmentClass>

REVISION NUMBER ASSIGNMENT CLASS

Example

"R2.2"

SheetFormat

Data Property

The sheet format.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SheetFormatAssignmentClass>

SHEET FORMAT ASSIGNMENT CLASS

Example

"DIN A3"

SheetNumber

Data Property

The sheet number of the drawing.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SheetNumberAssignmentClass>

SHEET NUMBER ASSIGNMENT CLASS

Example

"2a"

SubProjectName

Data Property

The name of the related sub-project.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SubProjectNameAssignmentClass>

SUB PROJECT NAME ASSIGNMENT CLASS

Example

"a sub-project"

SubProjectNumber**Data Property**

The number of the related sub-project.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SubProjectNumberAssignmentClass>

SUB PROJECT NUMBER ASSIGNMENT CLASS

Example

"P3.1415-SP2"

TotalNumberOfSheets**Data Property**

The total number of sheets.

Type: *Undefined* | *Integer*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/TotalNumberOfSheets>

TOTAL NUMBER OF SHEETS

Example

4

UnitIdentificationCode**Data Property**

The identification code of the related unit according to ISA-95.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UnitIdentificationCodeAssignmentClass>

UNIT IDENTIFICATION CODE ASSIGNMENT CLASS

Example

"U-923-463"

UnitName**Data Property**

The name of the related unit according to ISA-95.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UnitIsa95NameAssignmentClass>

UNIT ISA95 NAME ASSIGNMENT CLASS

Example

"a unit"

Example in DEXPI XML

```
<Object type="Core/Diagram.Metadata">
  <Data property="ApprovalDateRepresentation">
    <String>2016-04-01</String>
  </Data>
  <Data property="ApprovalDescription">
    <AggregatedDataValue type="Core/DataTypes.MultiLanguageString">
      <Data property="SingleLanguageStrings">
        <AggregatedDataValue type="Core/DataTypes.SingleLanguageString">
          <Data property="Language">
            <String>en</String>
          </Data>
          <Data property="Value">
            <String>approved</String>
          </Data>
        </AggregatedDataValue>
        <AggregatedDataValue type="Core/DataTypes.SingleLanguageString">
          <Data property="Language">
            <String>de</String>
          </Data>
          <Data property="Value">
            <String>genehmigt</String>
          </Data>
        </AggregatedDataValue>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="ApproverName">
    <String>A. P. Prover</String>
  </Data>
```

(continues on next page)

(continued from previous page)

```

<Data property="ArchiveNumber">
  <String>XY923-463</String>
</Data>
<Data property="BlockName">
  <String>a block</String>
</Data>
<Data property="BlockNumber">
  <String>B987-654</String>
</Data>
<Data property="CheckerName">
  <String>C. Hecker</String>
</Data>
<Data property="Confidentiality">
  <DataReference data="Core/Diagram.ConfidentialityClassification.
↪ ConfidentialInformation"/>
</Data>
<Data property="CreationDateRepresentation">
  <String>2016-04-01</String>
</Data>
<Data property="CreatorName">
  <String>A. Creator</String>
</Data>
<Data property="DesignerName">
  <String>D. E. Signer</String>
</Data>
<Data property="DrafterName">
  <String>D. Rafter</String>
</Data>
<Data property="DrawingName">
  <String>DEXPI example PID</String>
</Data>
<Data property="DrawingNumber">
  <String>123/A93</String>
</Data>
<Data property="DrawingSubTitle">
  <AggregatedDataValue type="Core/DataTypes.MultiLanguageString">
    <Data property="SingleLanguageStrings">
      <AggregatedDataValue type="Core/DataTypes.SingleLanguageString">
        <Data property="Language">
          <String>en</String>
        </Data>
        <Data property="Value">
          <String>DEXPI Example PID</String>
        </Data>
      </AggregatedDataValue>
      <AggregatedDataValue type="Core/DataTypes.SingleLanguageString">
        <Data property="Language">
          <String>de</String>
        </Data>
        <Data property="Value">
          <String>DEXPI Beispiel-R&I</String>
        </Data>
      </AggregatedDataValue>
    </Data>
  </AggregatedDataValue>

```

(continues on next page)

(continued from previous page)

```

        </AggregatedDataValue>
    </Data>
</AggregatedDataValue>
</Data>
<Data property="FileName">
    <String>DEXPI_example_PID.xml</String>
</Data>
<Data property="LastModificationDateRepresentation">
    <String>2016-04-02</String>
</Data>
<Data property="LocationName">
    <String>C1248</String>
</Data>
<Data property="ProcessCellIdentificationCode">
    <String>PC123</String>
</Data>
<Data property="ProcessCellName">
    <String>a process cell</String>
</Data>
<Data property="ProjectName">
    <String>a project</String>
</Data>
<Data property="ProjectNumber">
    <String>P3.1415</String>
</Data>
<Data property="ProjectRangeNumber">
    <String>PR321</String>
</Data>
<Data property="ReplacedDrawing">
    <String>D321</String>
</Data>
<Data property="ResponsibleDepartmentName">
    <String>R2-D2</String>
</Data>
<Data property="RevisionNumber">
    <String>R2.2</String>
</Data>
<Data property="SheetFormat">
    <String>DIN A3</String>
</Data>
<Data property="SheetNumber">
    <String>2a</String>
</Data>
<Data property="SubProjectName">
    <String>a sub-project</String>
</Data>
<Data property="SubProjectNumber">
    <String>P3.1415-SP2</String>
</Data>
<Data property="TotalNumberOfSheets">
    <Integer>4</Integer>
</Data>

```

(continues on next page)

(continued from previous page)

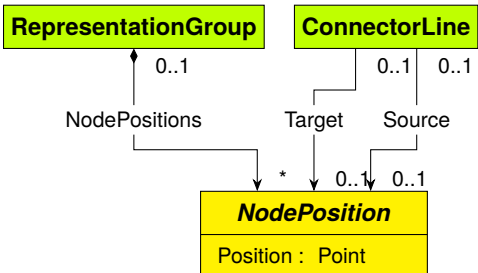
```
<Data property="UnitIdentificationCode">
  <String>U-923-463</String>
</Data>
<Data property="UnitName">
  <String>a unit</String>
</Data>
</Object>
```

NodePosition

Core::Diagram::NodePosition

Overview

Abstract Class



Subtypes

- InstrumentationNodePosition
- PipingNodePosition

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
Position	1	Point	no	no

Position

Data Property

Type: Point

Multiplicity: 1

Ordered: no

Unique: no

Example

Point(

$X = 3.5,$

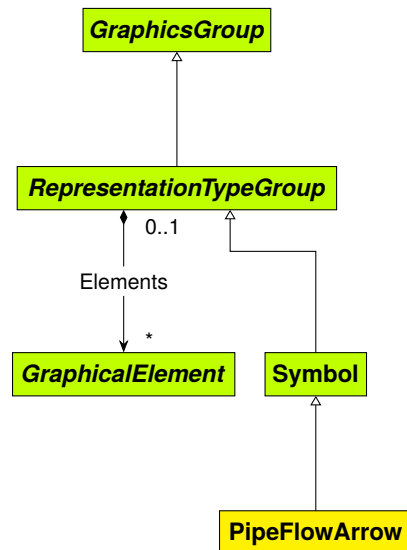
$Y = 5.5)$

PipeFlowArrow

Core::Diagram::PipeFlowArrow

Overview

Concrete Class



Supertypes

- *Symbol*

Example in DEXPI XML

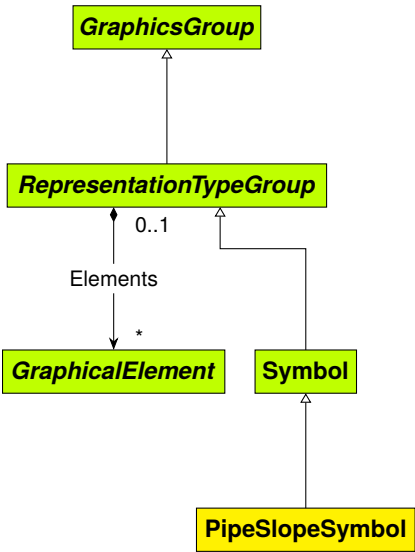
```
<Object type="Core/Diagram.PipeFlowArrow"/>
```

PipeSlopeSymbol

Core::Diagram::PipeSlopeSymbol

Overview

Concrete Class



Supertypes

- *Symbol*

Example in DEXPI XML

```
<Object type="Core/Diagram.PipeSlopeSymbol"/>
```

Point

Core::Diagram::Point

Overview

Aggregated Data Type

A point in the X-Y-plane.

<<datatype>>
Point

X : Double
Y : Double

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>X</i>	1	<i>Double</i>	no	no
<i>Y</i>	1	<i>Double</i>	no	no

X

Data Property

The X (horizontal) coordinate of the *Point* in mm.

Type: *Double*

Multiplicity: 1

Ordered: no

Unique: no

Example

50.5

Y

Data Property

The Y (horizontal) coordinate of the *Point* in mm.

Type: *Double*

Multiplicity: 1

Ordered: no

Unique: no

Example

-1.2

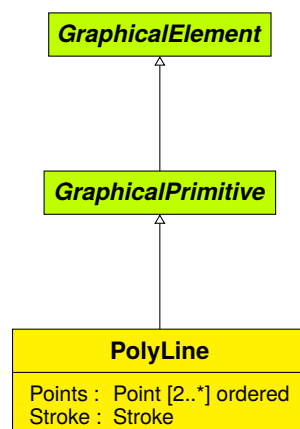
PolyLine

*Core::Diagram::*PolyLine

Overview

Concrete Class

A curve that consists of one or more chained straight sections.



Supertypes

- *GraphicalPrimitive*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Points</i>	2..*	<i>Point</i>	yes	no
<i>Stroke</i>	1	<i>Stroke</i>	no	no

Geometry

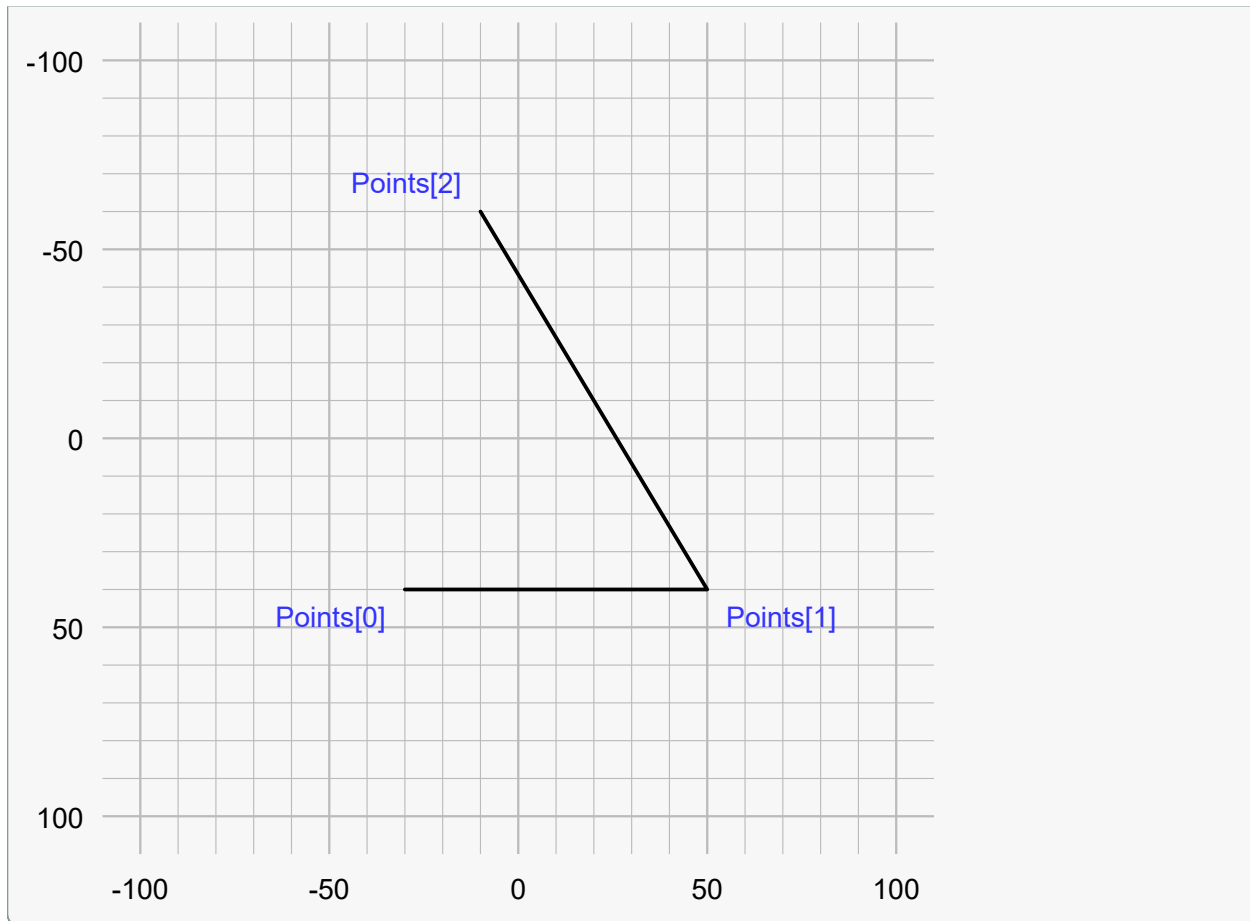
The first straight section of a *PolyLine* starts at its first *Point*, i.e., *Points[0]*, and ends at its second *Point*. For each further *Point*, another section is drawn that starts at the preceding *Point*.

example

We consider a *PolyLine* with 3 *Points*:

- (-30, 40),
- (50, 40), and
- (-10, -60).

The *PolyLine* will look like this:



Mapping to SVG

A *PolyLine* corresponds to an `svg:polyline` with these attributes:

- `points` contains the *Points* of the *PolyLine*.

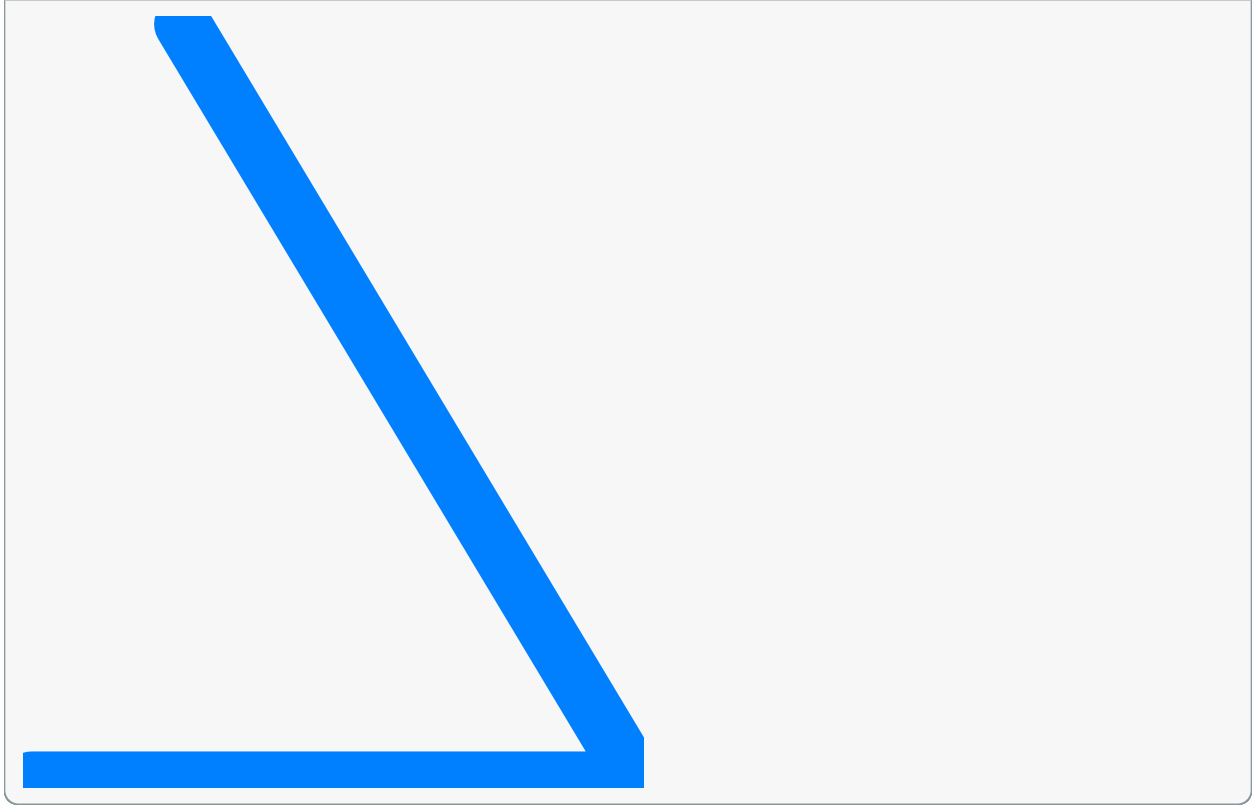
Each *Point* i with position x_i and y_i is converted to a string as `str( $x_i$ ) + , + str( $y_i$ )`. Finally, the `points` attribute is set to a single string that contains all positions, separated by spaces.

- `stroke`, `stroke-dasharray`, `stroke-dashoffset` and `stroke-width` are set according to the SVG mapping rules for the *Stroke* of the *PolyLine* (see *Stroke*).
- `stroke-linecap` = "round" and `stroke-linejoin` = "round" reflect the heuristic for line caps and line joins.
- `vector-effect` = "non-scaling-stroke" reflects the heuristic for scaled symbols.
- `fill` = "none" is required to avoid that the `svg:polyline` is filled.

example

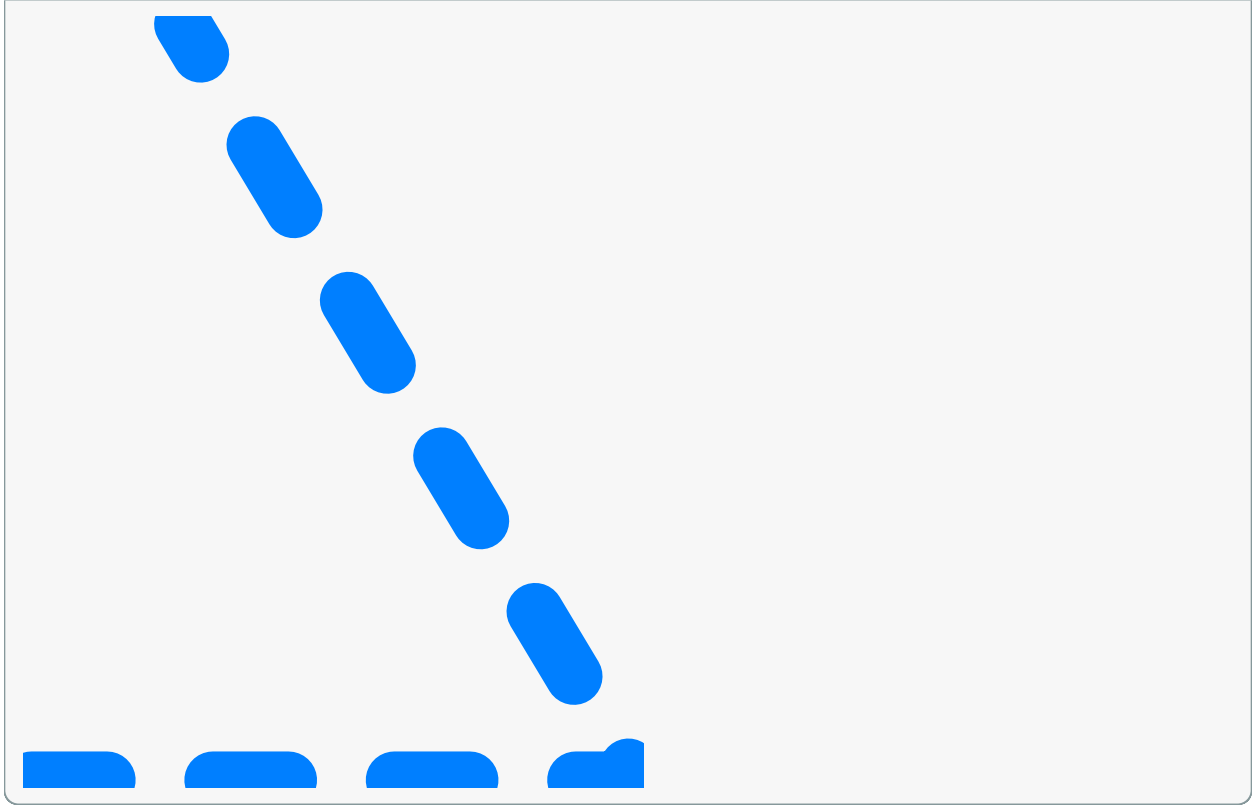
We assume that the *Stroke* of the *PolyLine* above is 2mm #007fff solid.

```
{PolyLineSolidSvgCode}
```

**example**

We consider the same *PolyLine* as in the previous example, except that the *Stroke* is 2mm #007fff Dash.

```
{PolyLineDashSvgCode}
```



Points

Data Property

The points of the *PolyLine*.

Type: *Point*

Multiplicity: 2..*

Ordered: yes

Unique: no

Example

```
[  
  Point(  
    X = 3.5,  
    Y = 5.5),  
  Point(  
    X = 5.5,  
    Y = 7.5)  
]
```


Stroke**Data Property**

The stroke of the *PolyLine*.

Type: *Stroke*

Multiplicity: 1

Ordered: no

Unique: no

Example

Stroke(

Color = *Color*(

B = 0,

G = 0,

R = 128),

DashStyle = *Solid*,

Width = 0.3)

Example in DEXPI XML

```
<Object type="Core/Diagram.PolyLine">
  <Data property="Points">
    <AggregatedDataValue type="Core/Diagram.Point">
      <Data property="X">
        <Double>3.5</Double>
      </Data>
      <Data property="Y">
        <Double>5.5</Double>
      </Data>
    </AggregatedDataValue>
    <AggregatedDataValue type="Core/Diagram.Point">
      <Data property="X">
        <Double>5.5</Double>
      </Data>
      <Data property="Y">
        <Double>7.5</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="Stroke">
    <AggregatedDataValue type="Core/Diagram.Stroke">
      <Data property="Color">
        <AggregatedDataValue type="Core/Diagram.Color">
          <Data property="B">
            <Integer>0</Integer>
          </Data>
```

(continues on next page)

(continued from previous page)

```

    <Data property="G">
      <Integer>0</Integer>
    </Data>
    <Data property="R">
      <Integer>128</Integer>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="DashStyle">
  <DataReference data="Core/Diagram.DashStyle.Solid"/>
</Data>
<Data property="Width">
  <Double>0.3</Double>
</Data>
</AggregatedDataValue>
</Data>
</Object>

```

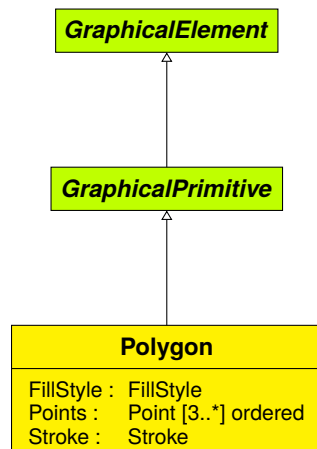
Polygon

Core::Diagram::Polygon

Overview

Concrete Class

A figure that is delimited by a closed chain of three or more straight sections.



Supertypes

- *GraphicalPrimitive*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>FillStyle</i>	1	<i>FillStyle</i>	no	no
<i>Points</i>	3..*	<i>Point</i>	yes	no
<i>Stroke</i>	1	<i>Stroke</i>	no	no

Geometry

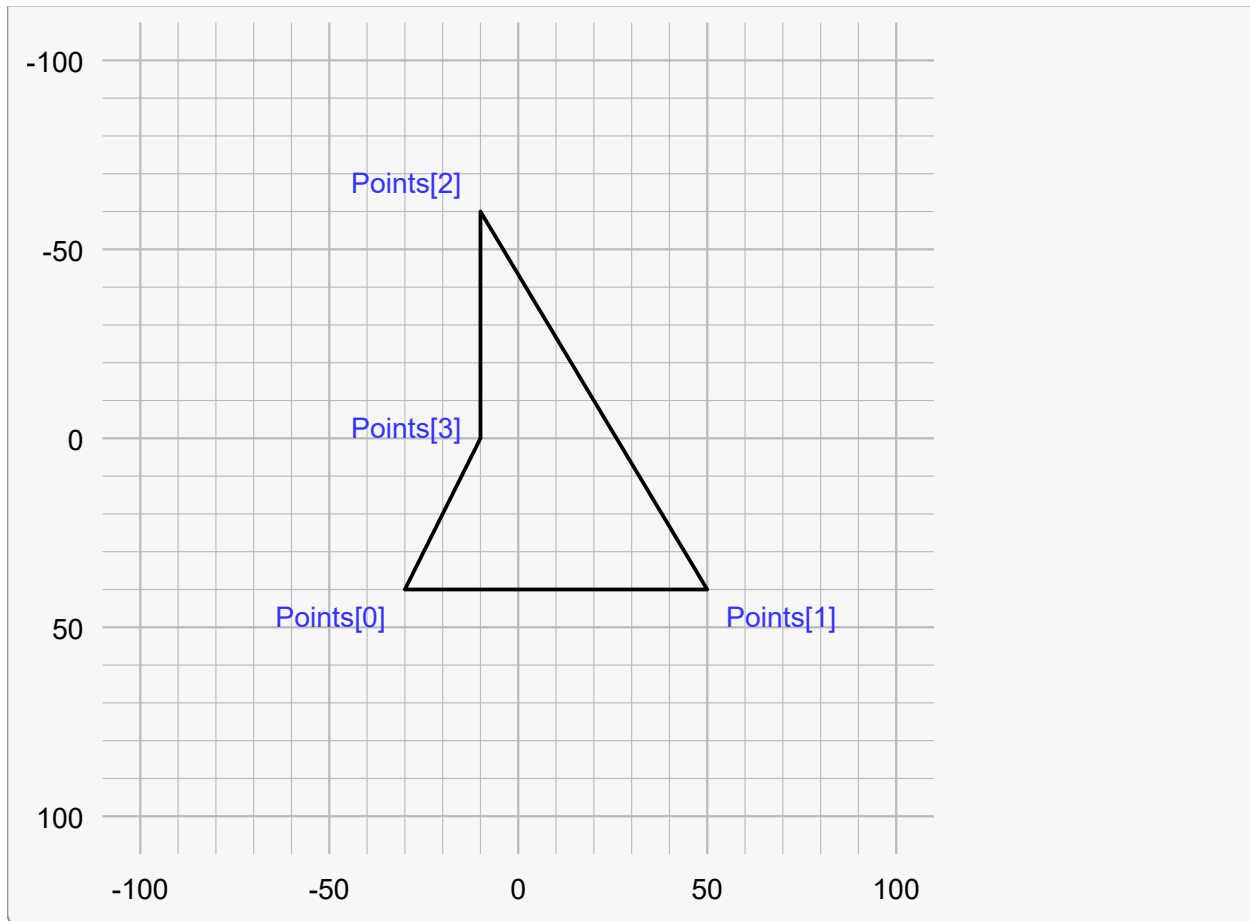
The first straight section that delimits a Polygon starts at its first *Point*, i.e., *Points[0]*, and ends at its second *Point*. For each further *Point*, another section is added that starts at the preceding *Point*. Finally, a section is added that starts at the last *Point* and goes back to the first *Point* such that the chain of sections is closed.

example

We consider a *Polygon* with 4 *Points*:

- (-30, 40),
- (50, 40),
- (-10, -60), and
- (-10, 0).

The *Polygon* will look like this:



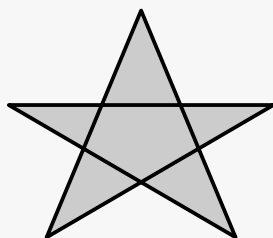
Interior of a Polygon

Depending on its *FillStyle*, the interior of a *Polygon* may be filled. There are several possibilities to define when “a point is inside a polygon”. DEXPI applies a simple intuitive rule:

- Choose an arbitrary point outside the geometric bounding box of the *Polygon*.
- Each point that can be reached from there without crossing any segment is outside the *Polygon*.
- Each other point is inside the *Polygon*.

example

The grey area in the figure visualizes the interior of the given *Polygon*.

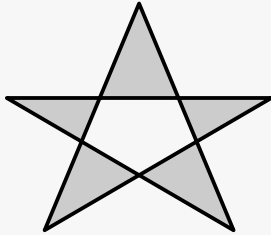


arbitrary point outside
the bounding box

technical

In case of polygons, the DEXPI definition for interior points coincides with the *nonzero rule* in SVG. See [Section 13.4.2. of the SVG 2 Recommendation](#) for a more precise definition of the *nonzero rule*.

The SVG 2 Recommendation also defines an alternative rule called *evenodd rule*. If we applied this rule, the pentagonal area in the center of this *Polygon* would be considered outside:



Interior of a polygon when the evenodd rule is applied.
For illustration only! DEXPI does not apply this rule!

Visual Center

Let n be the number of *Points*, and let x_i and y_i be the *X* and *Y* positions of *Point* i for $0 \leq i < n$.

Then the horizontal position c_x and the vertical position c_y of the visual center of the *Polygon* are

$$c_x = \frac{1}{2} \left[\min_{i=0}^{n-1} x_i + \max_{i=0}^{n-1} x_i \right] ,$$

$$c_y = \frac{1}{2} \left[\min_{i=0}^{n-1} y_i + \max_{i=0}^{n-1} y_i \right] .$$

Intuitively, the visual center is the center of the minimum bounding box that contains all *Points*.

If the *Polygon* has the *FillStyle Hatch*, then one of the hatch lines will cross through the visual center.

example

For the *Polygon* above, we have

$$\begin{aligned} c_x &= \frac{1}{2} [\min(-30, 50, -10, -10) + \max(-30, 50, -10, -10)] \\ &= \frac{1}{2} [-30 + 50] \\ &= 10 \end{aligned}$$

and

$$\begin{aligned} c_y &= \frac{1}{2} [\min(40, 40, -60, 0) + \max(40, 40, -60, 0)] \\ &= \frac{1}{2} [-60 + 40] \\ &= -10 . \end{aligned}$$

Mapping to SVG

A *Polygon* corresponds to an `svg:polygon` with these attributes:

- **transform** contains a translation from the origin to the visual center. This is required due to the modified positions in *points*.

`transform = "translate(+str( $c_x$ )+ , +str( $c_y$ )+)"`

- `points` contains the *Points* of the *Polygon* relative to the visual center.

For each *Point* i with (absolute) position x_i and y_i , the position relative to the visual center is

$$\begin{aligned}\tilde{x}_i &= x_i - c_x \quad , \\ \tilde{y}_i &= y_i - c_y \quad .\end{aligned}$$

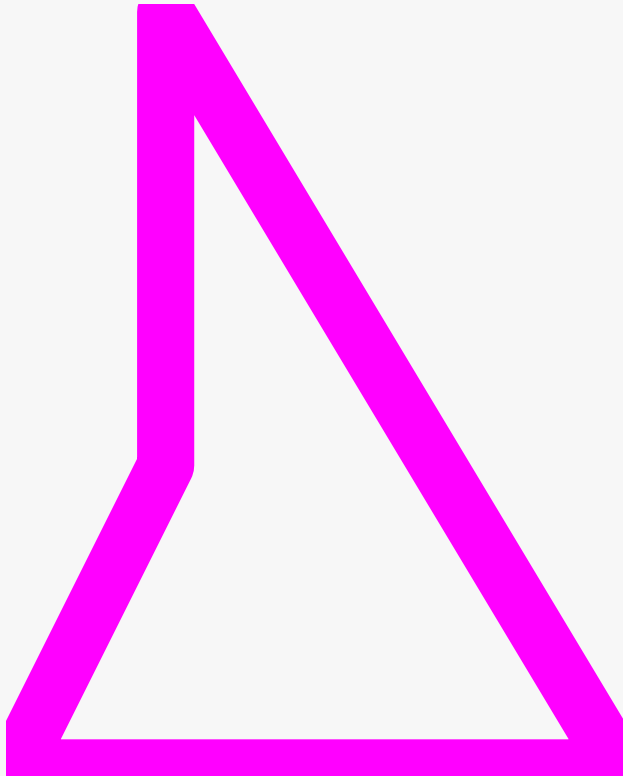
Each of the relative positions is converted to a string as $\text{str}(\tilde{x}_i) + , + \text{str}(\tilde{y}_i)$. Finally, the `points` attribute is set to a single string that contains all relative positions, separated by spaces.

- `stroke`, `stroke-dasharray`, `stroke-dashoffset` and `stroke-width` are set according to the SVG mapping rules for the *Stroke* of the *Polygon* (see *Stroke*).
- `stroke-linecap` = "round" and `stroke-linejoin` = "round" reflect the heuristic for line caps and line joins.
- `vector-effect` = "non-scaling-stroke" reflects the heuristic for scaled symbols.
- `fill` is set according to the SVG mapping rules for the *FillStyle* of the *Polygon* (see *FillStyle*).

example

We assume that the *Stroke* of the *Polygon* above is 2mm #ff00ff Solid and that its *FillStyle* is *Transparent*.

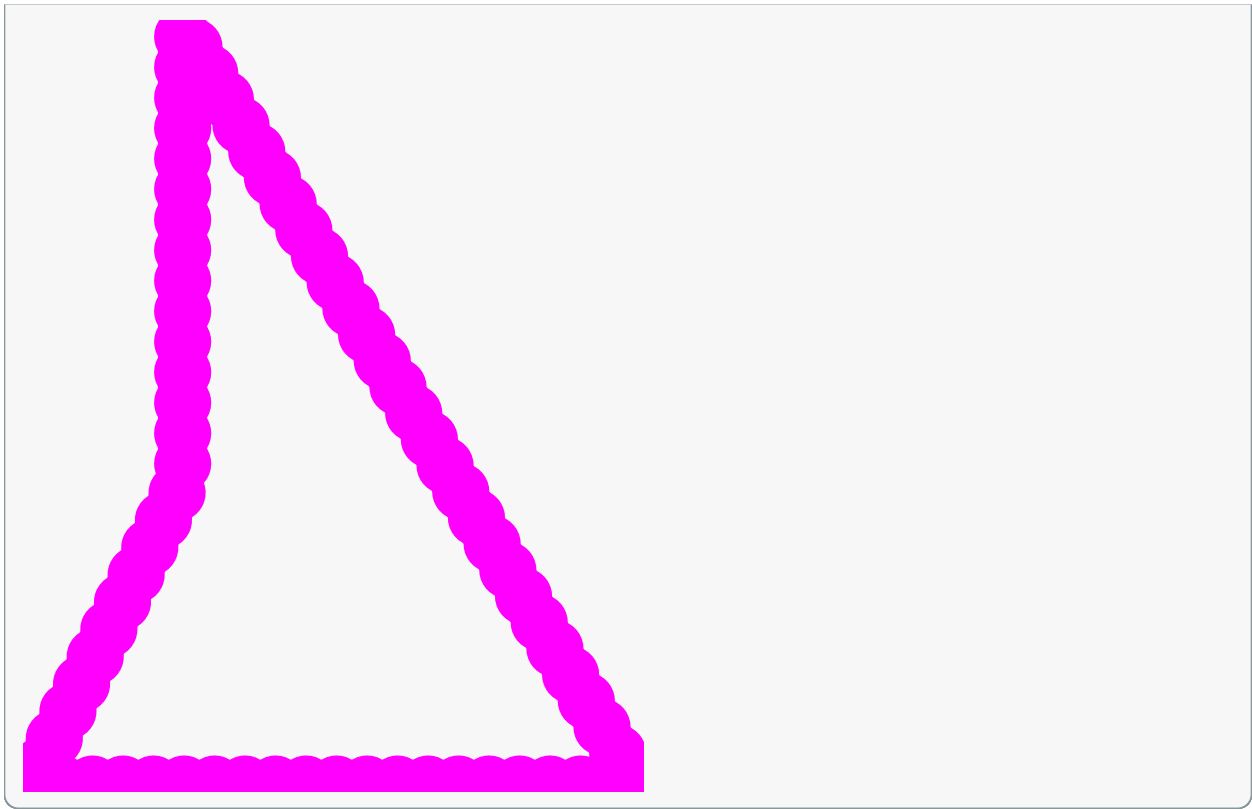
```
{PolygonTransparentSvgCode}
```



example

We consider the same *Polygon* as in the previous example, except that the *Stroke* is 2mm #ff00ff Dot.

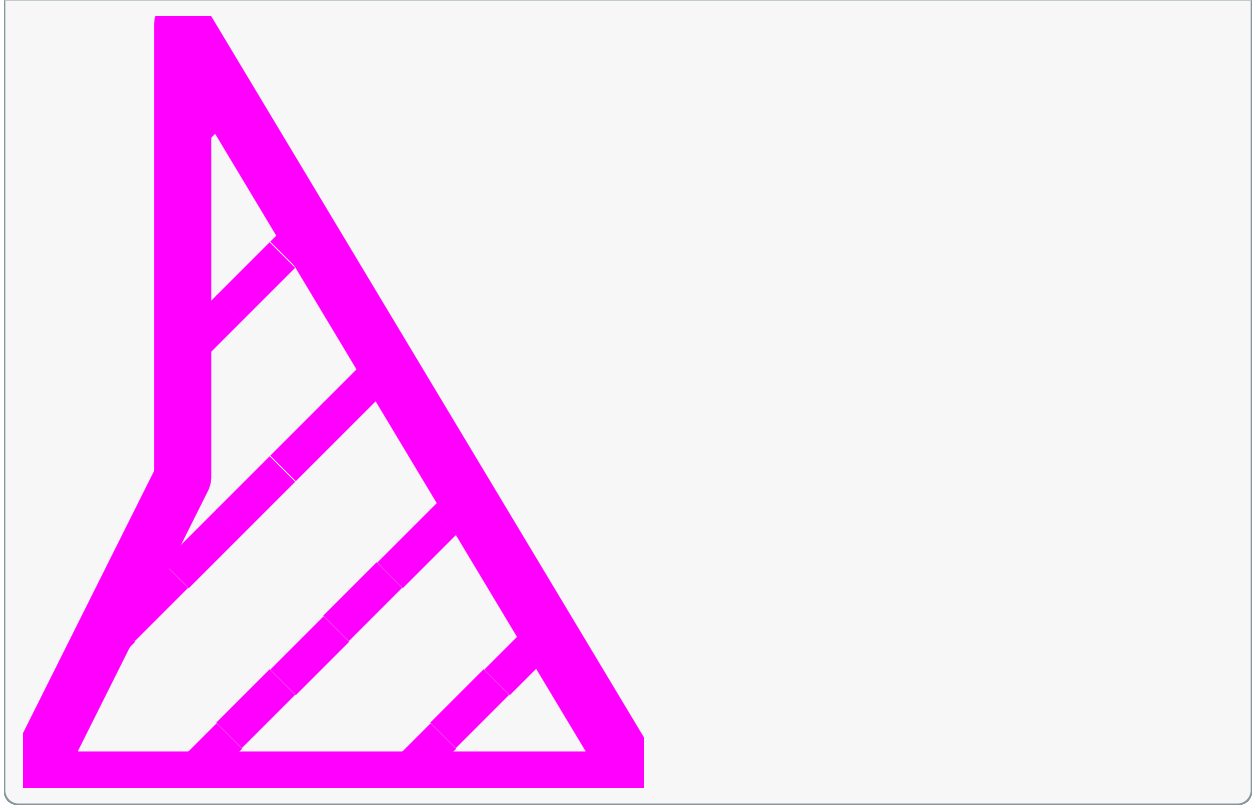
```
{PolygonDotSvgCode}
```



example

Finally, a variant with *Stroke* 2mm #ff00ff Solid and *FillStyle* Hatch.

```
{PolygonHatchSvgCode}
```



FillStyle

Data Property

The fill style of the *Polygon*.

Type: *FillStyle*

Multiplicity: 1

Ordered: no

Unique: no

Example

Transparent

Points

Data Property

The points (vertices) of the *Polygon*.

Type: *Point*

Multiplicity: 3..*

Ordered: yes

Unique: no

Example

```
[
  Point(
    X = 3.5,
    Y = 5.5),
  Point(
    X = 5.5,
    Y = 7.5),
  Point(
    X = 1.5,
    Y = 9.5)
]
```

Stroke**Data Property**

The stroke of the *Polygon*.

Type: *Stroke*

Multiplicity: 1

Ordered: no

Unique: no

Example

```
Stroke(
  Color = Color(
    B = 0,
    G = 0,
    R = 128),
  DashStyle = Solid,
  Width = 0.3)
```

Example in DEXPI XML

```
<Object type="Core/Diagram.Polygon">
  <Data property="FillStyle">
    <DataReference data="Core/Diagram.FillStyle.Transparent"/>
  </Data>
  <Data property="Points">
    <AggregatedDataValue type="Core/Diagram.Point">
      <Data property="X">
```

(continues on next page)

(continued from previous page)

```

    <Double>3.5</Double>
  </Data>
  <Data property="Y">
    <Double>5.5</Double>
  </Data>
</AggregatedDataValue>
<AggregatedDataValue type="Core/Diagram.Point">
  <Data property="X">
    <Double>5.5</Double>
  </Data>
  <Data property="Y">
    <Double>7.5</Double>
  </Data>
</AggregatedDataValue>
<AggregatedDataValue type="Core/Diagram.Point">
  <Data property="X">
    <Double>1.5</Double>
  </Data>
  <Data property="Y">
    <Double>9.5</Double>
  </Data>
</AggregatedDataValue>
</Data>
<Data property="Stroke">
  <AggregatedDataValue type="Core/Diagram.Stroke">
    <Data property="Color">
      <AggregatedDataValue type="Core/Diagram.Color">
        <Data property="B">
          <Integer>0</Integer>
        </Data>
        <Data property="G">
          <Integer>0</Integer>
        </Data>
        <Data property="R">
          <Integer>128</Integer>
        </Data>
      </AggregatedDataValue>
    </Data>
    <Data property="DashStyle">
      <DataReference data="Core/Diagram.DashStyle.Solid"/>
    </Data>
    <Data property="Width">
      <Double>0.3</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>

```

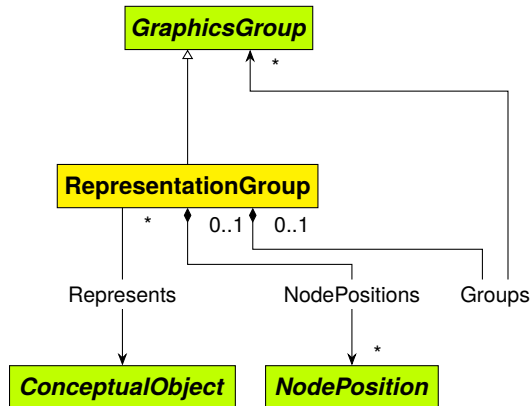
RepresentationGroup

Core::Diagram::RepresentationGroup

Overview

Concrete Class

A **RepresentationGroup** is a graphical representation of a *ConceptualObject*.



Supertypes

- *GraphicsGroup*

Subtypes

- *Diagram*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Groups</i>	*	<i>GraphicsGroup</i>	yes
<i>NodePositions</i>	*	<i>NodePosition</i>	no

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Represents</i>	1	<i>ConceptualObject</i>	no	no	*

Groups

Composition Property

The groups of the *RepresentationGroup*.

Type: *GraphicsGroup*

Multiplicity: *

Ordered: yes

NodePositions

Composition Property

The node positions of the *RepresentationGroup*.

Type: *NodePosition*

Multiplicity: *

Ordered: no

Represents

Reference Property

The object represented by the *RepresentationGroup*.

Type: *ConceptualObject*

Multiplicity: 1

Ordered: no

Unique: no

Opposite Multiplicity: *

Example in DEXPI XML

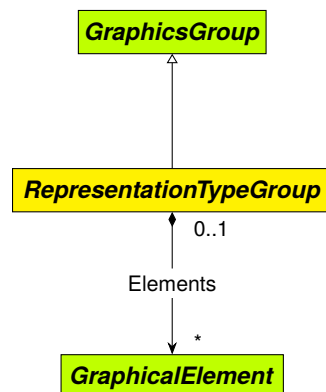
```
<Object type="Core/Diagram.RepresentationGroup"/>
```

RepresentationTypeGroup

Core::Diagram::RepresentationTypeGroup

Overview

Abstract Class



Supertypes

- *GraphicsGroup*

Subtypes

- *Border*
- *Label*
- *Static*
- *Symbol*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Elements</i>	*	<i>GraphicalElement</i>	yes

Elements

Composition Property

The graphical elements of the *RepresentationTypeGroup*.

Type: *GraphicalElement*

Multiplicity: *

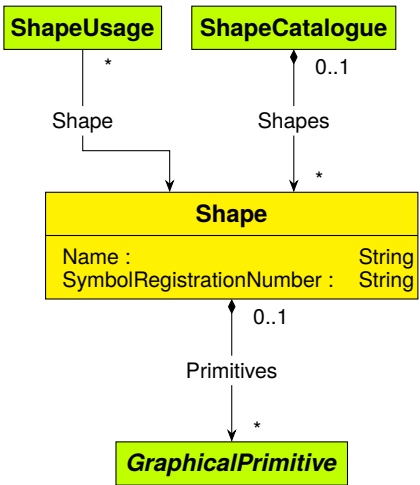
Ordered: yes

Shape

Core::Diagram::Shape

Overview

Concrete Class



Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Primitives</i>	*	<i>GraphicalPrimitive</i>	yes

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Name</i>	1	<i>String</i>	no	no
<i>SymbolRegistrationNumber</i>	1	<i>String</i>	no	no

Name

Data Property

Type: *String*

Multiplicity: 1

Ordered: no

Unique: no

Example

"ENTRIFUGAL_PUMP_SHAPE"

Primitives

Composition Property

The primitives of the *Shape*.

Type: *GraphicalPrimitive*

Multiplicity: *

Ordered: yes

SymbolRegistrationNumber

Data Property

A registration number that identifies the *Shape*.

The following format is recommended:

[standard]:[version]-[id]-[transform]

where

- **standard** is the name of a drawing standard,
- **version** is the version the drawing standard, e.g., a year,
- **id** is the identifier used in the standard for the *Shape*,

- **transform** gives the transformation of the *Shape* w.r.t. its depiction in the standard.

transform is one of the following values:

- A: no transformation,
- B: rotation by 90°,
- C: rotation by 180°,
- D: rotation by 270°,
- E: mirrored,
- F: mirrored, rotation by 90°,
- G: mirrored, rotation by 180°,
- H: mirrored, rotation by 270°.

Type: *String*

Multiplicity: 1

Ordered: no

Unique: no

Example

"ISO10628:2012-2322-A"

Example in DEXPI XML

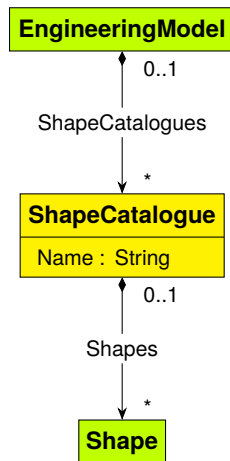
```
<Object type="Core/Diagram.Shape">
  <Data property="Name">
    <String>ENTRIFUGAL_PUMP_SHAPE</String>
  </Data>
  <Data property="SymbolRegistrationNumber">
    <String>ISO10628:2012-2322-A</String>
  </Data>
</Object>
```

ShapeCatalogue

Core::Diagram::ShapeCatalogue

Overview

Concrete Class



Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Shapes</i>	*	<i>Shape</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Name</i>	1	<i>String</i>	no	no

Name

Data Property

The name of the *ShapeCatalogue*.

Type: *String*

Multiplicity: 1

Ordered: no

Unique: no

Example

"Shapes for example P&ID"

Shapes

Composition Property

The shapes of the *ShapeCatalogue*.

Type: *Shape*

Multiplicity: *

Ordered: no

Example in DEXPI XML

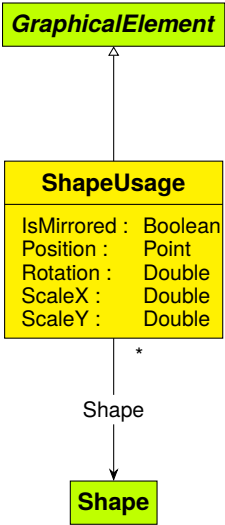
```
<Object type="Core/Diagram.ShapeCatalogue">
  <Data property="Name">
    <String>Shapes for example P&ID</String>
  </Data>
</Object>
```

ShapeUsage

Core::Diagram::ShapeUsage

Overview

Concrete Class



Supertypes

- *GraphicalElement*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Shape</i>	1	<i>Shape</i>	no	no	*

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>IsMirrored</i>	1	<i>Boolean</i>	no	no
<i>Position</i>	1	<i>Point</i>	no	no
<i>Rotation</i>	1	<i>Double</i>	no	no
<i>ScaleX</i>	1	<i>Double</i>	no	no
<i>ScaleY</i>	1	<i>Double</i>	no	no

IsMirrored

Data Property

Type: *Boolean*

Multiplicity: 1

Ordered: no

Unique: no

Example

false

Position

Data Property

Type: *Point*

Multiplicity: 1

Ordered: no

Unique: no

Example

```
Point(  
  X = 20.0,  
  Y = 20.0)
```

Rotation

Data Property

Type: *Double*

Multiplicity: 1

Ordered: no

Unique: no

Example

0.0

ScaleX**Data Property****Type:** *Double***Multiplicity:** 1**Ordered:** no**Unique:** no**Example**

1.0

ScaleY**Data Property****Type:** *Double***Multiplicity:** 1**Ordered:** no**Unique:** no**Example**

1.0

Shape**Reference Property**The shape used by the *ShapeUsage*.**Type:** *Shape***Multiplicity:** 1**Ordered:** no**Unique:** no**Opposite Multiplicity:** ***Example in DEXPI XML**

```
<Object type="Core/Diagram.ShapeUsage">
  <Data property="IsMirrored">
    <Boolean>false</Boolean>
  </Data>
  <Data property="Position">
    <AggregatedDataValue type="Core/Diagram.Point">
```

(continues on next page)

(continued from previous page)

```

<Data property="X">
  <Double>20.0</Double>
</Data>
<Data property="Y">
  <Double>20.0</Double>
</Data>
</AggregatedDataValue>
</Data>
<Data property="Rotation">
  <Double>0.0</Double>
</Data>
<Data property="ScaleX">
  <Double>1.0</Double>
</Data>
<Data property="ScaleY">
  <Double>1.0</Double>
</Data>
</Object>

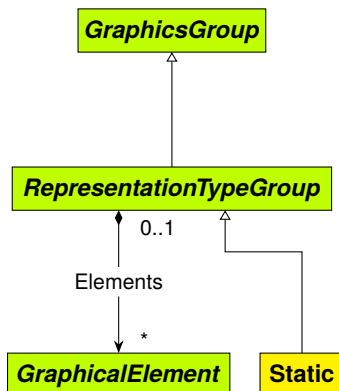
```

Static

Core::Diagram::Static

Overview

Concrete Class



Supertypes

- *RepresentationTypeGroup*

Example in DEXPI XML

```
<Object type="Core/Diagram.Static"/>
```

Stroke

Core::Diagram::Stroke

Overview

Aggregated Data Type

A stroke defines the visual appearance of a line.

<<datatype>> Stroke	
Color :	Color
DashStyle :	DashStyle
Width :	Double

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Color</i>	1	<i>Color</i>	no	no
<i>DashStyle</i>	1	<i>DashStyle</i>	no	no
<i>Width</i>	1	<i>Double</i>	no	no

Simplified Literals

Within this specification, a simplified type of literal is used to refer to a *Stroke*. This literal is composed of

- a decimal representation of the *Width*, followed by mm,
- the simplified literal of the *Color* of the *Stroke* (see *Color*),
- the name of the *DashStyle* of the *Stroke* (see *DashStyle*).

The three components are separated by spaces.

For example, a *Stroke* with

- *Width* = 2,
- *Color* = #00ff00, and
- *DashStyle* = *Dot*

is represented as 2mm #00ff00 Dot.

Mapping to SVG

A *Stroke* is mapped to a set of XML attributes, which are added to the SVG element (e.g., a `svg:polygon` or a `svg:polyline` element) that is drawn with the *Stroke*.

- The *Color* of the *Stroke* is mapped to the `stroke` attribute. For the attribute value, see the SVG mapping for *Color*.
- The *DashStyle* of the *Stroke* is mapped to the `stroke-dasharray` and `stroke-dashoffset` attributes as described in the SVG mapping for *DashStyle*. Note that these attribute values depend on the *Width* of the *Stroke* and on the *linecap* of the *GraphicalPrimitive* that uses the *Stroke*.

- The *Width* of the *Stroke* is mapped to the `stroke-width` attribute. The attribute value is the concatenation of the string representation of the *Width* and the fixed string `mm`. That is, in terms of standard-css-values-4 we use an *absolute length* with the unit millimeters.

example

For examples, see the SVG mapping sections of those subclasses of *GraphicalPrimitive* that have a *Stroke* attribute:

- *Ellipse*
- *EllipseArc*
- *Polygon*
- *PolyLine*

Color**Data Property**

The color of the *Stroke*.

Type: *Color*

Multiplicity: 1

Ordered: no

Unique: no

Example

Color(

B = 255,

G = 0,

R = 0)

DashStyle**Data Property**

The dash style of the *Stroke*.

Type: *DashStyle*

Multiplicity: 1

Ordered: no

Unique: no

Example

Solid

Width

Data Property

The width of the *Stroke* in mm.

Type: *Double*

Multiplicity: 1

Ordered: no

Unique: no

Example

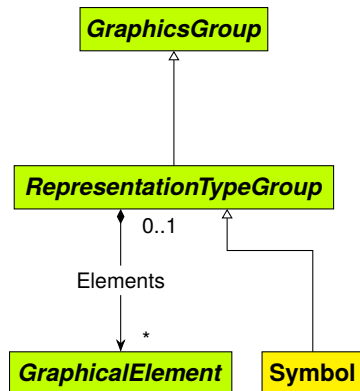
2.0

Symbol

Core::Diagram::Symbol

Overview

Concrete Class



Supertypes

- *RepresentationTypeGroup*

Subtypes

- *CustomSymbol*
- *InsulationSymbol*
- *PipeFlowArrow*
- *PipeSlopeSymbol*

Example in DEXPI XML

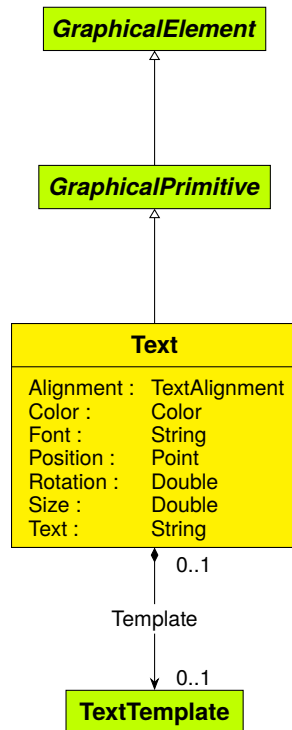
```
<Object type="Core/Diagram.Symbol"/>
```

Text

Core::Diagram::Text

Overview

Concrete Class



Supertypes

- *GraphicalPrimitive*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Template</i>	0..1	<i>TextTemplate</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Alignment</i>	1	<i>TextAlignment</i>	no	no
<i>Color</i>	1	<i>Color</i>	no	no
<i>Font</i>	1	<i>String</i>	no	no
<i>Position</i>	1	<i>Point</i>	no	no
<i>Rotation</i>	1	<i>Double</i>	no	no
<i>Size</i>	1	<i>Double</i>	no	no
<i>Text</i>	1	<i>String</i>	no	no

Alignment

Data Property

Type: *TextAlignment*

Multiplicity: 1

Ordered: no

Unique: no

Example

CenterBottom

Color

Data Property

Type: *Color*

Multiplicity: 1

Ordered: no

Unique: no

Example

Color(

B = 0,

G = 0,

R = 0)

Font

Data Property

Type: *String*

Multiplicity: 1

Ordered: no

Unique: no

Example

"Calibri"

Position

Data Property

Type: *Point*

Multiplicity: 1

Ordered: no

Unique: no

Example

```
Point(  
     $X = 9.7$ ,  
     $Y = 16.64$ )
```

Rotation

Data Property

Type: *Double*

Multiplicity: 1

Ordered: no

Unique: no

Example

0.0

Size

Data Property

Type: *Double*

Multiplicity: 1

Ordered: no

Unique: no

Example

3.0

Template

Composition Property

The template that describes how the text is built.

Type: *TextTemplate*

Multiplicity: 0..1

Ordered: no

Text

Data Property

Type: *String*

Multiplicity: 1

Ordered: no

Unique: no

Example

"N1"

Example in DEXPI XML

```

<Object type="Core/Diagram.Text">
  <Data property="Alignment">
    <DataReference data="Core/Diagram.TextAlignment.CenterBottom"/>
  </Data>
  <Data property="Color">
    <AggregatedDataValue type="Core/Diagram.Color">
      <Data property="B">
        <Integer>0</Integer>
      </Data>
      <Data property="G">
        <Integer>0</Integer>
      </Data>
      <Data property="R">
        <Integer>0</Integer>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="Font">
    <String>Calibri</String>
  </Data>
  <Data property="Position">
    <AggregatedDataValue type="Core/Diagram.Point">
      <Data property="X">
        <Double>9.7</Double>
      </Data>
      <Data property="Y">
        <Double>16.64</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="Rotation">
    <Double>0.0</Double>
  </Data>

```

(continues on next page)

(continued from previous page)

```
</Data>
<Data property="Size">
  <Double>3.0</Double>
</Data>
<Data property="Text">
  <String>N1</String>
</Data>
</Object>
```

TextAlignment

Core::Diagram::TextAlignment

Overview

Enumeration

<<enumeration>> TextAlignment
CenterBottom
CenterCenter
CenterTop
LeftBottom
LeftCenter
LeftTop
RightBottom
RightCenter
RightTop

Members

Literals

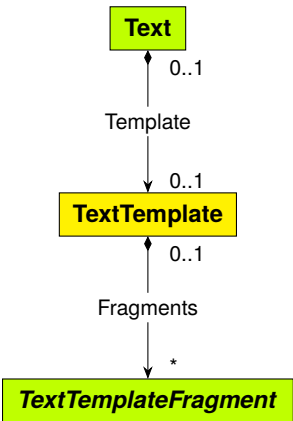
Name
CenterBottom
CenterCenter
CenterTop
LeftBottom
LeftCenter
LeftTop
RightBottom
RightCenter
RightTop

TextTemplate

Core::Diagram::TextTemplate

Overview

Concrete Class



Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Fragments</i>	*	<i>TextTemplateFragment</i>	yes

Fragments

Composition Property

The fragments of the *TextTemplate*.

Type: *TextTemplateFragment*

Multiplicity: *

Ordered: yes

Example in DEXPI XML

```
<Object type="Core/Diagram.TextTemplate"/>
```

TextTemplateFragment

Core::Diagram::TextTemplateFragment

Overview

Abstract Class



Subtypes

- *AttributeRepresentation*
- *LiteralText*

Core::Diagram

Overview

Package

Two-dimensional engineering diagrams.

The Diagram package provides concepts

1. for the graphical representation of a *ConceptualObject*,
2. for linking parts of this graphical representation to parts of the *ConceptualObject*, and
3. for semantic annotations of the graphical representation.

technical

This specification is not a drawing standard. It does not prescribe any drawing style or standard and it does not prescribe which information in a *ConceptualObject* has to be represented in a *Diagram*.

technical

Until version 1.2, graphics has not been in the scope of the DEXPI P&ID Specification. Graphics was handled as prescribed by standard-proteus. Over the years, some best practices concerning graphics have evolved in the DEXPI community, and fragmentary annotations have been added to the DEXPI P&ID Specification.

In DEXPI P&ID Specification 1.3, an informative Graphics package was added to cover conceptual and graphics information in a single comprehensive model.

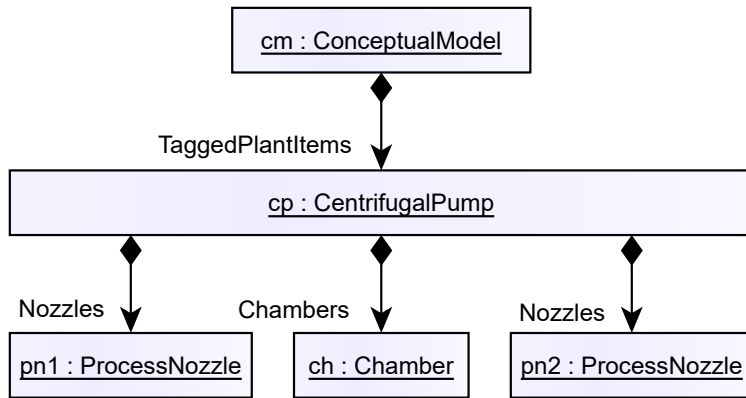
For the P&ID Specification 1.4, the Graphics package was further elaborated, and it is now a normative part of the specification. It serves to document the best practices that have emerged in the past, but no new features have been added.

The Graphics package is independent from Proteus Schema in the sense that it is a pure UML package. But, like for all other packages in this specification, mappings to Proteus Schema are defined that describe how instances of the package's types are serialized. In consequence, the features of the Graphics package are constrained to what is supported by Proteus Schema. Future versions of the DEXPI P&ID Specification will skip support for Proteus Schema, such that additional features can be added to the Graphics package depending on the DEXPI community's needs.

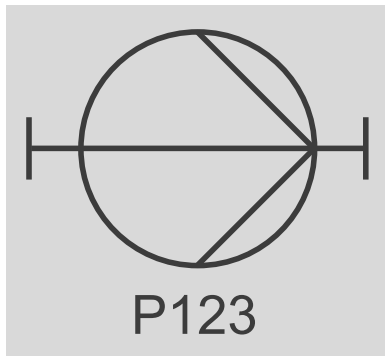
Since DEXPI 2.0, the Graphics package can be used for both P&IDs and process models.

Example

We consider a simple *ConceptualModel* cm that contains a single *CentrifugalPump* with two *ProcessNozzles* and a *Chamber*; in total, there are five *ConceptualObjects*:

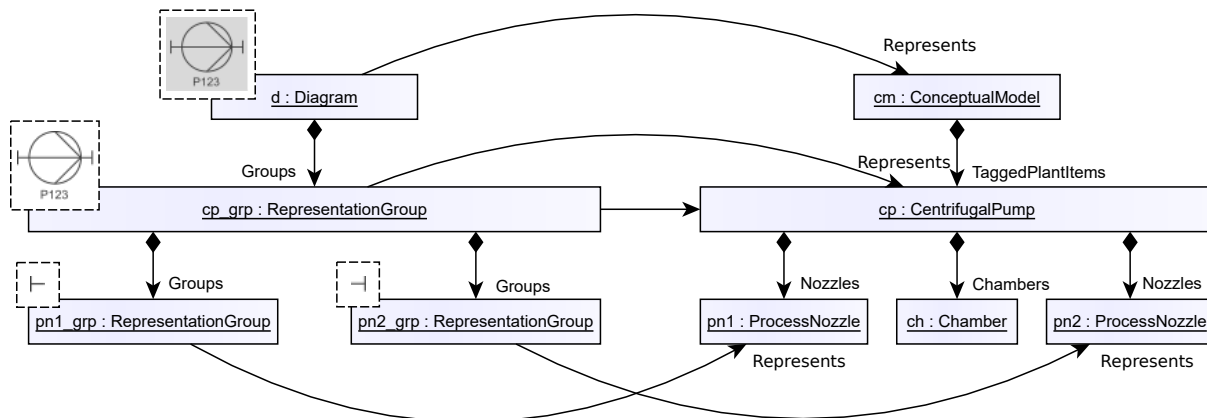


A typical graphical representation of this model may look like this:



Linking Graphics and Conceptual Objects

The graphics can be decomposed in parts each of which represents one of the *ConceptualObjects*. For each of these parts, we use a *RepresentationGroup*:



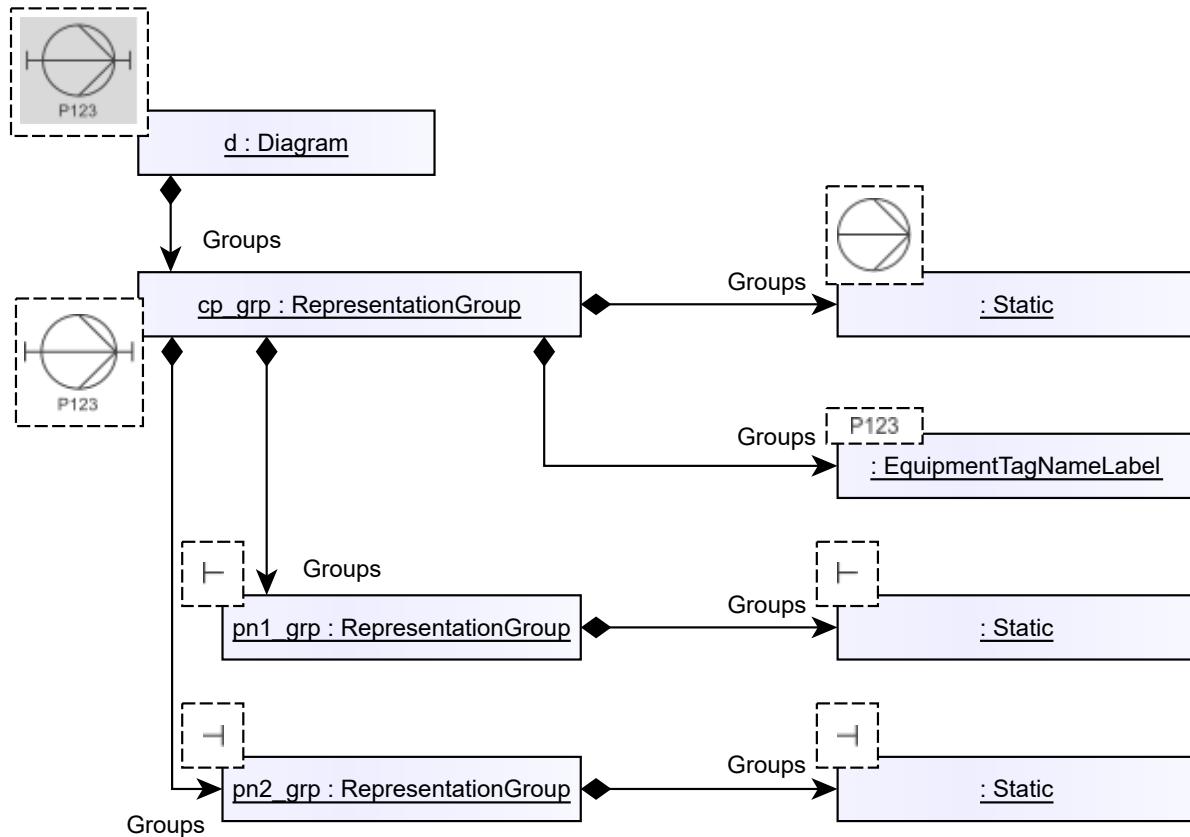
- The *Diagram* *d* represents the entire *ConceptualModel*. *Diagram* is a subclass of *RepresentationGroup* that is meant to be used for the entire graphics. In addition to a conventional *RepresentationGroup*, a *Diagram* has a size (width and height) and a background color.
- *cp_grp* is the *RepresentationGroup* for the *CentrifugalPump* *cp*.
- *pn1_grp* and *pn2_grp* represent the two *ProcessNozzles* *pn1* and *pn2*.

Note that the composition hierarchy of the Diagram corresponds exactly to the composition hierarchy of the *ConceptualModel*. In the *ConceptualModel*, pn1 is a component of cp, and in the Diagram, pn1_grp is a component of cp_grp.

However, if a *ConceptualObject* (including all its descendants) has no graphical representation at all, no *RepresentationGroup* is required: In the example above, there is no *RepresentationGroup* for the *Chamber* ch.

Adding Semantics

So far, we have decomposed the overall graphics in *RepresentationGroups* that correspond to certain *ConceptualObjects*. On a finer level of granularity, we can identify parts with different meanings. These meanings are covered by different subclasses of the abstract class *RepresentationTypeGroup*. In the example, two of these subclasses are used: *Static* and *EquipmentTagNameLabel*.

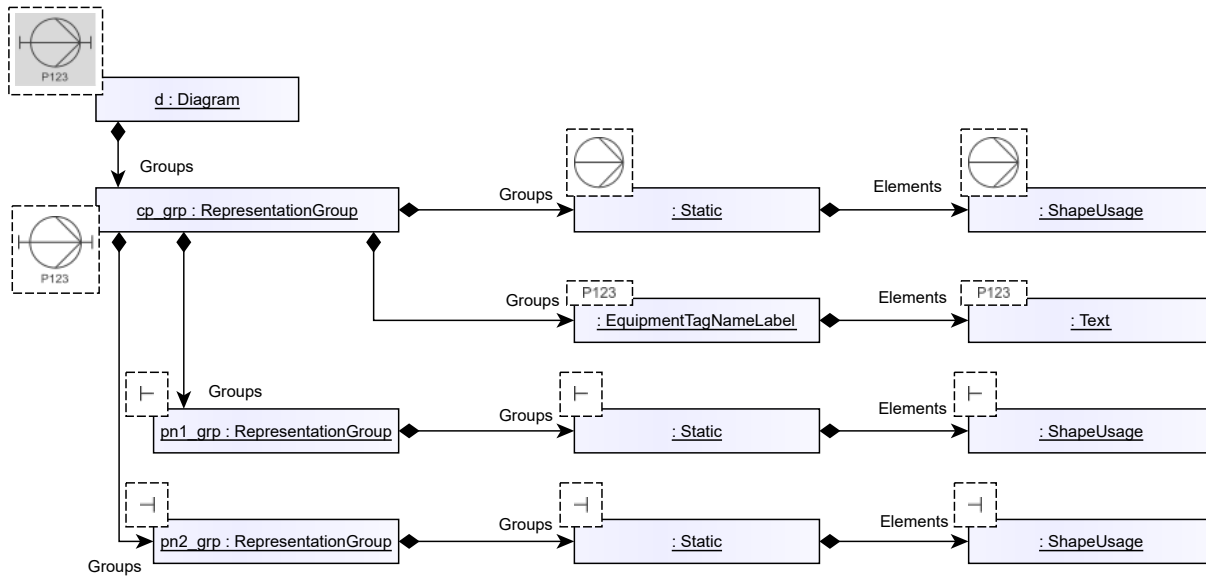


- Consider cp_grp, the *RepresentationGroup* of the *CentrifugalPump* cp. In addition to the two groups for the *ProcessNozzles*, cp_grp now contains two *RepresentationTypeGroups*.
 - The *Static* group contains the actual *CentrifugalPump* graphics without any annotations, labels, etc.
 - The *EquipmentTagNameLabel* group displays the tag name of the *CentrifugalPump*.
- pn1_grp contains a *Static* group for the basic symbol of the *ProcessNozzle*.
- pn2_grp contains another *Static* group for the second *ProcessNozzle* symbol.

Whereas *RepresentationGroups* can be decomposed in further groups, *RepresentationTypeGroups* cannot contain further groups.

Adding Graphical Elements

RepresentationTypeGroups rather contain *GraphicalElements* such as geometric primitives, texts, or *ShapeUsages*.



- The *Static* group for the *CentrifugalPump* contains a *ShapeUsage*, i.e., a reference to a reusable *Shape* with information about this specific usage (scaling, rotation, position).
- The *EquipmentTagNameLabel* for the *CentrifugalPump* contains a *Text*.
- Both *Static* groups for the two *ProcessNozzles* contain a *ShapeUsage*.

Coordinate System

The coordinate system used is a 2D Cartesian system whose horizontal axis is oriented to the right and whose vertical axis is oriented to the bottom. In consequence, angles are measured clockwise.

Lengths are dimensionless. In the examples, we assume they are in millimeters.

In Proteus Schema, the vertical axis to the top, and angles are measured anti-clockwise.

Mapping to SVG

The purpose of these mappings is to define the visualization of instances of the types in this package, i.e., “how they look like”.

In order to keep the mappings simple, the SVG mappings for all *GraphicalPrimitives* are defined such that they can be easily scaled when they appear in a *Shape* (in particular, they use `vector-effect = "non-scaling-stroke"`). In consequence, further scaling, e.g., in an application, will lead to unacceptable results.

The SVG mappings in this specification must not be misunderstood as a proposal for exchanging P&ID graphics via SVG.

Modulo Operator

In some calculations for the SVG and Proteus mappings, the *modulo operator* (or *remainder of division*) is used. As the modulo operator can be defined in different ways, we give a definition of how the operator is used in this specification. We content ourselves with a definition for positive quotients.

Let $q > 0, a \in \mathbb{R}$. Then

$$a \bmod q = a - \left\lfloor \frac{a}{q} \right\rfloor \cdot q \quad .$$

$\lfloor \cdot \rfloor$ is the *floor function* that for $x \in \mathbb{R}$ yields the largest integer n such that $n \leq x$.

For example,

$$\begin{aligned} 0 \bmod 360 &= 0 - \left\lfloor \frac{0}{360} \right\rfloor \cdot 360 = 0 - 0 \cdot 360 = 0 \quad , \\ 100 \bmod 360 &= 100 - \left\lfloor \frac{100}{360} \right\rfloor \cdot 360 = 100 - 0 \cdot 360 = 100 \quad , \\ 359 \bmod 360 &= 359 - \left\lfloor \frac{359}{360} \right\rfloor \cdot 360 = 359 - 0 \cdot 360 = 359 \quad , \\ 360 \bmod 360 &= 360 - \left\lfloor \frac{360}{360} \right\rfloor \cdot 360 = 360 - 1 \cdot 360 = 0 \quad , \\ (-100) \bmod 360 &= -100 - \left\lfloor \frac{-100}{360} \right\rfloor \cdot 360 = -100 - (-1) \cdot 360 = 260 \quad . \end{aligned}$$

technical

The definition of the modulo operator differs between programming languages. For example, the modulo operator in Python, written `%`, coincides with the definition above. The modulo operator in JavaScript (ECMAScript), also written `%`, behaves differently for negative dividends: `-100 % 360` is evaluated to `-100`.

Members

Classes

Name	Type
<i>AttributeRepresentation</i>	Concrete Class
<i>Border</i>	Concrete Class
<i>ConnectorLine</i>	Concrete Class
<i>CustomSymbol</i>	Concrete Class
<i>Diagram</i>	Concrete Class
<i>Ellipse</i>	Concrete Class
<i>EllipseArc</i>	Concrete Class
<i>GraphicalElement</i>	Abstract Class
<i>GraphicalPrimitive</i>	Abstract Class
<i>GraphicsGroup</i>	Abstract Class
<i>InsulationSymbol</i>	Concrete Class
<i>Label</i>	Concrete Class
<i>LiteralText</i>	Concrete Class
<i>MetaData</i>	Concrete Class
<i>NodePosition</i>	Abstract Class
<i>PipeFlowArrow</i>	Concrete Class
<i>PipeSlopeSymbol</i>	Concrete Class
<i>PolyLine</i>	Concrete Class
<i>Polygon</i>	Concrete Class
<i>RepresentationGroup</i>	Concrete Class
<i>RepresentationTypeGroup</i>	Abstract Class
<i>Shape</i>	Concrete Class
<i>ShapeCatalogue</i>	Concrete Class
<i>ShapeUsage</i>	Concrete Class
<i>Static</i>	Concrete Class
<i>Symbol</i>	Concrete Class
<i>Text</i>	Concrete Class
<i>TextTemplate</i>	Concrete Class
<i>TextTemplateFragment</i>	Abstract Class

Data Types

Name	Type
<i>AttributeRepresentationType</i>	Enumeration
<i>Color</i>	Aggregated Data Type
<i>ConfidentialityClassification</i>	Enumeration
<i>DashStyle</i>	Enumeration
<i>FillStyle</i>	Enumeration
<i>Point</i>	Aggregated Data Type
<i>Stroke</i>	Aggregated Data Type
<i>TextAlignment</i>	Enumeration

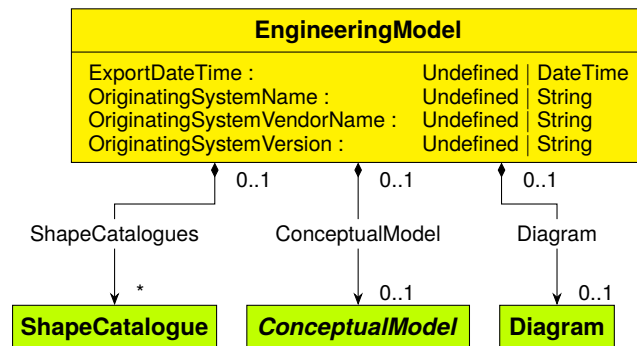
3.1.5 EngineeringModel

Core::EngineeringModel

Overview

Concrete Class

An entire DEXPI engineering model.



Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>ConceptualModel</i>	0..1	<i>ConceptualModel</i>	no
<i>Diagram</i>	0..1	<i>Diagram</i>	no
<i>ShapeCatalogues</i>	*	<i>ShapeCatalogue</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>ExportDateTime</i>	1	<i>Undefined DateTime</i>	no	no
<i>OriginatingSystemName</i>	1	<i>Undefined String</i>	no	no
<i>OriginatingSystemVendorName</i>	1	<i>Undefined String</i>	no	no
<i>OriginatingSystemVersion</i>	1	<i>Undefined String</i>	no	no

ConceptualModel

Composition Property

The conceptual model of the EngineeringModel.

Type: *ConceptualModel*

Multiplicity: 0..1

Ordered: no

Diagram

Composition Property

The diagram of the EngineeringModel.

Type: *Diagram*

Multiplicity: 0..1

Ordered: no

ExportDateTime

Data Property

The date time at which the *EngineeringModel* was exported by the originating system (see *OriginatingSystemName*).

Type: *Undefined* | *DateTime*

Multiplicity: 1

Ordered: no

Unique: no

Example

2020-12-07T15:32:42

OriginatingSystemName

Data Property

The name of the system from which the *EngineeringModel* originates, e.g., the name of a P&ID tool.

Type: *Undefined* | *String*

Multiplicity: 1

Ordered: no

Unique: no

Example

"PID Kit Professional"

OriginatingSystemVendorName

Data Property

The name of the vendor of the system from which the *EngineeringModel* originates, e.g., the name of a software company.

Type: *Undefined* | *String*

Multiplicity: 1

Ordered: no

Unique: no

Example

"Smart and Clever Systems, Inc."

OriginatingSystemVersion

Data Property

The version of the the system from which the *EngineeringModel* originates, e.g., the version number of a tool.

Type: *Undefined* | *String*

Multiplicity: 1

Ordered: no

Unique: no

Example

"1.1"

ShapeCatalogues

Composition Property

The shape catalogues of the EngineeringModel.

Type: *ShapeCatalogue*

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Core/EngineeringModel">
  <Data property="ExportDateTime">
    <DateTime>2020-12-07T15:32:42</DateTime>
  </Data>
  <Data property="OriginatingSystemName">
    <String>PID Kit Professional</String>
  </Data>
  <Data property="OriginatingSystemVendorName">
    <String>Smart and Clever Systems, Inc.</String>
  </Data>
  <Data property="OriginatingSystemVersion">
    <String>1.1</String>
  </Data>
</Object>
```

3.1.6 Note

*Core::*Note

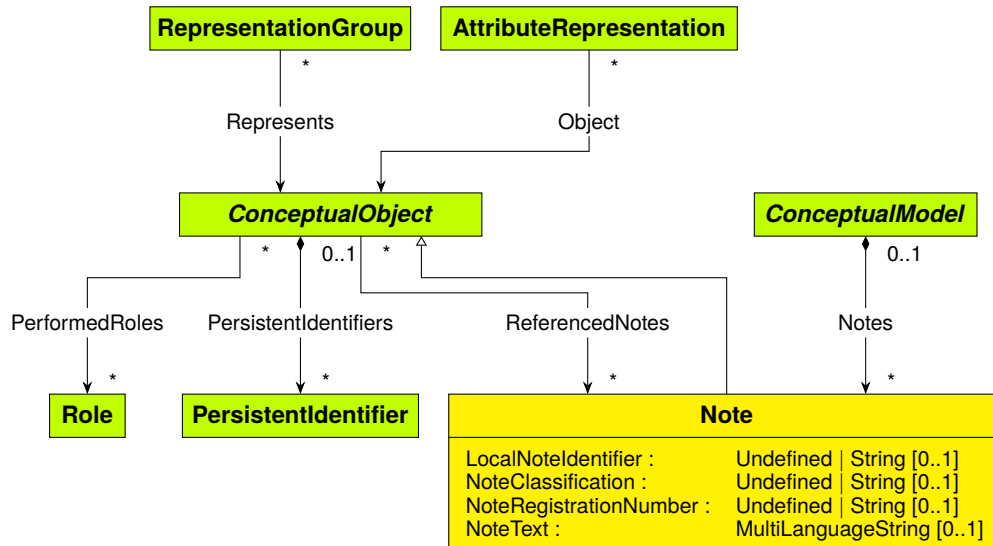
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/Note>

NOTE

A textual annotation about *ConceptualObjects*. See also *ReferencedNotes*.



Supertypes

- *ConceptualObject*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>LocalNoteIdentifier</i>	0..1	<i>Undefined String</i>	no	no
<i>NoteClassification</i>	0..1	<i>Undefined String</i>	no	no
<i>NoteRegistrationNumber</i>	0..1	<i>Undefined String</i>	no	no
<i>NoteText</i>	0..1	<i>MultiLanguageString</i>	no	no

LocalNoteIdentifier

Data Property

An identifier for the *Note* in a local context.

A typical usage of the *LocalNoteIdentifier* is to visualize the relation between a *ConceptualObject* shown on a diagram and the actual *NoteText* shown elsewhere on the diagram: The *LocalNoteIdentifier* would be drawn near the symbol of the *ConceptualObject*. A separate table would list the *LocalNoteIdentifier* and *NoteText* attributes of all *Notes* on the diagram.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LocalNoteIdentifierAssignmentClass>

LOCAL NOTE IDENTIFIER ASSIGNMENT CLASS

Example

"42"

NoteClassification

Data Property

A textual classification of the *Note*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NoteClassificationAssignmentClass>

NOTE CLASSIFICATION ASSIGNMENT CLASS

Example

"operational requirement"

NoteRegistrationNumber

Data Property

A registration number of the *Note*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NoteRegistrationNumberAssignmentClass>

NOTE REGISTRATION NUMBER ASSIGNMENT CLASS

Example

"XY-42-1"

NoteText

Data Property

The text of the *Note*.

Type: *MultiLanguageString*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NoteTextAssignmentClass>

NOTE TEXT ASSIGNMENT CLASS

Example

```
MultiLanguageString(
  SingleLanguageStrings = [
    SingleLanguageString(
      Language = "en",
      Value = "Water outlet is elevated to reduce sand entrainment."))]
```

Example in DEXPI XML

```
<Object type="Core/Note">
  <Data property="LocalNoteIdentifier">
    <String>42</String>
  </Data>
  <Data property="NoteClassification">
    <String>operational requirement</String>
  </Data>
  <Data property="NoteRegistrationNumber">
    <String>XY-42-1</String>
  </Data>
  <Data property="NoteText">
    <AggregatedDataValue type="Core/DataTypes.MultiLanguageString">
      <Data property="SingleLanguageStrings">
        <AggregatedDataValue type="Core/DataTypes.SingleLanguageString">
          <Data property="Language">
            <String>en</String>
          </Data>
          <Data property="Value">
            <String>Water outlet is elevated to reduce sand entrainment.</String>
          </Data>
        </AggregatedDataValue>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

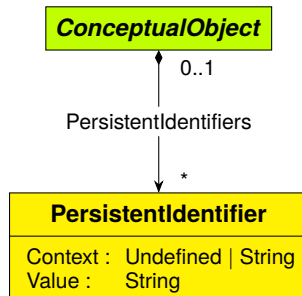
3.1.7 PersistentIdentifier

Core::PersistentIdentifier

Overview

Concrete Class

A persistent context-dependent identifier for a *ConceptualObject*.



Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Context</i>	1	<i>Undefined String</i>	no	no
<i>Value</i>	1	<i>String</i>	no	no

Context

Data Property

The context of the *PersistentIdentifier* such as a database, a project number or name, etc.

Type: *Undefined | String*

Multiplicity: 1

Ordered: no

Unique: no

Example

"mysql://myhost:1111/db"

Value

Data Property

The value of the *PersistentIdentifier*, i.e., the actual identifier string.

Type: *String*

Multiplicity: 1

Ordered: no

Unique: no

Example

"84187274"

Example in DEXPI XML

```
<Object type="Core/PersistentIdentifier">
  <Data property="Context">
    <String>mysql://myhost:1111/db</String>
  </Data>
  <Data property="Value">
    <String>84187274</String>
  </Data>
</Object>
```

3.1.8 PhysicalQuantities

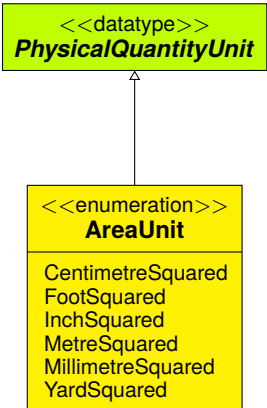
AreaUnit

Core::PhysicalQuantities::AreaUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *area* with dimension L^2 .



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Sym- bol	UN Code	RDL Label	RDL URI
Centime- treSquared	cm ²	CMK	CENTIMETRE SQUARED	http://data.posccaesar.org/rdl/RDS1357829
FootSquared	ft ²	FTK	FOOT SQUARED	http://data.posccaesar.org/rdl/RDS1342934
InchSquared	in ²	INK	INCH SQUARED	http://data.posccaesar.org/rdl/RDS1342979
MetreSquared	m ²	MTK	METRE SQUARED	http://data.posccaesar.org/rdl/RDS1358009
Millime- treSquared	mm ²	MMK	MILLIMETRE SQUARED	http://data.posccaesar.org/rdl/RDS1358189
YardSquared	yd ²	YDK	YARD SQUARED	http://data.posccaesar.org/rdl/RDS1343744

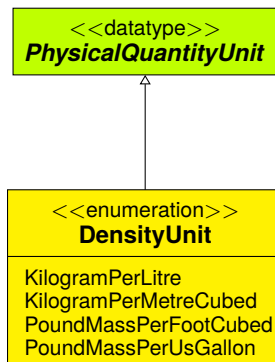
DensityUnit

Core::PhysicalQuantities::DensityUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *density* with dimension L⁻³M.



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Symbol	UN Code	RDL Label	RDL URI
KilogramPerLitre	kg/l	B35	KILOGRAM PER LITRE	http://data.posccaesar.org/rdl/RDS1355129
KilogramPerMetre-Cubed	kg/m ³	KMQ	KILOGRAM PER METRE CUBED	http://data.posccaesar.org/rdl/RDS1354454
PoundMassPer-FootCubed	lb/ft ³	87	POUND MASS PER FOOT CUBED	http://data.posccaesar.org/rdl/RDS1340909
PoundMassPerUs-Gallon	lb/gal (US)	GE	POUND MASS PER US GALLON	http://data.posccaesar.org/rdl/RDS1341629

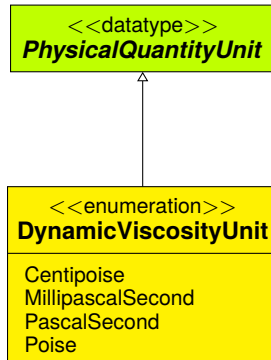
DynamicViscosityUnit

Core::PhysicalQuantities::DynamicViscosityUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *dynamic viscosity* with dimension $L^{-1}MT^{-1}$.



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Sym- bol	UN Code	RDL Label	RDL URI
Centipoise	cP	C7	CENTIPOISE	http://data.posccaesar.org/rdl/RDS1318274
MillipascalSec- ond	mPa·s	C24	MILLIPASCAL OND	http://data.posccaesar.org/rdl/RDS11617155
PascalSecond	Pa·s	C65	PASCAL SECOND	http://data.posccaesar.org/rdl/RDS1352204
Poise	P	89	POISE	http://data.posccaesar.org/rdl/RDS1339964

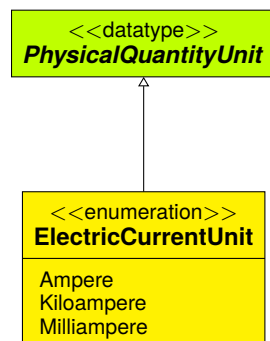
ElectricCurrentUnit

*Core::PhysicalQuantities::*ElectricCurrentUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *electric current* with dimension I.



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Symbol	UN Code	RDL Label	RDL URI
Ampere	A	AMP	AMPERE	http://data.posccaesar.org/rdl/RDS1313954
Kiloampere	kA	B22	KILOAMPERE	http://data.posccaesar.org/rdl/RDS4316756455
Milliampere	mA	4K	MILLIAMPERE	http://data.posccaesar.org/rdl/RDS4316806954

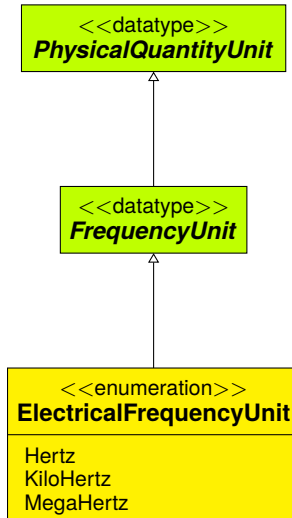
ElectricalFrequencyUnit

Core::PhysicalQuantities::ElectricalFrequencyUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *electrical frequency* with dimension T^{-1} .



Supertypes

- *FrequencyUnit*

Members

Literals

Name	Symbol	UN Code	RDL Label	RDL URI
Hertz	Hz	HTZ	HERTZ	http://data.posccaesar.org/rdl/RDS1326464
KiloHertz	kHz	KHZ	KILOHERTZ	http://data.posccaesar.org/rdl/RDS4316756612
MegaHertz	MHz	MHZ	MEGAHERTZ	http://data.posccaesar.org/rdl/RDS4316806716

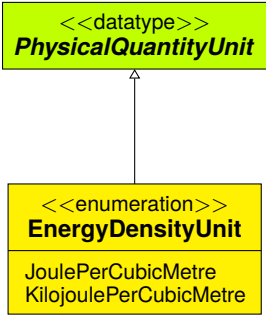
EnergyDensityUnit

Core::PhysicalQuantities::EnergyDensityUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *energy density* with dimension $L^{-1}MT^{-2}$.



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Symbol	RDL Label
JoulePerCubicMetre	J/m ³	JOULE PER CUBIC METRE
KilojoulePerCubicMetre	kJ/m ³	KILOJOULE PER CUBIC METRE

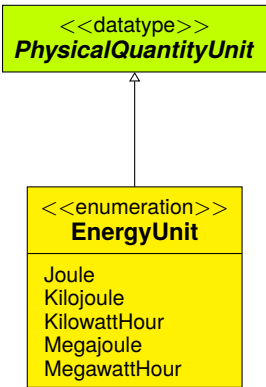
EnergyUnit

Core::PhysicalQuantities::EnergyUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *energy* with dimension L^2MT^{-2} .



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Sym- bol	UN Code	RDL Label	RDL URI
Joule	J	JOU	JOULE	http://data.posccaesar.org/rdl/RDS1327544
Kilojoule	kJ	KJO	KILOJOULE	http://data.posccaesar.org/rdl/RDS4316756626
KilowattHour	kW·h	KWH	KILOWATT HOUR	http://data.posccaesar.org/rdl/RDS4316756769
Megajoule	MJ	3B	MEGAJOULE	http://data.posccaesar.org/rdl/RDS4316806730
MegawattHour	MW·h	MWH	MEGAWATT HOUR	http://data.posccaesar.org/rdl/RDS43168067178

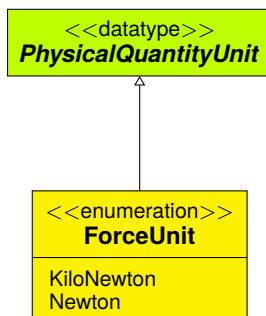
ForceUnit

Core::PhysicalQuantities::ForceUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *force* with dimension LMT^{-2} .



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Symbol	UN Code	RDL Label	RDL URI
KiloNewton	kN	B47	KILONEWTON	http://data.posccaesar.org/rdl/RDS1351034
Newton	N	NEW	NEWTON	http://data.posccaesar.org/rdl/RDS1337939

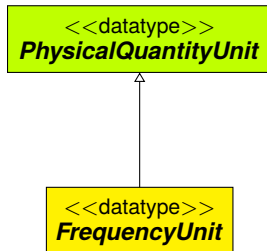
FrequencyUnit

Core::PhysicalQuantities::FrequencyUnit

Overview

AbstractDataType

A unit of measurement for a physical quantity of type *frequency* with dimension T^{-1} . FrequencyUnit has 3 subtypes for different application areas.



Supertypes

- *PhysicalQuantityUnit*

Subtypes

- *ElectricalFrequencyUnit*
- *NumberPerTimeIntervalUnit*
- *RotationalFrequencyUnit*

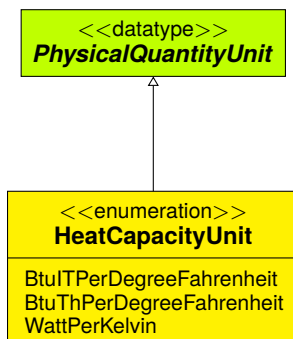
HeatCapacityUnit

Core::PhysicalQuantities::HeatCapacityUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *heat capacity* with dimension $L^2M\Theta^{-1}T^{-3}$.



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Symbol	UN Code	RDL Label	RDL URI
BtuITPerDegreeFahrenheit	Btu[IT]/°F	N60		
BtuThPerDegreeFahrenheit	Btu[th]/°F	N61		
WattPerKelvin	W/K	D52	WATT KELVIN	PER http://data.posccaesar.org/rdl/RDS1355759

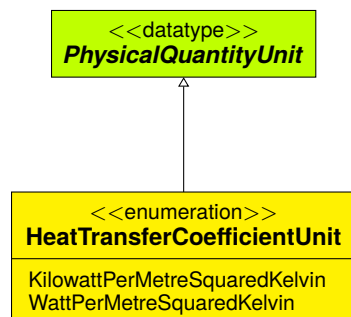
HeatTransferCoefficientUnit

Core::PhysicalQuantities::HeatTransferCoefficientUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *heat transfer coefficient* with dimension $MT^{-3}\Theta$.



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Sym- bol	UN Code	RDL Label	RDL URI
KilowattPerMe- treSquaredKelvin	kW/(m ² ·K)	N78	KILOWATT PER SQUARED KELVIN	http://data.posccaesar.org/rdl/RDS43167567170
WattPerMe- treSquaredKelvin	W/(m ² ·K)	D55	WATT PER METRE SQUARED KELVIN	http://data.posccaesar.org/rdl/RDS1348424

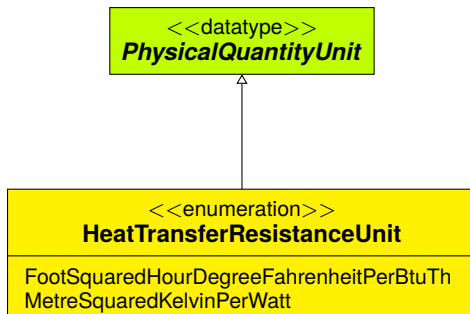
HeatTransferResistanceUnit

Core::PhysicalQuantities::HeatTransferResistanceUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *heat transfer resistance* with dimension $M^{-1}\Theta T^3$.



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Sym- bol	UN Code	RDL Label	RDL URI
FootSquaredHourDe- greeFahrenheitPerB- tuTh	°F·h·ft ² /E	J19	DEGREE FAHRENHEIT SQUARE FOOT HOUR PER BRITISH THERMAL UNIT	http://data.posccaesar.org/rdl/RDS43166353195
MetreSquaredKelvin- PerWatt	m ² ·K/W	D19	KELVIN SQUARE METRE PER WATT	http://data.posccaesar.org/rdl/RDS4316756411

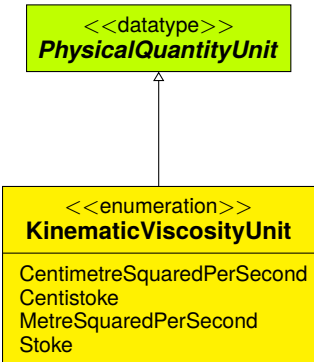
KinematicViscosityUnit

Core::PhysicalQuantities::KinematicViscosityUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *kinematic viscosity* with dimension L^2T^{-1} .



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Sym- bol	UN Code	RDL Label			RDL URI
CentimetreSquared- PerSecond	cm ² /s	M81	SQUARE	CENTIMETRE PER SECOND		http://data.posccaesar.org/rdl/RDS43168636161
Centistoke	cSt	4C	CENTISTOKE			http://data.posccaesar.org/rdl/RDS43165906236
MetreSquaredPerSec- ond	m ² /s	S4	METRE	SQUARED	PER	http://data.posccaesar.org/rdl/RDS1355399
Stoke	St	91	STOKE			http://data.posccaesar.org/rdl/RDS4316863983

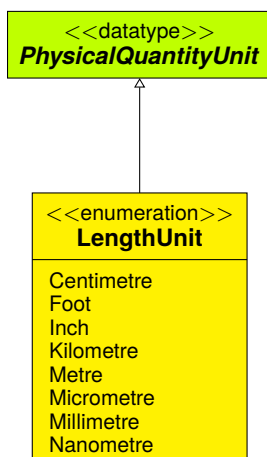
LengthUnit

Core::PhysicalQuantities::LengthUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *length* with dimension L.



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Symbol	UN Code	RDL Label	RDL URI
Centimetre	cm	CMT	CENTIMETRE	http://data.posccaesar.org/rdl/RDS1318004
Foot	ft	FOT	FOOT	http://data.posccaesar.org/rdl/RDS1324664
Inch	in	INH	INCH	http://data.posccaesar.org/rdl/RDS1326959
Kilometre	km	KMT	KILOMETRE	http://data.posccaesar.org/rdl/RDS1330199
Metre	m	MTR	METRE	http://data.posccaesar.org/rdl/RDS1332674
Micrometre	µm	4H	MICROMETRE	http://data.posccaesar.org/rdl/RDS1351529
Millimetre	mm	MMT	MILLIMETRE	http://data.posccaesar.org/rdl/RDS1357739
Nanometre	nm	C45	NANOMETRE	http://data.posccaesar.org/rdl/RDS1337669

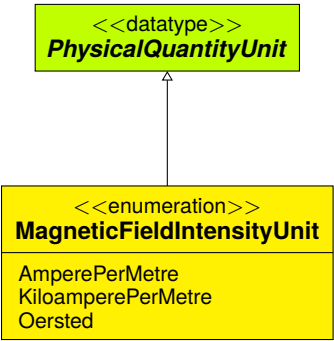
MagneticFieldIntensityUnit

Core::PhysicalQuantities::MagneticFieldIntensityUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *magnetic field intensity* with dimension IL^{-1} .



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Sym- bol	UN Code	RDL Label	RDL URI
AmperePerMetre	A/m	AE	AMPERE PER METRE	http://data.posccaesar.org/rdl/RDS1353734
KiloamperePer- Metre	kA/m	B24	KILOAMPERE PER ME- TRE	http://data.posccaesar.org/rdl/RDS4316756466
Oersted	Oe	66	OERSTED	http://data.posccaesar.org/rdl/RDS1338479

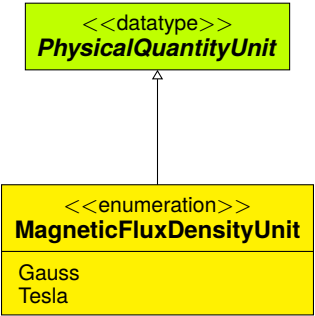
MagneticFluxDensityUnit

Core::PhysicalQuantities::MagneticFluxDensityUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *magnetic flux density* with dimension $MT^{-2}I^{-1}$.



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Symbol	UN Code	RDL Label	RDL URI
Gauss	Gs	76	GAUSS	http://data.posccaesar.org/rdl/RDS4316755955
Tesla	T	D33	TESLA	http://data.posccaesar.org/rdl/RDS1344644

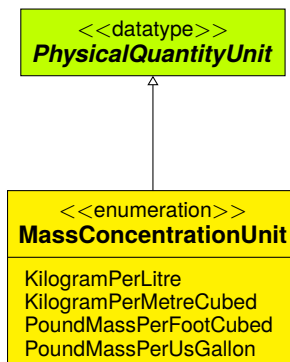
MassConcentrationUnit

*Core::PhysicalQuantities::*MassConcentrationUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *mass concentration* with dimension $L^{-3}M$.



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Symbol	UN Code	RDL Label	RDL URI
KilogramPerLitre	kg/l	B35	KILOGRAM PER LITRE	http://data.posccaesar.org/rdl/RDS1355129
KilogramPerMetre-Cubed	kg/m ³	KMQ	KILOGRAM PER METRE CUBED	http://data.posccaesar.org/rdl/RDS1354454
PoundMassPer-FootCubed	lb/ft ³	87	POUND MASS PER FOOT CUBED	http://data.posccaesar.org/rdl/RDS1340909
PoundMassPerUs-Gallon	lb/gal (US)	GE	POUND MASS PER US GALLON	http://data.posccaesar.org/rdl/RDS1341629

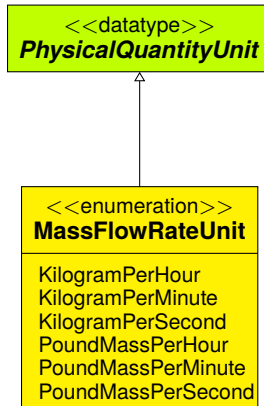
MassFlowRateUnit

Core::PhysicalQuantities::MassFlowRateUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *mass flow rate* with dimension MT^{-1} .



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Sym- bol	UN Code	RDL Label			RDL URI
KilogramPerHour	kg/h	E93	KILOGRAM PER HOUR			http://data.posccaesar.org/rdl/RDS1329344
KilogramPerMinute	kg/min	F31	KILOGRAM	PER	MINUTE	http://data.posccaesar.org/rdl/RDS1350719
KilogramPerSecond	kg/s	KGS	KILOGRAM	PER	SECOND	http://data.posccaesar.org/rdl/RDS1329659
PoundMassPerHour	lb/h	4U	POUND	MASS	PER HOUR	http://data.posccaesar.org/rdl/RDS43168250123
PoundMassPerMinute	lb/min	K78	POUND	MASS	PER MINUTE	http://data.posccaesar.org/rdl/RDS43168250156
PoundMassPerSecond	lb/s	K81	POUND	MASS	PER SECOND	http://data.posccaesar.org/rdl/RDS43168250167

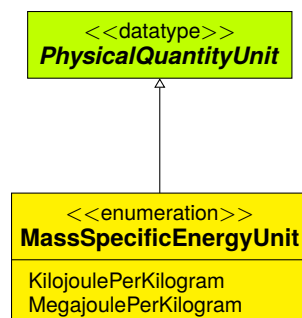
MassSpecificEnergyUnit

*Core::PhysicalQuantities::*MassSpecificEnergyUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *mass specific energy* with dimension L^2T^{-2} .



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Sym- bol	UN Code	RDL Label	RDL URI
KilojoulePerKilo- gram	kJ/kg	B42	KILOJOULE PER KILO- GRAM	http://data.posccaesar.org/rdl/RDS1329884
MegajoulePerKilo- gram	MJ/kg	JK	MEGAJOULE PER KILO- GRAM	http://data.posccaesar.org/rdl/RDS1332224

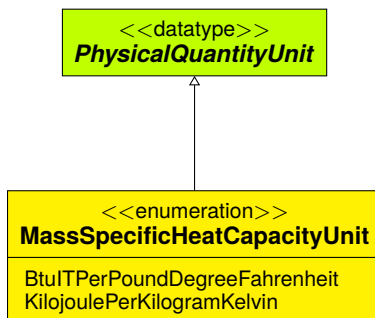
MassSpecificHeatCapacityUnit

Core::PhysicalQuantities::MassSpecificHeatCapacityUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *mass specific heat capacity* with dimension $L^2\Theta^{-1}T^{-2}$.



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Sym- bol	UN Code	RDL Label	RDL URI
BtuITPerPound- DegreeFahrenheit	Btu[IT]/(l	J43	BRITISH THERMAL UNIT PER POUND MASS DEGREE FAHRENHEIT	http://data.posccaesar.org/rdl/RDS43164893116
KilojoulePerKilo- gramKelvin	kJ/(kg·K)	B43	KILOJOULE PER KILOGRAM KELVIN	http://data.posccaesar.org/rdl/RDS43167566130

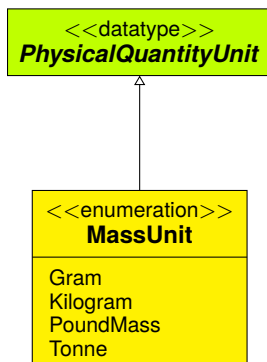
MassUnit

Core::PhysicalQuantities::MassUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *mass* with dimension M.



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Symbol	UN Code	RDL Label	RDL URI
Gram	g	GRM	GRAM	http://data.posccaesar.org/rdl/RDS1325789
Kilogram	kg	KGM	KILOGRAM	http://data.posccaesar.org/rdl/RDS1328669
PoundMass	lb	LBR	POUND MASS	http://data.posccaesar.org/rdl/RDS11617515
Tonne	t	TNE	TONNE	http://data.posccaesar.org/rdl/RDS1344689

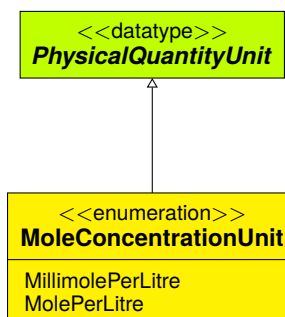
MoleConcentrationUnit

Core::PhysicalQuantities::MoleConcentrationUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *mole concentration* with dimension NL^{-3} .



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Symbol	UN Code	RDL Label	RDL URI
MillimolePerLitre	mmol/l	M33		
MolePerLitre	mol/l	C38	MOLE PER LITRE	http://data.posccaesar.org/rdl/RDS1357964

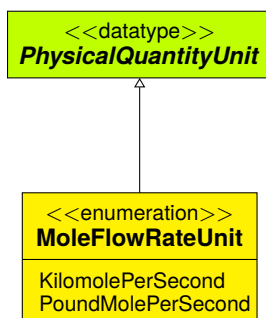
MoleFlowRateUnit

Core::PhysicalQuantities::MoleFlowRateUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *mole flow rate* with dimension NT^{-1} .



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Sym- bol	UN Code	RDL Label	RDL URI
KilomolePerSec- ond	kmol/s	E94	KILOMOLE PER SECOND	http://data.posccaesar.org/rdl/RDS43167566235
PoundMolePer- Second	lb- mol/s	P45	MOLE-POUND MASS PER SECOND	http://data.posccaesar.org/rdl/RDS4316807294

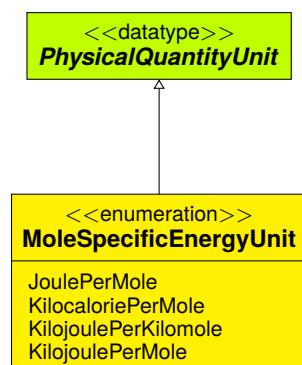
MoleSpecificEnergyUnit

Core::PhysicalQuantities::MoleSpecificEnergyUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *mole specific energy* with dimension $N^{-1}L^2MT^{-2}$.



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Sym- bol	UN Code	RDL Label	RDL URI
JoulePerMole	J/mol	B15	JOULE PER MOLE	http://data.posccaesar.org/rdl/RDS1354364
KilocaloriePerMole			KILOCALORIE PER MOLE	http://data.posccaesar.org/rdl/RDS43167564201
KilojoulePerKilo-mole	J/mol	B15	KILOJOULE PER KILO-MOLE	http://data.posccaesar.org/rdl/RDS43167566144
KilojoulePerMole	kJ/mol	B44	KILOJOULE PER MOLE	http://data.posccaesar.org/rdl/RDS43167566155

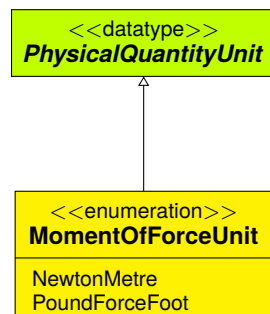
MomentOfForceUnit

Core::PhysicalQuantities::MomentOfForceUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *moment of force* with dimension L^2MT^{-2} .



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Sym- bol	UN Code	RDL Label	RDL URI
NewtonMetre	N·m	NU	NEWTON METRE	http://data.posccaesar.org/rdl/RDS1337984
PoundForceFoot	lbf·ft	M92	FOOT POUND FORCE	http://data.posccaesar.org/rdl/RDS1324754

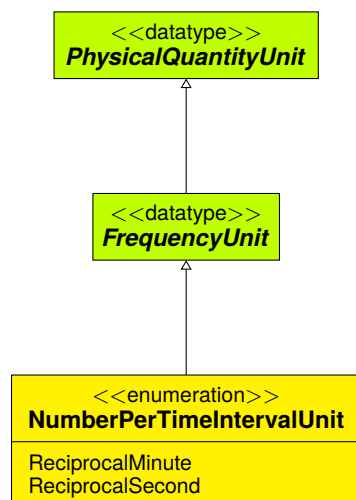
NumberPerTimeIntervalUnit

Core::PhysicalQuantities::NumberPerTimeIntervalUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *number per time interval* with dimension T^{-1} .



Supertypes

- *FrequencyUnit*

Members

Literals

Name	Sym- bol	UN Code	RDL Label	RDL URI
ReciprocalMinute	min^{-1}	C94	RECIPROCAL MINUTE	http://data.posccaesar.org/rdl/RDS4316851589
ReciprocalSecond	s^{-1}	C97	RECIPROCAL SECOND	http://data.posccaesar.org/rdl/RDS1355489

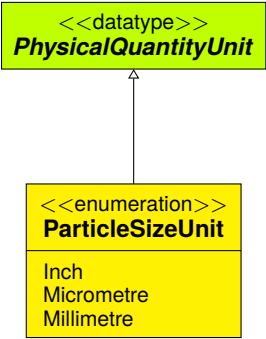
ParticleSizeUnit

Core::PhysicalQuantities::ParticleSizeUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *particle size* with dimension L.



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Symbol	UN Code	RDL Label	RDL URI
Inch	in	INH	INCH	http://data.posccaesar.org/rdl/RDS1326959
Micrometre	µm	4H	MICROMETRE	http://data.posccaesar.org/rdl/RDS1351529
Millimetre	mm	MMT	MILLIMETRE	http://data.posccaesar.org/rdl/RDS1357739

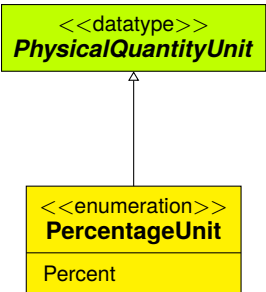
PercentageUnit

Core::PhysicalQuantities::PercentageUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *percentage* with dimension 1.



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Symbol	UN Code	RDL Label	RDL URI
Percent	???	???	PERCENT	http://data.posccaesar.org/rdl/RDS1317959

PhysicalQuantity

Core::PhysicalQuantities::PhysicalQuantity

Overview

Aggregated Data Type

A quantification of a scalar physical property such as a length or a force. A *PhysicalQuantity* comprises a numerical *Value* and a *Unit*.

UnitType : PhysicalQuantityUnit
<<datatype>> PhysicalQuantity
Unit : UnitType Value : Double

Members

Type Parameters

Name	Default Type
<i>UnitType</i>	<i>PhysicalQuantityUnit</i>

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Unit</i>	1	<i>UnitType</i>	no	no
<i>Value</i>	1	<i>Double</i>	no	no

Unit

Data Property

The unit of measurement of the *PhysicalQuantity*.

Type: *UnitType*

Multiplicity: 1

Ordered: no

Unique: no

Example*KilomolePerSecond***UnitType****Data Type Parameter**

A type parameter that gives the type of the *Unit*.

Default Type: *PhysicalQuantityUnit*

Value**Data Property**

The numerical value of the *PhysicalQuantity*.

Type: *Double*

Multiplicity: 1

Ordered: no

Unique: no

Example

8.5

PhysicalQuantityUnit

Core::PhysicalQuantities::PhysicalQuantityUnit

Overview**AbstractDataType**

<<datatype>> <i>PhysicalQuantityUnit</i>

Subtypes

- *AreaUnit*
- *DensityUnit*
- *DynamicViscosityUnit*
- *ElectricCurrentUnit*
- *EnergyDensityUnit*
- *EnergyUnit*
- *ForceUnit*
- *FrequencyUnit*
- *HeatCapacityUnit*
- *HeatTransferCoefficientUnit*

- *HeatTransferResistanceUnit*
- *KinematicViscosityUnit*
- *LengthUnit*
- *MagneticFieldIntensityUnit*
- *MagneticFluxDensityUnit*
- *MassConcentrationUnit*
- *MassFlowRateUnit*
- *MassSpecificEnergyUnit*
- *MassSpecificHeatCapacityUnit*
- *MassUnit*
- *MoleConcentrationUnit*
- *MoleFlowRateUnit*
- *MoleSpecificEnergyUnit*
- *MomentOfForceUnit*
- *ParticleSizeUnit*
- *PercentageUnit*
- *PowerUnit*
- *PressureUnit*
- *SurfaceTensionUnit*
- *TemperatureUnit*
- *ThermalConductivityUnit*
- *TimeIntervalUnit*
- *VelocityUnit*
- *VoltageUnit*
- *VolumeFlowRateUnit*
- *VolumeUnit*
- *pHUnit*

PhysicalQuantityVector

Core::PhysicalQuantities::PhysicalQuantityVector

Overview

Aggregated Data Type

A quantification of a vector of physical properties such as length or force. A `PhysicalQuantityVector` comprises an arbitrary number of *Values* and a *Unit*.

UnitType : PhysicalQuantityUnit
<<datatype>>
PhysicalQuantityVector
Unit : UnitType
Values : Undefined Double [*] ordered

Members

Type Parameters

Name	Default Type
<i>UnitType</i>	<i>PhysicalQuantityUnit</i>

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Unit</i>	1	<i>UnitType</i>	no	no
<i>Values</i>	*	<i>Undefined</i> <i>Double</i>	yes	no

Unit

Data Property

The unit of measurement of the *PhysicalQuantityVector*.

Type: *UnitType*

Multiplicity: 1

Ordered: no

Unique: no

Example

KilomolePerSecond

UnitType

Data Type Parameter

A type parameter that gives the type of the *Unit*.

Default Type: *PhysicalQuantityUnit*

Values

Data Property

The numerical values of the *PhysicalQuantityVector*.

Type: *Undefined* | *Double*

Multiplicity: *

Ordered: yes

Unique: no

Example

```
[  
  8.5,  
  17.3  
]
```

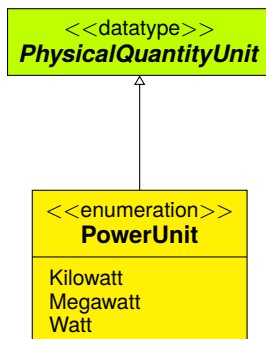
PowerUnit

Core::PhysicalQuantities::PowerUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *power* with dimension L^2MT^{-3} .



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Symbol	RDL Label	RDL URI
Kilowatt	kW	KILOWATT	http://data.posccaesar.org/rdl/RDS1330919
Megawatt	MW	MEGAWATT	http://data.posccaesar.org/rdl/RDS1332584
Watt	W	WATT	http://data.posccaesar.org/rdl/RDS1348154

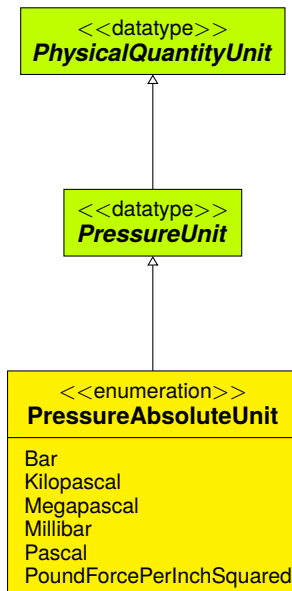
PressureAbsoluteUnit

Core::PhysicalQuantities::PressureAbsoluteUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *pressure absolute* with dimension $L^{-1}MT^{-2}$.



Supertypes

- *PressureUnit*

Members

Literals

Name	Sym- bol	UN Code	RDL Label	RDL URI
Bar	bar	BAR	BAR	http://data.posccaesar.org/rdl/RDS1314539
Kilopascal	kPa	KPA	KILOPASCAL	http://data.posccaesar.org/rdl/RDS1330559
Megapascal	MPa	MPA	MEGAPASCAL	http://data.posccaesar.org/rdl/RDS1332404
Millibar	mbar	MBR	MILLIBAR	http://data.posccaesar.org/rdl/RDS11617875
Pascal	Pa	PAL	PASCAL	http://data.posccaesar.org/rdl/RDS1338749
PoundForcePerInch-Squared	lbf/in ²	PS	POUND FORCE PER INCH SQUARED	http://data.posccaesar.org/rdl/RDS1341809

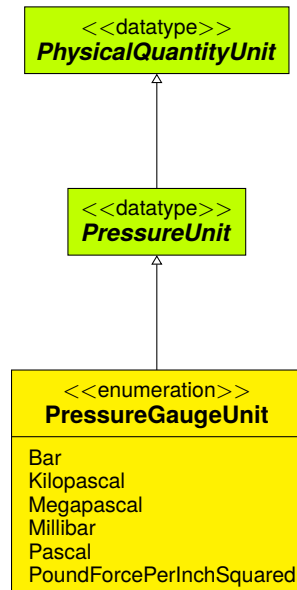
PressureGaugeUnit

Core::PhysicalQuantities::PressureGaugeUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *pressure gauge* with dimension $L^{-1}MT^{-2}$.



Supertypes

- *PressureUnit*

Members

Literals

Name	Sym- bol	UN Code	RDL Label	RDL URI
Bar	bar	BAR	BAR	http://data.posccaesar.org/rdl/RDS1314539
Kilopascal	kPa	KPA	KILOPASCAL	http://data.posccaesar.org/rdl/RDS1330559
Megapascal	MPa	MPA	MEGAPASCAL	http://data.posccaesar.org/rdl/RDS1332404
Millibar	mbar	MBR	MILLIBAR	http://data.posccaesar.org/rdl/RDS11617875
Pascal	Pa	PAL	PASCAL	http://data.posccaesar.org/rdl/RDS1338749
PoundForcePerInch-Squared	lbf/in ²	PS	POUND FORCE PER INCH SQUARED	http://data.posccaesar.org/rdl/RDS1341809

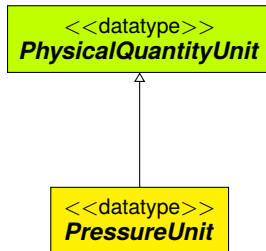
PressureUnit

Core::PhysicalQuantities::PressureUnit

Overview

AbstractDataType

A unit of measurement for a physical quantity of type *pressure* with dimension $L^{-1}MT^{-2}$. PressureUnit has 2 subtypes for different application areas.



Supertypes

- *PhysicalQuantityUnit*

Subtypes

- *PressureAbsoluteUnit*
- *PressureGaugeUnit*

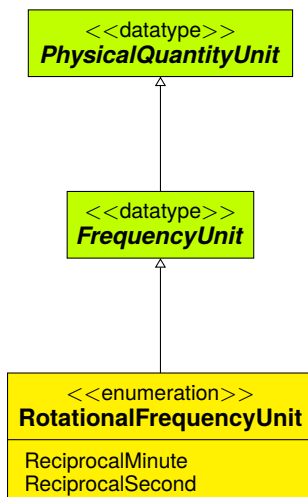
RotationalFrequencyUnit

Core::PhysicalQuantities::RotationalFrequencyUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *rotational frequency* with dimension T^{-1} .



Supertypes

- *FrequencyUnit*

Members

Literals

Name	Sym- bol	UN Code	RDL Label	RDL URI
ReciprocalMinute	min^{-1}	C94	RECIPROCAL MINUTE	http://data.posccaesar.org/rdl/RDS4316851589
ReciprocalSecond	s^{-1}	C97	RECIPROCAL SECOND	http://data.posccaesar.org/rdl/RDS1355489

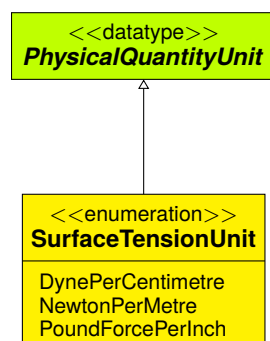
SurfaceTensionUnit

*Core::PhysicalQuantities::*SurfaceTensionUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *surface tension* with dimension MT^{-2} .



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Sym- bol	UN Code	RDL Label	RDL URI
DynePerCentime- tre	dyn/cm	DX	DYNE PER CENTIME- TRE	http://data.posccaesar.org/rdl/RDS1323359
NewtonPerMetre	N/m	4P	NEWTON PER METRE	http://data.posccaesar.org/rdl/RDS1338299
PoundForcePer- Inch	lbf/in	F48	POUND FORCE PER INCH	http://data.posccaesar.org/rdl/RDS4316825046

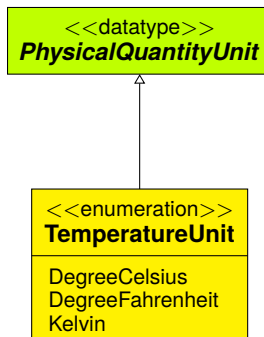
TemperatureUnit

Core::PhysicalQuantities::TemperatureUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *temperature* with dimension Θ .



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Sym- bol	UN Code	RDL Label	RDL URI
DegreeCelsius	°C	CEL	DEGREE CELSIUS	http://data.posccaesar.org/rdl/RDS1322684
DegreeFahren- heit	°F	FAH	DEGREE FAHREN- HEIT	http://data.posccaesar.org/rdl/RDS1322549
Kelvin	K	KEL	KELVIN	http://data.posccaesar.org/rdl/RDS1327904

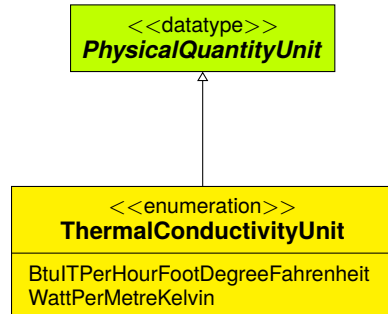
ThermalConductivityUnit

Core::PhysicalQuantities::ThermalConductivityUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *thermal conductivity* with dimension $\text{LM}\Theta^{-1}\text{T}^{-3}$.



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Symbol	UN Code	RDL Label	RDL URI
BtuITPerHourFoot-DegreeFahrenheit	Btu[IT]·ft/(t	J40	BRITISH THERMAL UNIT PER HOUR FOOT DEGREE FAHRENHEIT	http://data.posccaesar.org/rdl/RDS431648930
WattPerMe-treKelvin	W/(m·K)	D53	WATT PER METRE KELVIN	http://data.posccaesar.org/rdl/RDS1348334

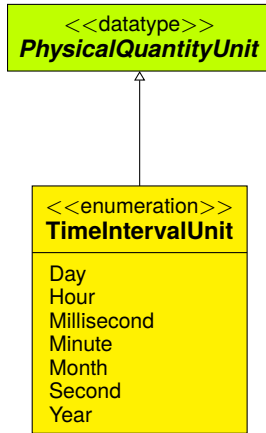
TimeIntervalUnit

Core::PhysicalQuantities::TimeIntervalUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *time interval* with dimension T.



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Symbol	UN Code	RDL Label	RDL URI
Day	d	DAY	DAY	http://data.posccaesar.org/rdl/RDS8945301
Hour	h	HUR	HOUR	http://data.posccaesar.org/rdl/RDS1326734
Millisecond	ms	C26	MILLISECOND	http://data.posccaesar.org/rdl/RDS1336364
Minute	min	MIN	MINUTE	http://data.posccaesar.org/rdl/RDS1336814
Month	mo	MON	MONTH	http://data.posccaesar.org/rdl/RDS1370700781
Second	s	SEC	SECOND	http://data.posccaesar.org/rdl/RDS1342529
Year	y	ANN	YEAR	http://data.posccaesar.org/rdl/RDS4316908552

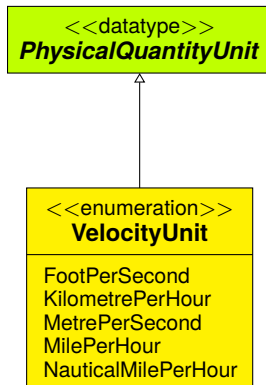
VelocityUnit

Core::PhysicalQuantities::VelocityUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *velocity* with dimension LT^{-1} .



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Sym- bol	UN Code	RDL Label	RDL URI
FootPerSecond	ft/s	FS	FOOT PER SECOND	http://data.posccaesar.org/rdl/RDS1324439
KilometrePerHour	km/h	KMH	KILOMETRE PER HOUR	http://data.posccaesar.org/rdl/RDS1330289
MetrePerSecond	m/s	MTS	METRE PER SECOND	http://data.posccaesar.org/rdl/RDS1354499
MilePerHour	mile/h	HM	MILE PER HOUR	http://data.posccaesar.org/rdl/RDS1334204
NauticalMilePer-Hour	kn	KNT	KNOT	http://data.posccaesar.org/rdl/RDS1331099

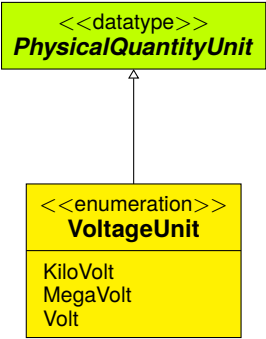
VoltageUnit

Core::PhysicalQuantities::VoltageUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *voltage* with dimension $L^2MT^{-3}I^{-1}$.



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Symbol	UN Code	RDL Label	RDL URI
KiloVolt	kV	KVT	KILOVOLT	http://data.posccaesar.org/rdl/RDS1359653041
MegaVolt	MV	B78	MEGAVOLT	http://data.posccaesar.org/rdl/RDS1359661910
Volt	V	VLT	VOLT	http://data.posccaesar.org/rdl/RDS1347974

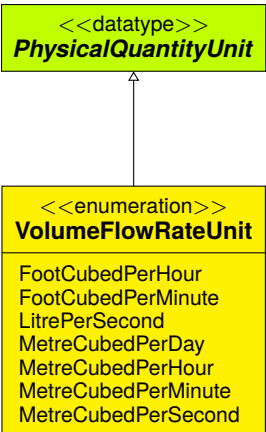
VolumeFlowRateUnit

*Core::PhysicalQuantities::*VolumeFlowRateUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *volume flow rate* with dimension L^3T^{-1} .



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Sym- bol	UN Code	RDL Label	RDL URI
FootCubedPerHour	ft ³ /h	2K	FOOT CUBED PER HOUR	http://data.posccaesar.org/rdl/RDS1320029
FootCubedPerMinute	ft ³ /min	2L	FOOT CUBED PER MINUTE	http://data.posccaesar.org/rdl/RDS1320164
LitrePerSecond	l/s	G51	LITRE PER SECOND	http://data.posccaesar.org/rdl/RDS1331369
MetreCubedPerDay	m ³ /d	G52	METRE CUBED PER DAY	http://data.posccaesar.org/rdl/RDS1320839
MetreCubedPerHour	m ³ /h	MQH	METRE CUBED PER HOUR	http://data.posccaesar.org/rdl/RDS1321064
MetreCubedPerMinute	m ³ /min	G53	METRE CUBED PER MINUTE	http://data.posccaesar.org/rdl/RDS1349909
MetreCubedPerSecond	m ³ /s	MQS	METRE CUBED PER SECOND	http://data.posccaesar.org/rdl/RDS1321379

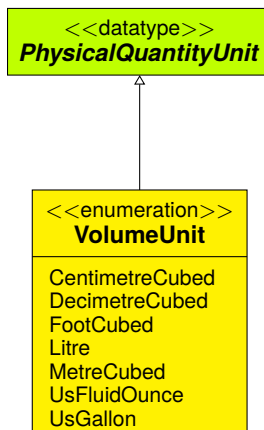
VolumeUnit

Core::PhysicalQuantities::VolumeUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *volume* with dimension L³.



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Symbol	UN Code	RDL Label	RDL URI
Centimetre-Cubed	cm ³	CMQ	CENTIMETRE CUBED	http://data.posccaesar.org/rdl/RDS1357874
DecimetreCubed	dm ³	DMQ	DECIMETRE CUBED	http://data.posccaesar.org/rdl/RDS1319174
FootCubed	ft ³	FTQ	FOOT CUBED	http://data.posccaesar.org/rdl/RDS1319669
Litre	l	LTR	LITRE	http://data.posccaesar.org/rdl/RDS1331144
MetreCubed	m ³	MTQ	METRE CUBED	http://data.posccaesar.org/rdl/RDS1349099
UsFluidOunce	fl oz (US)	OZA	US FLUID OUNCE	http://data.posccaesar.org/rdl/RDS11619315
UsGallon	gal (US)	GLL	US GALLON	http://data.posccaesar.org/rdl/RDS11615400

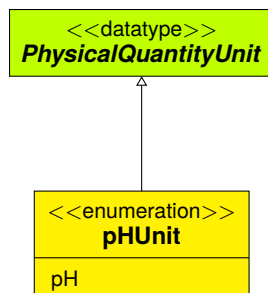
pHUnit

Core::PhysicalQuantities::pHUnit

Overview

Enumeration

A unit of measurement for a physical quantity of type *pH* with dimension 1.



Supertypes

- *PhysicalQuantityUnit*

Members

Literals

Name	Symbol	UN Code	RDL Label	RDL URI
pH	pH	Q30	pH	http://data.posccaesar.org/rdl/RDS1060066911

Core::PhysicalQuantities

Overview

Package

Data types for representing physical quantities such as the area of a surface and the frequency of a repeating event.

Members

Data Types

Name	Type
<i>AreaUnit</i>	Enumeration
<i>DensityUnit</i>	Enumeration
<i>DynamicViscosityUnit</i>	Enumeration
<i>ElectricCurrentUnit</i>	Enumeration
<i>ElectricalFrequencyUnit</i>	Enumeration
<i>EnergyDensityUnit</i>	Enumeration
<i>EnergyUnit</i>	Enumeration
<i>ForceUnit</i>	Enumeration
<i>FrequencyUnit</i>	AbstractDataType
<i>HeatCapacityUnit</i>	Enumeration
<i>HeatTransferCoefficientUnit</i>	Enumeration
<i>HeatTransferResistanceUnit</i>	Enumeration
<i>KinematicViscosityUnit</i>	Enumeration
<i>LengthUnit</i>	Enumeration
<i>MagneticFieldIntensityUnit</i>	Enumeration
<i>MagneticFluxDensityUnit</i>	Enumeration
<i>MassConcentrationUnit</i>	Enumeration
<i>MassFlowRateUnit</i>	Enumeration
<i>MassSpecificEnergyUnit</i>	Enumeration
<i>MassSpecificHeatCapacityUnit</i>	Enumeration
<i>MassUnit</i>	Enumeration
<i>MoleConcentrationUnit</i>	Enumeration
<i>MoleFlowRateUnit</i>	Enumeration
<i>MoleSpecificEnergyUnit</i>	Enumeration
<i>MomentOfForceUnit</i>	Enumeration
<i>NumberPerTimeIntervalUnit</i>	Enumeration
<i>ParticleSizeUnit</i>	Enumeration
<i>PercentageUnit</i>	Enumeration
<i>PhysicalQuantity</i>	Aggregated Data Type
<i>PhysicalQuantityUnit</i>	AbstractDataType
<i>PhysicalQuantityVector</i>	Aggregated Data Type

continues on next page

Table 3 – continued from previous page

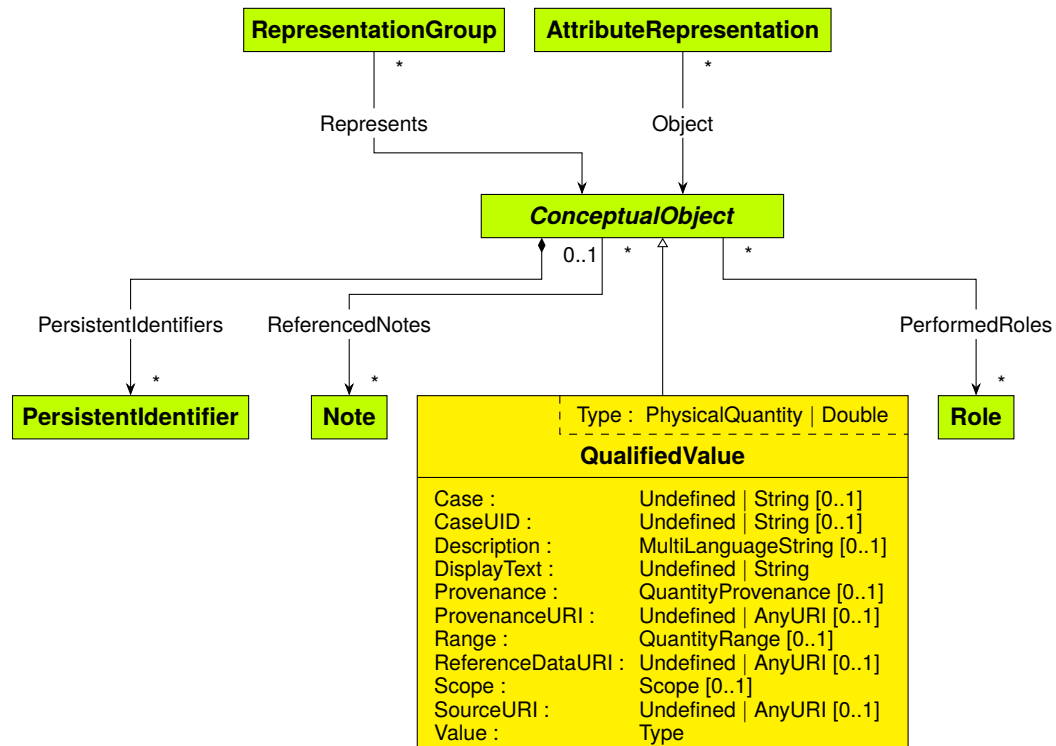
Name	Type
<i>PowerUnit</i>	Enumeration
<i>PressureAbsoluteUnit</i>	Enumeration
<i>PressureGaugeUnit</i>	Enumeration
<i>PressureUnit</i>	AbstractDataType
<i>RotationalFrequencyUnit</i>	Enumeration
<i>SurfaceTensionUnit</i>	Enumeration
<i>TemperatureUnit</i>	Enumeration
<i>ThermalConductivityUnit</i>	Enumeration
<i>TimeIntervalUnit</i>	Enumeration
<i>VelocityUnit</i>	Enumeration
<i>VoltageUnit</i>	Enumeration
<i>VolumeFlowRateUnit</i>	Enumeration
<i>VolumeUnit</i>	Enumeration
<i>pHUnit</i>	Enumeration

3.1.9 QualifiedValue

Core::QualifiedValue

Overview

Concrete Class



Supertypes

- *ConceptualObject*

Members

Type Parameters

Name	Default Type
<i>Type</i>	<i>PhysicalQuantity</i> <i>Double</i>

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Case</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>CaseUID</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>Description</i>	0..1	<i>MultiLanguageString</i>	no	no
<i>DisplayText</i>	1	<i>Undefined</i> <i>String</i>	no	no
<i>Provenance</i>	0..1	<i>QuantityProvenance</i>	no	no
<i>ProvenanceURI</i>	0..1	<i>Undefined</i> <i>AnyURI</i>	no	no
<i>Range</i>	0..1	<i>QuantityRange</i>	no	no
<i>ReferenceDataURI</i>	0..1	<i>Undefined</i> <i>AnyURI</i>	no	no
<i>Scope</i>	0..1	<i>Scope</i>	no	no
<i>SourceURI</i>	0..1	<i>Undefined</i> <i>AnyURI</i>	no	no
<i>Value</i>	1	<i>Type</i>	no	no

Case

Data Property

The identifier string for the case definition.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

Example

"Special case"

CaseUID

Data Property

Globally unique identifier to the case referred to.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

Example

"1723-001"

Description**Data Property**

A description.

Type: *MultiLanguageString*

Multiplicity: 0..1

Ordered: no

Unique: no

Example

```
MultiLanguageString(
  SingleLanguageStrings = [
    SingleLanguageString(
      Language = "en",
      Value = "Lower limit ambient operating temperature."),
    SingleLanguageString(
      Language = "de",
      Value = "Untergrenze für die Umgebungstemperatur bei Betrieb.")]
```

DisplayText**Data Property**

The display text of the *QualifiedValue*, typically shown in a diagram.

Type: *Undefined* | *String*

Multiplicity: 1

Ordered: no

Unique: no

Example

"Lower limit ambient operating temperature."

Provenance**Data Property**

Type: *QuantityProvenance*

Multiplicity: 0..1

Ordered: no

Unique: no

Example

Specified

ProvenanceURI

Data Property

Type: *Undefined* | *AnyURI*

Multiplicity: 0..1

Ordered: no

Unique: no

Example

"abc://exampleURI"

Range

Data Property

Type: *QuantityRange*

Multiplicity: 0..1

Ordered: no

Unique: no

Example

LowerLimit

ReferenceDataURI

Data Property

A reference to the definition of the *QualifiedValue*.

Type: *Undefined* | *AnyURI*

Multiplicity: 0..1

Ordered: no

Unique: no

Example

"abc://referenceURI"

Scope

Data Property

Type: *Scope*

Multiplicity: 0..1

Ordered: no

Unique: no

Example

Operating

SourceURI

Data Property

A reference to the source of the value.

Type: *Undefined* | *AnyURI*

Multiplicity: 0..1

Ordered: no

Unique: no

Example

"abc://sourceURI"

Type

Data Type Parameter

Default Type: *PhysicalQuantity* | *Double*

Value

Data Property

Type: *Type*

Multiplicity: 1

Ordered: no

Unique: no

Example

PhysicalQuantity(

Unit = *Kelvin*,

Value = 230.2)

Example in DEXPI XML

```
<Object type="Core/QualifiedValue">
  <Data property="Case">
    <String>Special case</String>
  </Data>
  <Data property="CaseUID">
    <String>1723-001</String>
  </Data>
  <Data property="Description">
```

(continues on next page)

(continued from previous page)

```

<AggregatedDataValue type="Core/DataTypes.MultiLanguageString">
  <Data property="SingleLanguageStrings">
    <AggregatedDataValue type="Core/DataTypes.SingleLanguageString">
      <Data property="Language">
        <String>en</String>
      </Data>
      <Data property="Value">
        <String>Lower limit ambient operating temperature.</String>
      </Data>
    </AggregatedDataValue>
    <AggregatedDataValue type="Core/DataTypes.SingleLanguageString">
      <Data property="Language">
        <String>de</String>
      </Data>
      <Data property="Value">
        <String>Untergrenze f&#252;r die Umgebungstemperatur bei Betrieb.</String>
      </Data>
    </AggregatedDataValue>
  </Data>
</AggregatedDataValue>
<Data property="DisplayText">
  <String>Lower limit ambient operating temperature.</String>
</Data>
<Data property="Provenance">
  <DataReference data="Core/DataTypes.QuantityProvenance.Specified"/>
</Data>
<Data property="ProvenanceURI">
  <String>abc://exampleURI</String>
</Data>
<Data property="Range">
  <DataReference data="Core/DataTypes.QuantityRange.LowerLimit"/>
</Data>
<Data property="ReferenceDataURI">
  <String>abc://referenceURI</String>
</Data>
<Data property="Scope">
  <DataReference data="Core/DataTypes.Scope.Operating"/>
</Data>
<Data property="SourceURI">
  <String>abc://sourceURI</String>
</Data>
<Data property="Value">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.TemperatureUnit.Kelvin"/>
    </Data>
    <Data property="Value">
      <Double>230.2</Double>
    </Data>
  </AggregatedDataValue>
</Data>

```

(continues on next page)

(continued from previous page)

</Object>

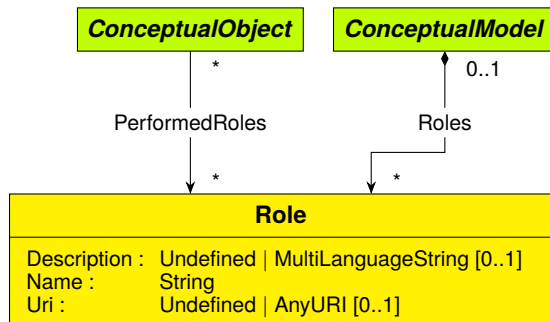
3.1.10 Role

Core::Role

Overview

Concrete Class

The role fulfilled by a *Core.ConceptualObject*.



Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Description</i>	0..1	<i>Undefined MultiLanguageString</i>	no	no
<i>Name</i>	1	<i>String</i>	no	no
<i>Uri</i>	0..1	<i>Undefined AnyURI</i>	no	no

Description

Data Property

A description of the *Role*.

Type: *Undefined | MultiLanguageString*

Multiplicity: 0..1

Ordered: no

Unique: no

Name

Data Property

The name of the *Role*.

Type: *String*

Multiplicity: 1

Ordered: no

Unique: no

Example

"Bottom Product"

Uri

Data Property

The URI of the *Role*.

Type: *Undefined* | *AnyURI*

Multiplicity: 0..1

Ordered: no

Unique: no

Example

"http://www.exampleProcessOntology/BottomProduct"

Example in DEXPI XML

```
<Object type="Core/Role">
  <Data property="Name">
    <String>Bottom Product</String>
  </Data>
  <Data property="Uri">
    <String>http://www.exampleProcessOntology/BottomProduct</String>
  </Data>
</Object>
```

Core

3.1.11 Overview

Model

URI: <https://data.dexpi.org/models/2.0.0/Core.xml>

Core.xml

General types for the representation of engineering information and associated graphical representations.

3.1.12 Members

Packages

Name
<i>DataTypes</i>
<i>Diagram</i>
<i>PhysicalQuantities</i>

Classes

Name	Type
<i>ConceptualModel</i>	Abstract Class
<i>ConceptualObject</i>	Abstract Class
<i>EngineeringModel</i>	Concrete Class
<i>Note</i>	Concrete Class
<i>PersistentIdentifier</i>	Concrete Class
<i>QualifiedValue</i>	Concrete Class
<i>Role</i>	Concrete Class

3.2 Plant

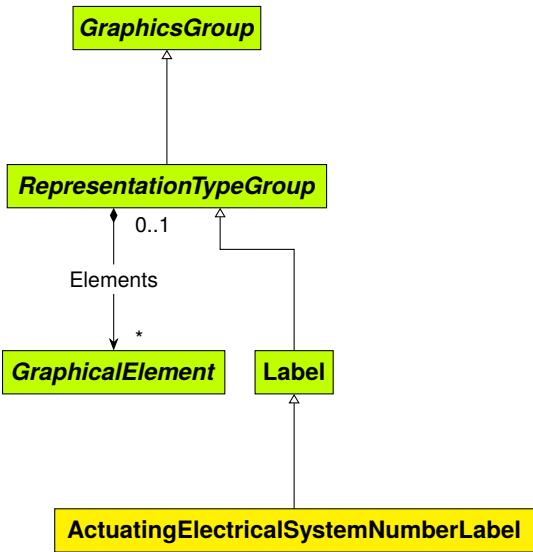
3.2.1 Diagram

ActuatingElectricalSystemNumberLabel

Plant::Diagram::ActuatingElectricalSystemNumberLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

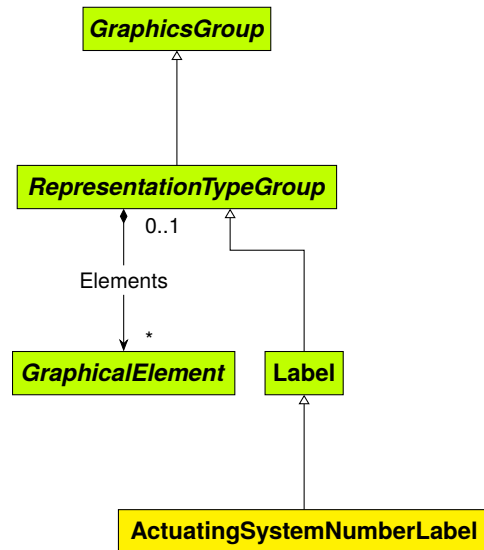
```
<Object type="Plant/Diagram.ActuatingElectricalSystemNumberLabel"/>
```

ActuatingSystemNumberLabel

Plant::Diagram::ActuatingSystemNumberLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

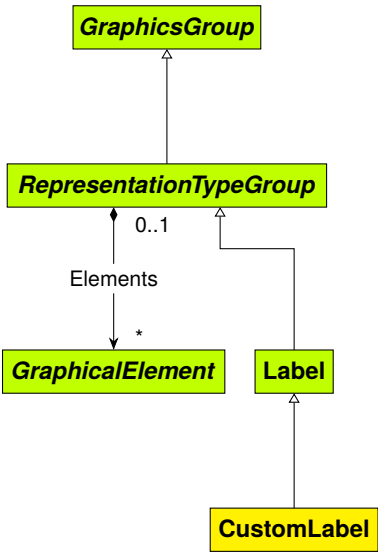
```
<Object type="Plant/Diagram.ActuatingSystemNumberLabel"/>
```

CustomLabel

Plant::Diagram::CustomLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

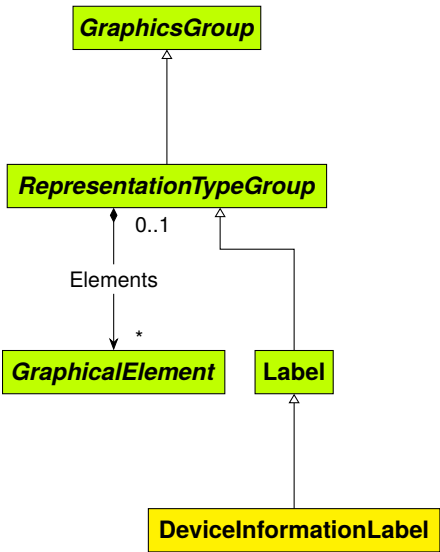
```
<Object type="Plant/Diagram.CustomLabel"/>
```

DeviceInformationLabel

Plant::Diagram::DeviceInformationLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

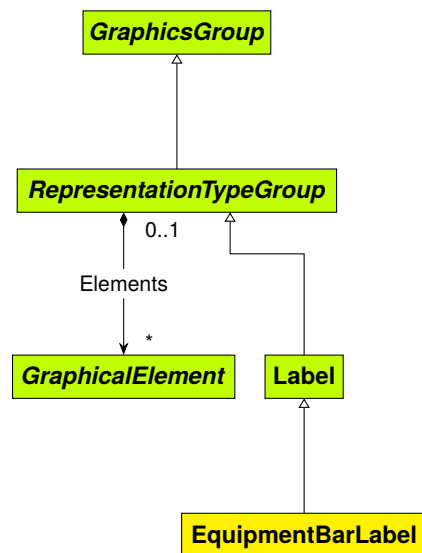
```
<Object type="Plant/Diagram.DeviceInformationLabel"/>
```

EquipmentBarLabel

Plant::Diagram::EquipmentBarLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

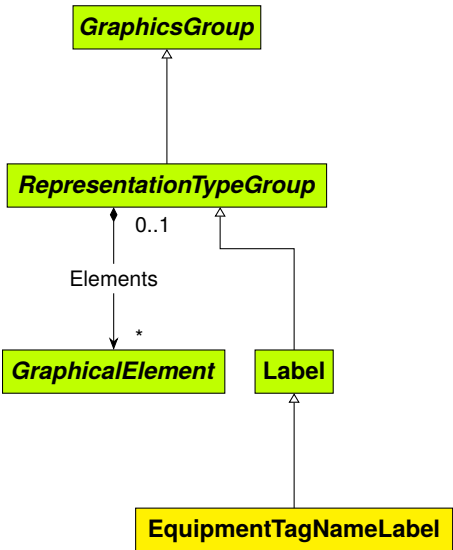
```
<Object type="Plant/Diagram.EquipmentBarLabel"/>
```

EquipmentTagNameLabel

Plant::Diagram::EquipmentTagNameLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

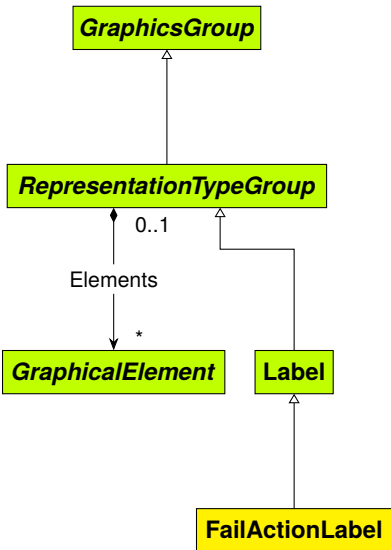
```
<Object type="Plant/Diagram.EquipmentTagNameLabel"/>
```

FailActionLabel

Plant::Diagram::FailActionLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

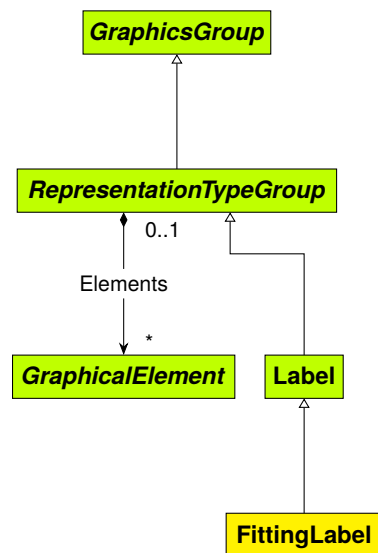
```
<Object type="Plant/Diagram.FailActionLabel"/>
```

FittingLabel

Plant::Diagram::FittingLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

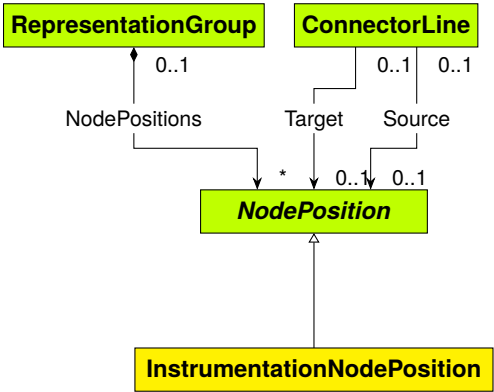
```
<Object type="Plant/Diagram.FittingLabel"/>
```

InstrumentationNodePosition

Plant::Diagram::InstrumentationNodePosition

Overview

Concrete Class



Supertypes

- *NodePosition*

Example in DEXPI XML

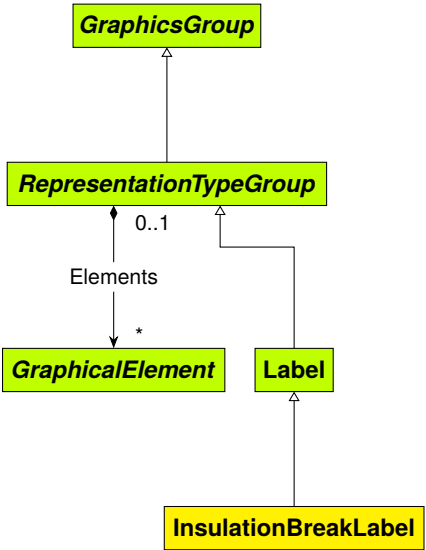
```
<Object type="Plant/Diagram.InstrumentationNodePosition"/>
```

InsulationBreakLabel

Plant::Diagram::InsulationBreakLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

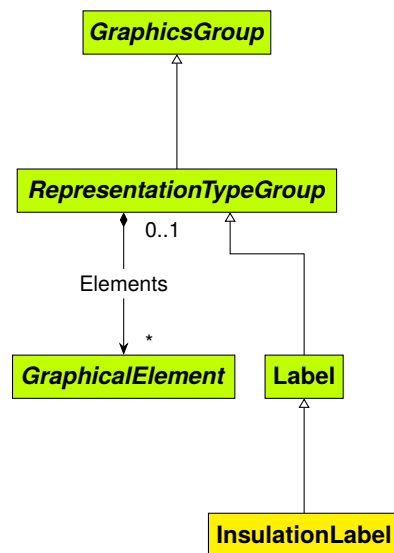
```
<Object type="Plant/Diagram.InsulationBreakLabel"/>
```

InsulationLabel

Plant::Diagram::InsulationLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

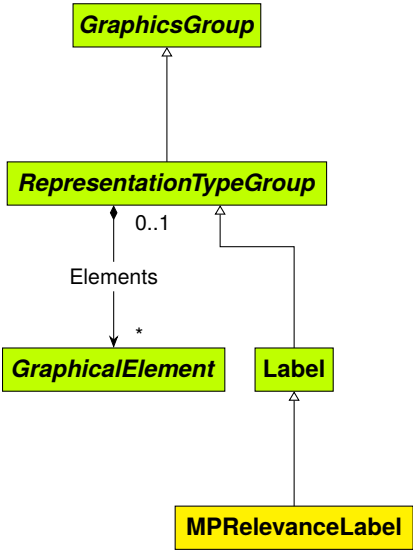
```
<Object type="Plant/Diagram.InsulationLabel"/>
```

MPRelevanceLabel

Plant::Diagram::MPRelevanceLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

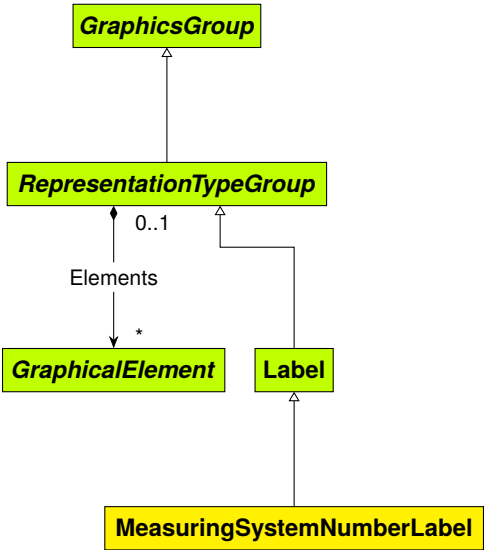
```
<Object type="Plant/Diagram.MPRelevanceLabel"/>
```

MeasuringSystemNumberLabel

Plant::Diagram::MeasuringSystemNumberLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

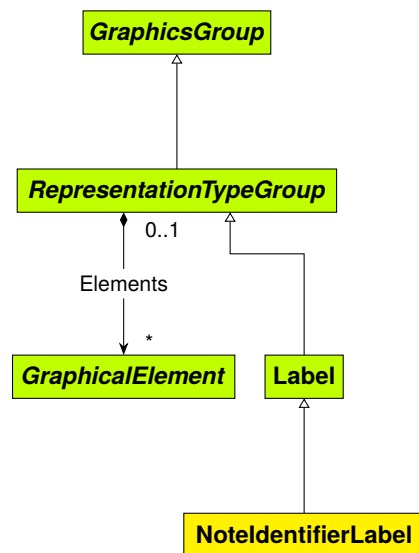
```
<Object type="Plant/Diagram.MeasuringSystemNumberLabel"/>
```

NoteIdentifierLabel

Plant::Diagram::NoteIdentifierLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

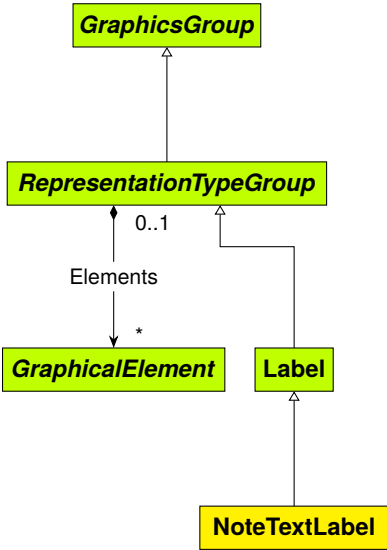
```
<Object type="Plant/Diagram.NoteIdentifierLabel"/>
```

NoteTextLabel

Plant::Diagram::NoteTextLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

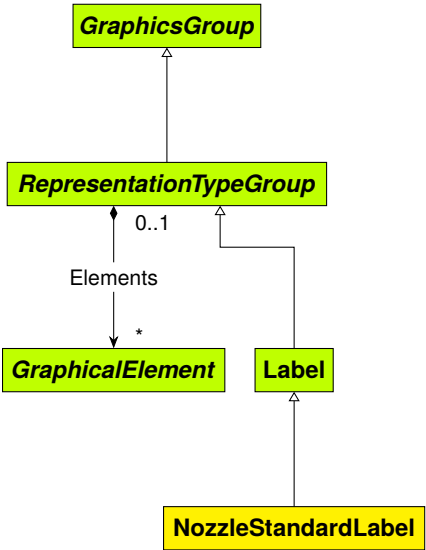
```
<Object type="Plant/Diagram.NoteTextLabel"/>
```

NozzleStandardLabel

Plant::Diagram::NozzleStandardLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

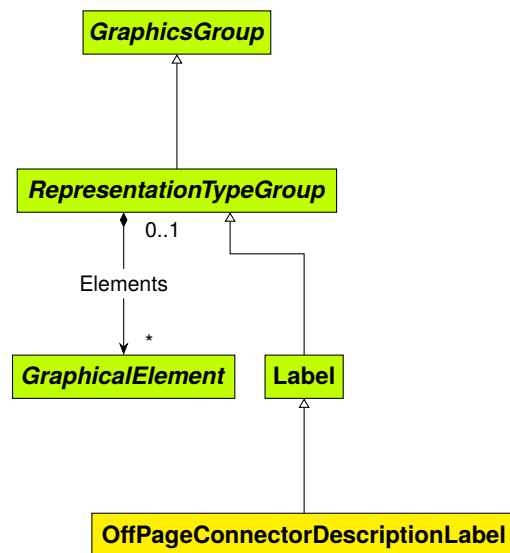
```
<Object type="Plant/Diagram.NozzleStandardLabel"/>
```

OffPageConnectorDescriptionLabel

*Plant::Diagram::*OffPageConnectorDescriptionLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

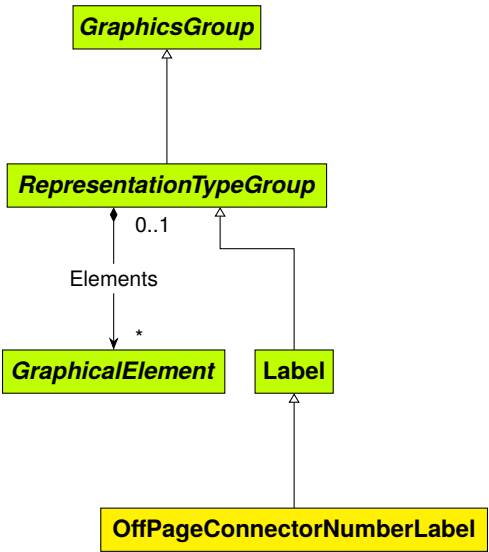
```
<Object type="Plant/Diagram.OffPageConnectorDescriptionLabel"/>
```

OffPageConnectorNumberLabel

*Plant::Diagram::*OffPageConnectorNumberLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

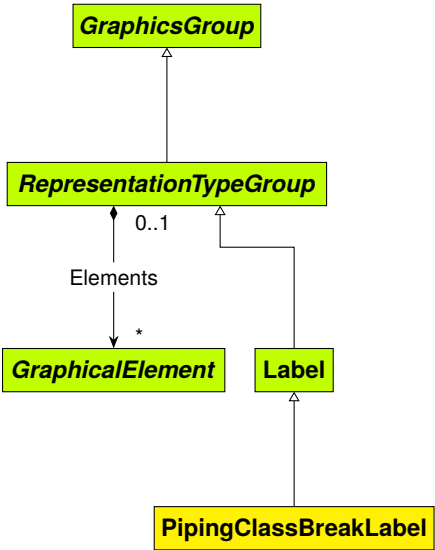
```
<Object type="Plant/Diagram.OffPageConnectorNumberLabel"/>
```

PipingClassBreakLabel

Plant::Diagram::PipingClassBreakLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

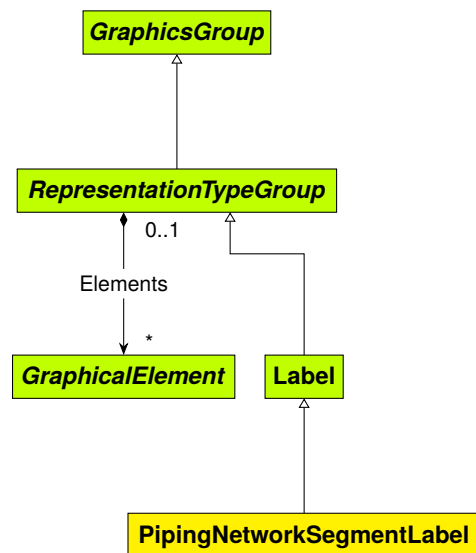
```
<Object type="Plant/Diagram.PipingClassBreakLabel"/>
```

PipingNetworkSegmentLabel

Plant::Diagram::PipingNetworkSegmentLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

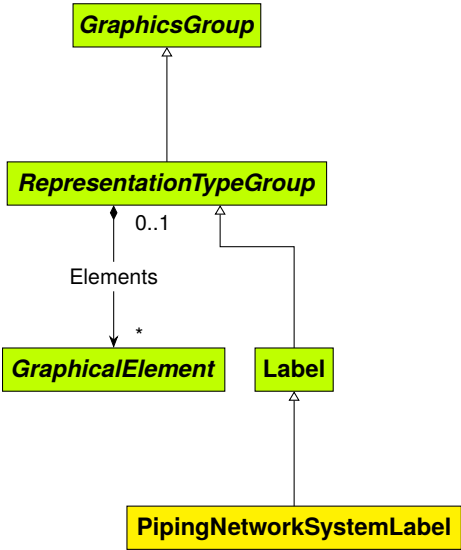
```
<Object type="Plant/Diagram.PipingNetworkSegmentLabel"/>
```

PipingNetworkSystemLabel

Plant::Diagram::PipingNetworkSystemLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

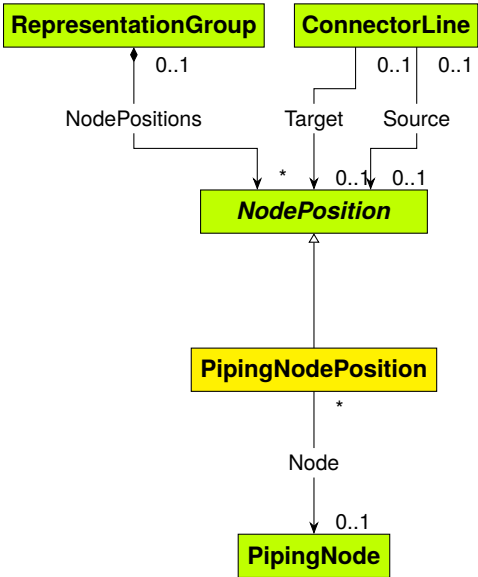
```
<Object type="Plant/Diagram.PipingNetworkSystemLabel"/>
```

PipingNodePosition

Plant::Diagram::PipingNodePosition

Overview

Concrete Class



Supertypes

- *NodePosition*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Node</i>	0..1	<i>PipingNode</i>	no	no	*

Node

Reference Property

Type: *PipingNode*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

Example in DEXPI XML

```
<Object type="Plant/Diagram.PipingNodePosition"/>
```

PlantMetaData

Plant::Diagram::PlantMetaData

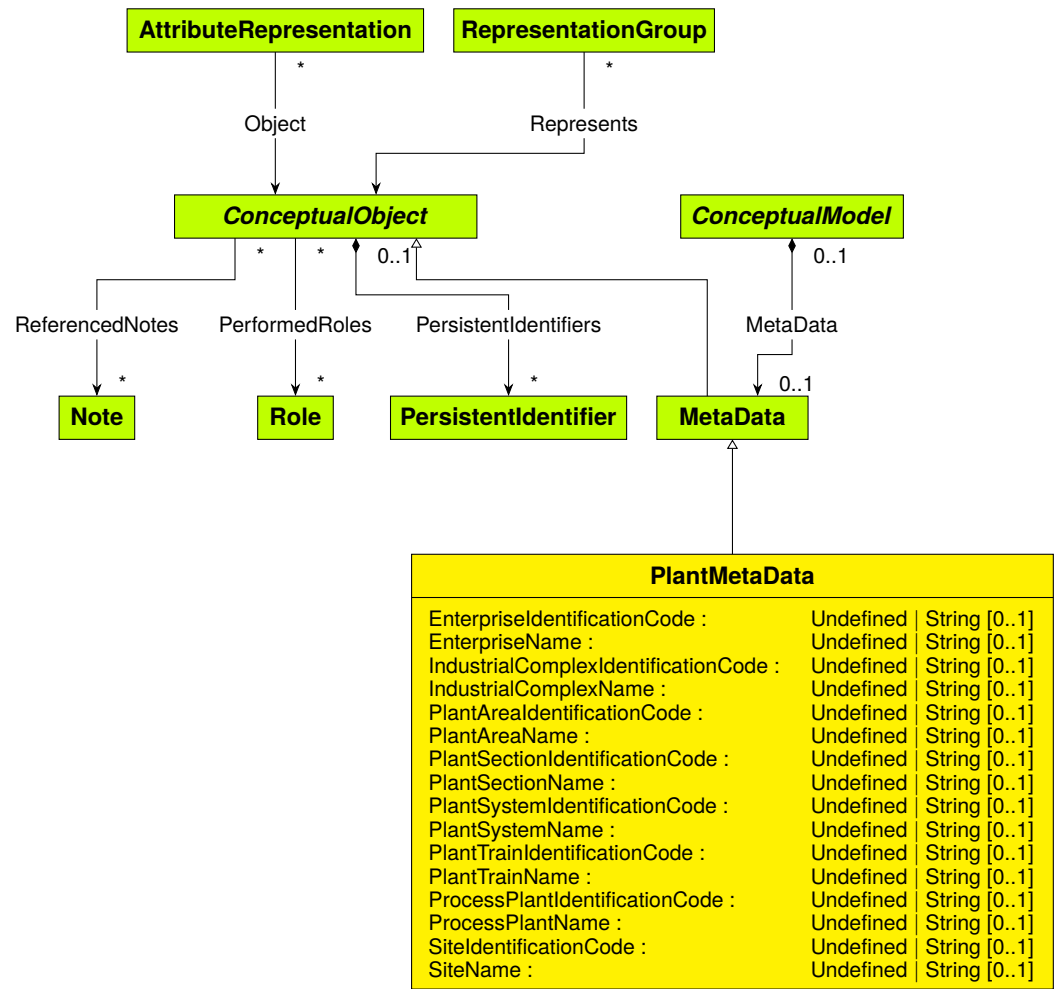
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/PlantMetaData>

PLANT META DATA

Meta data about a Core.Container with a *PlantModel*.



Supertypes

- *Metadata*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>EnterpriseIdentificationCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>EnterpriseName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>IndustrialComplexIdentificationCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>IndustrialComplexName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>PlantAreaIdentificationCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>PlantAreaName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>PlantSectionIdentificationCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>PlantSectionName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>PlantSystemIdentificationCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>PlantSystemName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>PlantTrainIdentificationCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>PlantTrainName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>ProcessPlantIdentificationCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>ProcessPlantName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>SiteIdentificationCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>SiteName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

EnterpriseIdentificationCode

Data Property

The identification code of the enterprise.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/EnterpriseIdentificationCodeAssignmentClass>

ENTERPRISE IDENTIFICATION CODE ASSIGNMENT CLASS

Example

"C1248"

EnterpriseName

Data Property

The name of the enterprise.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/EnterpriseNameAssignmentClass>

ENTERPRISE NAME ASSIGNMENT CLASS

Example

"CompAny Ltd."

IndustrialComplexIdentificationCode**Data Property**

The identification code of the industrial complex.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/IndustrialComplexIdentificationCodeAssignmentClass>

INDUSTRIAL COMPLEX IDENTIFICATION CODE ASSIGNMENT CLASS

Example

"I-Chain"

IndustrialComplexName**Data Property**

The name of the industrial complex.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/IndustrialComplexNameAssignmentClass>

INDUSTRIAL COMPLEX NAME ASSIGNMENT CLASS

Example

"Isophorone Chain"

PlantAreaIdentificationCode**Data Property**

The identification code of the plant area according to ISA-95.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PlantAreaIdentificationCodeAssignmentClass>

PLANT AREA IDENTIFICATION CODE ASSIGNMENT CLASS

Example

"F4"

PlantAreaName

Data Property

The name of the plant area according to ISA-95.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/AreaIsa95NameAssignmentClass>

AREA ISA95 NAME ASSIGNMENT CLASS

Example

"Area F4"

PlantSectionIdentificationCode

Data Property

The identification code of the plant section.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PlantSectionIdentificationCodeAssignmentClass>

PLANT SECTION IDENTIFICATION CODE ASSIGNMENT CLASS

Example

"10"

PlantSectionName

Data Property

The name of the plant section.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PlantSectionNameAssignmentClass>

PLANT SECTION NAME ASSIGNMENT CLASS

Example

"Utilities"

PlantSystemIdentificationCode**Data Property**

The identification code of the plant system.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PlantSystemIdentificationCodeAssignmentClass>

PLANT SYSTEM IDENTIFICATION CODE ASSIGNMENT CLASS

Example

"X123"

PlantSystemName**Data Property**

The name of the plant system.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PlantSystemNameAssignmentClass>

PLANT SYSTEM NAME ASSIGNMENT CLASS

Example

"System X123"

PlantTrainIdentificationCode**Data Property**

The identification code of the plant train.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PlantTrainIdentificationCodeAssignmentClass>

PLANT TRAIN IDENTIFICATION CODE ASSIGNMENT CLASS

Example

"T456"

PlantTrainName

Data Property

The name of the plant train.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PlantTrainNameAssignmentClass>

PLANT TRAIN NAME ASSIGNMENT CLASS

Example

"Train T456"

ProcessPlantIdentificationCode

Data Property

The identification code of the process plant.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ProcessPlantIdentificationCodeAssignmentClass>

PROCESS PLANT IDENTIFICATION CODE ASSIGNMENT CLASS

Example

"ABC"

ProcessPlantName

Data Property

The name of the process plant.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ProcessPlantNameAssignmentClass>

PROCESS PLANT NAME ASSIGNMENT CLASS

Example

"ABC Plant"

SiteIdentificationCode**Data Property**

The identification code of the site.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SiteIdentificationCodeAssignmentClass>

SITE IDENTIFICATION CODE ASSIGNMENT CLASS

Example

"DC"

SiteName**Data Property**

The name of the site.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SiteNameAssignmentClass>

SITE NAME ASSIGNMENT CLASS

Example

"Dexpi City"

Example in DEXPI XML

```
<Object type="Plant/Diagram.PlantMetaData">
  <Data property="EnterpriseIdentificationCode">
    <String>C1248</String>
  </Data>
  <Data property="EnterpriseName">
    <String>CompAny Ltd.</String>
  </Data>
  <Data property="IndustrialComplexIdentificationCode">
    <String>I-Chain</String>
  </Data>
```

(continues on next page)

(continued from previous page)

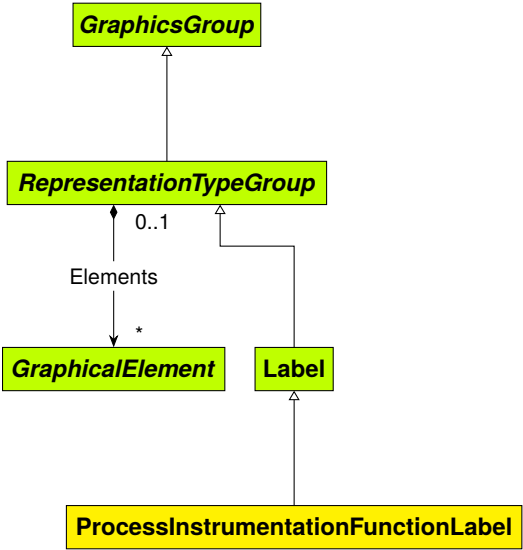
```
<Data property="IndustrialComplexName">
  <String>Isophorone Chain</String>
</Data>
<Data property="PlantAreaIdentificationCode">
  <String>F4</String>
</Data>
<Data property="PlantAreaName">
  <String>Area F4</String>
</Data>
<Data property="PlantSectionIdentificationCode">
  <String>10</String>
</Data>
<Data property="PlantSectionName">
  <String>Utilities</String>
</Data>
<Data property="PlantSystemIdentificationCode">
  <String>X123</String>
</Data>
<Data property="PlantSystemName">
  <String>System X123</String>
</Data>
<Data property="PlantTrainIdentificationCode">
  <String>T456</String>
</Data>
<Data property="PlantTrainName">
  <String>Train T456</String>
</Data>
<Data property="ProcessPlantIdentificationCode">
  <String>ABC</String>
</Data>
<Data property="ProcessPlantName">
  <String>ABC Plant</String>
</Data>
<Data property="SiteIdentificationCode">
  <String>DC</String>
</Data>
<Data property="SiteName">
  <String>Dexpi City</String>
</Data>
</Object>
```

ProcessInstrumentationFunctionLabel

Plant::Diagram::ProcessInstrumentationFunctionLabel

Overview

Concrete Class



Supertypes

- *Label*

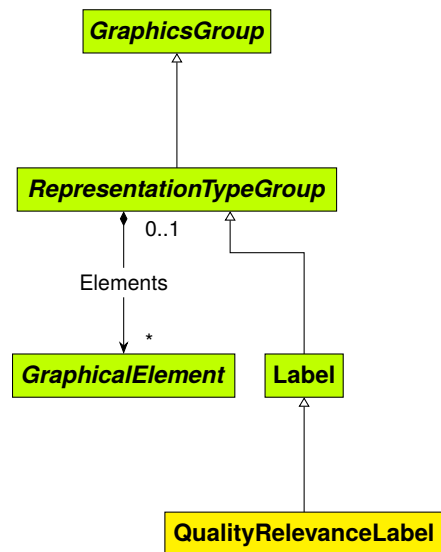
Example in DEXPI XML

```
<Object type="Plant/Diagram.ProcessInstrumentationFunctionLabel"/>
```

QualityRelevanceLabel

Plant::Diagram::QualityRelevanceLabel

Overview

Concrete Class**Supertypes**

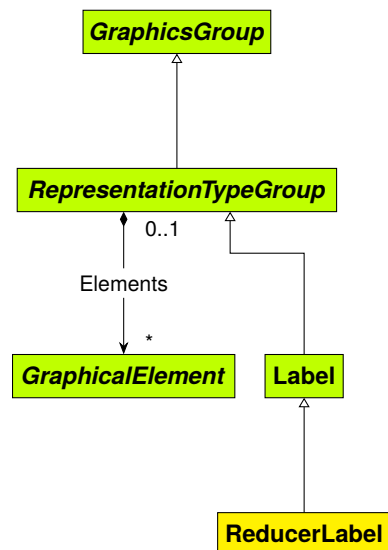
- *Label*

Example in DEXPI XML

```
<Object type="Plant/Diagram.QualityRelevanceLabel"/>
```

ReducerLabel

Plant::Diagram::ReducerLabel

Overview**Concrete Class**

Supertypes

- *Label*

Example in DEXPI XML

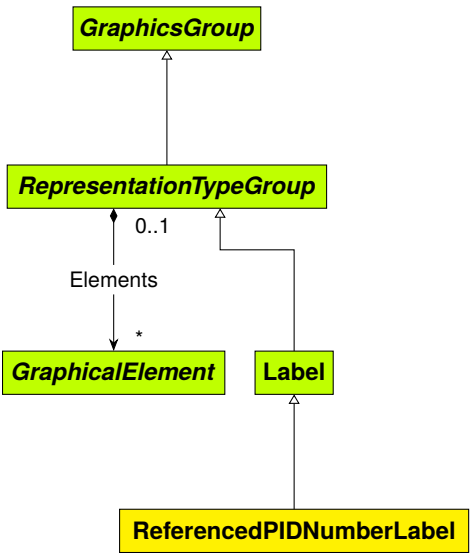
```
<Object type="Plant/Diagram.ReducerLabel"/>
```

ReferencedPIDNumberLabel

Plant::Diagram::ReferencedPIDNumberLabel

Overview

Concrete Class



Supertypes

- *Label*

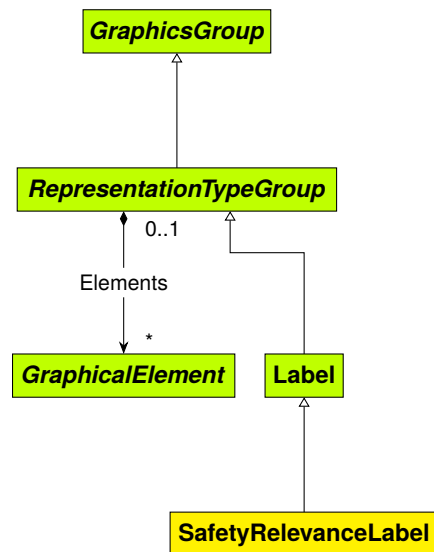
Example in DEXPI XML

```
<Object type="Plant/Diagram.ReferencedPIDNumberLabel"/>
```

SafetyRelevanceLabel

Plant::Diagram::SafetyRelevanceLabel

Overview

Concrete Class**Supertypes**

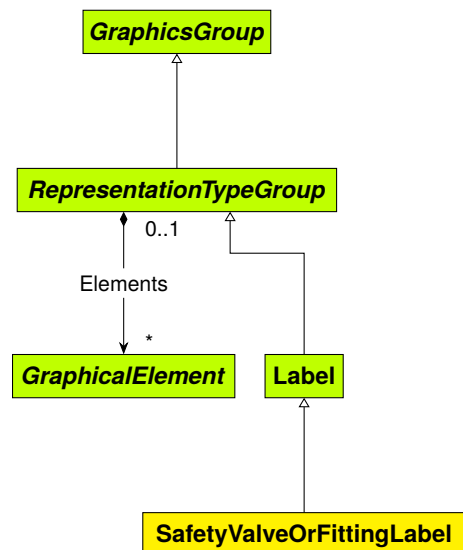
- *Label*

Example in DEXPI XML

```
<Object type="Plant/Diagram.SafetyRelevanceLabel"/>
```

SafetyValveOrFittingLabel

Plant::Diagram::SafetyValveOrFittingLabel

Overview**Concrete Class**

Supertypes

- *Label*

Example in DEXPI XML

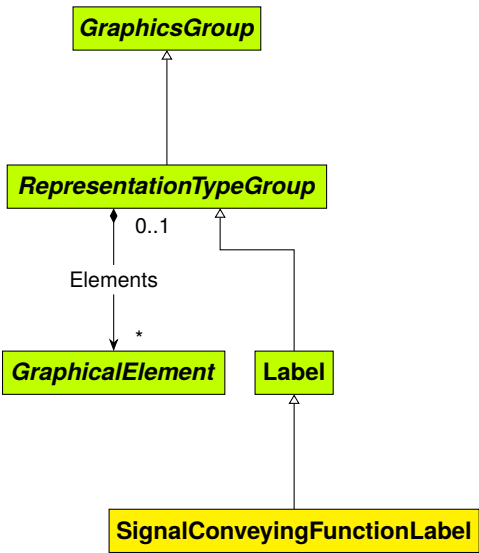
```
<Object type="Plant/Diagram.SafetyValveOrFittingLabel"/>
```

SignalConveyingFunctionLabel

Plant::Diagram::SignalConveyingFunctionLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

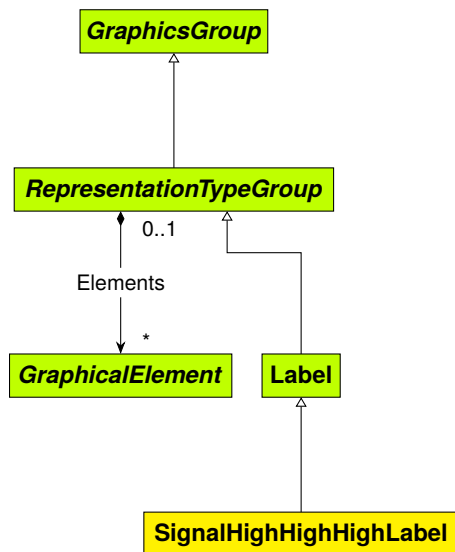
```
<Object type="Plant/Diagram.SignalConveyingFunctionLabel"/>
```

SignalHighHighHighLabel

Plant::Diagram::SignalHighHighHighLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

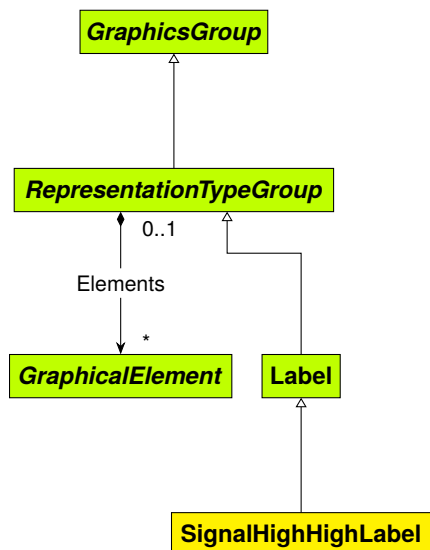
```
<Object type="Plant/Diagram.SignalHighHighHighLabel"/>
```

SignalHighHighLabel

Plant::Diagram::SignalHighHighLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

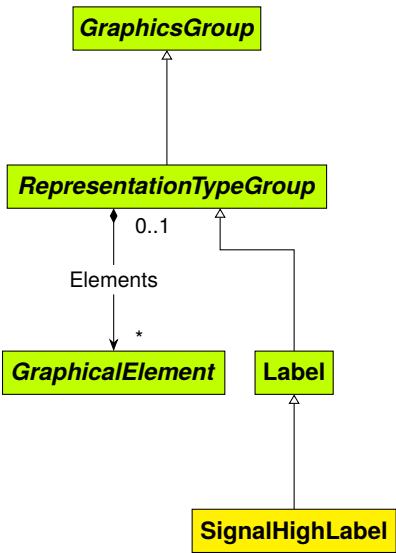
```
<Object type="Plant/Diagram.SignalHighHighLabel"/>
```

SignalHighLabel

Plant::Diagram::SignalHighLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

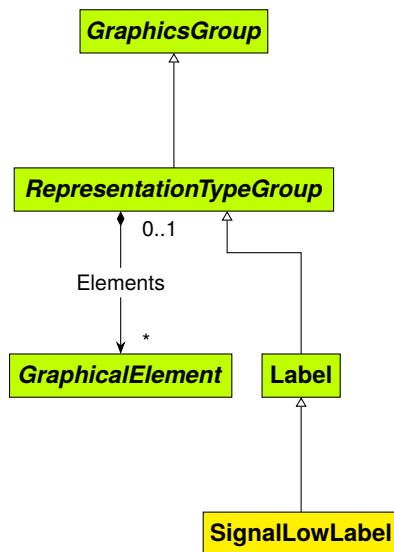
```
<Object type="Plant/Diagram.SignalHighLabel"/>
```

SignalLowLabel

Plant::Diagram::SignalLowLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

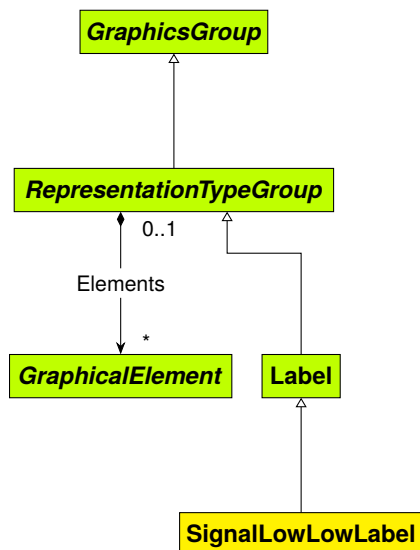
```
<Object type="Plant/Diagram.SignalLowLabel"/>
```

SignalLowLowLabel

Plant::Diagram::SignalLowLowLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

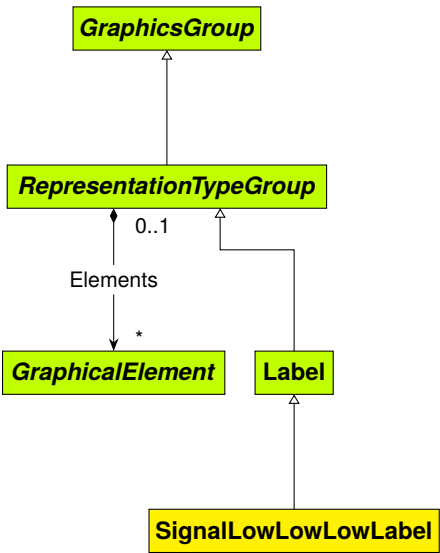
```
<Object type="Plant/Diagram.SignalLowLowLabel"/>
```

SignalLowLowLowLabel

Plant::Diagram::SignalLowLowLowLabel

Overview

Concrete Class



Supertypes

- *Label*

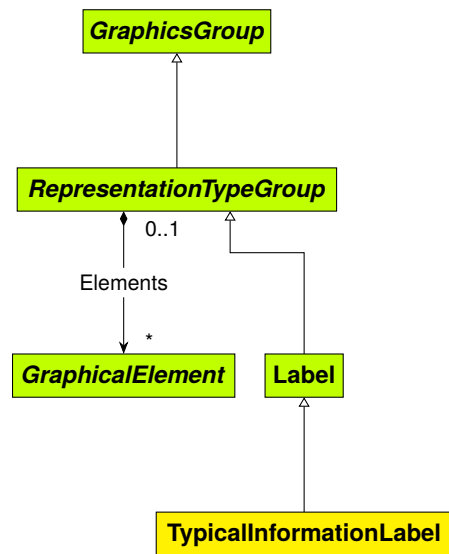
Example in DEXPI XML

```
<Object type="Plant/Diagram.SignalLowLowLowLabel"/>
```

TypicalInformationLabel

Plant::Diagram::TypicalInformationLabel

Overview

Concrete Class**Supertypes**

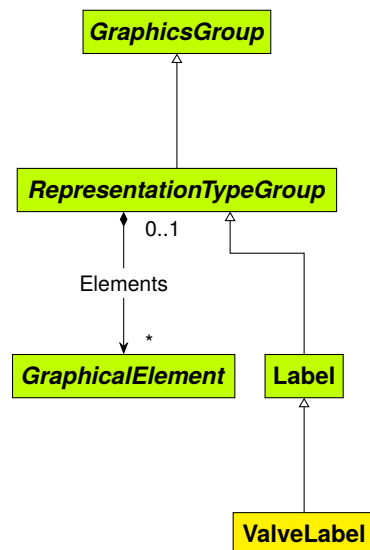
- *Label*

Example in DEXPI XML

```
<Object type="Plant/Diagram.TypicalInformationLabel"/>
```

ValveLabel

Plant::Diagram::ValveLabel

Overview**Concrete Class**

Supertypes

- *Label*

Example in DEXPI XML

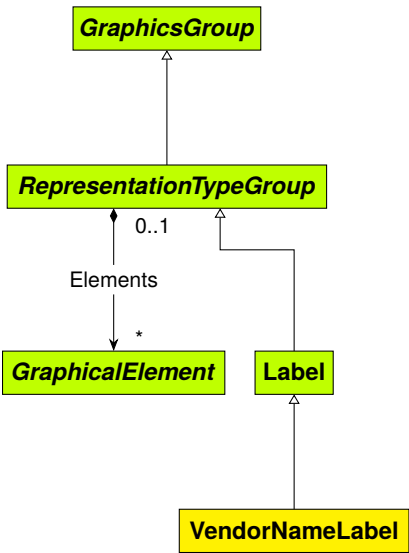
```
<Object type="Plant/Diagram.ValveLabel"/>
```

VendorNameLabel

*Plant::Diagram::*VendorNameLabel

Overview

Concrete Class



Supertypes

- *Label*

Example in DEXPI XML

```
<Object type="Plant/Diagram.VendorNameLabel"/>
```

*Plant::*Diagram

Overview

Package

Members

Classes

Name	Type
<i>ActuatingElectricalSystemNumberLabel</i>	Concrete Class
<i>ActuatingSystemNumberLabel</i>	Concrete Class
<i>CustomLabel</i>	Concrete Class
<i>DeviceInformationLabel</i>	Concrete Class
<i>EquipmentBarLabel</i>	Concrete Class
<i>EquipmentTagNameLabel</i>	Concrete Class
<i>FailActionLabel</i>	Concrete Class
<i>FittingLabel</i>	Concrete Class
<i>InstrumentationNodePosition</i>	Concrete Class
<i>InsulationBreakLabel</i>	Concrete Class
<i>InsulationLabel</i>	Concrete Class
<i>MPRelevanceLabel</i>	Concrete Class
<i>MeasuringSystemNumberLabel</i>	Concrete Class
<i>NoteIdentifierLabel</i>	Concrete Class
<i>NoteTextLabel</i>	Concrete Class
<i>NozzleStandardLabel</i>	Concrete Class
<i>OffPageConnectorDescriptionLabel</i>	Concrete Class
<i>OffPageConnectorNumberLabel</i>	Concrete Class
<i>PipingClassBreakLabel</i>	Concrete Class
<i>PipingNetworkSegmentLabel</i>	Concrete Class
<i>PipingNetworkSystemLabel</i>	Concrete Class
<i>PipingNodePosition</i>	Concrete Class
<i>PlantMetaData</i>	Concrete Class
<i>ProcessInstrumentationFunctionLabel</i>	Concrete Class
<i>QualityRelevanceLabel</i>	Concrete Class
<i>ReducerLabel</i>	Concrete Class
<i>ReferencedPIDNumberLabel</i>	Concrete Class
<i>SafetyRelevanceLabel</i>	Concrete Class
<i>SafetyValveOrFittingLabel</i>	Concrete Class
<i>SignalConveyingFunctionLabel</i>	Concrete Class
<i>SignalHighHighHighLabel</i>	Concrete Class
<i>SignalHighHighLabel</i>	Concrete Class
<i>SignalHighLabel</i>	Concrete Class
<i>SignalLowLabel</i>	Concrete Class
<i>SignalLowLowLabel</i>	Concrete Class
<i>SignalLowLowLowLabel</i>	Concrete Class
<i>TypicalInformationLabel</i>	Concrete Class
<i>ValveLabel</i>	Concrete Class
<i>VendorNameLabel</i>	Concrete Class

3.2.2 Enumerations

ChamberFunctionClassification

Plant::Enumerations::ChamberFunctionClassification

Overview

Enumeration

<<enumeration>> ChamberFunctionClassification
Cooling Heating Processing Tempering

Members

Literals

Name	Symbol	RDL Label	RDL URI
Cooling	cooling	COOLING	http://data.posccaesar.org/rdl/RDS9684422
Heating	heating	HEATING	http://data.posccaesar.org/rdl/RDS9666872
Processing	processing	PROCESSING	http://data.posccaesar.org/rdl/RDS9658367
Tempering	tempering	TEMPERING	http://sandbox.dexpi.org/rdl/Tempering

CompositionBreakClassification

*Plant::Enumerations::*CompositionBreakClassification

Overview

Enumeration

<<enumeration>> CompositionBreakClassification
CompositionBreak NoCompositionBreak

Members

Literals

Name	Symbol	RDL Label	RDL URI
CompositionBreak	composition break	COMPOSITION BREAK	http://sandbox.dexpi.org/rdl/CompositionBreak
NoComposition-Break	no composition break	NO COMPOSITION BREAK	http://sandbox.dexpi.org/rdl/NoCompositionBreak

DetonationProofArtefactClassification

Plant::Enumerations::DetonationProofArtefactClassification

Overview

Enumeration

<<enumeration>> DetonationProofArtefactClassification	
DetonationProofArtefact	NonDetonationProofArtefact

Members

Literals

Name	Symbol	RDL Label		RDL URI
DetonationProofArtefact	detonation-proof artefact	DETONATION ARTEFACT	PROOF	http://sandbox.dexpi.org/rdl/DetonationProofArtefact
NonDetonationProofArtefact	non detonation-proof artefact	NON DETONATION PROOF ARTEFACT		http://sandbox.dexpi.org/rdl/NonDetonationProofArtefact

ExplosionProofArtefactClassification

Plant::Enumerations::ExplosionProofArtefactClassification

Overview

Enumeration

<<enumeration>> ExplosionProofArtefactClassification	
ExplosionProofArtefact	NonExplosionProofArtefact

Members

Literals

Name	Symbol	RDL Label		RDL URI
ExplosionProofArtefact	explosion-proof artefact	EXPLOSION ARTEFACT	PROOF	http://sandbox.dexpi.org/rdl/ExplosionProofArtefact
NonExplosionProofArtefact	non explosion-proof artefact	NON EXPLOSION PROOF ARTEFACT		http://sandbox.dexpi.org/rdl/NonExplosionProofArtefact

FailActionClassification

Plant::Enumerations::FailActionClassification

Overview

Enumeration

<<enumeration>> FailActionClassification
FailClose FailOpen FailRetainPosition

Members

Literals

Name	Symbol	RDL Label	RDL URI
FailClose	fail close	FAIL CLOSE	http://data.posccaesar.org/rdl/RDS5921400
FailOpen	fail open	FAIL OPEN	http://data.posccaesar.org/rdl/RDS5921445
FailRetainPosition	fail retain position	FAIL RETAIN POSITION	http://sandbox.dexpi.org/rdl/FailRetainPosition

FireResistantArtefactClassification

Plant::Enumerations::FireResistantArtefactClassification

Overview

Enumeration

<<enumeration>> FireResistantArtefactClassification
FireResistantArtefact NonFireResistantArtefact

Members

Literals

Name	Symbol	RDL Label	RDL URI
FireResistantArtefact	fire-resistant artefact	FIRE RESISTANT ARTEFACT	http://data.posccaesar.org/rdl/RDS7907520
NonFireResistantArtefact	non fire-resistant artefact	NON FIRE RESISTANT ARTEFACT	http://sandbox.dexpi.org/rdl/NonFireResistantArtefact

GmpRelevanceClassification

Plant::Enumerations::GmpRelevanceClassification

Overview

Enumeration

<<enumeration>> GmpRelevanceClassification	
GmpRelevantFunction	NonGmpRelevantFunction

Members

Literals

Name	Symbol	RDL Label	RDL URI
GmpRelevantFunction	GMP relevant	GMP RELEVANT FUNCTION	http://sandbox.dexpi.org/rdl/GmpRelevantFunction
NonGmpRelevantFunction	not GMP relevant	NON GMP RELEVANT FUNCTION	http://sandbox.dexpi.org/rdl/NonGmpRelevantFunction

GuaranteedSupplyFunctionClassification

Plant::Enumerations::GuaranteedSupplyFunctionClassification

Overview

Enumeration

<<enumeration>> GuaranteedSupplyFunctionClassification	
GuaranteedSupplyFunction	NonGuaranteedSupplyFunction

Members

Literals

Name	Symbol	RDL Label	RDL URI
GuaranteedSupplyFunction	guaranteed supply	GUARANTEED SUPPLY FUNCTION	http://sandbox.dexpi.org/rdl/GuaranteedSupplyFunction
NonGuaranteedSupplyFunction	no guaranteed supply	NON GUARANTEED SUPPLY FUNCTION	http://sandbox.dexpi.org/rdl/NonGuaranteedSupplyFunction

HeatTracingTypeClassification

Plant::Enumerations::HeatTracingTypeClassification

Overview

Enumeration

<<enumeration>> HeatTracingTypeClassification
ElectricalHeatTracingSystem HeatTracingSystem NoHeatTracingSystem SteamHeatTracingSystem TubularHeatTracingSystem

Members

Literals

Name	Symbol	RDL Label	RDL URI
ElectricalHeatTracingSystem	electrical heat tracing system	ELECTRICAL HEAT TRACING SYSTEM	http://data.posccaesar.org/rdl/RDS11854600
HeatTracingSystem	heat tracing system	HEAT TRACING SYSTEM	http://data.posccaesar.org/rdl/RDS267434
NoHeatTracingSystem	no heat tracing system	NO HEAT TRACING SYSTEM	http://sandbox.dexpi.org/rdl/NoHeatTracingSystem
SteamHeatTracingSystem	steam heat tracing system	STEAM HEAT TRACING SYSTEM	http://data.posccaesar.org/rdl/RDS11854690
TubularHeatTracingSystem	tubular heat tracing system	TUBULAR HEAT TRACING SYSTEM	http://data.posccaesar.org/rdl/RDS11854645

InsulationBreakClassification

Plant::Enumerations::InsulationBreakClassification

Overview

Enumeration

<<enumeration>> InsulationBreakClassification
InsulationBreak NoInsulationBreak

Members

Literals

Name	Symbol	RDL Label	RDL URI
InsulationBreak	insulation break	INSULATION BREAK	http://sandbox.dexpi.org/rdl/InsulationBreak
NoInsulation-Break	no insulation break	NO INSULATION BREAK	http://sandbox.dexpi.org/rdl/NoInsulationBreak

JacketedPipeClassification

Plant::Enumerations::JacketedPipeClassification

Overview

Enumeration

<<enumeration>> JacketedPipeClassification
JacketedPipe UnjacketedPipe

Members

Literals

Name	Symbol	RDL Label	RDL URI
JacketedPipe	jacketed	JACKETED PIPE	http://sandbox.dexpi.org/rdl/JacketedPipe
UnjacketedPipe	not jacketed	UNJACKETED PIPE	http://sandbox.dexpi.org/rdl/UnjacketedPipe

LocationClassification

Plant::Enumerations::LocationClassification

Overview

Enumeration

<<enumeration>> LocationClassification
CentralLocation ControlPanel Field

Members

Literals

Name	Symbol	RDL Label	RDL URI
CentralLocation	central	CENTRAL LOCATION	http://sandbox.dexpi.org/rdl/CentralLocation
ControlPanel	panel	CONTROL PANEL	http://data.posccaesar.org/rdl/RDS874124
Field	field	FIELD	http://data.posccaesar.org/rdl/RDS409545541

NominalDiameterBreakClassification

*Plant::Enumerations::*NominalDiameterBreakClassification

Overview

Enumeration

<<enumeration>>
NominalDiameterBreakClassification
NoNominalDiameterBreak
NominalDiameterBreak

Members

Literals

Name	Symbol	RDL Label	RDL URI
NoNominalDiameterBreak	no nominal diameter break	NO NOMINAL DIAMETER BREAK	http://sandbox.dexpi.org/rdl/NoNominalDiameterBreak
NominalDiameterBreak	nominal diameter break	NOMINAL DIAMETER BREAK	http://sandbox.dexpi.org/rdl/NominalDiameterBreak

NominalDiameterStandardClassification

*Plant::Enumerations::*NominalDiameterStandardClassification

Overview

Enumeration

Members

Literals

Name	Symbol	RDL Label	RDL URI
Din2448ObjectDn100	DN 100 (DIN 2448)	DIN 2448 OBJECT DN 100	http://sandbox.dexpi.org/rdl/Din2448ObjectDn100
Din2448ObjectDn125	DN 125 (DIN 2448)	DIN 2448 OBJECT DN 125	http://sandbox.dexpi.org/rdl/Din2448ObjectDn125
Din2448ObjectDn15	DN 15 (DIN 2448)	DIN 2448 OBJECT DN 15	http://sandbox.dexpi.org/rdl/Din2448ObjectDn15

continues on next page

Table 5 – continued from previous page

Name	Symbol	RDL Label	RDL URI
Din2448ObjectDn150	DN 150 (DIN 2448)	DIN 2448 OBJECT DN 150	http://sandbox.dexpi.org/rdl/Din2448ObjectDn150
Din2448ObjectDn20	DN 20 (DIN 2448)	DIN 2448 OBJECT DN 20	http://sandbox.dexpi.org/rdl/Din2448ObjectDn20
Din2448ObjectDn200	DN 200 (DIN 2448)	DIN 2448 OBJECT DN 200	http://sandbox.dexpi.org/rdl/Din2448ObjectDn200
Din2448ObjectDn25	DN 25 (DIN 2448)	DIN 2448 OBJECT DN 25	http://sandbox.dexpi.org/rdl/Din2448ObjectDn25
Din2448ObjectDn32	DN 32 (DIN 2448)	DIN 2448 OBJECT DN 32	http://sandbox.dexpi.org/rdl/Din2448ObjectDn32
Din2448ObjectDn40	DN 40 (DIN 2448)	DIN 2448 OBJECT DN 40	http://sandbox.dexpi.org/rdl/Din2448ObjectDn40
Din2448ObjectDn50	DN 50 (DIN 2448)	DIN 2448 OBJECT DN 50	http://sandbox.dexpi.org/rdl/Din2448ObjectDn50
Din2448ObjectDn65	DN 65 (DIN 2448)	DIN 2448 OBJECT DN 65	http://sandbox.dexpi.org/rdl/Din2448ObjectDn65
Din2448ObjectDn80	DN 80 (DIN 2448)	DIN 2448 OBJECT DN 80	http://sandbox.dexpi.org/rdl/Din2448ObjectDn80
Iso6708ObjectDn100	DN 100 (ISO 6708)	ISO 6708 OBJECT DN 100	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn100
Iso6708ObjectDn1000	DN 1000 (ISO 6708)	ISO 6708 OBJECT DN 1000	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn1000
Iso6708ObjectDn1200	DN 1200 (ISO 6708)	ISO 6708 OBJECT DN 1200	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn1200
Iso6708ObjectDn125	DN 125 (ISO 6708)	ISO 6708 OBJECT DN 125	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn125
Iso6708ObjectDn1400	DN 1400 (ISO 6708)	ISO 6708 OBJECT DN 1400	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn1400
Iso6708ObjectDn15	DN 15 (ISO 6708)	ISO 6708 OBJECT DN 15	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn15
Iso6708ObjectDn150	DN 150 (ISO 6708)	ISO 6708 OBJECT DN 150	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn150
Iso6708ObjectDn1600	DN 1600 (ISO 6708)	ISO 6708 OBJECT DN 1600	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn1600
Iso6708ObjectDn20	DN 20 (ISO 6708)	ISO 6708 OBJECT DN 20	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn20
Iso6708ObjectDn200	DN 200 (ISO 6708)	ISO 6708 OBJECT DN 200	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn200
Iso6708ObjectDn25	DN 25 (ISO 6708)	ISO 6708 OBJECT DN 25	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn25
Iso6708ObjectDn250	DN 250 (ISO 6708)	ISO 6708 OBJECT DN 250	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn250
Iso6708ObjectDn300	DN 300 (ISO 6708)	ISO 6708 OBJECT DN 300	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn300
Iso6708ObjectDn32	DN 32 (ISO 6708)	ISO 6708 OBJECT DN 32	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn32
Iso6708ObjectDn350	DN 350 (ISO 6708)	ISO 6708 OBJECT DN 350	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn350
Iso6708ObjectDn40	DN 40 (ISO 6708)	ISO 6708 OBJECT DN 40	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn40
Iso6708ObjectDn400	DN 400 (ISO 6708)	ISO 6708 OBJECT DN 400	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn400
Iso6708ObjectDn450	DN 450 (ISO 6708)	ISO 6708 OBJECT DN 450	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn450
Iso6708ObjectDn50	DN 50 (ISO 6708)	ISO 6708 OBJECT DN 50	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn50
Iso6708ObjectDn500	DN 500 (ISO 6708)	ISO 6708 OBJECT DN 500	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn500
Iso6708ObjectDn600	DN 600 (ISO 6708)	ISO 6708 OBJECT DN 600	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn600
Iso6708ObjectDn65	DN 65 (ISO 6708)	ISO 6708 OBJECT DN 65	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn65
Iso6708ObjectDn700	DN 700 (ISO 6708)	ISO 6708 OBJECT DN 700	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn700
Iso6708ObjectDn80	DN 80 (ISO 6708)	ISO 6708 OBJECT DN 80	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn80
Iso6708ObjectDn800	DN 800 (ISO 6708)	ISO 6708 OBJECT DN 800	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn800
Iso6708ObjectDn900	DN 900 (ISO 6708)	ISO 6708 OBJECT DN 900	http://sandbox.dexpi.org/rdl/Iso6708ObjectDn900
Nps10Artefact	NPS 10	NPS 10 ARTEFACT	http://data.posccaesar.org/rdl/RDS20863408298
Nps12Artefact	NPS 12	NPS 12 ARTEFACT	http://data.posccaesar.org/rdl/RDS208634082110
Nps14Artefact	NPS 14	NPS 14 ARTEFACT	http://data.posccaesar.org/rdl/RDS208634082122
Nps16Artefact	NPS 16	NPS 16 ARTEFACT	http://data.posccaesar.org/rdl/RDS208634082134
Nps18Artefact	NPS 18	NPS 18 ARTEFACT	http://data.posccaesar.org/rdl/RDS208634082146
Nps1Artefact	NPS 1	NPS 1 ARTEFACT	http://data.posccaesar.org/rdl/RDS20863408137
Nps1_1_PER_2Artefact	NPS 1 1/2	NPS 1 1/2 ARTEFACT	http://data.posccaesar.org/rdl/RDS2086340822
Nps1_1_PER_4Artefact	NPS 1 1/4	NPS 1 1/4 ARTEFACT	http://data.posccaesar.org/rdl/RDS20863408321
Nps1_PER_2Artefact	NPS 1/2	NPS 1/2 ARTEFACT	http://data.posccaesar.org/rdl/RDS20863408113
Nps1_PER_4Artefact	NPS 1/4	NPS 1/4 ARTEFACT	http://data.posccaesar.org/rdl/RDS2086340839
Nps20Artefact	NPS 20	NPS 20 ARTEFACT	http://data.posccaesar.org/rdl/RDS208634082158
Nps24Artefact	NPS 24	NPS 24 ARTEFACT	http://data.posccaesar.org/rdl/RDS208634082170
Nps2Artefact	NPS 2	NPS 2 ARTEFACT	http://data.posccaesar.org/rdl/RDS20863408214
Nps2_1_PER_2Artefact	NPS 2 1/2	NPS 2 1/2 ARTEFACT	http://data.posccaesar.org/rdl/RDS20863408226
Nps30Artefact	NPS 30	NPS 30 ARTEFACT	http://data.posccaesar.org/rdl/RDS208634082182

continues on next page

Table 5 – continued from previous page

Name	Symbol	RDL Label	RDL URI
Nps36Artefact	NPS 36	NPS 36 ARTEFACT	http://data.posccaesar.org/rdl/RDS208634082194
Nps3Artefact	NPS 3	NPS 3 ARTEFACT	http://data.posccaesar.org/rdl/RDS20863408238
Nps3_1_PER_2Artefact	NPS 3 1/2	NPS 3 1/2 ARTEFACT	http://data.posccaesar.org/rdl/RDS20863408333
Nps3_PER_4Artefact	NPS 3/4	NPS 3/4 ARTEFACT	http://data.posccaesar.org/rdl/RDS20863408125
Nps42Artefact	NPS 42	NPS 42 ARTEFACT	http://data.posccaesar.org/rdl/RDS208634082206
Nps48Artefact	NPS 48	NPS 48 ARTEFACT	http://data.posccaesar.org/rdl/RDS208634082218
Nps4Artefact	NPS 4	NPS 4 ARTEFACT	http://data.posccaesar.org/rdl/RDS20863408250
Nps54Artefact	NPS 54	NPS 54 ARTEFACT	http://data.posccaesar.org/rdl/RDS208634082230
Nps5Artefact	NPS 5	NPS 5 ARTEFACT	http://data.posccaesar.org/rdl/RDS20863408262
Nps60Artefact	NPS 60	NPS 60 ARTEFACT	http://data.posccaesar.org/rdl/RDS208634082242
Nps6Artefact	NPS 6	NPS 6 ARTEFACT	http://data.posccaesar.org/rdl/RDS20863408274
Nps8Artefact	NPS 8	NPS 8 ARTEFACT	http://data.posccaesar.org/rdl/RDS20863408286

NominalPressureStandardClassification

*Plant::Enumerations::*NominalPressureStandardClassification

Overview

Enumeration

Members

Literals

Name	Symbol	RDL Label	RDL URI
Class10000PsiArtefact	Class 10000 psi	CLASS 10000 PSI ARTEFACT	http://sandbox.dexpi.org/rdl/Class10000PsiArtefact
Class1000KpaArtefact	Class 1000 kpa	CLASS 1000 KPA ARTEFACT	http://sandbox.dexpi.org/rdl/Class1000KpaArtefact
Class125LbsArtefact	Class 125 lbs	CLASS 125 LBS ARTEFACT	http://sandbox.dexpi.org/rdl/Class125LbsArtefact
Class15000PsiArtefact	Class 15000 psi	CLASS 15000 PSI ARTEFACT	http://sandbox.dexpi.org/rdl/Class15000PsiArtefact
Class1500LbsArtefact	Class 1500 lbs	CLASS 1500 LBS ARTEFACT	http://sandbox.dexpi.org/rdl/Class1500LbsArtefact
Class150LbsArtefact	Class 150 lbs	CLASS 150 LBS ARTEFACT	http://sandbox.dexpi.org/rdl/Class150LbsArtefact
Class16BarArtefact	Class 16 bar	CLASS 16 BAR ARTEFACT	http://sandbox.dexpi.org/rdl/Class16BarArtefact
Class20000PsiArtefact	Class 20000 psi	CLASS 20000 PSI ARTEFACT	http://sandbox.dexpi.org/rdl/Class20000PsiArtefact
Class2000PsiArtefact	Class 2000 psi	CLASS 2000 PSI ARTEFACT	http://sandbox.dexpi.org/rdl/Class2000PsiArtefact
Class2500LbsArtefact	Class 2500 lbs	CLASS 2500 LBS ARTEFACT	http://sandbox.dexpi.org/rdl/Class2500LbsArtefact
Class250PsiArtefact	Class 250 psi	CLASS 250 PSI ARTEFACT	http://sandbox.dexpi.org/rdl/Class250PsiArtefact
Class3000PsiArtefact	Class 3000 psi	CLASS 3000 PSI ARTEFACT	http://sandbox.dexpi.org/rdl/Class3000PsiArtefact
Class300LbsArtefact	Class 300 lbs	CLASS 300 LBS ARTEFACT	http://sandbox.dexpi.org/rdl/Class300LbsArtefact
Class300PsiArtefact	Class 300 psi	CLASS 300 PSI ARTEFACT	http://sandbox.dexpi.org/rdl/Class300PsiArtefact
Class315BarArtefact	Class 315 bar	CLASS 315 BAR ARTEFACT	http://sandbox.dexpi.org/rdl/Class315BarArtefact
Class345BarArtefact	Class 345 bar	CLASS 345 BAR ARTEFACT	http://sandbox.dexpi.org/rdl/Class345BarArtefact
Class350BarArtefact	Class 350 bar	CLASS 350 BAR ARTEFACT	http://sandbox.dexpi.org/rdl/Class350BarArtefact
Class4000PsiArtefact	Class 4000 psi	CLASS 4000 PSI ARTEFACT	http://sandbox.dexpi.org/rdl/Class4000PsiArtefact
Class400LbsArtefact	Class 400 lbs	CLASS 400 LBS ARTEFACT	http://sandbox.dexpi.org/rdl/Class400LbsArtefact
Class4500LbsArtefact	Class 4500 lbs	CLASS 4500 LBS ARTEFACT	http://sandbox.dexpi.org/rdl/Class4500LbsArtefact
Class4500PsiArtefact	Class 4500 psi	CLASS 4500 PSI ARTEFACT	http://sandbox.dexpi.org/rdl/Class4500PsiArtefact
Class5000PsiArtefact	Class 5000 psi	CLASS 5000 PSI ARTEFACT	http://sandbox.dexpi.org/rdl/Class5000PsiArtefact
Class50BarArtefact	Class 50 bar	CLASS 50 BAR ARTEFACT	http://sandbox.dexpi.org/rdl/Class50BarArtefact
Class517BarArtefact	Class 517 bar	CLASS 517 BAR ARTEFACT	http://sandbox.dexpi.org/rdl/Class517BarArtefact

continues on

Table 6 – continued from previous page

Name	Symbol	RDL Label	RDL URI
Class6000PsiArtefact	Class 6000 psi	CLASS 6000 PSI ARTEFACT	http://sandbox.dexpi.org/rdl/Class6000PsiArtefact
Class600LbsArtefact	Class 600 lbs	CLASS 600 LBS ARTEFACT	http://sandbox.dexpi.org/rdl/Class600LbsArtefact
Class690BarArtefact	Class 690 bar	CLASS 690 BAR ARTEFACT	http://sandbox.dexpi.org/rdl/Class690BarArtefact
Class800LbsArtefact	Class 800 lbs	CLASS 800 LBS ARTEFACT	http://sandbox.dexpi.org/rdl/Class800LbsArtefact
Class800PsiArtefact	Class 800 psi	CLASS 800 PSI ARTEFACT	http://sandbox.dexpi.org/rdl/Class800PsiArtefact
Class850KpaArtefact	Class 850 kpa	CLASS 850 KPA ARTEFACT	http://sandbox.dexpi.org/rdl/Class850KpaArtefact
Class9000LbsArtefact	Class 9000 lbs	CLASS 9000 LBS ARTEFACT	http://sandbox.dexpi.org/rdl/Class9000LbsArtefact
Class900LbsArtefact	Class 900 lbs	CLASS 900 LBS ARTEFACT	http://sandbox.dexpi.org/rdl/Class900LbsArtefact
En1333Pn100Artefact	PN 100 (EN 1333)	EN 1333 PN 100 ARTEFACT	http://sandbox.dexpi.org/rdl/En1333Pn100Artefact
En1333Pn10Artefact	PN 10 (EN 1333)	EN 1333 PN 10 ARTEFACT	http://sandbox.dexpi.org/rdl/En1333Pn10Artefact
En1333Pn160Artefact	PN 160 (EN 1333)	EN 1333 PN 160 ARTEFACT	http://sandbox.dexpi.org/rdl/En1333Pn160Artefact
En1333Pn16Artefact	PN 16 (EN 1333)	EN 1333 PN 16 ARTEFACT	http://sandbox.dexpi.org/rdl/En1333Pn16Artefact
En1333Pn250Artefact	PN 250 (EN 1333)	EN 1333 PN 250 ARTEFACT	http://sandbox.dexpi.org/rdl/En1333Pn250Artefact
En1333Pn25Artefact	PN 25 (EN 1333)	EN 1333 PN 25 ARTEFACT	http://sandbox.dexpi.org/rdl/En1333Pn25Artefact
En1333Pn2_COMMA_5Artefact	PN 2,5 (EN 1333)	EN 1333 PN 2,5 ARTEFACT	http://sandbox.dexpi.org/rdl/En1333Pn2_COMMA_5Artefact
En1333Pn320Artefact	PN 320 (EN 1333)	EN 1333 PN 320 ARTEFACT	http://sandbox.dexpi.org/rdl/En1333Pn320Artefact
En1333Pn400Artefact	PN 400 (EN 1333)	EN 1333 PN 400 ARTEFACT	http://sandbox.dexpi.org/rdl/En1333Pn400Artefact
En1333Pn40Artefact	PN 40 (EN 1333)	EN 1333 PN 40 ARTEFACT	http://sandbox.dexpi.org/rdl/En1333Pn40Artefact
En1333Pn63Artefact	PN 63 (EN 1333)	EN 1333 PN 63 ARTEFACT	http://sandbox.dexpi.org/rdl/En1333Pn63Artefact
En1333Pn6Artefact	PN 6 (EN 1333)	EN 1333 PN 6 ARTEFACT	http://sandbox.dexpi.org/rdl/En1333Pn6Artefact

NumberOfPortsClassification

Plant::Enumerations::NumberOfPortsClassification

Overview

Enumeration

<<enumeration>> NumberOfPortsClassification
FourPortValve ThreePortValve TwoPortValve

Members

Literals

Name	Symbol	RDL Label	RDL URI
FourPortValve	4 port valve	FOUR PORT VALVE	http://data.posccaesar.org/rdl/RDS6330166
ThreePortValve	3 port valve	THREE PORT VALVE	http://data.posccaesar.org/rdl/RDS6331437
TwoPortValve	2 port valve	TWO PORT VALVE	http://data.posccaesar.org/rdl/RDS11506315

OnHoldClassification

Plant::Enumerations::OnHoldClassification

Overview

Enumeration

<<enumeration>> OnHoldClassification
NotOnHold OnHold

Members

Literals

Name	Symbol	RDL Label	RDL URI
NotOnHold	not on hold	NOT ON HOLD	http://sandbox.dexpi.org/rdl/NotOnHold
OnHold	on hold	ON HOLD	http://sandbox.dexpi.org/rdl/OnHold

OperationClassification

Plant::Enumerations::OperationClassification

Overview

Enumeration

<<enumeration>> OperationClassification
ContinuousOperation IntermittentOperation

Members

Literals

Name	Symbol	RDL Label	RDL URI
ContinuousOperation	continuous operation	CONTINUOUS OPERATION	http://data.posccaesar.org/rdl/RDS9710162
IntermittentOperation	intermittent operation	INTERMITTENT OPERATION	http://data.posccaesar.org/rdl/RDS9705752

PipingClassArtefactClassification

Plant::Enumerations::PipingClassArtefactClassification

Overview

Enumeration

<<enumeration>> PipingClassArtefactClassification
NonPipingClassArtefact PipingClassArtefact

Members

Literals

Name	Symbol	RDL Label	RDL URI
NonPipingClassArtefact	non-piping-class artefact	NON PIPING CLASS ARTEFACT	http://sandbox.dexpi.org/rdl/NonPipingClassArtefact
PipingClassArtefact	pipng class artefact	PIPING CLASS ARTEFACT	http://sandbox.dexpi.org/rdl/PipingClassArtefact

PipingClassBreakClassification

Plant::Enumerations::PipingClassBreakClassification

Overview

Enumeration

<<enumeration>> PipingClassBreakClassification
NoPipingClassBreak PipingClassBreak

Members

Literals

Name	Symbol	RDL Label	RDL URI
NoPipingClassBreak	no pipng class break	NO PIPING CLASS BREAK	http://sandbox.dexpi.org/rdl/NoPipingClassBreak
PipingClassBreak	pipng class break	PIPING CLASS BREAK	http://sandbox.dexpi.org/rdl/PipingClassBreak

PipingNetworkSegmentFlowClassification

Plant::Enumerations::PipingNetworkSegmentFlowClassification

Overview

Enumeration

<<enumeration>>	
PipingNetworkSegmentFlowClassification	
DualFlowPipingNetworkSegment	SingleFlowPipingNetworkSegment

Members

Literals

Name	Sym- bol	RDL Label	RDL URI
DualFlowPipingNetworkSegment	dual flow	DUAL FLOW PIPING NETWORK SEGMENT	http://sandbox.dexpi.org/rdl/DualFlowPipingNetworkSegment
SingleFlowPipingNetworkSegment	single flow	SINGLE FLOW PIPING NETWORK SEGMENT	http://sandbox.dexpi.org/rdl/SingleFlowPipingNetworkSegment

PipingNetworkSegmentSlopeClassification

Plant::Enumerations::PipingNetworkSegmentSlopeClassification

Overview

Enumeration

<<enumeration>>	
PipingNetworkSegmentSlopeClassification	
SlopedPipingNetworkSegment	UnslopedPipingNetworkSegment

Members

Literals

Name	Sym- bol	RDL Label	RDL URI
SlopedPipingNetworkSegment	sloped	SLOPED PIPING NETWORK SEGMENT	http://sandbox.dexpi.org/rdl/SlopedPipingNetworkSegment
UnslopedPipingNetworkSegment	not sloped	UNSLOPED PIPING NETWORK SEGMENT	http://sandbox.dexpi.org/rdl/UnslopedPipingNetworkSegment

PortStatusClassification

Plant::Enumerations::PortStatusClassification

Overview

Enumeration

<<enumeration>> PortStatusClassification
StatusHighHighHighPort StatusHighHighPort StatusHighPort StatusLowLowLowPort StatusLowLowPort StatusLowPort

Members

Literals

Name	Sym- bol	RDL Label	RDL URI
StatusHighHigh-HighPort	HHH	STATUS HIGH HIGH HIGH PORT	http://sandbox.dexpi.org/rdl/StatusHighHighHighPort
StatusHighHighPort	HH	STATUS HIGH HIGH PORT	http://data.posccaesar.org/rdl/RDS323099
StatusHighPort	H	STATUS HIGH PORT	http://data.posccaesar.org/rdl/RDS323144
StatusLowLowLow-Port	LLL	STATUS LOW LOW LOW PORT	http://sandbox.dexpi.org/rdl/StatusLowLowLowPort
StatusLowLowPort	LL	STATUS LOW LOW PORT	http://data.posccaesar.org/rdl/RDS323189
StatusLowPort	L	STATUS LOW PORT	http://data.posccaesar.org/rdl/RDS323234

PrimarySecondaryPipingNetworkSegmentClassification

Plant::Enumerations::PrimarySecondaryPipingNetworkSegmentClassification

Overview

Enumeration

<<enumeration>> PrimarySecondaryPipingNetworkSegmentClassification
PrimaryPipingNetworkSegment SecondaryPipingNetworkSegment

Members

Literals

Name	Symbol	RDL Label	RDL URI
PrimaryPipingNetworkSegment	primary segment	PRIMARY PIPING NETWORK SEGMENT	http://sandbox.dexpi.org/rdl/PrimaryPipingNetworkSegment
SecondaryPipingNetworkSegment	secondary segment	SECONDARY PIPING NETWORK SEGMENT	http://sandbox.dexpi.org/rdl/SecondaryPipingNetworkSegment

QualityRelevanceClassification

Plant::Enumerations::QualityRelevanceClassification

Overview

Enumeration

<<enumeration>> QualityRelevanceClassification
NonQualityRelevantFunction QualityRelevantFunction

Members

Literals

Name	Symbol	RDL Label	RDL URI
NonQualityRelevantFunction	not quality relevant	NON QUALITY RELEVANT FUNCTION	http://sandbox.dexpi.org/rdl/NonQualityRelevantFunction
QualityRelevantFunction	quality relevant	QUALITY RELEVANT FUNCTION	http://sandbox.dexpi.org/rdl/QualityRelevantFunction

SignalConveyingTypeClassification

Plant::Enumerations::SignalConveyingTypeClassification

Overview

Enumeration

<<enumeration>> SignalConveyingTypeClassification
CapillarySignalConveying ConductedRadiationSignalConveying ElectricalSignalConveying HydraulicSignalConveying PneumaticSignalConveying

Members

Literals

Name	Symbol	RDL Label	RDL URI
CapillarySignalConveying	capillary	CAPILLARY SIGNAL CONVEYING	http://sandbox.dexpi.org/rdl/CapillarySignalConveying
ConductedRadiationSignalConveying	conducted radiation	CONDUCTED RADIATION SIGNAL CONVEYING	http://sandbox.dexpi.org/rdl/ConductedRadiationSignalConveying
ElectricalSignalConveying	electrical	ELECTRICAL SIGNAL CONVEYING	http://sandbox.dexpi.org/rdl/ElectricalSignalConveying
HydraulicSignalConveying	hydraulic	HYDRAULIC SIGNAL CONVEYING	http://sandbox.dexpi.org/rdl/HydraulicSignalConveying
PneumaticSignalConveying	pneumatic	PNEUMATIC SIGNAL CONVEYING	http://sandbox.dexpi.org/rdl/PneumaticSignalConveying

SiphonClassification

Plant::Enumerations::SiphonClassification

Overview

Enumeration

<<enumeration>> SiphonClassification
NoSiphon Siphon

Members

Literals

Name	Symbol	RDL Label	RDL URI
NoSiphon	no siphon	NO SIPHON	http://sandbox.dexpi.org/rdl/NoSiphon
Siphon	siphon	SIPHON	http://data.posccaesar.org/rdl/RDS311084

Plant::Enumerations

Overview

Package

The Enumerations package contains enumerations for various aspects of engineering information in a P&ID. Enumerations that are relevant for P&ID graphics only are part of the Graphics package.

Members

Data Types

Name	Type
<i>ChamberFunctionClassification</i>	Enumeration
<i>CompositionBreakClassification</i>	Enumeration
<i>DetonationProofArtefactClassification</i>	Enumeration
<i>ExplosionProofArtefactClassification</i>	Enumeration
<i>FailActionClassification</i>	Enumeration
<i>FireResistantArtefactClassification</i>	Enumeration
<i>GmpRelevanceClassification</i>	Enumeration
<i>GuaranteedSupplyFunctionClassification</i>	Enumeration
<i>HeatTracingTypeClassification</i>	Enumeration
<i>InsulationBreakClassification</i>	Enumeration
<i>JacketedPipeClassification</i>	Enumeration
<i>LocationClassification</i>	Enumeration
<i>NominalDiameterBreakClassification</i>	Enumeration
<i>NominalDiameterStandardClassification</i>	Enumeration
<i>NominalPressureStandardClassification</i>	Enumeration
<i>NumberOfPortsClassification</i>	Enumeration
<i>OnHoldClassification</i>	Enumeration
<i>OperationClassification</i>	Enumeration
<i>PipingClassArtefactClassification</i>	Enumeration
<i>PipingClassBreakClassification</i>	Enumeration
<i>PipingNetworkSegmentFlowClassification</i>	Enumeration
<i>PipingNetworkSegmentSlopeClassification</i>	Enumeration
<i>PortStatusClassification</i>	Enumeration
<i>PrimarySecondaryPipingNetworkSegmentClassification</i>	Enumeration
<i>QualityRelevanceClassification</i>	Enumeration
<i>SignalConveyingTypeClassification</i>	Enumeration
<i>SiphonClassification</i>	Enumeration

3.2.3 Instrumentation

ActuatingElectricalFunction

Plant::Instrumentation::ActuatingElectricalFunction

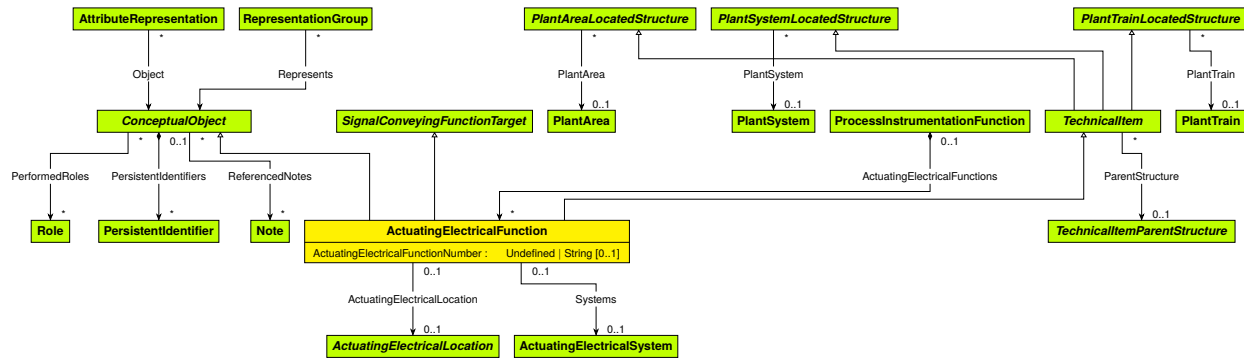
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/ActuatingElectricalFunction>

ACTUATING ELECTRICAL FUNCTION

An actuation setting electrical function. It covers all types of electrical consumers, e.g., motors and heaters.



Supertypes

- *ConceptualObject*
- *SignalConveyingFunctionTarget*
- *TechnicalItem*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>ActuatingElectricalLocation</i>	0..1	<i>ActuatingElectricalLocation</i>	no	no	0..1
<i>Systems</i>	0..1	<i>ActuatingElectricalSystem</i>	no	no	0..1

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>ActuatingElectricalFunctionNumber</i>	0..1	<i>Undefined String</i>	no	no

ActuatingElectricalFunctionNumber

Data Property

An identifier for the *ActuatingElectricalFunction*. It usually contains the identifier of the *ProcessInstrumentationFunction* that includes the *ActuatingElectricalFunction* (see *ProcessInstrumentationFunctionNumber*).

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ActuatingElectricalFunctionNumberAssignmentClass>

ACTUATING ELECTRICAL FUNCTION NUMBER ASSIGNMENT CLASS

Example

"E4750.01"

ActuatingElectricalLocation**Reference Property**

The actuating electrical location of the ActuatingElectricalFunction.

Type: *ActuatingElectricalLocation*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

Systems**Reference Property**

The ActuatingElectricalSystem that implements the ActuatingElectricalFunction.

Type: *ActuatingElectricalSystem*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

Example in DEXPI XML

```
<Object type="Plant/Instrumentation.ActuatingElectricalFunction">
  <Data property="ActuatingElectricalFunctionNumber">
    <String>E4750.01</String>
  </Data>
</Object>
```

ActuatingElectricalLocation

Plant::Instrumentation::ActuatingElectricalLocation

Overview**Abstract Class**

An object suitable as the ActuatingElectricalLocation of an *ActuatingElectricalFunction*.

ActuatingElectricalLocation

Subtypes

- *Nozzle*
- *PipingNetworkSegment*

ActuatingElectricalSystem

Plant::Instrumentation::ActuatingElectricalSystem

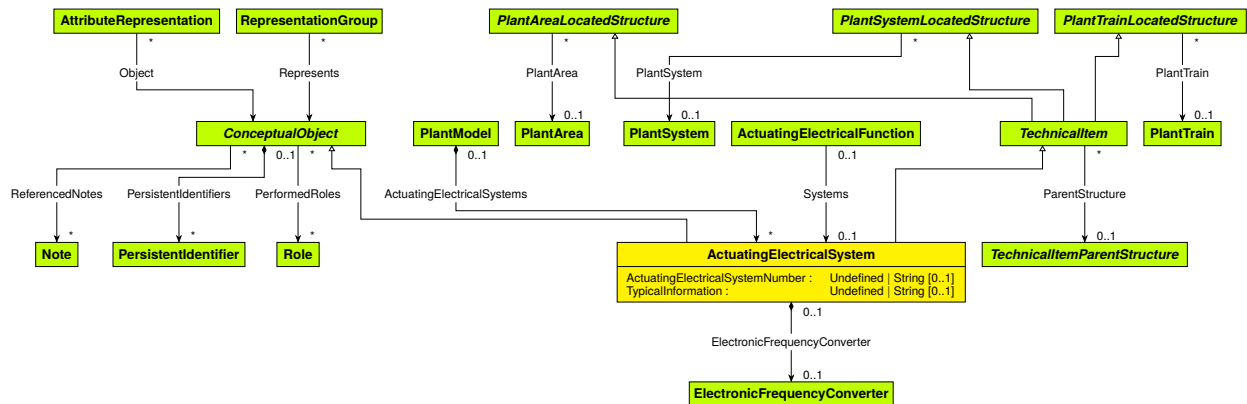
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/ActuatingElectricalSystem>

ACTUATING ELECTRICAL SYSTEM

An assembly of artefacts that is designed to fulfill an *ActuatingElectricalFunction*.



Supertypes

- *ConceptualObject*
- *TechnicalItem*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>ElectronicFrequencyConverter</i>	0..1	<i>ElectronicFrequencyConverter</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>ActuatingElectricalSystemNumber</i>	0..1	<i>Undefined String</i>	no	no
<i>TypicalInformation</i>	0..1	<i>Undefined String</i>	no	no

ActuatingElectricalSystemNumber

Data Property

The number of *ActuatingElectricalSystem*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ActuatingSystemNumberAssignmentClass>

ACTUATING SYSTEM NUMBER ASSIGNMENT CLASS

Example

"E0001"

ElectronicFrequencyConverter

Composition Property

The *ElectronicFrequencyConverter* of the *ActuatingElectricalSystem*.

Type: *ElectronicFrequencyConverter*

Multiplicity: 0..1

Ordered: no

TypicalInformation

Data Property

Typical information about the *ActuatingElectricalSystem*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/TypicalInformationAssignmentClass>

TYPICAL INFORMATION ASSIGNMENT CLASS

Example

"F4"

Example in DEXPI XML

```
<Object type="Plant/Instrumentation.ActuatingElectricalSystem">
  <Data property="ActuatingElectricalSystemNumber">
    <String>E0001</String>
  </Data>
```

(continues on next page)

(continued from previous page)

```

<Data property="TypicalInformation">
  <String>F4</String>
</Data>
</Object>

```

ActuatingFunction

Plant::Instrumentation::ActuatingFunction

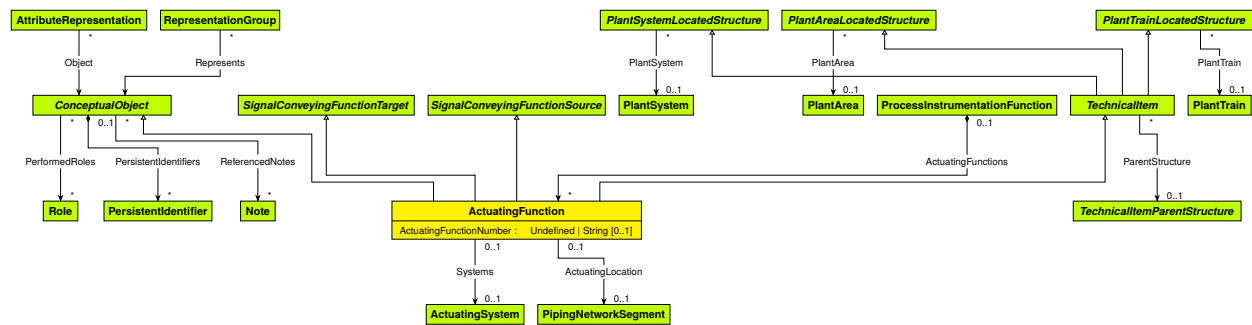
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/ActuatingFunction>

ACTUATING FUNCTION

A function for acting control structures relating to the process.



Supertypes

- *ConceptualObject*
- *SignalConveyingFunctionSource*
- *SignalConveyingFunctionTarget*
- *TechnicalItem*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>ActuatingLocation</i>	0..1	<i>PipingNetworkSegment</i>	no	no	0..1
<i>Systems</i>	0..1	<i>ActuatingSystem</i>	no	no	0..1

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>ActuatingFunctionNumber</i>	0..1	<i>Undefined String</i>	no	no

ActuatingFunctionNumber

Data Property

An identifier for the *ActuatingFunction*. It usually contains the identifier of the *ProcessInstrumentationFunction* that includes the *ActuatingFunction* (see *ProcessInstrumentationFunctionNumber*).

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ActuatingFunctionNumberAssignmentClass>

ACTUATING FUNCTION NUMBER ASSIGNMENT CLASS

Example

"HV4750.01"

ActuatingLocation

Reference Property

The actuating location of the ActuatingFunction.

Type: *PipingNetworkSegment*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

Systems

Reference Property

The ActuatingSystem that implements the ActuatingFunction.

Type: *ActuatingSystem*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

Example in DEXPI XML

```
<Object type="Plant/Instrumentation.ActuatingFunction">
  <Data property="ActuatingFunctionNumber">
    <String>HV4750.01</String>
  </Data>
</Object>
```

ActuatingSystem

Plant::Instrumentation::ActuatingSystem

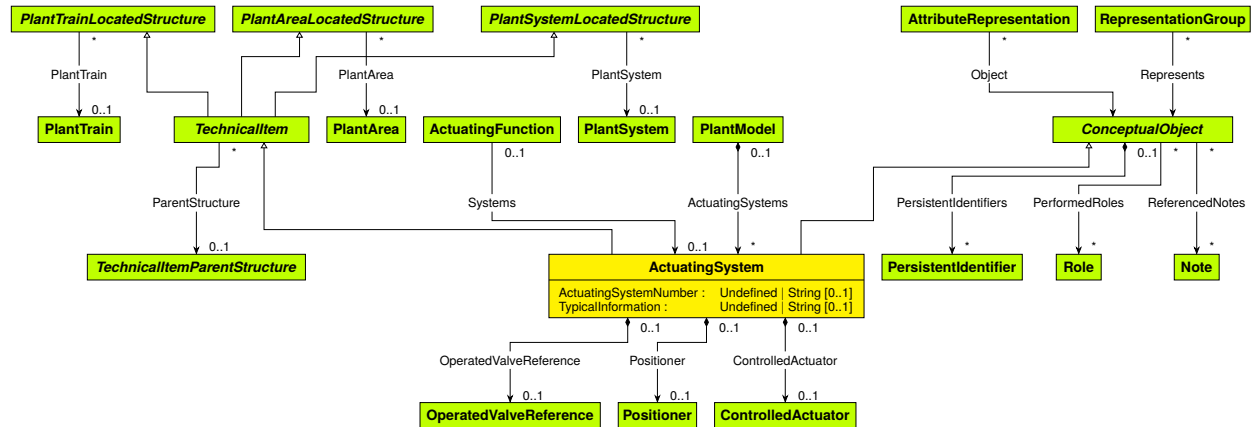
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/ActuatingSystem>

ACTUATING SYSTEM

An assembly of artefacts that is designed to fulfill an *ActuatingFunction*.



Supertypes

- *ConceptualObject*
- *TechnicalItem*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>ControlledActuator</i>	0..1	<i>ControlledActuator</i>	no
<i>OperatedValveReference</i>	0..1	<i>OperatedValveReference</i>	no
<i>Positioner</i>	0..1	<i>Positioner</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>ActuatingSystemNumber</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>TypicalInformation</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

ActuatingSystemNumber

Data Property

The number of the *ActuatingSystem*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ActuatingSystemNumberAssignmentClass>

ACTUATING SYSTEM NUMBER ASSIGNMENT CLASS

Example

"FT0001"

ControlledActuator

Composition Property

The *ControlledActuator* of the *ActuatingSystem*.

Type: *ControlledActuator*

Multiplicity: 0..1

Ordered: no

OperatedValveReference

Composition Property

The *OperatedValveReference* of the *ActuatingSystem*.

Type: *OperatedValveReference*

Multiplicity: 0..1

Ordered: no

Positioner

Composition Property

The *Positioner* of the *ActuatingSystem*.

Type: *Positioner*

Multiplicity: 0..1

Ordered: no

TypicalInformation

Data Property

Typical information about the *ActuatingSystem*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/TypicalInformationAssignmentClass>

TYPICAL INFORMATION ASSIGNMENT CLASS

Example

"F4"

Example in DEXPI XML

```
<Object type="Plant/Instrumentation.ActuatingSystem">
  <Data property="ActuatingSystemNumber">
    <String>FT0001</String>
  </Data>
  <Data property="TypicalInformation">
    <String>F4</String>
  </Data>
</Object>
```

ControlledActuator

Plant::Instrumentation::ControlledActuator

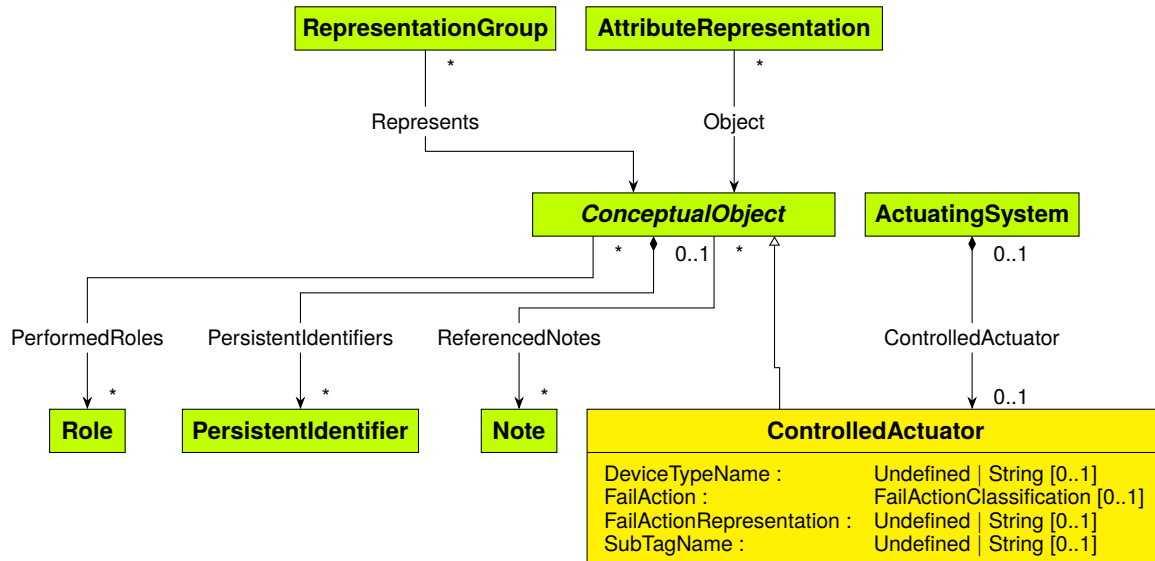
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/ControlledActuator>

CONTROLLED ACTUATOR

A transducer that is intended to convert energy (electric, mechanical, pneumatic or hydraulic) from an external source into kinetic energy (motion) in response to a signal or power input.



Supertypes

- *ConceptualObject*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DeviceTypeName</i>	0..1	<i>Undefined String</i>	no	no
<i>FailAction</i>	0..1	<i>FailActionClassification</i>	no	no
<i>FailActionRepresentation</i>	0..1	<i>Undefined String</i>	no	no
<i>SubTagName</i>	0..1	<i>Undefined String</i>	no	no

DeviceTypeName

Data Property

The device type of the *ControlledActuator*.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DeviceTypeNameAssignmentClass>

DEVICE TYPE NAME ASSIGNMENT CLASS

Example

"pressure transmitter"

FailAction

Data Property

The fail action of the *ControlledActuator*.

Type: *FailActionClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/FailActionSpecialization>

FAIL ACTION SPECIALIZATION

Example

FailOpen

FailActionRepresentation

Data Property

A readable representation of the fail action of the *ControlledActuator*. This attribute should also be referenced in the graphics if applicable.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/FailActionRepresentationAssignmentClass>

FAIL ACTION REPRESENTATION ASSIGNMENT CLASS

Example

"F.O."

SubTagName

Data Property

The sub tag name of the *ControlledActuator*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SubTagNameAssignmentClass>

SUB TAG NAME ASSIGNMENT CLASS

Example

"ST1"

Example in DEXPI XML

```
<Object type="Plant/Instrumentation.ControlledActuator">
  <Data property="DeviceTypeName">
    <String>pressure transmitter</String>
  </Data>
  <Data property="FailAction">
    <DataReference data="Plant/Enumerations.FailActionClassification.FailOpen"/>
  </Data>
  <Data property="FailActionRepresentation">
    <String>F.O.</String>
  </Data>
  <Data property="SubTagName">
    <String>ST1</String>
  </Data>
</Object>
```

ElectronicFrequencyConverter

Plant::Instrumentation::ElectronicFrequencyConverter

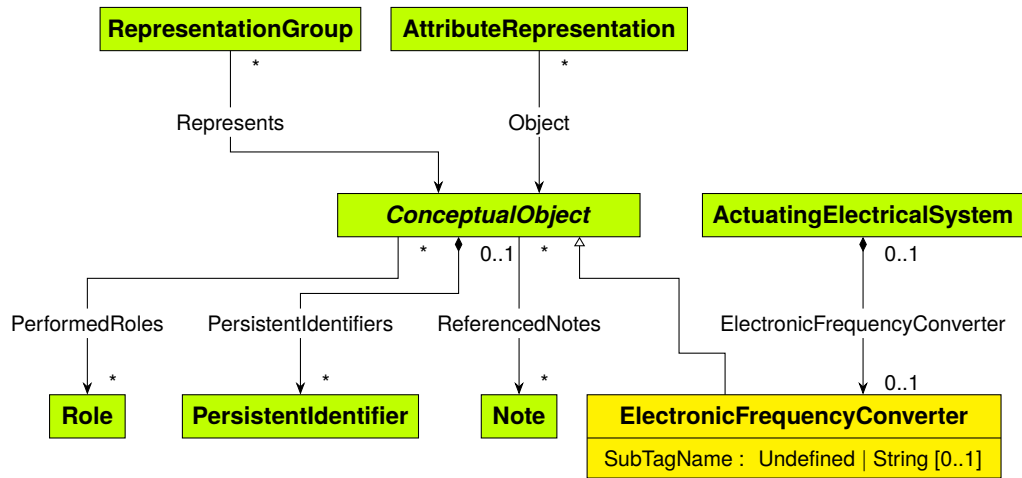
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/ElectronicFrequencyConverter>

ELECTRONIC FREQUENCY CONVERTER

An electronic AC converter for changing the frequency.



Supertypes

- *ConceptualObject*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>SubTagName</i>	0..1	<i>Undefined String</i>	no	no

SubTagName

Data Property

The sub tag name of the *ElectronicFrequencyConverter*.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SubTagNameAssignmentClass>

SUB TAG NAME ASSIGNMENT CLASS

Example

"ST1"

Example in DEXPI XML

```
<Object type="Plant/Instrumentation.ElectronicFrequencyConverter">
  <Data property="SubTagName">
    <String>ST1</String>
  </Data>
</Object>
```

FlowDetector

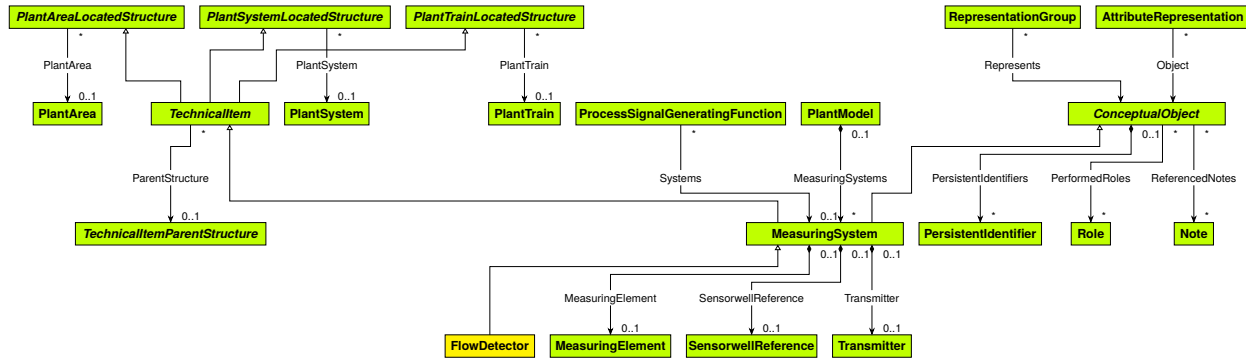
Plant::Instrumentation::FlowDetector

Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS1008719>

FLOW DETECTOR



Supertypes

- *MeasuringSystem*

Example in DEXPI XML

```
<Object type="Plant/Instrumentation.FlowDetector"/>
```

FlowInSignalOffPageConnector

Plant::Instrumentation::FlowInSignalOffPageConnector

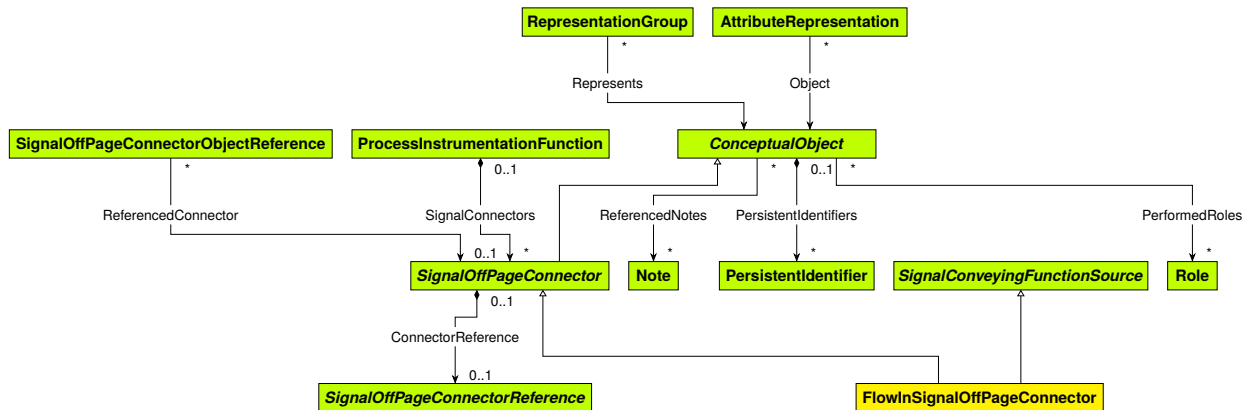
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/FlowInSignalOffPageConnector>

FLOW IN SIGNAL OFF PAGE CONNECTOR

A signal connector that indicates that a preceding part of a signal conveying function is represented somewhere else, either on the same PID, or on some other PID.



Supertypes

- *SignalConveyingFunctionSource*
- *SignalOffPageConnector*

Example in DEXPI XML

```
<Object type="Plant/Instrumentation.FlowInSignalOffPageConnector"/>
```

FlowOutSignalOffPageConnector

Plant::Instrumentation::FlowOutSignalOffPageConnector

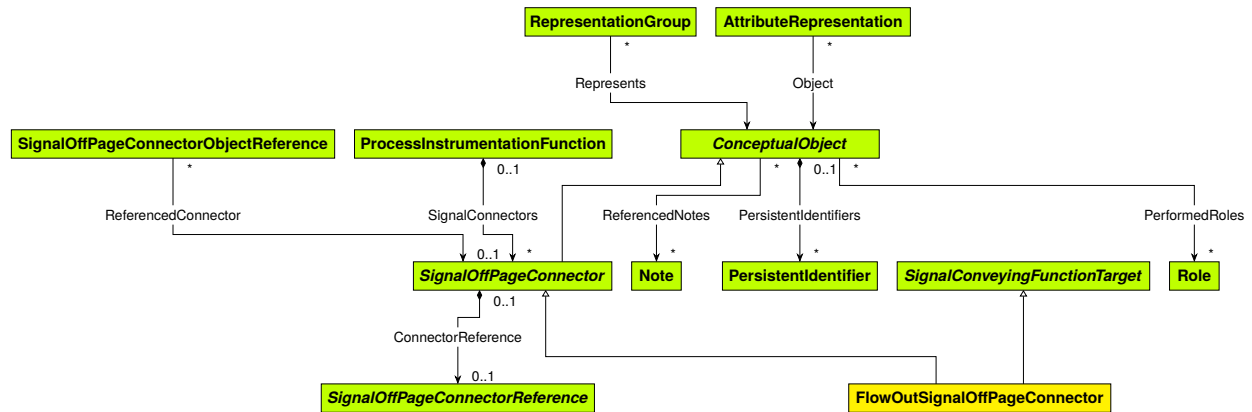
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/FlowOutSignalOffPageConnector>

FLOW OUT SIGNAL OFF PAGE CONNECTOR

A signal connector that indicates that a subsequent part of a signal conveying function is represented somewhere else, either on the same PID, or on some other PID.



Supertypes

- *SignalConveyingFunctionTarget*
- *SignalOffPageConnector*

Example in DEXPI XML

```
<Object type="Plant/Instrumentation.FlowOutSignalOffPageConnector"/>
```

InlineMeasuringElementReference

Plant::Instrumentation::InlineMeasuringElementReference

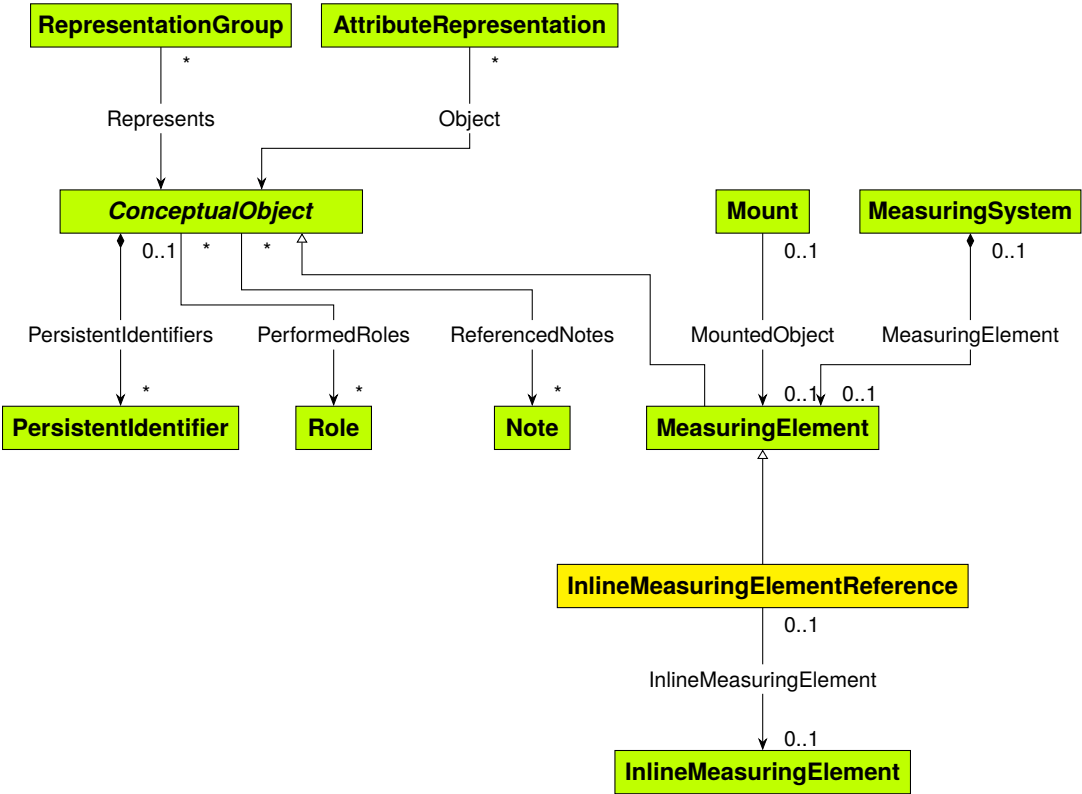
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/InlineMeasuringElementReference>

INLINE MEASURING ELEMENT REFERENCE

A reference to an *InlineMeasuringElement* that is part of a *PipingNetworkSegment*.



Supertypes

- *MeasuringElement*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite	Multiplicity
<i>InlineMeasuringElement</i>	0..1	<i>InlineMeasuringElement</i>	no	no	0..1	

InlineMeasuringElement

Reference Property

The *InlineMeasuringElement* referenced by the InlineMeasuringElementReference.

Type: *InlineMeasuringElement*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

Example in DEXPI XML

```
<Object type="Plant/Instrumentation.InlineMeasuringElementReference"/>
```

InstrumentationLoopFunction

Plant::Instrumentation::InstrumentationLoopFunction

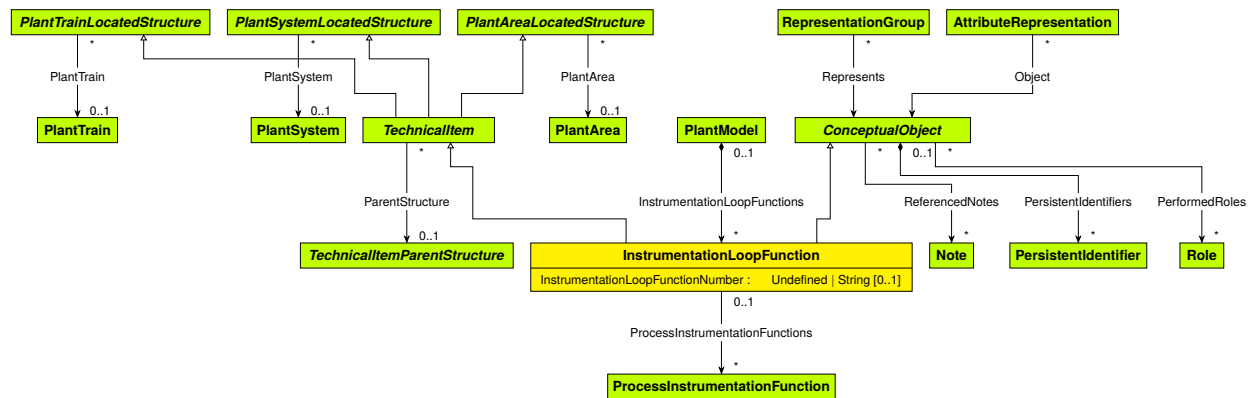
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/InstrumentationLoopFunction>

INSTRUMENTATION LOOP FUNCTION

An identified collection of related *ProcessInstrumentationFunctions* that interact for a known purpose.



Supertypes

- *ConceptualObject*
- *TechnicalItem*

Members

Reference Properties

Name	Multi- plicity	Type	Or- dered	Unique	Opposite plicity	Multi- plicity
<i>ProcessInstrumentation- Functions</i>	*	<i>ProcessInstrumentation- Function</i>	no	no	0..1	

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>InstrumentationLoopFunctionNumber</i>	0..1	<i>Undefined String</i>	no	no

InstrumentationLoopFunctionNumber

Data Property

The identification number of the *InstrumentationLoopFunction*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/InstrumentationLoopFunctionNumberAssignmentClass>

INSTRUMENTATION LOOP FUNCTION NUMBER ASSIGNMENT CLASS

Example

"4750.01"

ProcessInstrumentationFunctions

Reference Property

The *ProcessInstrumentationFunctions* that constitute this *InstrumentationLoopFunction*.

Type: *ProcessInstrumentationFunction*

Multiplicity: *

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

Example in DEXPI XML

```

<Object type="Plant/Instrumentation.InstrumentationLoopFunction">
  <Data property="InstrumentationLoopFunctionNumber">
    <String>4750.01</String>
  </Data>
</Object>

```

MeasuringElement

Plant::Instrumentation::MeasuringElement

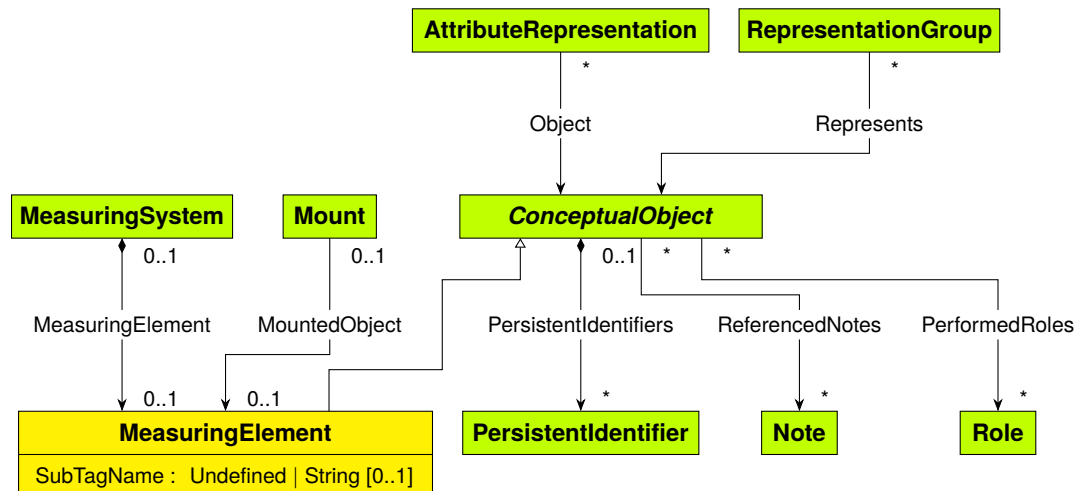
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/MeasuringElement>

MEASURING ELEMENT

An artefact used for the measurement of a Property.



Supertypes

- *ConceptualObject*

Subtypes

- *InlineMeasuringElementReference*
- *OfflineMeasuringElement*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>SubTagName</i>	0..1	<i>Undefined String</i>	no	no

SubTagName

Data Property

The sub tag name of the *MeasuringElement*.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SubTagNameAssignmentClass>

SUB TAG NAME ASSIGNMENT CLASS

Example

"ST1"

Example in DEXPI XML

```
<Object type="Plant/Instrumentation.MeasuringElement">
  <Data property="SubTagName">
    <String>ST1</String>
  </Data>
</Object>
```

MeasuringLineFunction

Plant::Instrumentation::MeasuringLineFunction

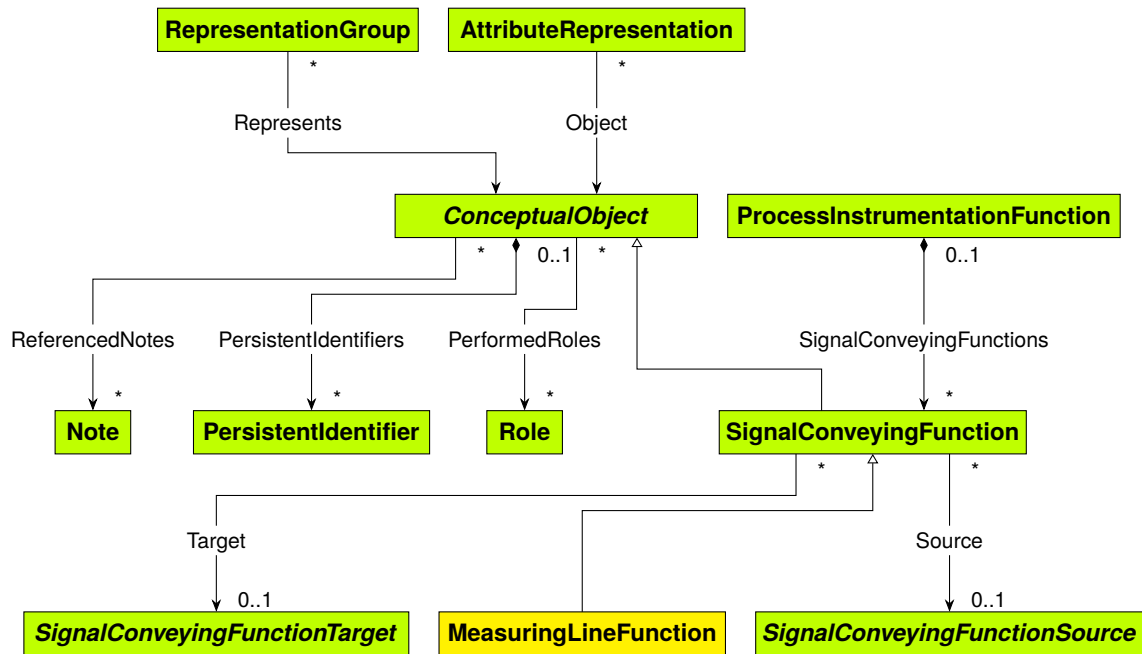
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/MeasuringLineFunction>

MEASURING LINE FUNCTION

Information flow function for measured values.



Supertypes

- *SignalConveyingFunction*

Example in DEXPI XML

```
<Object type="Plant/Instrumentation.MeasuringLineFunction"/>
```

MeasuringSystem

Plant::Instrumentation::MeasuringSystem

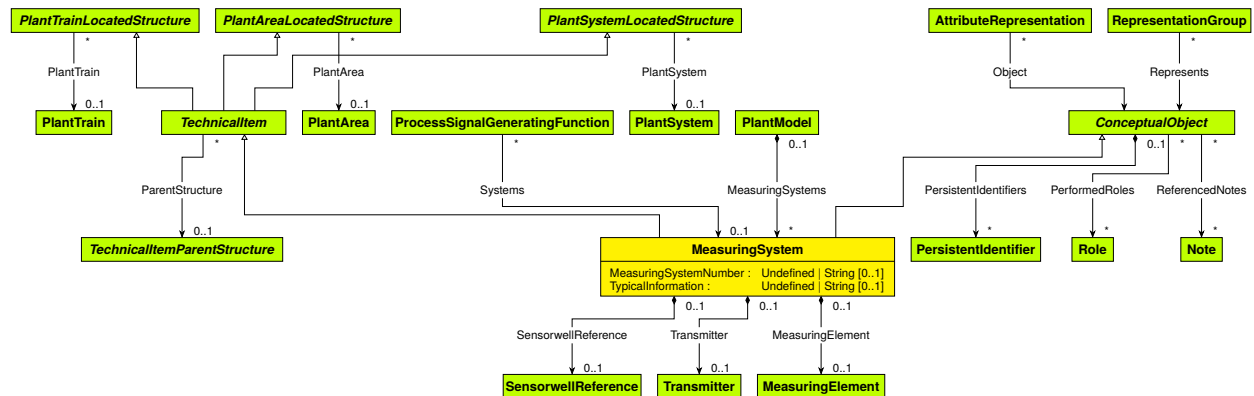
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/MeasuringSystem>

MEASURING SYSTEM

An assembly of artefacts that is designed to measure and transmit a property in order to fulfill one or more *ProcessSignalGeneratingFunctions*.



Supertypes

- *ConceptualObject*
- *TechnicalItem*

Subtypes

- *FlowDetector*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>MeasuringElement</i>	0..1	<i>MeasuringElement</i>	no
<i>SensorwellReference</i>	0..1	<i>SensorwellReference</i>	no
<i>Transmitter</i>	0..1	<i>Transmitter</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>MeasuringSystemNumber</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>TypicalInformation</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

MeasuringElement

Composition Property

The *MeasuringElement* of the MeasuringSystem.

Type: *MeasuringElement*

Multiplicity: 0..1

Ordered: no

MeasuringSystemNumber

Data Property

The number of the *MeasuringSystem*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/MeasuringSystemNumberAssignmentClass>

MEASURING SYSTEM NUMBER ASSIGNMENT CLASS

Example

"FE0001"

SensorwellReference

Composition Property

The *SensorwellReference* of the MeasuringSystem.

Type: *SensorwellReference*

Multiplicity: 0..1

Ordered: no

Transmitter

Composition Property

The *Transmitter* of the MeasuringSystem.

Type: *Transmitter*

Multiplicity: 0..1

Ordered: no

TypicalInformation

Data Property

Typical information about the *MeasuringSystem*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/TypicalInformationAssignmentClass>

TYPICAL INFORMATION ASSIGNMENT CLASS

Example

"F4"

Example in DEXPI XML

```
<Object type="Plant/Instrumentation.MeasuringSystem">
  <Data property="MeasuringSystemNumber">
    <String>FE0001</String>
  </Data>
  <Data property="TypicalInformation">
    <String>F4</String>
  </Data>
</Object>
```

OfflineMeasuringElement

Plant::Instrumentation::OfflineMeasuringElement

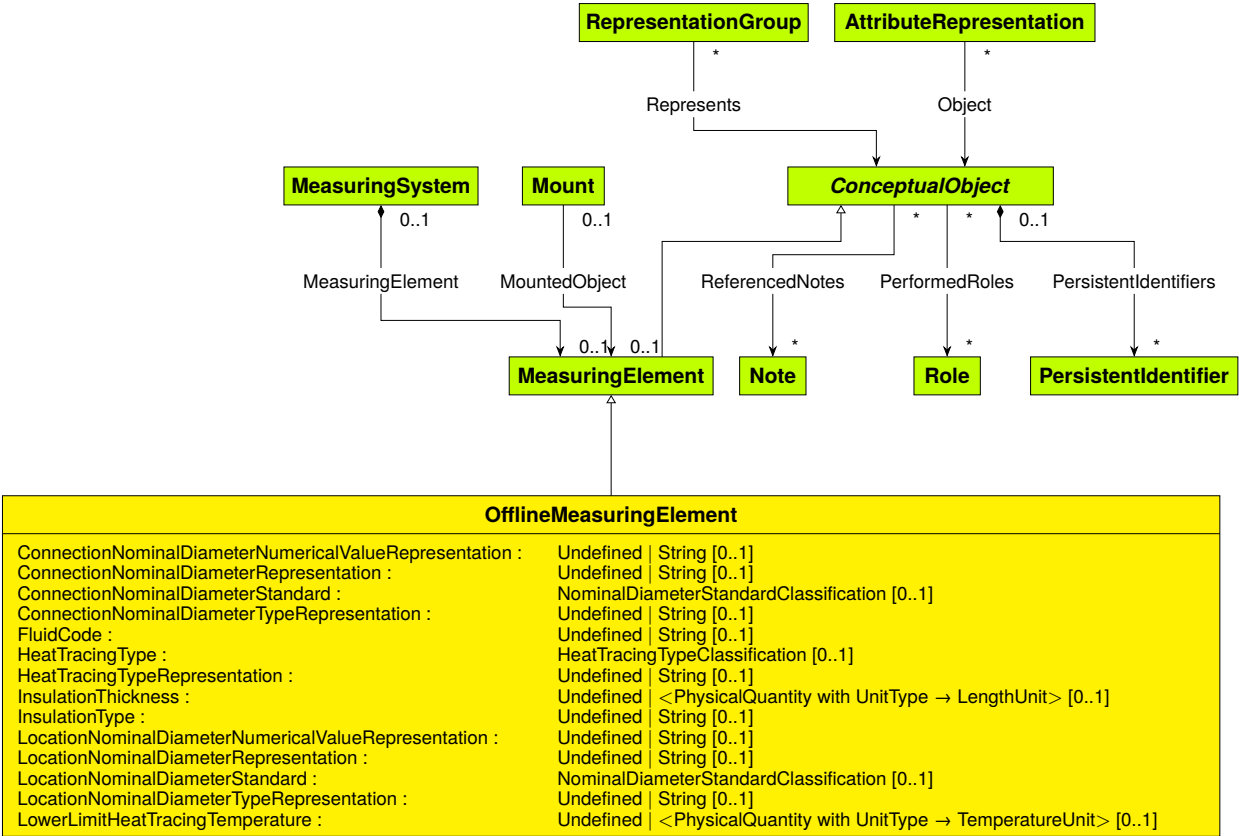
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/OfflineMeasuringElement>

OFFLINE MEASURING ELEMENT

A *MeasuringElement* that is not part of a *PipingNetworkSegment*.



Supertypes

- *MeasuringElement*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>ConnectionNominalDiameterNumericalValueRepresentation</i>	0..1	<i>Undefined String</i>	no	no
<i>ConnectionNominalDiameterRepresentation</i>	0..1	<i>Undefined String</i>	no	no
<i>ConnectionNominalDiameterStandard</i>	0..1	<i>NominalDiameterStandardClassification</i>	no	no
<i>ConnectionNominalDiameterTypeRepresentation</i>	0..1	<i>Undefined String</i>	no	no
<i>FluidCode</i>	0..1	<i>Undefined String</i>	no	no
<i>HeatTracingType</i>	0..1	<i>HeatTracingTypeClassification</i>	no	no
<i>HeatTracingTypeRepresentation</i>	0..1	<i>Undefined String</i>	no	no
<i>InsulationThickness</i>	0..1	<i>Undefined <PhysicalQuantity with Unit-Type → LengthUnit></i>	no	no
<i>InsulationType</i>	0..1	<i>Undefined String</i>	no	no
<i>LocationNominalDiameterNumericalValueRepresentation</i>	0..1	<i>Undefined String</i>	no	no
<i>LocationNominalDiameterRepresentation</i>	0..1	<i>Undefined String</i>	no	no
<i>LocationNominalDiameterStandard</i>	0..1	<i>NominalDiameterStandardClassification</i>	no	no
<i>LocationNominalDiameterTypeRepresentation</i>	0..1	<i>Undefined String</i>	no	no
<i>LowerLimitHeatTracingTemperature</i>	0..1	<i>Undefined <PhysicalQuantity with Unit-Type → TemperatureUnit></i>	no	no

ConnectionNominalDiameterNumericalValueRepresentation

Data Property

A readable representation of the numerical value of the nominal diameter at the device connection of the *OfflineMeasuringElement*. The purpose of this value is to give a textual representation of the nominal diameter to be used in the graphics of a PID.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ConnectionNominalDiameterNumericalValueRepresentationAssignmentClass>

CONNECTION NOMINAL DIAMETER NUMERICAL VALUE REPRESENTATION ASSIGNMENT CLASS

Example

"25"

ConnectionNominalDiameterRepresentation

Data Property

A readable representation of the nominal diameter at the device connection of the *OfflineMeasuringElement*. The purpose of this value is to give a textual representation of the nominal diameter to be used in the graphics of a PID.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ConnectionNominalDiameterRepresentationAssignmentClass>

CONNECTION NOMINAL DIAMETER REPRESENTATION ASSIGNMENT CLASS

Example

"DN 25"

ConnectionNominalDiameterStandard

Data Property

The nominal diameter of the device connection of the *OfflineMeasuringElement*, given as a reference to a nominal diameter standard and value.

Type: *NominalDiameterStandardClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ConnectionNominalDiameterStandardSpecialization>

CONNECTION NOMINAL DIAMETER STANDARD SPECIALIZATION

Example

Din2448ObjectDn25

ConnectionNominalDiameterTypeRepresentation

Data Property

A readable representation of the type of the nominal diameter at the device connection of the *OfflineMeasuringElement*. The purpose of this value is to give a textual representation of the nominal diameter to be used in the graphics of a PID.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ConnectionNominalDiameterTypeRepresentationAssignmentClass>

CONNECTION NOMINAL DIAMETER TYPE REPRESENTATION ASSIGNMENT CLASS

Example

"DN"

FluidCode**Data Property**

The identification code of the fluid related to the *OfflineMeasuringElement*. So far, DEXPI does not define restrictions for valid values.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/FluidCodeAssignmentClass>

FLUID CODE ASSIGNMENT CLASS

Example

"MNB"

HeatTracingType**Data Property**

A specialization indicating the heat tracing type related to the *OfflineMeasuringElement*.

Type: *HeatTracingTypeClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/HeatTracingTypeSpecialization>

HEAT TRACING TYPE SPECIALIZATION

Example

ElectricalHeatTracingSystem

HeatTracingTypeRepresentation**Data Property**

The heat tracing type related to the *OfflineMeasuringElement*, represented as a string.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/HeatTracingTypeRepresentationAssignmentClass>

HEAT TRACING TYPE REPRESENTATION ASSIGNMENT CLASS

Example

"E"

InsulationThickness**Data Property**

The insulation thickness of the *OfflineMeasuringElement*.

Type: *Undefined* | *<PhysicalQuantity* with *UnitType* → *LengthUnit**>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS4238040>

INSULATION THICKNESS

Example

PhysicalQuantity(
 Unit = *Millimetre*,
 Value = 40.0)

InsulationType**Data Property**

The identification code for the insulation type related to the *OfflineMeasuringElement*. So far, DEXPI does not define restrictions for valid values.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/InsulationTypeAssignmentClass>

INSULATION TYPE ASSIGNMENT CLASS

Example

"Q"

LocationNominalDiameterNumericalValueRepresentation**Data Property**

A readable representation of the numerical value of the nominal diameter at the location of the *OfflineMeasuringElement*. The purpose of this value is to give a textual representation of the nominal diameter to be used in the graphics of a PID.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LocationNominalDiameterNumericalValueRepresentationAssignmentClass>

LOCATION NOMINAL DIAMETER NUMERICAL VALUE REPRESENTATION ASSIGNMENT CLASS

Example

"25"

LocationNominalDiameterRepresentation

Data Property

A readable representation of the nominal diameter at the location of the *OfflineMeasuringElement*. The purpose of this value is to give a textual representation of the nominal diameter to be used in the graphics of a PID.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LocationNominalDiameterRepresentationAssignmentClass>

LOCATION NOMINAL DIAMETER REPRESENTATION ASSIGNMENT CLASS

Example

"DN 25"

LocationNominalDiameterStandard

Data Property

The nominal diameter of the location of the *OfflineMeasuringElement*, given as a reference to a nominal diameter standard and value.

Type: *NominalDiameterStandardClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LocationNominalDiameterStandardSpecialization>

LOCATION NOMINAL DIAMETER STANDARD SPECIALIZATION

Example

Din2448ObjectDn25

LocationNominalDiameterTypeRepresentation

Data Property

A readable representation of the type of the nominal diameter at the location of the *OfflineMeasuringElement*. The purpose of this value is to give a textual representation of the nominal diameter to be used in the graphics of a PID.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LocationNominalDiameterTypeRepresentationAssignmentClass>

LOCATION NOMINAL DIAMETER TYPE REPRESENTATION ASSIGNMENT CLASS

Example

"DN"

LowerLimitHeatTracingTemperature

Data Property

The lower limit for the temperature that a heat tracing system must ensure for the *OfflineMeasuringElement*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{TemperatureUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LowerLimitHeatTracingTemperature>

LOWER LIMIT HEAT TRACING TEMPERATURE

Example

```
PhysicalQuantity(
  Unit = DegreeCelsius,
  Value = 100.0)
```

Example in DEXPI XML

```
<Object type="Plant/Instrumentation.OfflineMeasuringElement">
  <Data property="ConnectionNominalDiameterNumericalValueRepresentation">
    <String>25</String>
  </Data>
  <Data property="ConnectionNominalDiameterRepresentation">
    <String>DN 25</String>
  </Data>
  <Data property="ConnectionNominalDiameterStandard">
    <DataReference data="Plant/Enumerations.NominalDiameterStandardClassification.
    ↪Din24480ObjectDn25"/>
  </Data>
</Object>
```

(continues on next page)

(continued from previous page)

```

</Data>
<Data property="ConnectionNominalDiameterTypeRepresentation">
  <String>DN</String>
</Data>
<Data property="FluidCode">
  <String>MNb</String>
</Data>
<Data property="HeatTracingType">
  <DataReference data="Plant/Enumerations.HeatTracingTypeClassification.
↪ElectricalHeatTracingSystem"/>
</Data>
<Data property="HeatTracingTypeRepresentation">
  <String>E</String>
</Data>
<Data property="InsulationThickness">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.LengthUnit.Millimetre"/>
    </Data>
    <Data property="Value">
      <Double>40.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="InsulationType">
  <String>Q</String>
</Data>
<Data property="LocationNominalDiameterNumericalValueRepresentation">
  <String>25</String>
</Data>
<Data property="LocationNominalDiameterRepresentation">
  <String>DN 25</String>
</Data>
<Data property="LocationNominalDiameterStandard">
  <DataReference data="Plant/Enumerations.NominalDiameterStandardClassification.
↪Din2448ObjectDn25"/>
</Data>
<Data property="LocationNominalDiameterTypeRepresentation">
  <String>DN</String>
</Data>
<Data property="LowerLimitHeatTracingTemperature">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.TemperatureUnit.DegreeCelsius"/>
    </Data>
    <Data property="Value">
      <Double>100.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>

```

OperatedValveReference

Plant::Instrumentation::OperatedValveReference

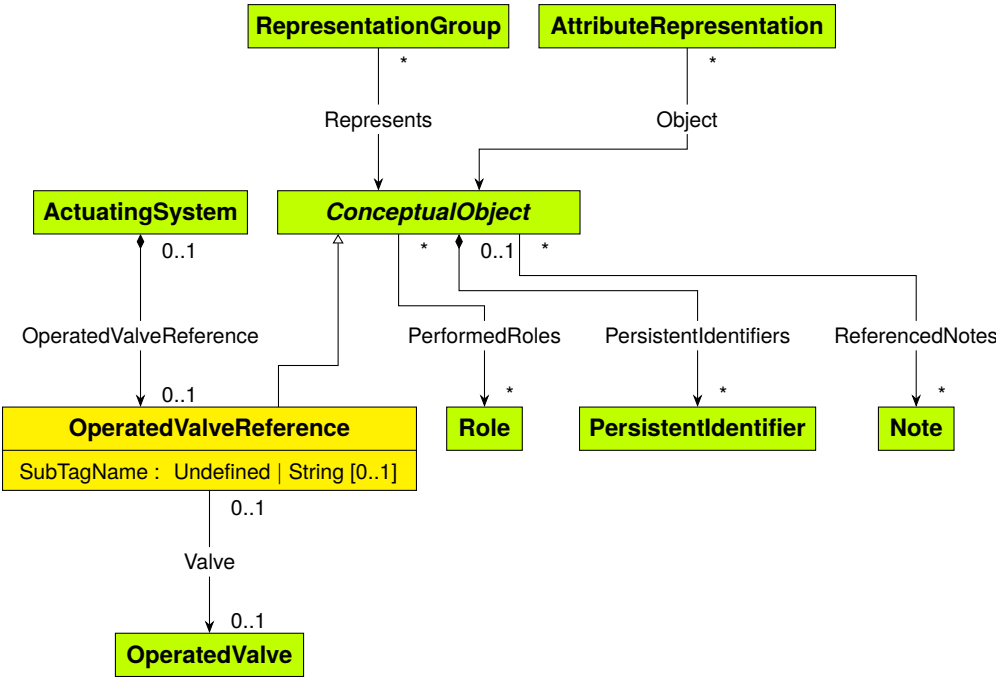
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/OperatedValveReference>

OPERATED VALVE REFERENCE

A reference to an *OperatedValve*.



Supertypes

- *ConceptualObject*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Valve</i>	0..1	<i>OperatedValve</i>	no	no	0..1

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>SubTagName</i>	0..1	<i>Undefined String</i>	no	no

SubTagName

Data Property

The sub tag name of the *OperatedValveReference*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SubTagNameAssignmentClass>

SUB TAG NAME ASSIGNMENT CLASS

Example

"ST1"

Valve

Reference Property

The actual valve referenced by the *OperatedValveReference*.

Type: *OperatedValve*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

Example in DEXPI XML

```
<Object type="Plant/Instrumentation.OperatedValveReference">
  <Data property="SubTagName">
    <String>ST1</String>
  </Data>
</Object>
```

Positioner

Plant::Instrumentation::Positioner

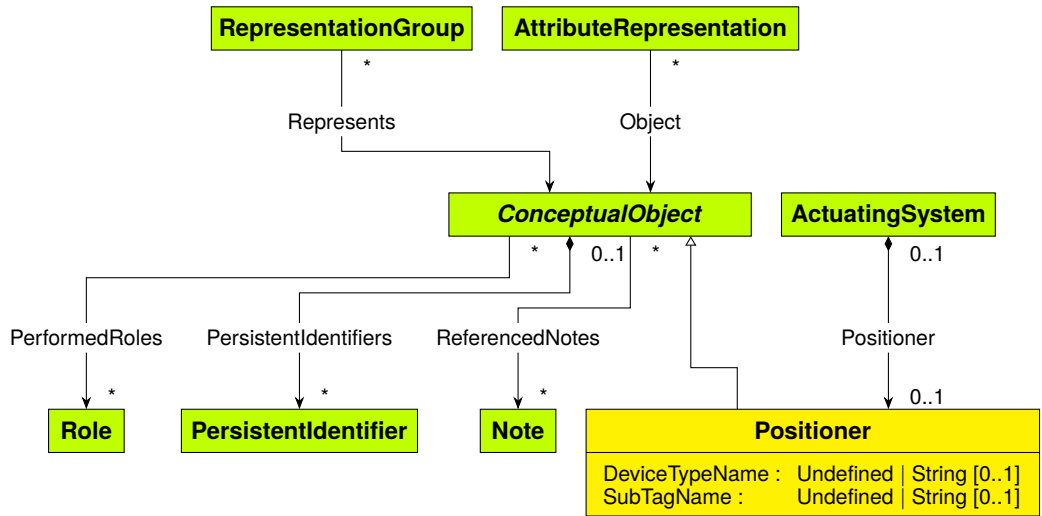
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/Positioner>

POSITIONER

A positioner.



Supertypes

- *ConceptualObject*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DeviceTypeName</i>	0..1	<i>Undefined String</i>	no	no
<i>SubTagName</i>	0..1	<i>Undefined String</i>	no	no

DeviceTypeName

Data Property

The device type of the *Positioner*.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DeviceTypeNameAssignmentClass>

DEVICE TYPE NAME ASSIGNMENT CLASS

Example

"pressure transmitter"

SubTagName

Data Property

The sub tag name of the *Positioner*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SubTagNameAssignmentClass>

SUB TAG NAME ASSIGNMENT CLASS

Example

"ST1"

Example in DEXPI XML

```

<Object type="Plant/Instrumentation.Positioner">
  <Data property="DeviceTypeName">
    <String>pressure transmitter</String>
  </Data>
  <Data property="SubTagName">
    <String>ST1</String>
  </Data>
</Object>

```

ProcessControlFunction

Plant::Instrumentation::ProcessControlFunction

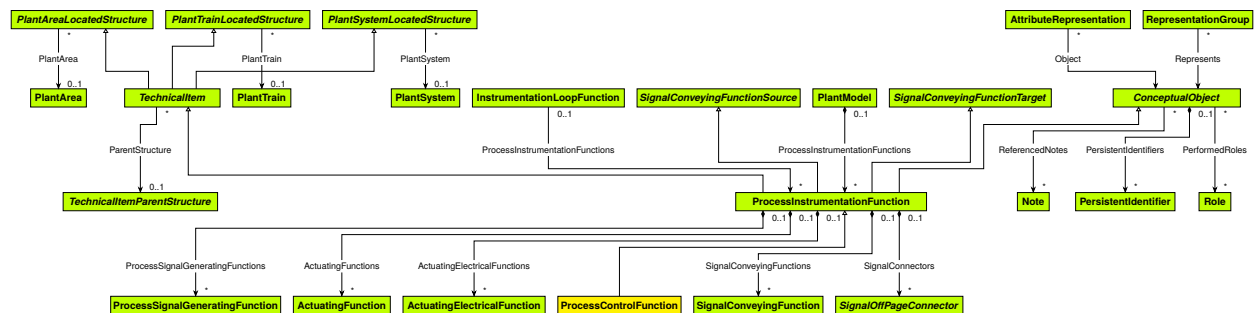
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/ProcessControlFunction>

PROCESS CONTROL FUNCTION

A requirement for control structures relating to Process Engineering.



Supertypes

- *ProcessInstrumentationFunction*

Example in DEXPI XML

```
<Object type="Plant/Instrumentation.ProcessControlFunction"/>
```

ProcessInstrumentationFunction

Plant::Instrumentation::ProcessInstrumentationFunction

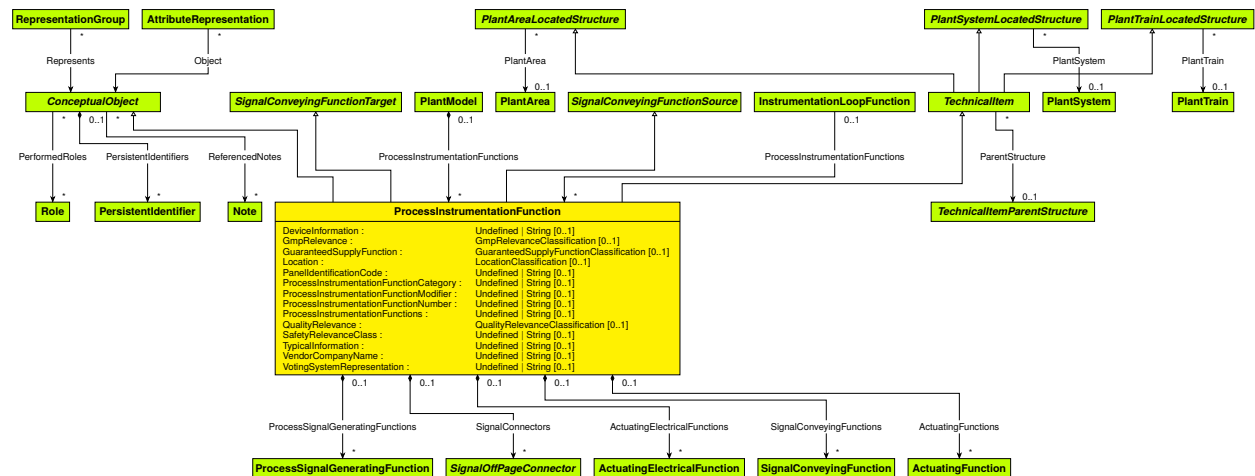
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/ProcessInstrumentationFunction>

PROCESS INSTRUMENTATION FUNCTION

A requirement for instrumentation and/or control structures relating to Process Engineering.



Supertypes

- *ConceptualObject*
- *SignalConveyingFunctionSource*
- *SignalConveyingFunctionTarget*
- *TechnicalItem*

Subtypes

- *ProcessControlFunction*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>ActuatingElectricalFunctions</i>	*	<i>ActuatingElectricalFunction</i>	no
<i>ActuatingFunctions</i>	*	<i>ActuatingFunction</i>	no
<i>ProcessSignalGeneratingFunctions</i>	*	<i>ProcessSignalGeneratingFunction</i>	no
<i>SignalConnectors</i>	*	<i>SignalOffPageConnector</i>	no
<i>SignalConveyingFunctions</i>	*	<i>SignalConveyingFunction</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DeviceInformation</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>GmpRelevance</i>	0..1	<i>GmpRelevanceClassification</i>	no	no
<i>GuaranteedSupplyFunction</i>	0..1	<i>GuaranteedSupplyFunctionClassification</i>	no	no
<i>Location</i>	0..1	<i>LocationClassification</i>	no	no
<i>PanelIdentificationCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>ProcessInstrumentationFunctionCategory</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>ProcessInstrumentationFunctionModifier</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>ProcessInstrumentationFunctionNumber</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>ProcessInstrumentationFunctions</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>QualityRelevance</i>	0..1	<i>QualityRelevanceClassification</i>	no	no
<i>SafetyRelevanceClass</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>TypicalInformation</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>VendorCompanyName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>VotingSystemRepresentation</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

ActuatingElectricalFunctions

Composition Property

The *ActuatingElectricalFunctions* that are part of this *ProcessInstrumentationFunction*.

Type: *ActuatingElectricalFunction*

Multiplicity: *

Ordered: no

ActuatingFunctions

Composition Property

The *ActuatingFunctions* that are part of this *ProcessInstrumentationFunction*.

Type: *ActuatingFunction*

Multiplicity: *

Ordered: no

DeviceInformation

Data Property

Device information the *ProcessInstrumentationFunction*, e.g., for a detector.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DeviceInformationAssignmentClass>

DEVICE INFORMATION ASSIGNMENT CLASS

Example

"MDM"

GmpRelevance

Data Property

A classification indicating if the *ProcessInstrumentationFunction* is relevant for GMP (good manufacturing practice).

Type: *GmpRelevanceClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/GmpRelevanceSpecialization>

GMP RELEVANCE SPECIALIZATION

Example

GmpRelevantFunction

GuaranteedSupplyFunction

Data Property

A classification indicating if the *ProcessInstrumentationFunction* is a guaranteed supply function.

Type: *GuaranteedSupplyFunctionClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/GuaranteedSupplyFunctionSpecialization>

GUARANTEED SUPPLY FUNCTION SPECIALIZATION

Example*GuaranteedSupplyFunction***Location****Data Property**

A specialization indicating the location of the *ProcessInstrumentationFunction*.

Type: *LocationClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LocationSpecialization>

LOCATION SPECIALIZATION

Example*Field***PanelIdentificationCode****Data Property**

The panel identification code of the *ProcessInstrumentationFunction*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PanelIdentificationCodeAssignmentClass>

PANEL IDENTIFICATION CODE ASSIGNMENT CLASS

Example

"P 3A"

ProcessInstrumentationFunctionCategory**Data Property**

The function category of the *ProcessInstrumentationFunction*. The value is a string, typically one or two letters. Recent standards for PIDs normally enforce a single letter from a fixed list. However, there are no formal DEXPI restrictions for valid strings.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ProcessInstrumentationFunctionCategoryAssignmentClass>

PROCESS INSTRUMENTATION FUNCTION CATEGORY ASSIGNMENT CLASS

Example

"H"

ProcessInstrumentationFunctionModifier

Data Property

The modifier of the *ProcessInstrumentationFunction*. The value is a string, typically a single letter, e.g., D for difference. So far, there are no formal DEXPI restrictions for valid strings.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ProcessInstrumentationFunctionModifierAssignmentClass>

PROCESS INSTRUMENTATION FUNCTION MODIFIER ASSIGNMENT CLASS

Example

"D"

ProcessInstrumentationFunctionNumber

Data Property

A unique identifier for the *ProcessInstrumentationFunction*. If the *ProcessInstrumentationFunction* is part of a *InstrumentationLoopFunction*, the identifier of the *ProcessInstrumentationFunction* usually contains the identifier of the *InstrumentationLoopFunction* (see *InstrumentationLoopFunctionNumber*).

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ProcessInstrumentationFunctionNumberAssignmentClass>

PROCESS INSTRUMENTATION FUNCTION NUMBER ASSIGNMENT CLASS

Example

"H4750.01"

ProcessInstrumentationFunctions

Data Property

Additional functions of the *ProcessInstrumentationFunction* (i.e., in addition to the function category, see *ProcessInstrumentationFunctionCategory*).

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ProcessInstrumentationFunctionsAssignmentClass>

PROCESS INSTRUMENTATION FUNCTIONS ASSIGNMENT CLASS

Example

"HS"

ProcessSignalGeneratingFunctions

Composition Property

The *ProcessSignalGeneratingFunctions* that are part of this *ProcessInstrumentationFunction*.

Type: *ProcessSignalGeneratingFunction*

Multiplicity: *

Ordered: no

QualityRelevance

Data Property

A classification indicating if the *ProcessInstrumentationFunction* is quality relevant.

Type: *QualityRelevanceClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/QualityRelevanceSpecialization>

QUALITY RELEVANCE SPECIALIZATION

Example

QualityRelevantFunction

SafetyRelevanceClass

Data Property

The safety relevance class the *ProcessInstrumentationFunction*.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SafetyRelevanceClassAssignmentClass>

SAFETY RELEVANCE CLASS ASSIGNMENT CLASS

Example

"SIL3"

SignalConnectors**Composition Property**

The Plant.Instrumentation.SignalOffPageConnectors that are part of this ProcessInstrumentationFunction.

Type: *SignalOffPageConnector*

Multiplicity: *

Ordered: no

SignalConveyingFunctions**Composition Property**

The *SignalConveyingFunctions* that are part of this ProcessInstrumentationFunction.

Type: *SignalConveyingFunction*

Multiplicity: *

Ordered: no

TypicalInformation**Data Property**

Typical information about the *ProcessInstrumentationFunction*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/TypicalInformationAssignmentClass>

TYPICAL INFORMATION ASSIGNMENT CLASS

Example

"F4"

VendorCompanyName**Data Property**

The vendor company name the *ProcessInstrumentationFunction*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/VendorCompanyNameAssignmentClass>

VENDOR COMPANY NAME ASSIGNMENT CLASS

Example

"Emerson"

VotingSystemRepresentation

Data Property

A representation of the voting system of the *ProcessInstrumentationFunction*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/VotingSystemRepresentationAssignmentClass>

VOTING SYSTEM REPRESENTATION ASSIGNMENT CLASS

Example

"1002"

Example in DEXPI XML

```
<Object type="Plant/Instrumentation.ProcessInstrumentationFunction">
  <Data property="DeviceInformation">
    <String>MDM</String>
  </Data>
  <Data property="GmpRelevance">
    <DataReference data="Plant/Enumerations.GmpRelevanceClassification.
↳ GmpRelevantFunction"/>
  </Data>
  <Data property="GuaranteedSupplyFunction">
    <DataReference data="Plant/Enumerations.GuaranteedSupplyFunctionClassification.
↳ GuaranteedSupplyFunction"/>
  </Data>
  <Data property="Location">
    <DataReference data="Plant/Enumerations.LocationClassification.Field"/>
  </Data>
  <Data property="PanelIdentificationCode">
    <String>P 3A</String>
  </Data>
  <Data property="ProcessInstrumentationFunctionCategory">
    <String>H</String>
  </Data>
  <Data property="ProcessInstrumentationFunctionModifier">
    <String>D</String>
  </Data>
</Object>
```

(continues on next page)

(continued from previous page)

```

</Data>
<Data property="ProcessInstrumentationFunctionNumber">
  <String>H4750.01</String>
</Data>
<Data property="ProcessInstrumentationFunctions">
  <String>HS</String>
</Data>
<Data property="QualityRelevance">
  <DataReference data="Plant/Enumerations.QualityRelevanceClassification.
  QualityRelevantFunction"/>
</Data>
<Data property="SafetyRelevanceClass">
  <String>SIL3</String>
</Data>
<Data property="TypicalInformation">
  <String>F4</String>
</Data>
<Data property="VendorCompanyName">
  <String>Emerson</String>
</Data>
<Data property="VotingSystemRepresentation">
  <String>1002</String>
</Data>
</Object>

```

ProcessSignalGeneratingFunction

Plant::Instrumentation::ProcessSignalGeneratingFunction

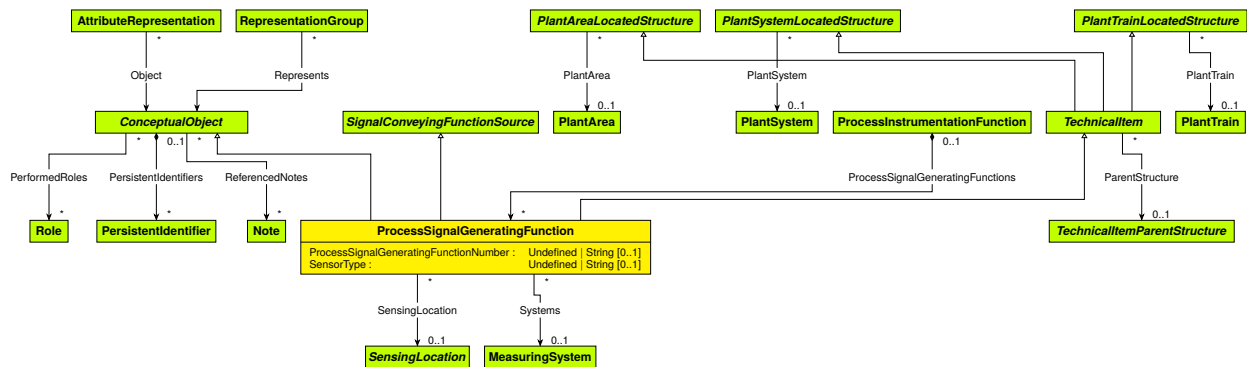
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/ProcessSignalGeneratingFunction>

PROCESS SIGNAL GENERATING FUNCTION

A function for instrumentation and/or control structures relating to Process Engineering.



Supertypes

- *ConceptualObject*
- *SignalConveyingFunctionSource*
- *TechnicalItem*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>SensingLocation</i>	0..1	<i>SensingLocation</i>	no	no	*
<i>Systems</i>	0..1	<i>MeasuringSystem</i>	no	no	*

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>ProcessSignalGeneratingFunctionNumber</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>SensorType</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

ProcessSignalGeneratingFunctionNumber

Data Property

An identifier for the *ProcessSignalGeneratingFunction*. It usually contains the identifier of the *ProcessInstrumentationFunction* that includes the *ProcessSignalGeneratingFunction* (see *ProcessInstrumentationFunctionNumber*).

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ProcessSignalGeneratingFunctionNumberAssignmentClass>

PROCESS SIGNAL GENERATING FUNCTION NUMBER ASSIGNMENT CLASS

Example

"TT4750.03"

SensingLocation

Reference Property

The sensing location of the *ProcessSignalGeneratingFunction*.

Type: *SensingLocation*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

SensorType

Data Property

The sensor type of the *ProcessSignalGeneratingFunction*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SensorTypeAssignmentClass>

SENSOR TYPE ASSIGNMENT CLASS

Example

"MDM"

Systems

Reference Property

The MeasuringSystem that implements the ProcessSignalGeneratingFunction.

Type: *MeasuringSystem*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

Example in DEXPI XML

```
<Object type="Plant/Instrumentation.ProcessSignalGeneratingFunction">
  <Data property="ProcessSignalGeneratingFunctionNumber">
    <String>TT4750.03</String>
  </Data>
  <Data property="SensorType">
    <String>MDM</String>
  </Data>
</Object>
```

SensingLocation

Plant::Instrumentation::SensingLocation

Overview

Abstract Class

An object that can act as a *SensingLocation* of a *ProcessSignalGeneratingFunction*.

SensingLocation

Subtypes

- *Mount*
- *Nozzle*
- *PipingComponent*
- *PipingNetworkSegment*

SensorwellReference

Plant::Instrumentation::SensorwellReference

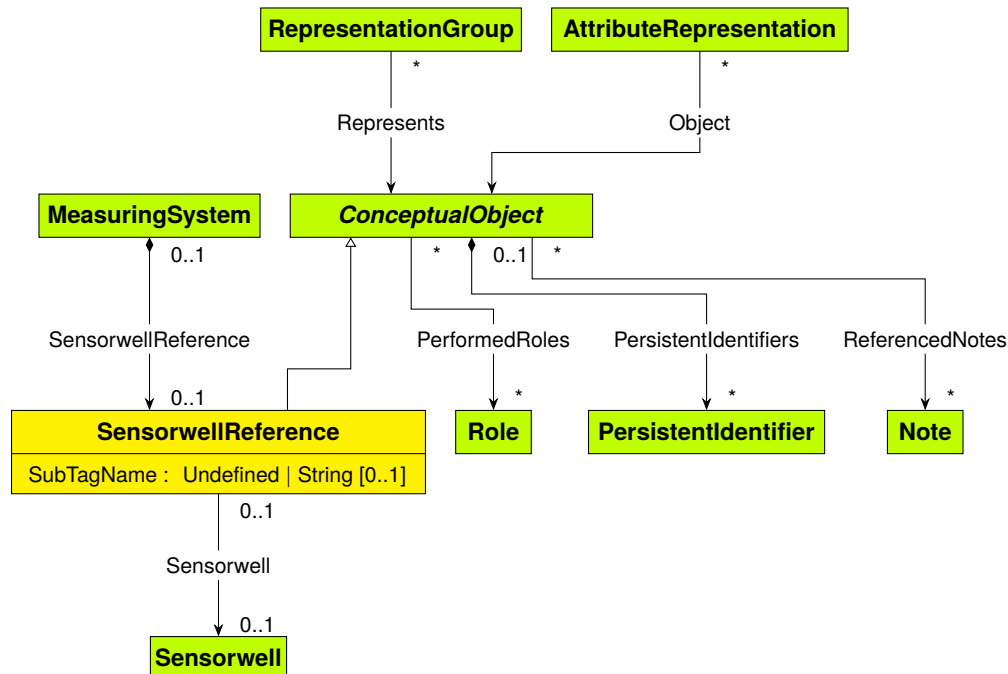
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/SensorwellReference>

SENSORWELL REFERENCE

A reference to a *Sensorwell* that is part of a *PipingNetworkSegment*.



Supertypes

- *ConceptualObject*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Sensorwell</i>	0..1	<i>Sensorwell</i>	no	no	0..1

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>SubTagName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

Sensorwell

Reference Property

The *Sensorwell* referenced by the SensorwellReference.

Type: *Sensorwell*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

SubTagName

Data Property

The sub tag name of the *SensorwellReference*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SubTagNameAssignmentClass>

SUB TAG NAME ASSIGNMENT CLASS

Example

"ST1"

Example in DEXPI XML

```
<Object type="Plant/Instrumentation.SensorwellReference">
  <Data property="SubTagName">
    <String>ST1</String>
  </Data>
</Object>
```

SignalConveyingFunction

Plant::Instrumentation::SignalConveyingFunction

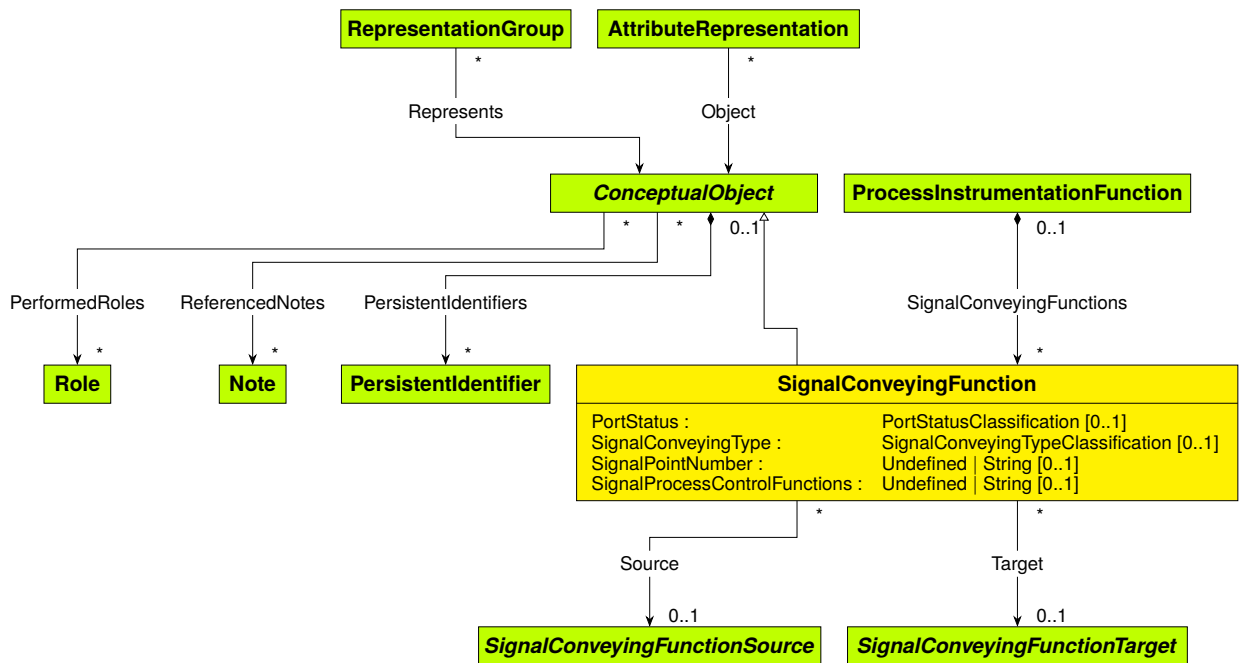
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/SignalConveyingFunction>

SIGNAL CONVEYING FUNCTION

A function for conveying a signal.



Supertypes

- *ConceptualObject*

Subtypes

- *MeasuringLineFunction*
- *SignalLineFunction*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Source</i>	0..1	<i>SignalConveyingFunctionSource</i>	no	no	*
<i>Target</i>	0..1	<i>SignalConveyingFunctionTarget</i>	no	no	*

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>PortStatus</i>	0..1	<i>PortStatusClassification</i>	no	no
<i>SignalConveyingType</i>	0..1	<i>SignalConveyingTypeClassification</i>	no	no
<i>SignalPointNumber</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>SignalProcessControlFunctions</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

PortStatus

Data Property

A classification indicating the port status of the *SignalConveyingFunction*.

Type: *PortStatusClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PortStatusSpecialization>

PORT STATUS SPECIALIZATION

Example

StatusHighHighPort

SignalConveyingType

Data Property

A classification indicating the signal conveying type of the *SignalConveyingFunction*.

Type: *SignalConveyingTypeClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SignalConveyingTypeSpecialization>

SIGNAL CONVEYING TYPE SPECIALIZATION

Example

ElectricalSignalConveying

SignalPointNumber

Data Property

The signal point number of the *SignalConveyingFunction*. Typical values are 1 to 6.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SignalPointNumberAssignmentClass>

SIGNAL POINT NUMBER ASSIGNMENT CLASS

Example

"2"

SignalProcessControlFunctions

Data Property

The process control functions of the *SignalConveyingFunction*. Values are combinations of characters.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SignalProcessControlFunctionsAssignmentClass>

SIGNAL PROCESS CONTROL FUNCTIONS ASSIGNMENT CLASS

Example

"SA"

Source

Reference Property

The source of the signal conveyed by this *SignalConveyingFunction*.

Type: *SignalConveyingFunctionSource*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

Target

Reference Property

The target of the signal conveyed by this *SignalConveyingFunction*.

Type: *SignalConveyingFunctionTarget*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

Example in DEXPI XML

```

<Object type="Plant/Instrumentation.SignalConveyingFunction">
  <Data property="PortStatus">
    <DataReference data="Plant/Enumerations.PortStatusClassification.StatusHighHighPort"/>
  </Data>
  <Data property="SignalConveyingType">
    <DataReference data="Plant/Enumerations.SignalConveyingTypeClassification.
ElectricalSignalConveying"/>
  </Data>
  <Data property="SignalPointNumber">
    <String>2</String>
  </Data>
  <Data property="SignalProcessControlFunctions">
    <String>SA</String>
  </Data>
</Object>

```

SignalConveyingFunctionSource

Plant::Instrumentation::SignalConveyingFunctionSource

Overview

Abstract Class

An object than can act as the *Source* of a *SignalConveyingFunction*.

SignalConveyingFunctionSource

Subtypes

- *ActuatingFunction*
- *FlowInSignalOffPageConnector*
- *ProcessInstrumentationFunction*
- *ProcessSignalGeneratingFunction*

SignalConveyingFunctionTarget

Plant::Instrumentation::SignalConveyingFunctionTarget

Overview

Abstract Class

An object than can act as the *Target* of a *SignalConveyingFunction*.

SignalConveyingFunctionTarget

Subtypes

- *ActuatingElectricalFunction*
- *ActuatingFunction*
- *FlowOutSignalOffPageConnector*
- *ProcessInstrumentationFunction*

SignalLineFunction

Plant::Instrumentation::SignalLineFunction

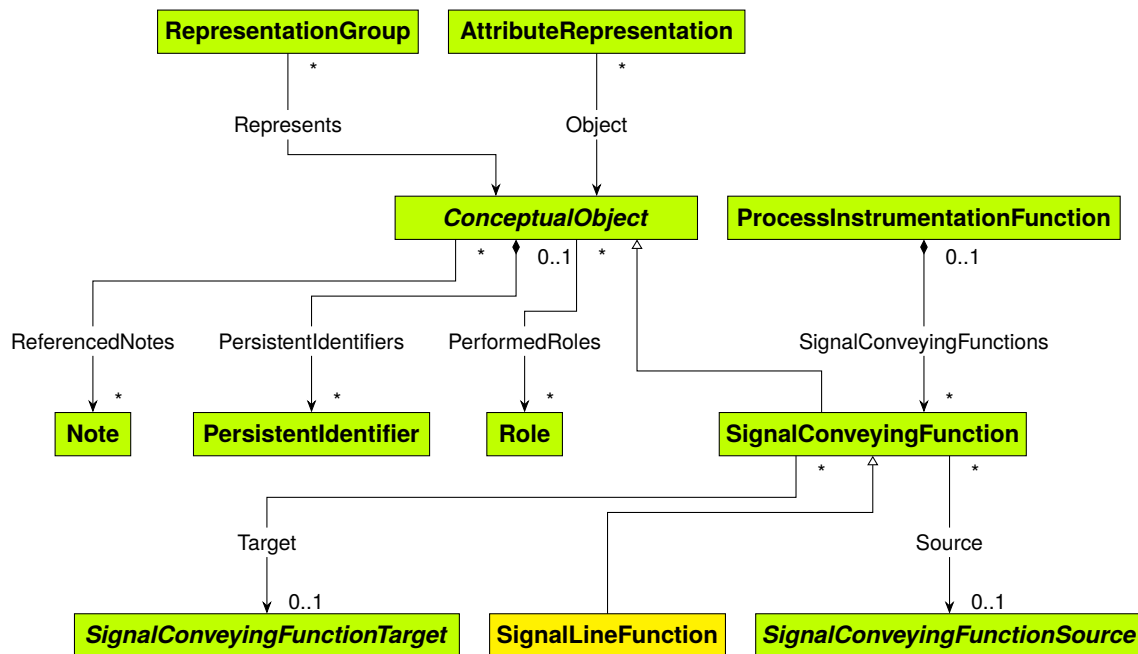
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/SignalLineFunction>

SIGNAL LINE FUNCTION

Information flow function for signals.



Supertypes

- *SignalConveyingFunction*

Example in DEXPI XML

<Object type="Plant/Instrumentation.SignalLineFunction"/>

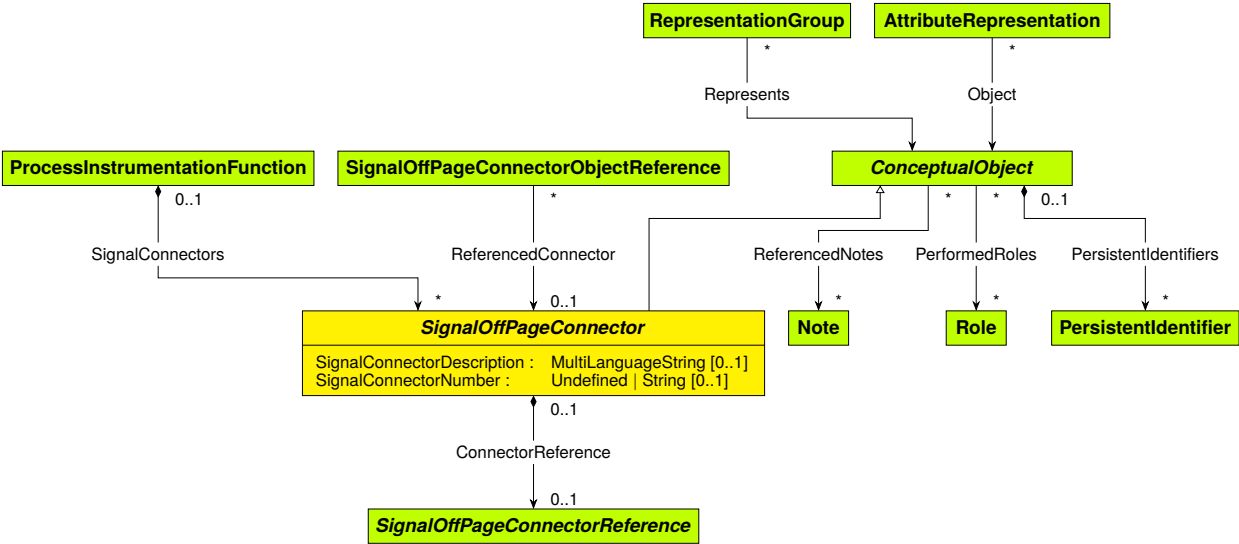
SignalOffPageConnector

Plant::Instrumentation::SignalOffPageConnector

Overview

Abstract Class

A signal connector that indicates that a *SignalConveyingFunction* is continued elsewhere, either on the same P&ID or on another P&ID. Graphically, it is usually represented as an arrow.



Supertypes

- *ConceptualObject*

Subtypes

- *FlowInSignalOffPageConnector*
- *FlowOutSignalOffPageConnector*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>ConnectorReference</i>	0..1	<i>SignalOffPageConnectorReference</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>SignalConnectorDescription</i>	0..1	<i>MultiLanguageString</i>	no	no
<i>SignalConnectorNumber</i>	0..1	<i>Undefined String</i>	no	no

ConnectorReference

Composition Property

A reference indicating to which other *SignalOffPageConnector* this *SignalOffPageConnector* is connected.

Type: *SignalOffPageConnectorReference*

Multiplicity: 0..1

Ordered: no

SignalConnectorDescription

Data Property

A description of the *SignalOffPageConnector*.

Type: *MultiLanguageString*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SignalConnectorDescriptionAssignmentClass>

SIGNAL CONNECTOR DESCRIPTION ASSIGNMENT CLASS

Example

```
" (('en', 'reactor temperature'), ('de', 'Reaktortemperatur')) "
```

SignalConnectorNumber

Data Property

The number of the *SignalOffPageConnector*.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SignalConnectorNumberAssignmentClass>

SIGNAL CONNECTOR NUMBER ASSIGNMENT CLASS

Example

"321"

SignalOffPageConnectorObjectReference

Plant::Instrumentation::SignalOffPageConnectorObjectReference

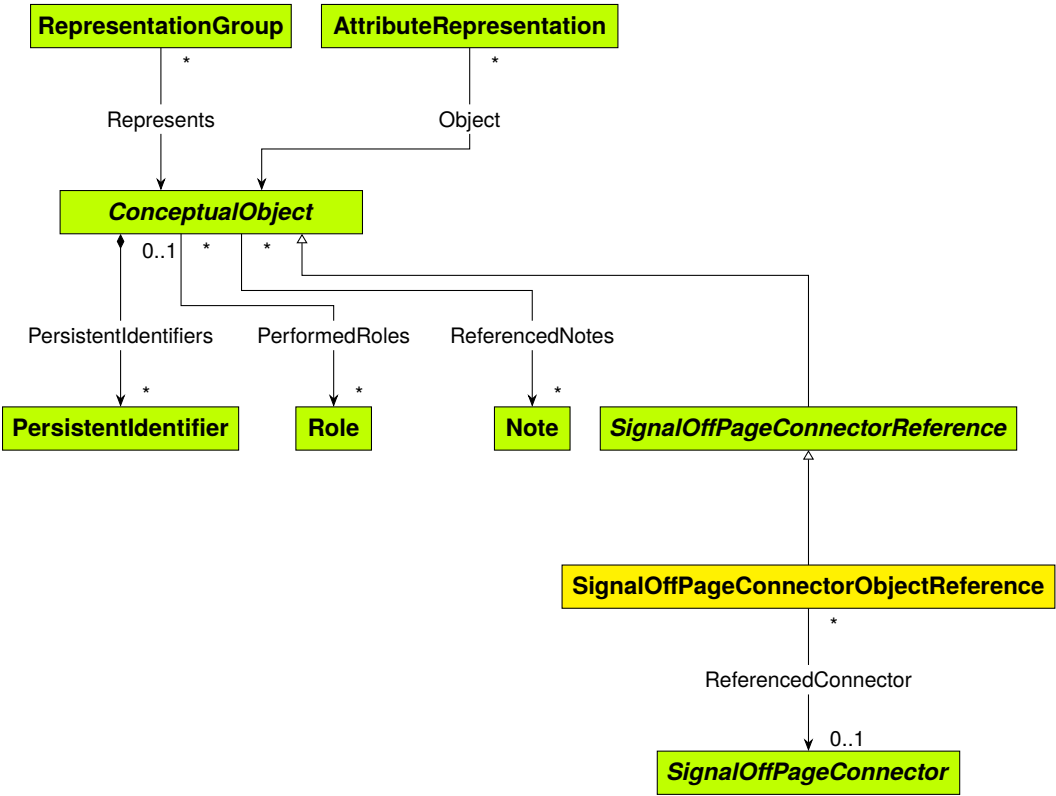
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/SignalOffPageConnectorObjectReference>

SIGNAL OFF PAGE CONNECTOR OBJECT REFERENCE

A reference to a *SignalOffPageConnector* by an association.



Supertypes

- *SignalOffPageConnectorReference*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>ReferencedConnector</i>	0..1	<i>SignalOffPageConnector</i>	no	no	*

ReferencedConnector

Reference Property

The *SignalOffPageConnector* referenced.

Type: *SignalOffPageConnector*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

Example in DEXPI XML

```
<Object type="Plant/Instrumentation.SignalOffPageConnectorObjectReference"/>
```

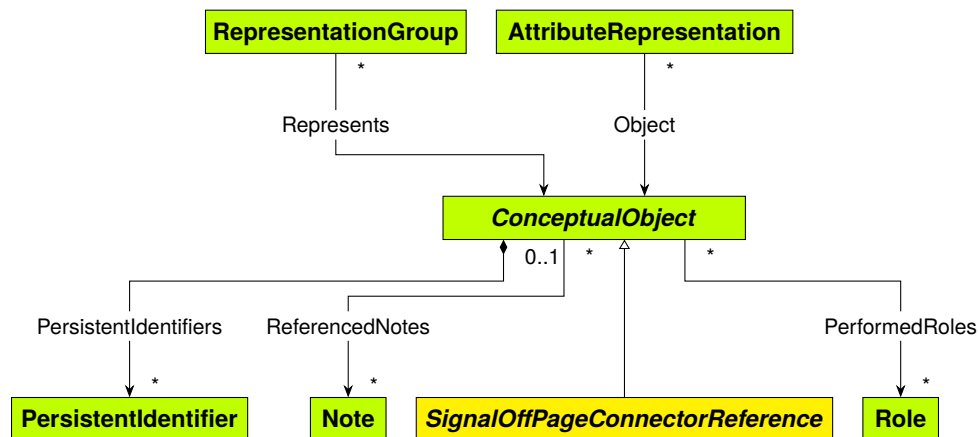
SignalOffPageConnectorReference

Plant::Instrumentation::SignalOffPageConnectorReference

Overview

Abstract Class

A reference to a *SignalOffPageConnector*.



Supertypes

- *ConceptualObject*

Subtypes

- *SignalOffPageConnectorObjectReference*
- *SignalOffPageConnectorReferenceByNumber*

SignalOffPageConnectorReferenceByNumber

Plant::Instrumentation::SignalOffPageConnectorReferenceByNumber

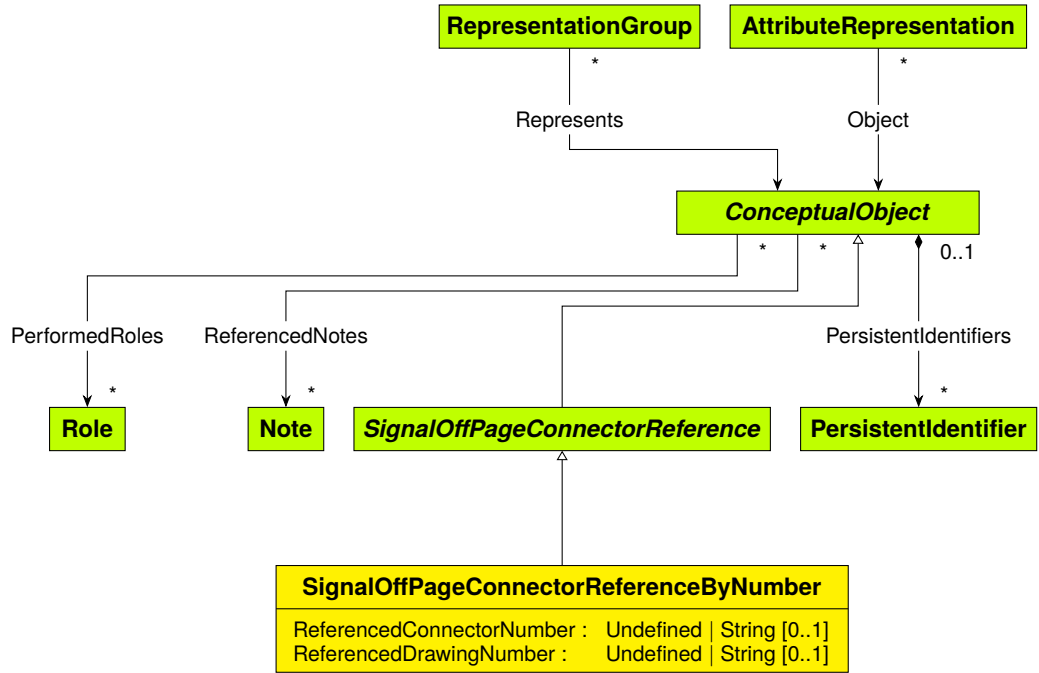
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/SignalOffPageConnectorReferenceByNumber>

SIGNAL OFF PAGE CONNECTOR REFERENCE BY NUMBER

A reference to a *SignalOffPageConnector* by drawing and connector number.



Supertypes

- *SignalOffPageConnectorReference*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>ReferencedConnectorNumber</i>	0..1	<i>Undefined String</i>	no	no
<i>ReferencedDrawingNumber</i>	0..1	<i>Undefined String</i>	no	no

ReferencedConnectorNumber

Data Property

The connector number of the referenced connector.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ReferencedConnectorNumberAssignmentClass>

REFERENCED CONNECTOR NUMBER ASSIGNMENT CLASS

Example

"97"

ReferencedDrawingNumber

Data Property

The Core.Diagram.MetaData.DrawingNumberAssignmentClass of the PID that contains the referenced connector.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ReferencedDrawingNumberAssignmentClass>

REFERENCED DRAWING NUMBER ASSIGNMENT CLASS

Example

"123/A93"

Example in DEXPI XML

```
<Object type="Plant/Instrumentation.SignalOffPageConnectorReferenceByNumber">
  <Data property="ReferencedConnectorNumber">
    <String>97</String>
  </Data>
  <Data property="ReferencedDrawingNumber">
    <String>123/A93</String>
  </Data>
</Object>
```

Transmitter

Plant::Instrumentation::Transmitter

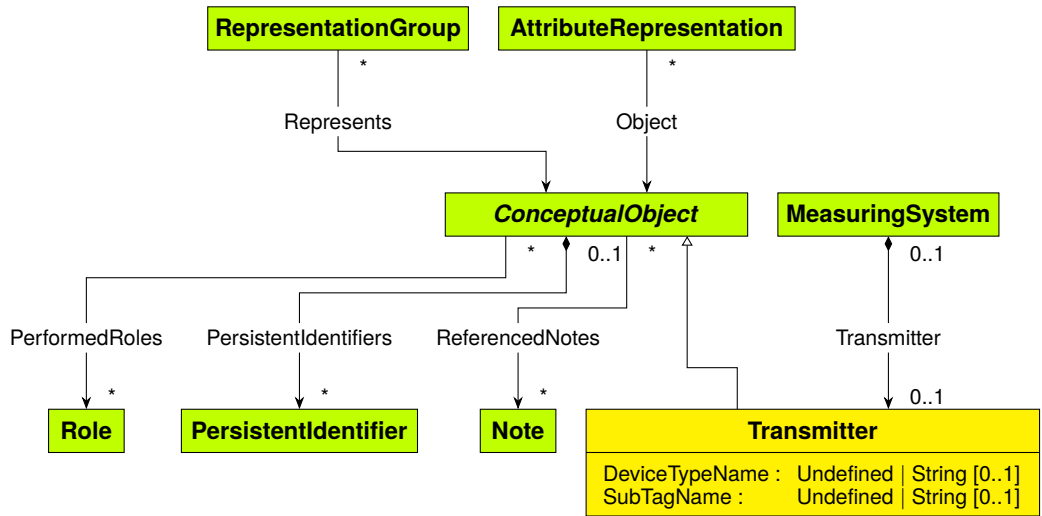
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS267929>

TRANSMITTER

A detecting instrument that generates a process variable signal and converts it into an output signal.



Supertypes

- *ConceptualObject*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DeviceTypeName</i>	0..1	<i>Undefined String</i>	no	no
<i>SubTagName</i>	0..1	<i>Undefined String</i>	no	no

DeviceTypeName

Data Property

The device type of the *Transmitter*.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DeviceTypeNameAssignmentClass>

DEVICE TYPE NAME ASSIGNMENT CLASS

Example

"pressure transmitter"

SubTagName

Data Property

The sub tag name of the *Transmitter*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SubTagNameAssignmentClass>

SUB TAG NAME ASSIGNMENT CLASS

Example

"ST1"

Example in DEXPI XML

```
<Object type="Plant/Instrumentation.Transmitter">
  <Data property="DeviceTypeName">
    <String>pressure transmitter</String>
  </Data>
  <Data property="SubTagName">
    <String>ST1</String>
  </Data>
</Object>
```

Plant::Instrumentation

Overview

Package

Members

Classes

Name	Type
<i>ActuatingElectricalFunction</i>	Concrete Class
<i>ActuatingElectricalLocation</i>	Abstract Class
<i>ActuatingElectricalSystem</i>	Concrete Class
<i>ActuatingFunction</i>	Concrete Class
<i>ActuatingSystem</i>	Concrete Class
<i>ControlledActuator</i>	Concrete Class
<i>ElectronicFrequencyConverter</i>	Concrete Class
<i>FlowDetector</i>	Concrete Class
<i>FlowInSignalOffPageConnector</i>	Concrete Class
<i>FlowOutSignalOffPageConnector</i>	Concrete Class
<i>InlineMeasuringElementReference</i>	Concrete Class
<i>InstrumentationLoopFunction</i>	Concrete Class

continues on next page

Table 7 – continued from previous page

Name	Type
<i>MeasuringElement</i>	Concrete Class
<i>MeasuringLineFunction</i>	Concrete Class
<i>MeasuringSystem</i>	Concrete Class
<i>OfflineMeasuringElement</i>	Concrete Class
<i>OperatedValveReference</i>	Concrete Class
<i>Positioner</i>	Concrete Class
<i>ProcessControlFunction</i>	Concrete Class
<i>ProcessInstrumentationFunction</i>	Concrete Class
<i>ProcessSignalGeneratingFunction</i>	Concrete Class
<i>SensingLocation</i>	Abstract Class
<i>SensorwellReference</i>	Concrete Class
<i>SignalConveyingFunction</i>	Concrete Class
<i>SignalConveyingFunctionSource</i>	Abstract Class
<i>SignalConveyingFunctionTarget</i>	Abstract Class
<i>SignalLineFunction</i>	Concrete Class
<i>SignalOffPageConnector</i>	Abstract Class
<i>SignalOffPageConnectorObjectReference</i>	Concrete Class
<i>SignalOffPageConnectorReference</i>	Abstract Class
<i>SignalOffPageConnectorReferenceByNumber</i>	Concrete Class
<i>Transmitter</i>	Concrete Class

3.2.4 Piping

AngleBallValve

Plant::Piping::AngleBallValve

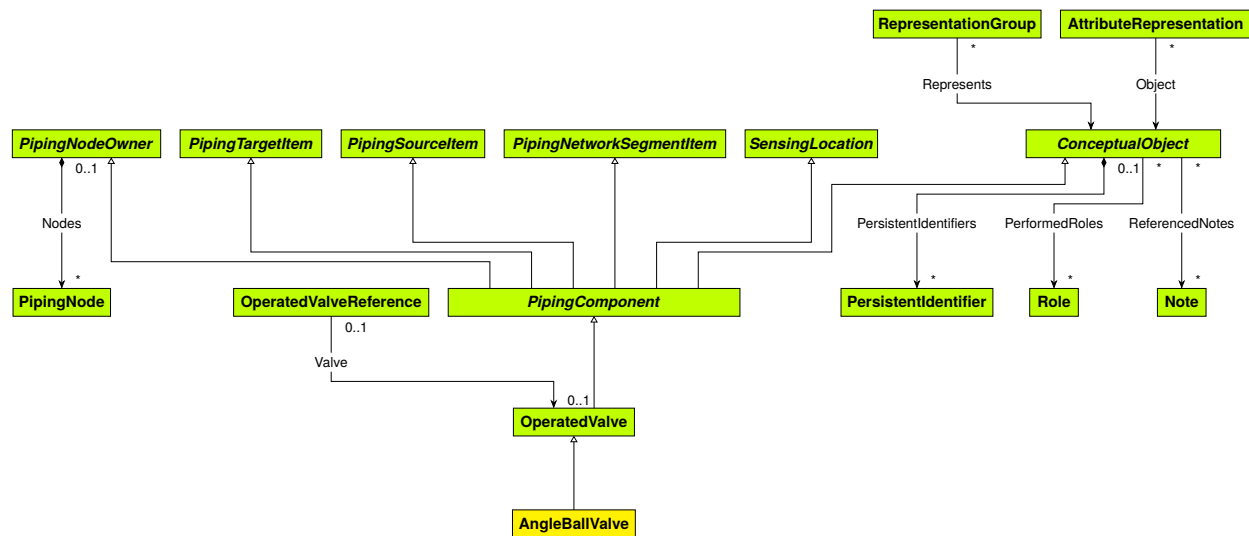
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/AngleBallValve>

ANGLE BALL VALVE

A valve that has valve ports which are not in-line and that has a ball closure member.



Supertypes

- *OperatedValve*

Example in DEXPI XML

```
<Object type="Plant/Piping.AngleBallValve"/>
```

AngleGlobeValve

Plant::Piping::AngleGlobeValve

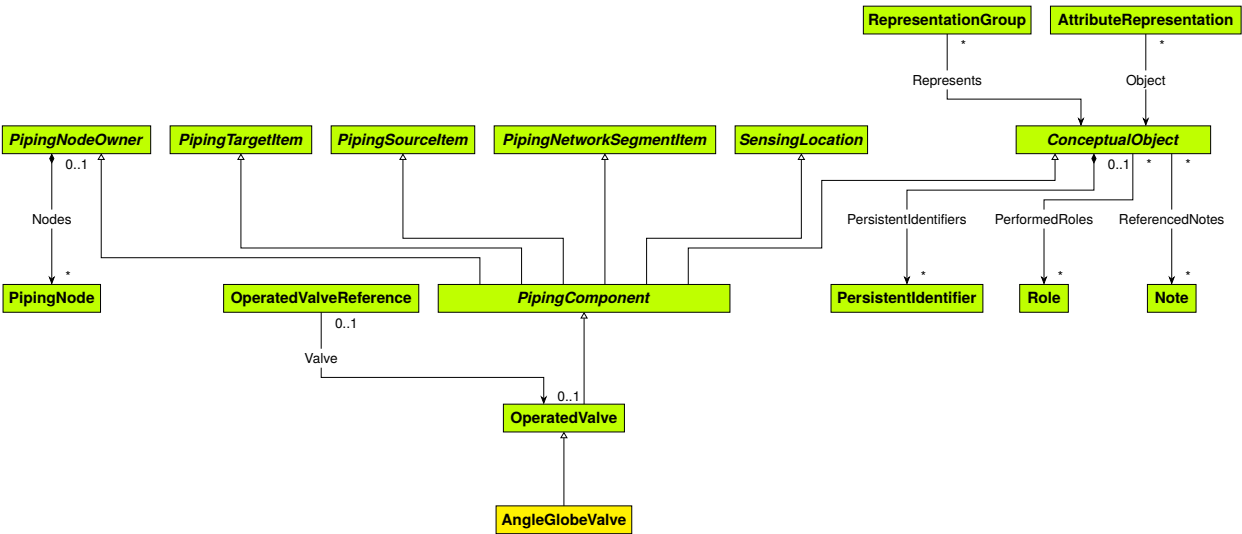
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS882944>

ANGLE GLOBE VALVE

A globe valve that deviates from the in-line design, i.e. with a body shape designed to adjust the flow direction with a specified angle relative to the straight through-flow an in-line valve would have provided for (from <http://data.posccaesar.org/rdl/RDS882944>).



Supertypes

- *OperatedValve*

Example in DEXPI XML

```
<Object type="Plant/Piping.AngleGlobeValve"/>
```

AnglePlugValve

Plant::Piping::AnglePlugValve

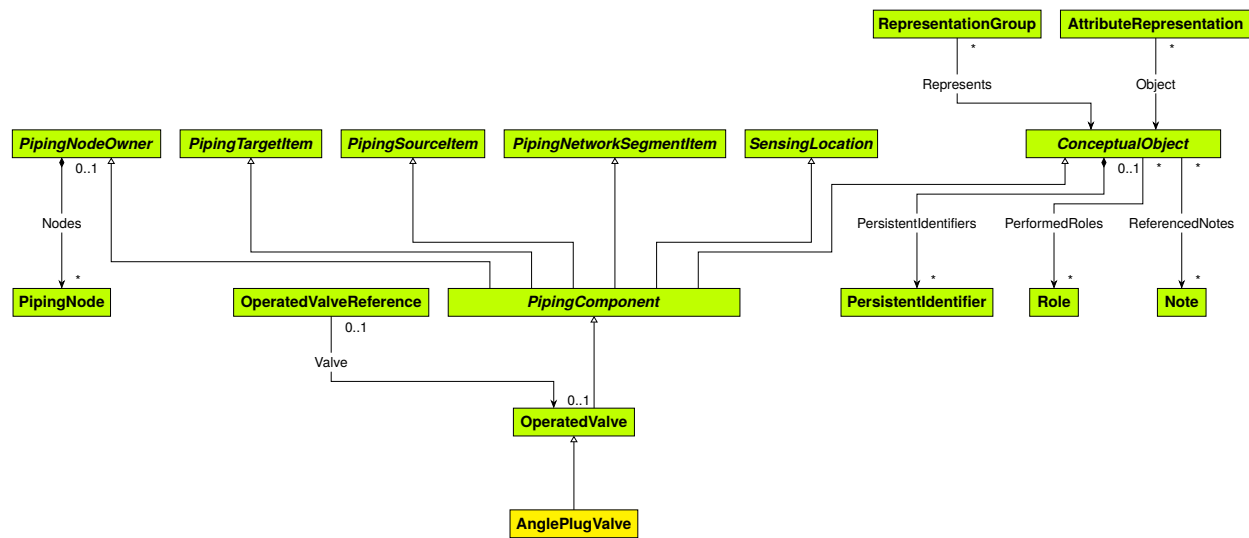
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/AnglePlugValve>

ANGLE PLUG VALVE

A valve that has valve ports which are not in-line and that has a quarter turn action in which the closure member is a cylindrical or tapered plug which operates by rotating on its axis and sealing against a downstream seat.



Supertypes

- *OperatedValve*

Example in DEXPI XML

```
<Object type="Plant/Piping.AnglePlugValve"/>
```

AngleValve

Plant::Piping::AngleValve

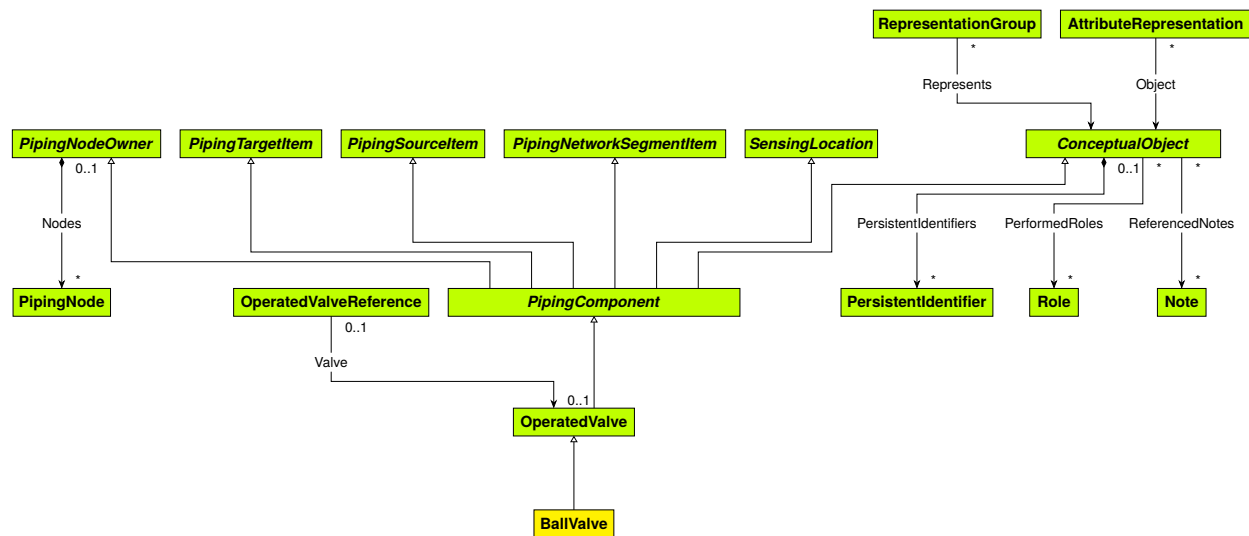
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS5789384>

ANGLE VALVE

A valve that has valve ports which are not in-line (from <http://data.posccaesar.org/rdl/RDS5789384>).



Supertypes

- *OperatedValve*

Example in DEXPI XML

```
<Object type="Plant/Piping.BallValve"/>
```

BlindFlange

Plant::Piping::BlindFlange

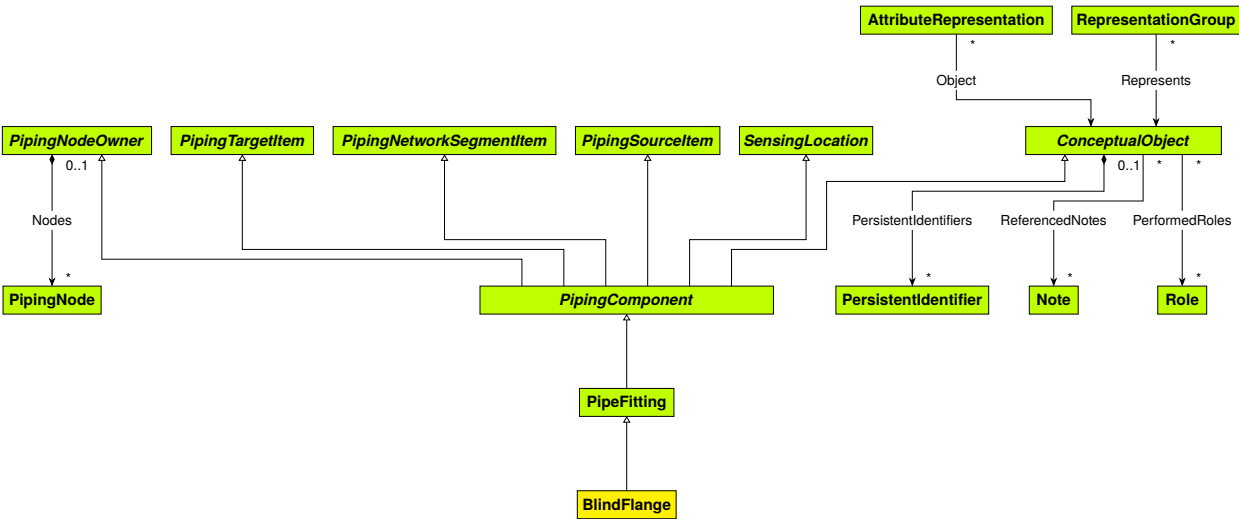
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS414719>

BLIND FLANGE

A pipe flange that is without a central opening and used to shut off a flanged pipe end (from <http://data.posccaesar.org/rdl/RDS414719>).



Supertypes

- *PipeFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.BlindFlange"/>
```

BreatherValve

Plant::Piping::BreatherValve

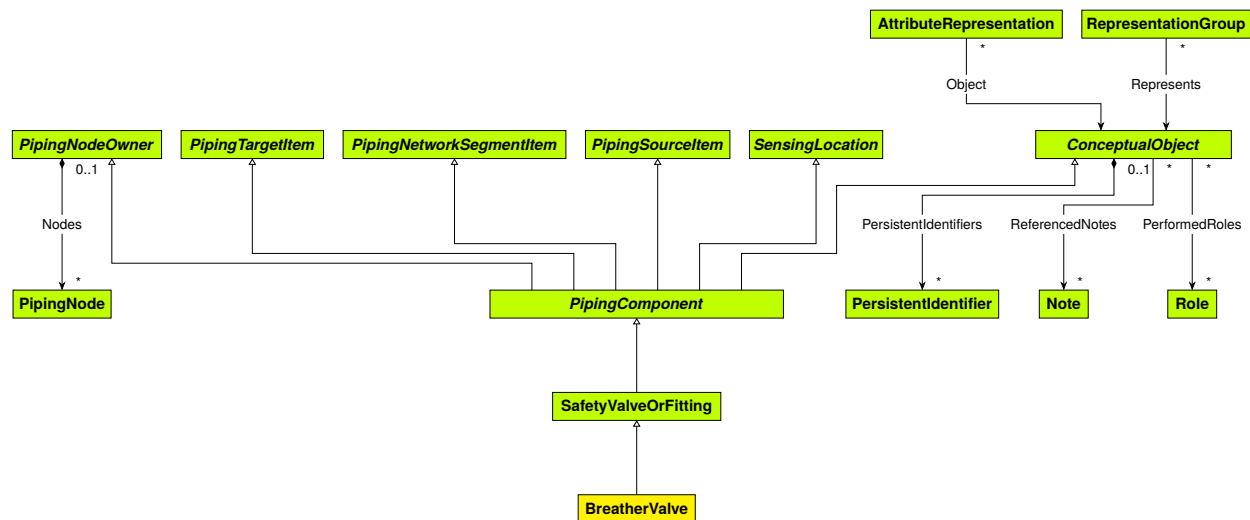
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/BreatherValve>

BREATHER VALVE

A BreatherValve is PRESSURE RELIEF DEVICE and a VACUUM RELIEF DEVICE that automatically releases excess pressure or admits pressure to prevent a vacuum (from <http://data.15926.org/rdl/RDS6330659>).



Supertypes

- *SafetyValveOrFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.BreatherValve"/>
```

ButterflyValve

Plant::Piping::ButterflyValve

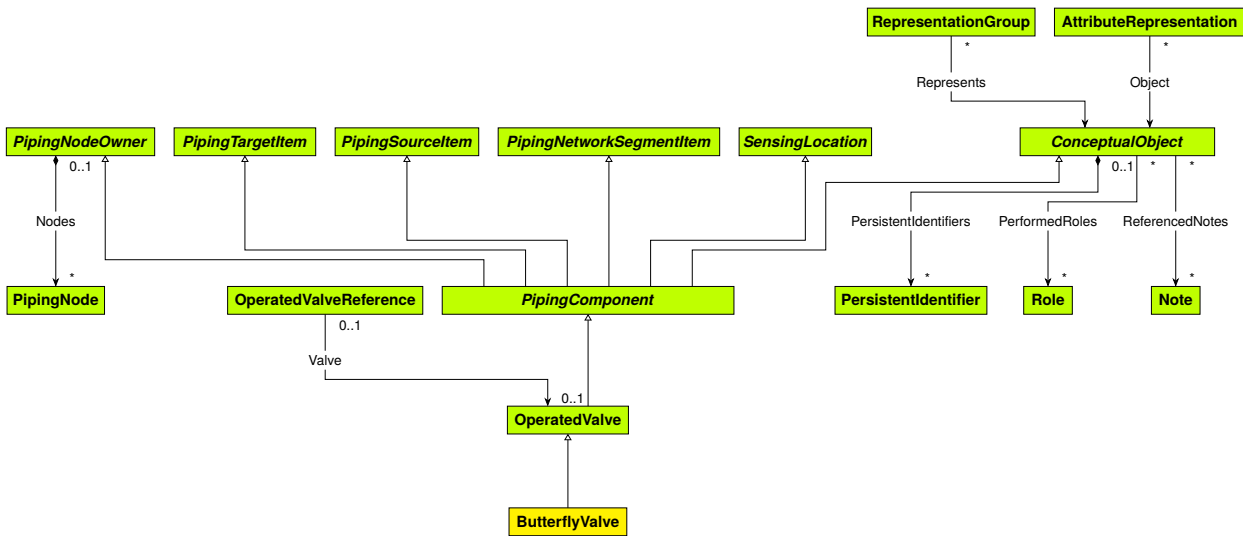
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS416609>

BUTTERFLY VALVE

A rotary valve that has a closure member of a disc type with a shaft parallel, or near parallel, to the plane of the disc, with an axis of rotation transverse to the flow direction (from <http://data.posccaesar.org/rdl/RDS416609>).



Supertypes

- *OperatedValve*

Example in DEXPI XML

```
<Object type="Plant/Piping.ButterflyValve"/>
```

CheckValve

Plant::Piping::CheckValve

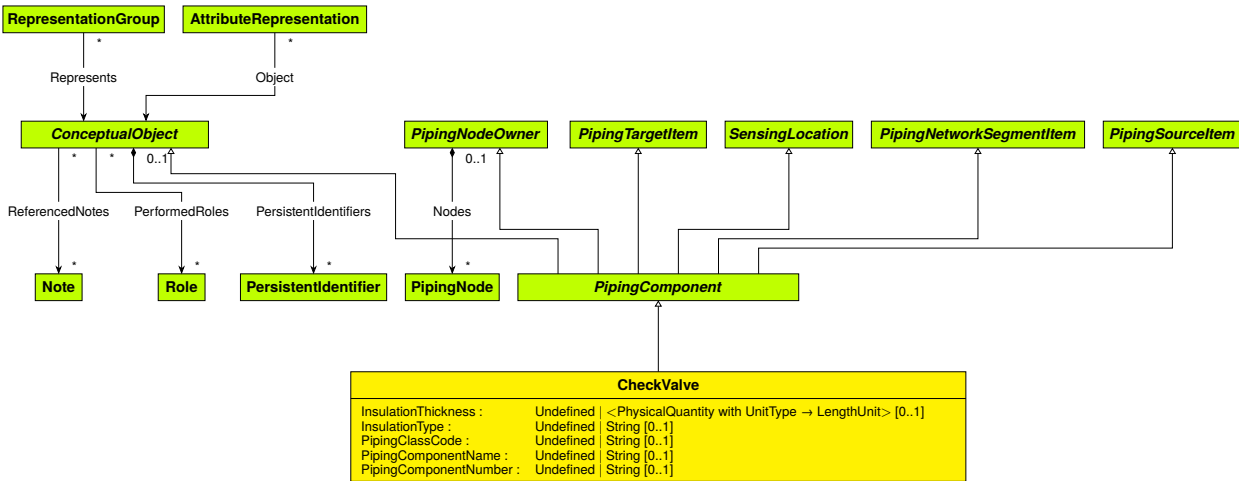
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS292229>

CHECK VALVE

A valve that permits fluid to flow in one direction only (from <http://data.posccaesar.org/rdl/RDS292229>).



Supertypes

- *PipingComponent*

Subtypes

- *GlobeCheckValve*
- *SwingCheckValve*

Members

Data Properties

Name	Multiplic-ity	Type	Or-dered	Unique
<i>InsulationThickness</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no
<i>InsulationType</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>PipingClassCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>PipingComponent-Name</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>PipingComponent-Number</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

InsulationThickness

Data Property

The insulation thickness of the *CheckValve*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → LengthUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS4238040>

INSULATION THICKNESS

Example

```
PhysicalQuantity(  
  Unit = Millimetre,  
  Value = 40.0)
```

InsulationType

Data Property

The identification code for the insulation type related to the *CheckValve*. So far, DEXPI does not define restrictions for valid values.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/InsulationTypeAssignmentClass>

INSULATION TYPE ASSIGNMENT CLASS

Example

"Q"

PipingClassCode

Data Property

The identification code of the piping class of the *CheckValve*. So far, DEXPI does not define restrictions for valid values.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PipingClassCodeAssignmentClass>

PIPING CLASS CODE ASSIGNMENT CLASS

Example

"75HB13"

PipingComponentName

Data Property

A string to classify the *CheckValve*. DEXPI does not prescribe the classification system. Typically, company or site standards are used.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PipingComponentNameAssignmentClass>

PIPING COMPONENT NAME ASSIGNMENT CLASS

Example

"73KH12"

PipingComponentNumber

Data Property

An identifier of the *CheckValve*. DEXPI does not prescribe the scope of the identifier, i.e., whether it should be unique in, e.g., a *PipingNetworkSegment* or a *PipingNetworkSystem*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RD1: <http://sandbox.dexpi.org/rdl/PipingComponentNumberAssignmentClass>

PIPING COMPONENT NUMBER ASSIGNMENT CLASS

Example

"C2"

Example in DEXPI XML

```
<Object type="Plant/Piping.CheckValve">
  <Data property="InsulationThickness">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Millimetre"/>
      </Data>
      <Data property="Value">
        <Double>40.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="InsulationType">
    <String>Q</String>
  </Data>
  <Data property="PipingClassCode">
    <String>75HB13</String>
  </Data>
  <Data property="PipingComponentName">
    <String>73KH12</String>
  </Data>
  <Data property="PipingComponentNumber">
    <String>C2</String>
  </Data>
</Object>
```

ClampedFlangeCoupling

Plant::Piping::ClampedFlangeCoupling

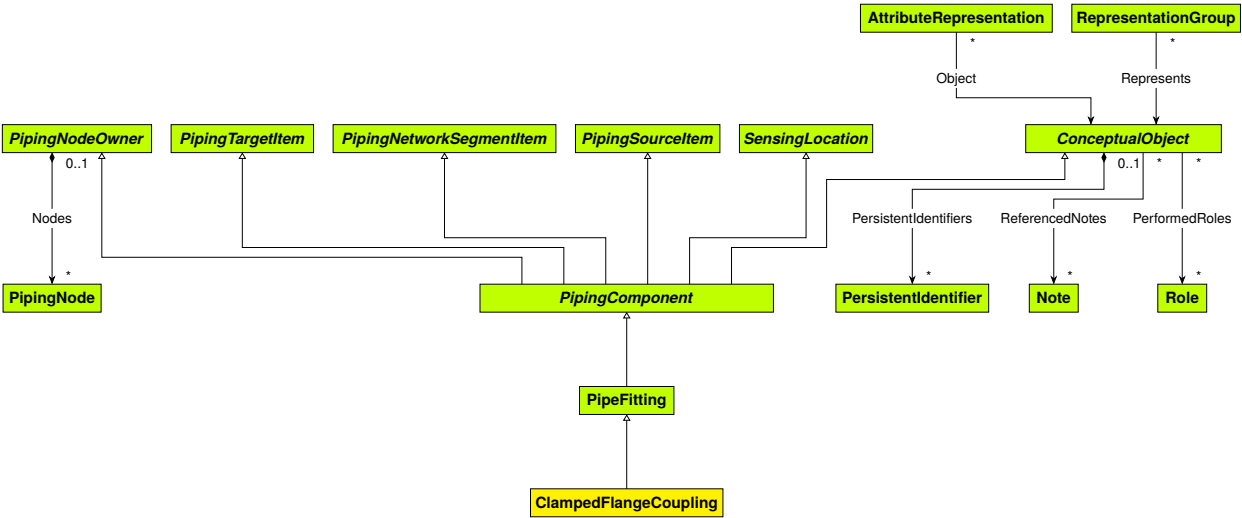
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/ClampedFlangeCoupling>

CLAMPED FLANGE COUPLING

A clamped flange coupling.



Supertypes

- *PipeFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.ClampedFlangeCoupling"/>
```

Compensator

Plant::Piping::Compensator

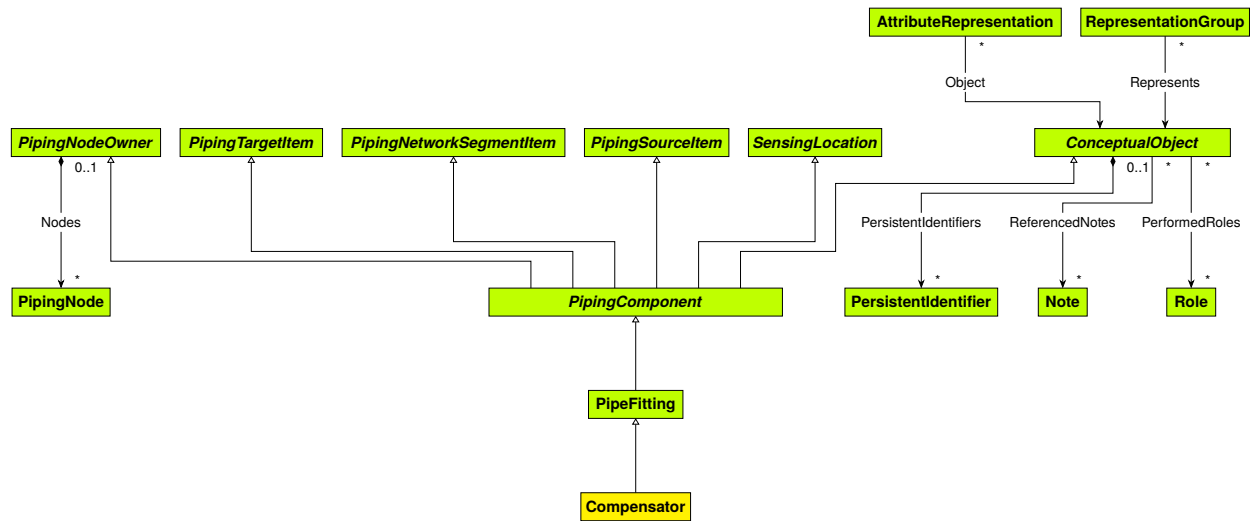
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS1280084541>

COMPENSATOR

A device compensating for axial or radial movement between two elements that is connected (from <http://data.posccaesar.org/rdl/RDS1280084541>).



Supertypes

- *PipeFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.Compensator"/>
```

ConicalStrainer

Plant::Piping::ConicalStrainer

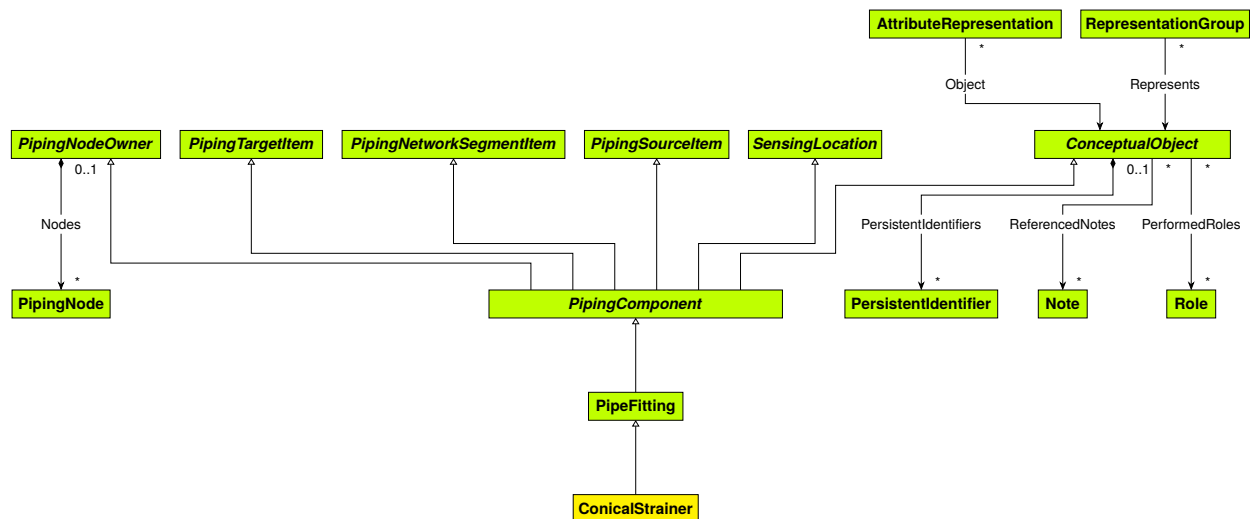
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS16044540>

CONICAL STRAINER

A strainer where the screen has a conical tubular shape (from <http://data.posccaesar.org/rdl/RDS16044540>).



Supertypes

- *PipeFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.ConicalStrainer"/>
```

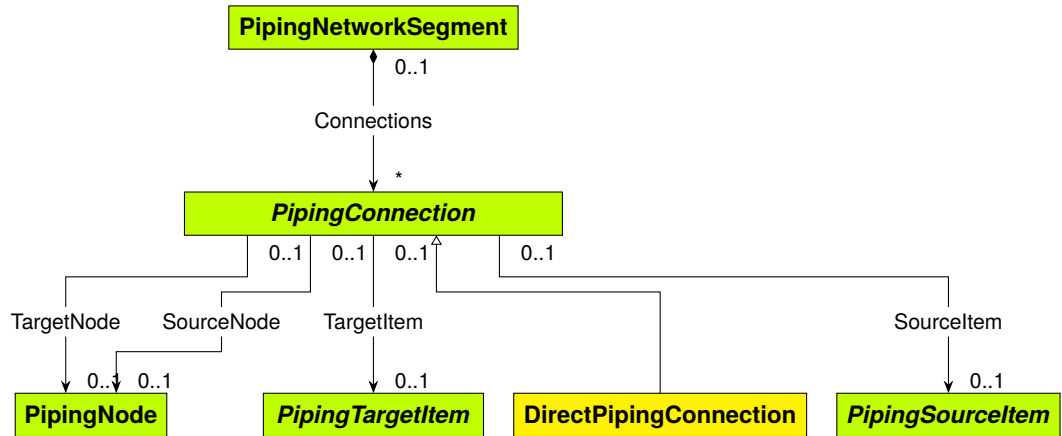
DirectPipingConnection

Plant::Piping::DirectPipingConnection

Overview

Concrete Class

A direct connection between two piping items, i.e. a connection that is not realized by a pipe.



Supertypes

- *PipingConnection*

Example in DEXPI XML

```
<Object type="Plant/Piping.DirectPipingConnection"/>
```

ElectromagneticFlowMeter

Plant::Piping::ElectromagneticFlowMeter

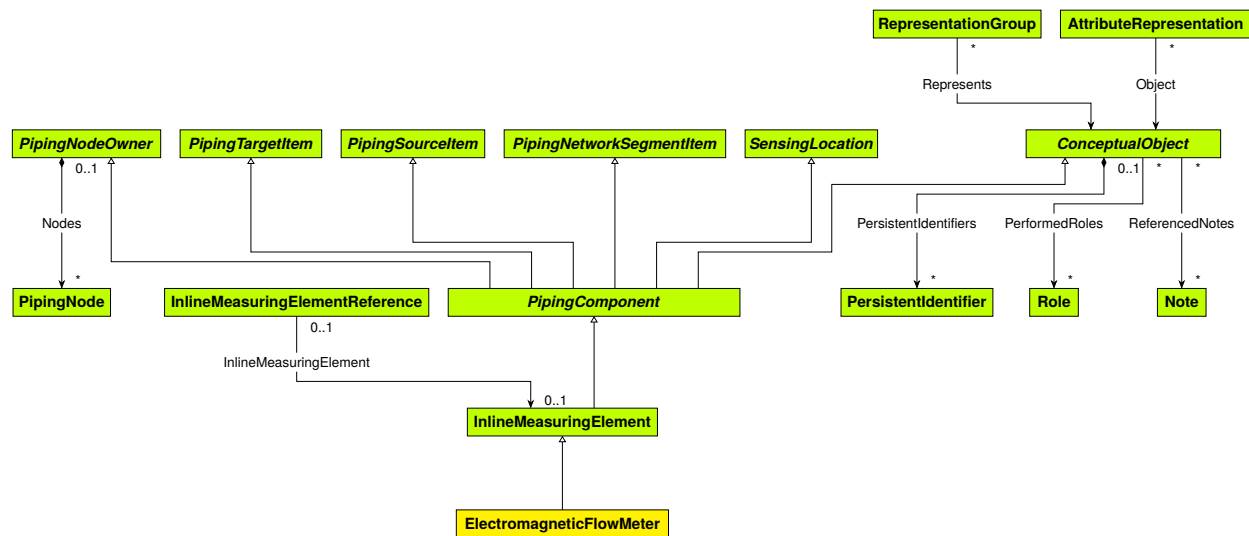
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS1009664>

ELECTROMAGNETIC FLOW METER

A velocity flow meter that is measuring flow rate of a conductive fluid running through a magnetic field by measuring the charge created when fluid interacting with the field (from <http://data.posccaesar.org/rdl/RDS1009664>).



Supertypes

- *InlineMeasuringElement*

Example in DEXPI XML

```
<Object type="Plant/Piping.ElectromagneticFlowMeter"/>
```

FlameArrestor

Plant::Piping::FlameArrestor

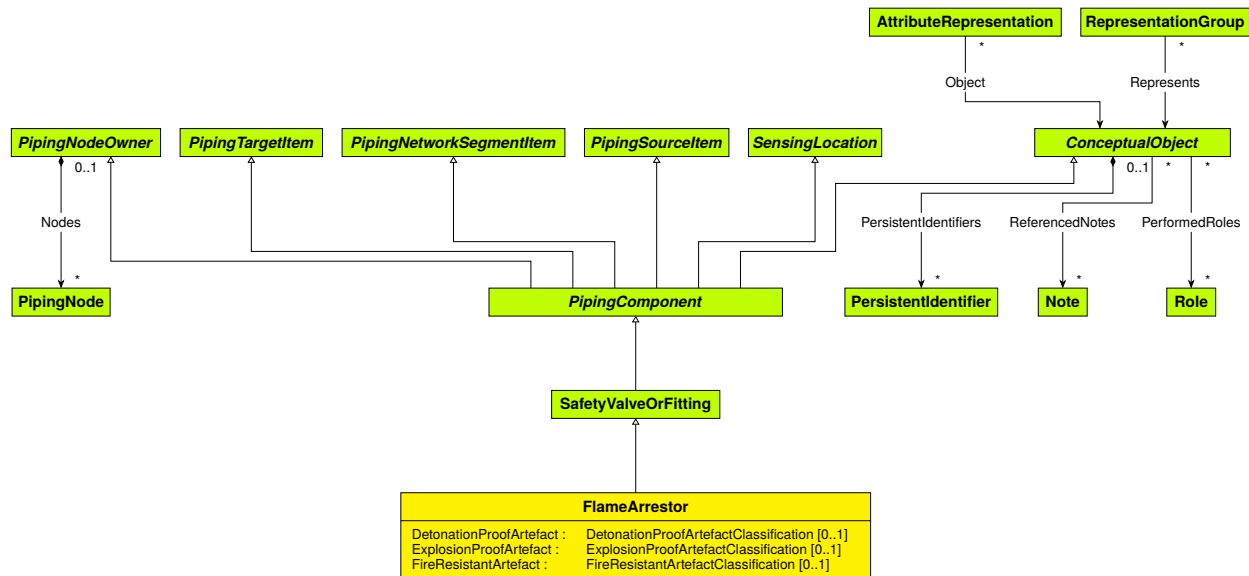
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS1325028651>

FLAME ARRESTOR

An ‘arrestor’ which is a trap covering an opening, e.g of a ventilation system or a pipe, to prevent flames from entering the system (from <http://data.posccaesar.org/rdl/RDS1325028651>).



Supertypes

- *SafetyValveOrFitting*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DetonationProofArtefact</i>	0..1	<i>DetonationProofArtefactClassification</i>	no	no
<i>ExplosionProofArtefact</i>	0..1	<i>ExplosionProofArtefactClassification</i>	no	no
<i>FireResistantArtefact</i>	0..1	<i>FireResistantArtefactClassification</i>	no	no

DetonationProofArtefact

Data Property

A specialization indicating if the *FlameArrestor* is detonation-proof.

Type: *DetonationProofArtefactClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DetonationProofArtefactSpecialization>

DETONATION PROOF ARTEFACT SPECIALIZATION

Example

NonDetonationProofArtefact

ExplosionProofArtefact

Data Property

A specialization indicating if the *FlameArrestor* is explosion-proof.

Type: *ExplosionProofArtefactClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ExplosionProofArtefactSpecialization>

EXPLOSION PROOF ARTEFACT SPECIALIZATION

Example

ExplosionProofArtefact

FireResistantArtefact

Data Property

A specialization indicating if the *FlameArrestor* is fire-resistant.

Type: *FireResistantArtefactClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/FireResistantArtefactSpecialization>

FIRE RESISTANT ARTEFACT SPECIALIZATION

Example

FireResistantArtefact

Example in DEXPI XML

```
<Object type="Plant/Piping.FlameArrestor">
  <Data property="DetonationProofArtefact">
    <DataReference data="Plant/Enumerations.DetonationProofArtefactClassification.
↪NonDetonationProofArtefact"/>
  </Data>
  <Data property="ExplosionProofArtefact">
    <DataReference data="Plant/Enumerations.ExplosionProofArtefactClassification.
↪ExplosionProofArtefact"/>
  </Data>
  <Data property="FireResistantArtefact">
    <DataReference data="Plant/Enumerations.FireResistantArtefactClassification.
↪FireResistantArtefact"/>
  </Data>
</Object>
```

Flange

Plant::Piping::Flange

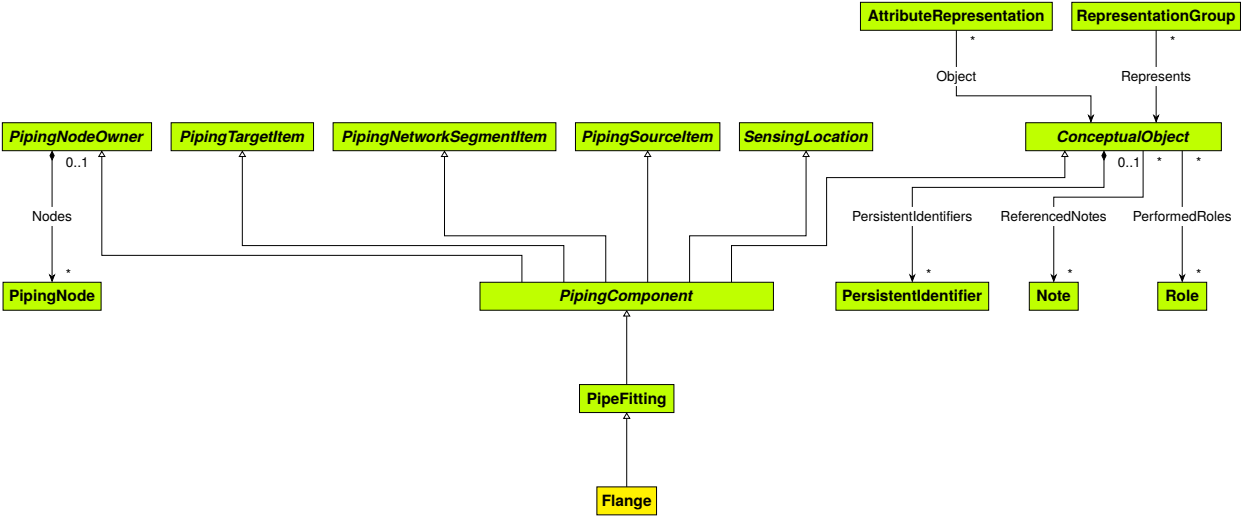
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS13307654>

FLANGE

A physical object that is a projecting flat rim, plate, collar, or rib



Supertypes

- *PipeFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.Flange"/>
```

FlangedConnection

Plant::Piping::FlangedConnection

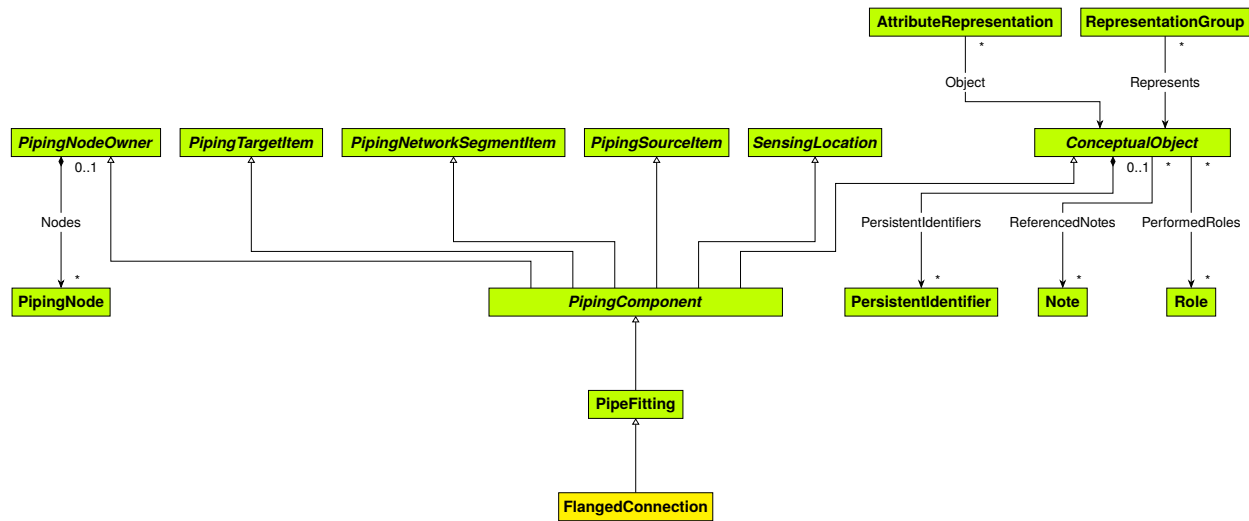
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/FlangedConnection>

FLANGED CONNECTION

A flanged connection.



Supertypes

- *PipeFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.FlangedConnection"/>
```

FlowInPipeOffPageConnector

Plant::Piping::FlowInPipeOffPageConnector

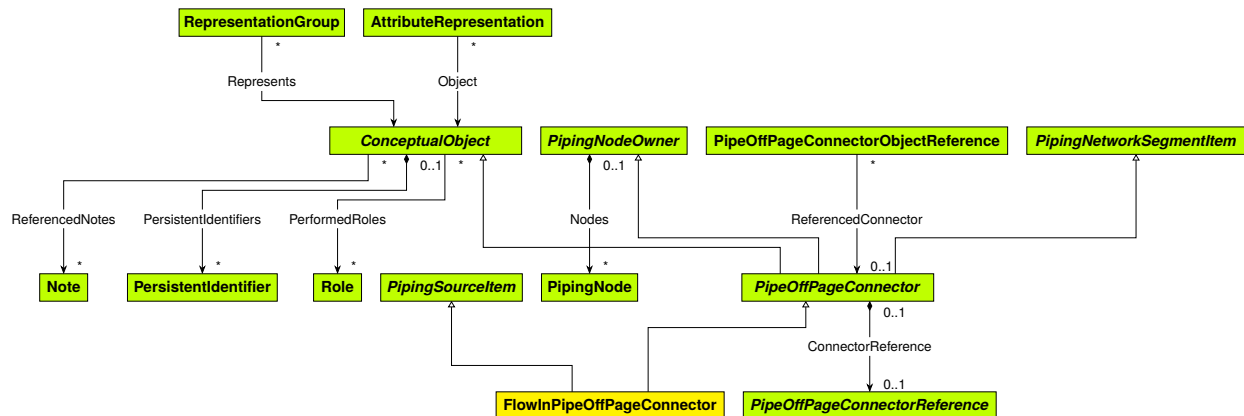
Overview

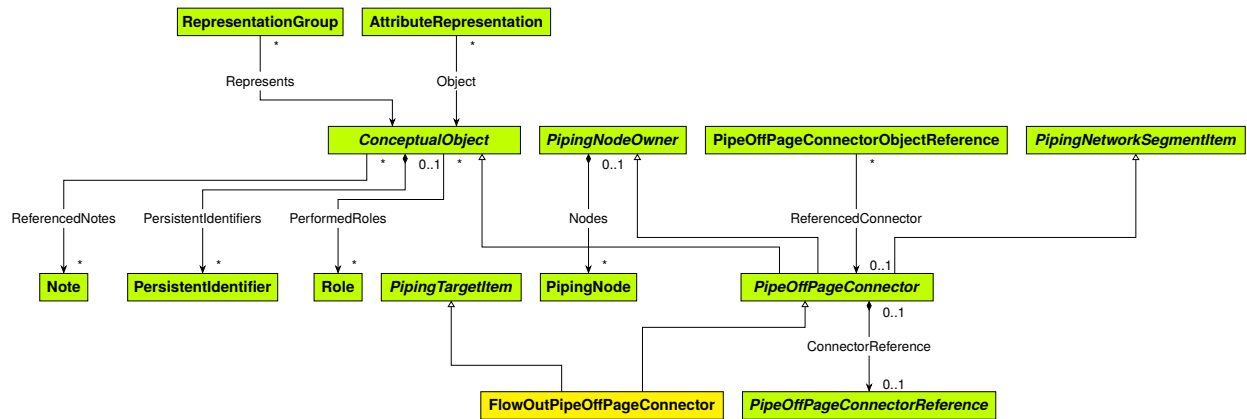
Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/FlowInPipeOffPageConnector>

FLOW IN PIPE OFF PAGE CONNECTOR

A pipe connector that indicates that a preceding part of a piping network segment is represented somewhere else, either on the same PID, or on some other PID.





Supertypes

- *PipeOffPageConnector*
- *PipingTargetItem*

Example in DEXPI XML

```
<Object type="Plant/Piping.FlowOutPipeOffPageConnector"/>
```

Funnel

Plant::Piping::Funnel

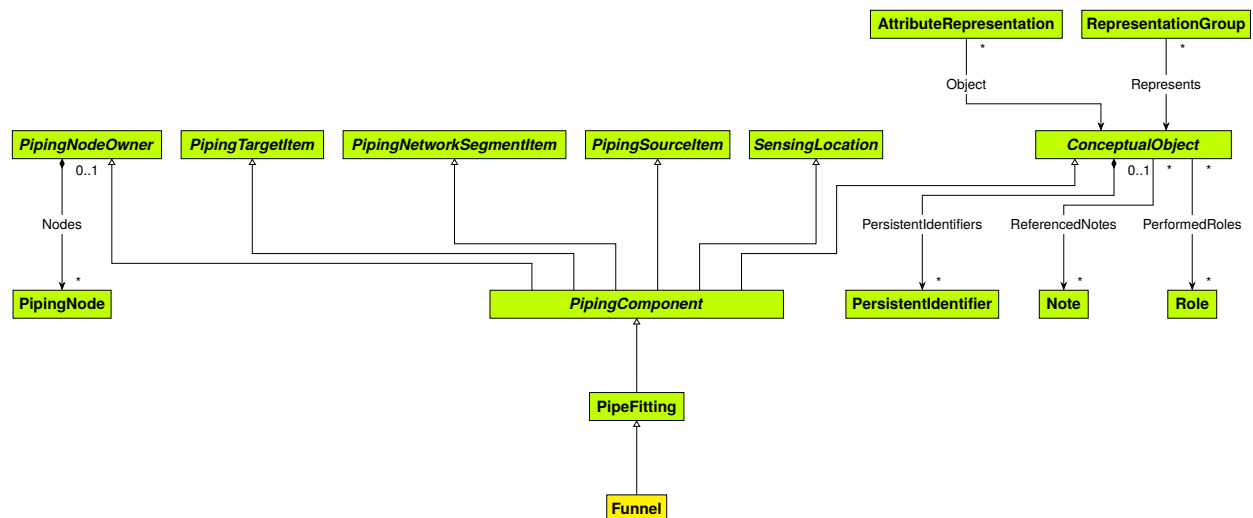
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS6689917>

FUNNEL

A hollow cone with a tube extending from the smaller end and that is designed to catch and direct a downward flow



Supertypes

- *PipeFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.Funnel"/>
```

GateValve

Plant::Piping::GateValve

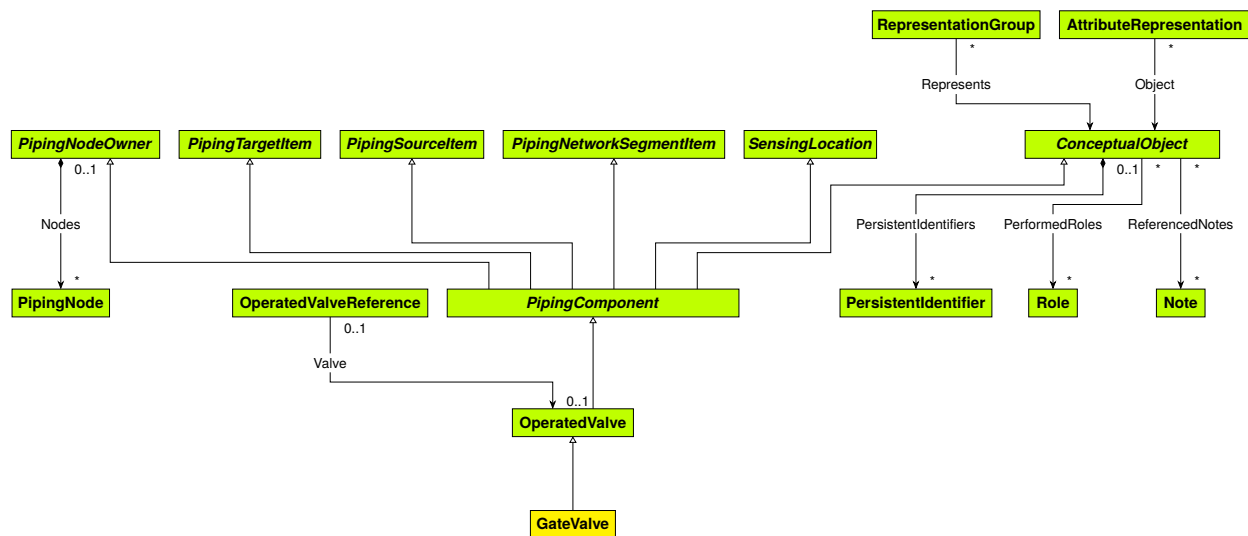
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS416519>

GATE VALVE

A valve that is a valve where the closure member is a gate or disc with a linear motion parallel, or nearly parallel, to the plane of flat seats, which are transverse to the direction of flow (from <http://data.posccaesar.org/rdl/RDS416519>).



Supertypes

- *OperatedValve*

Example in DEXPI XML

```
<Object type="Plant/Piping.GateValve"/>
```

GlobeCheckValve

Plant::Piping::GlobeCheckValve

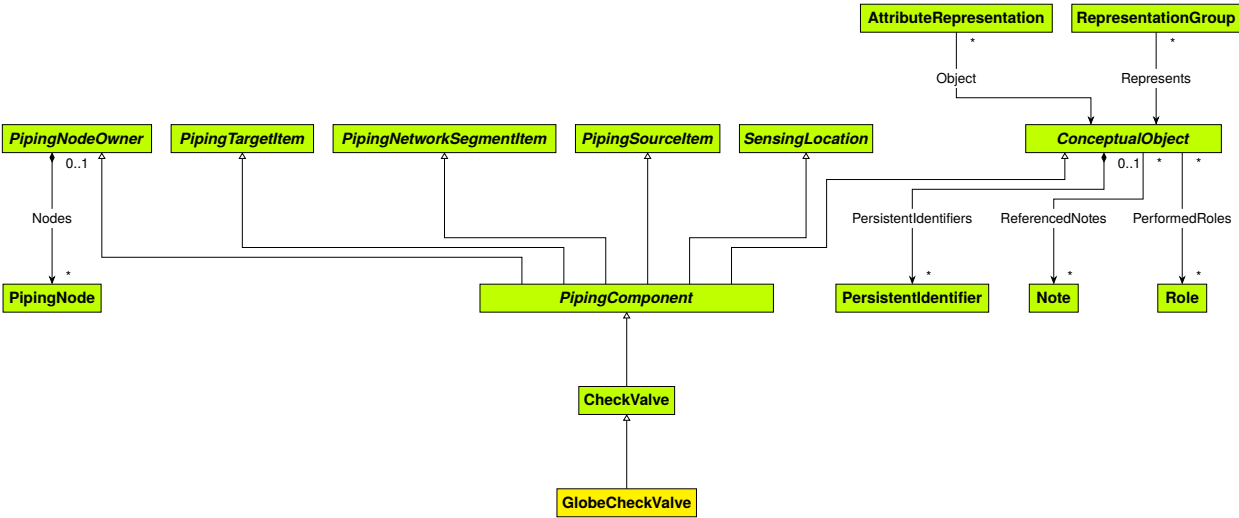
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/GlobeCheckValve>

GLOBE CHECK VALVE

A globe check valve.



Supertypes

- *CheckValve*

Example in DEXPI XML

```
<Object type="Plant/Piping.GlobeCheckValve"/>
```

GlobeValve

Plant::Piping::GlobeValve

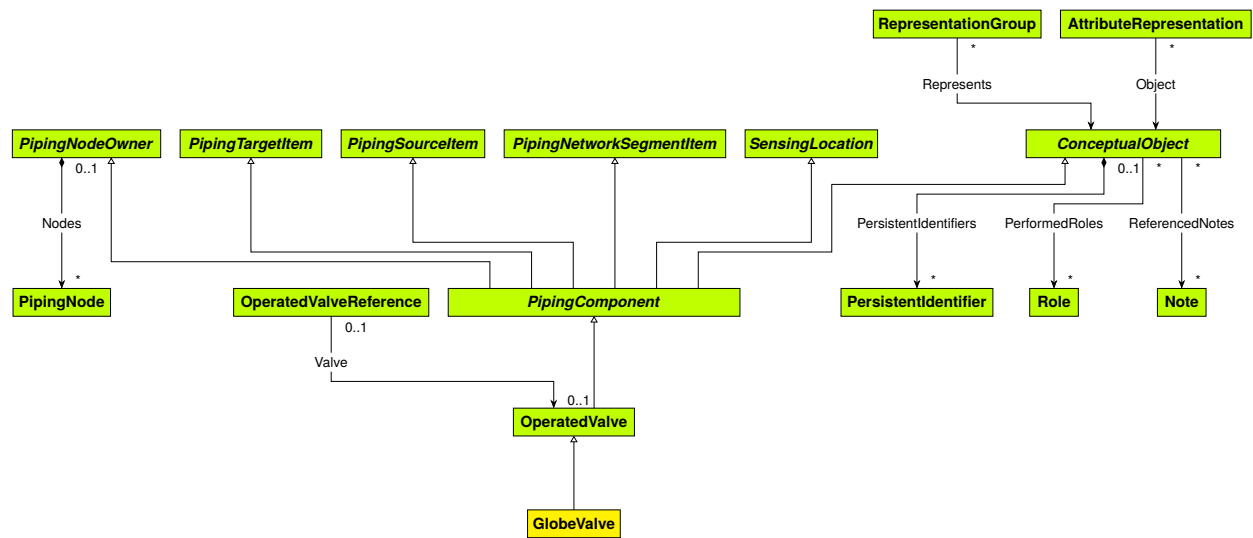
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS416204>

GLOBE VALVE

A valve that is a valve where the closure member is a disc or piston operating with linear motion normal to the flat or shaped seat (from <http://data.posccaesar.org/rdl/RDS416204>).



Supertypes

- *OperatedValve*

Example in DEXPI XML

```
<Object type="Plant/Piping.GlobeValve"/>
```

Hose

Plant::Piping::Hose

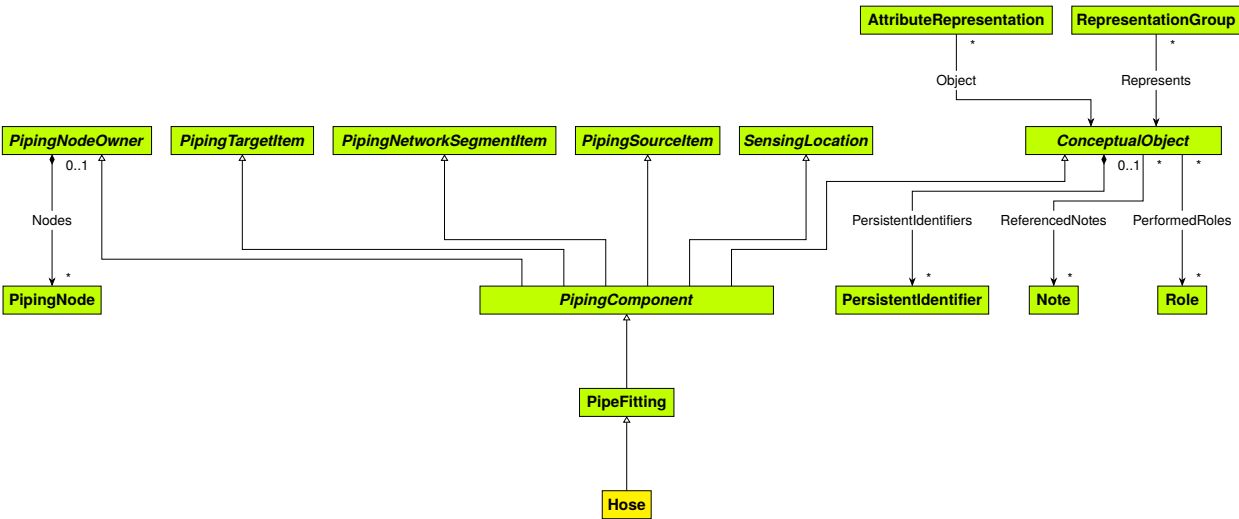
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS302174>

HOSE

A tubular which is flexible and capable of conveying liquids under pressure (from <http://data.posccaesar.org/rd1/RDS302174>).



Supertypes

- *PipeFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.Hose"/>
```

IlluminatedSightGlass

Plant::Piping::IlluminatedSightGlass

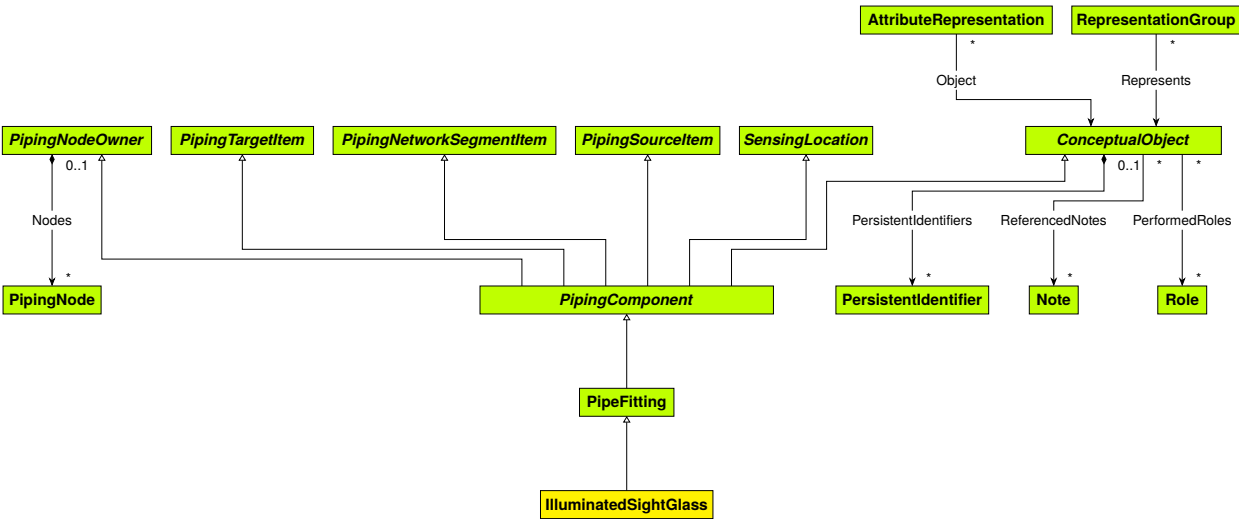
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/IlluminatedSightGlass>

ILLUMINATED SIGHT GLASS

An illuminated sight glass.



Supertypes

- *PipeFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.IlluminatedSightGlass"/>
```

InLineMixer

Plant::Piping::InLineMixer

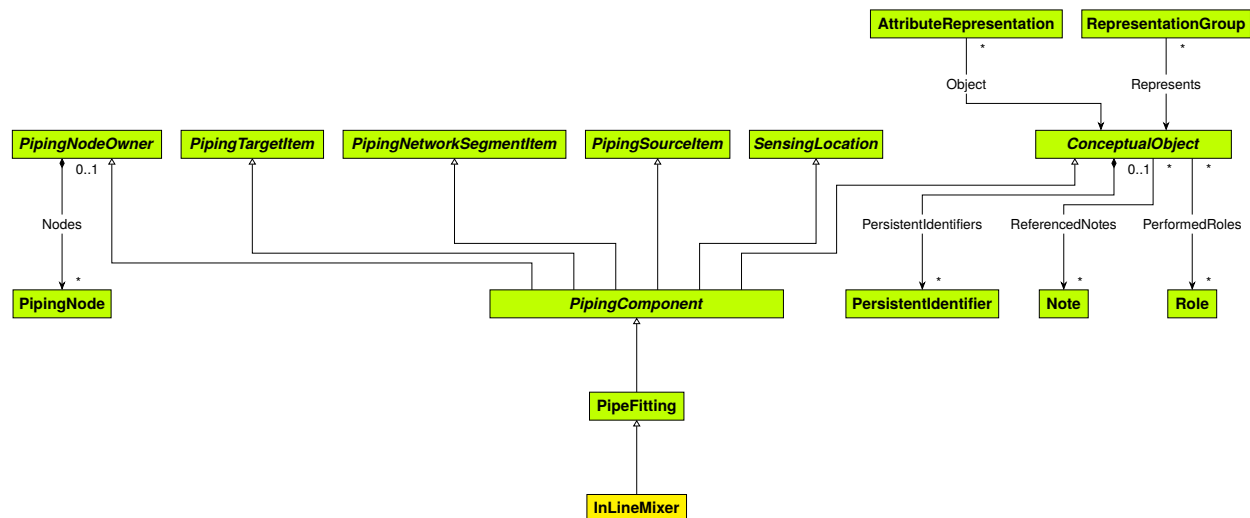
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS43167562195>

IN-LINE MIXER

A static mixer that is intended to be supported by connected equipment. Typically supported by piping



Supertypes

- *PipeFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.InLineMixer"/>
```

InlineMeasuringElement

Plant::Piping::InlineMeasuringElement

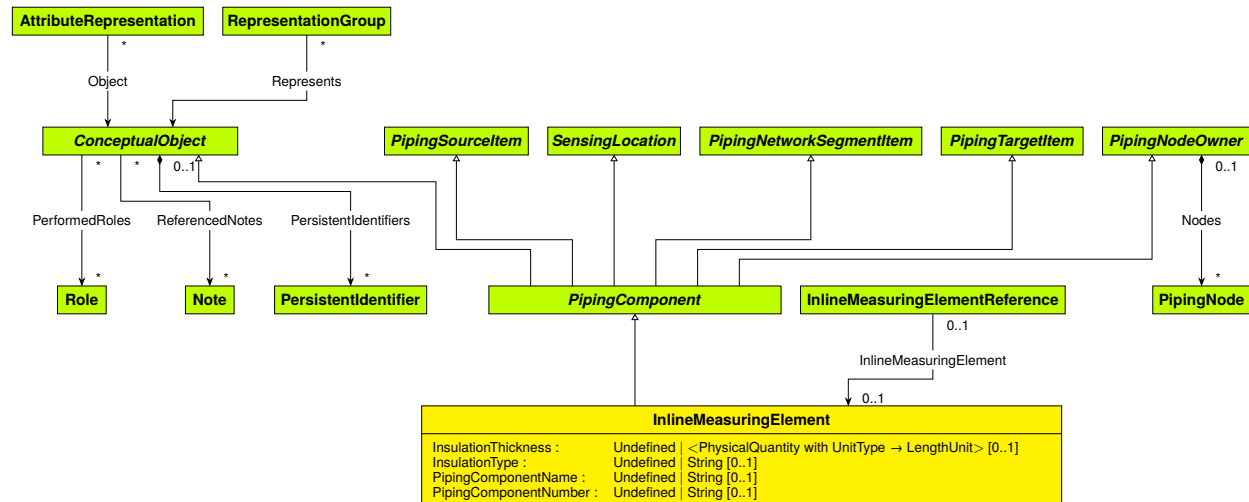
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/InlineMeasuringElement>

INLINE MEASURING ELEMENT

An inline measuring element.



Supertypes

- *PipingComponent*

Subtypes

- *ElectromagneticFlowMeter*
- *FlowMeasuringElement*
- *FlowNozzle*
- *MassFlowMeasuringElement*
- *PositiveDisplacementFlowMeter*
- *TurbineFlowMeter*
- *VariableAreaFlowMeter*
- *VenturiTube*
- *VolumeFlowMeasuringElement*

Members

Data Properties

Name	Multiplic- ity	Type	Or- dered	Unique
<i>InsulationThickness</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no
<i>InsulationType</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>PipingComponent- Name</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>PipingComponent- Number</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

InsulationThickness

Data Property

The insulation thickness of the *InlineMeasuringElement*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → LengthUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS4238040>

INSULATION THICKNESS

Example

```
PhysicalQuantity(
  Unit = Millimetre,
  Value = 40.0)
```

InsulationType

Data Property

The identification code for the insulation type related to the *InlineMeasuringElement*. So far, DEXPI does not define restrictions for valid values.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/InsulationTypeAssignmentClass>

INSULATION TYPE ASSIGNMENT CLASS

Example

"Q"

PipingComponentName**Data Property**

A string to classify the *InlineMeasuringElement*. DEXPI does not prescribe the classification system. Typically, company or site standards are used.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PipingComponentNameAssignmentClass>

PIPING COMPONENT NAME ASSIGNMENT CLASS

Example

"73KH12"

PipingComponentNumber**Data Property**

An identifier of the *InlineMeasuringElement*. DEXPI does not prescribe the scope of the identifier, i.e., whether it should be unique in, e.g., a *PipingNetworkSegment* or a *PipingNetworkSystem*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PipingComponentNumberAssignmentClass>

PIPING COMPONENT NUMBER ASSIGNMENT CLASS

Example

"C2"

Example in DEXPI XML

```

<Object type="Plant/Piping.InlineMeasuringElement">
  <Data property="InsulationThickness">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Millimetre"/>
      </Data>
      <Data property="Value">
        <Double>40.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>

```

(continues on next page)

(continued from previous page)

```

    </Data>
  </AggregatedDataValue>
</Data>
<Data property="InsulationType">
  <String>Q</String>
</Data>
<Data property="PipingComponentName">
  <String>73KH12</String>
</Data>
<Data property="PipingComponentNumber">
  <String>C2</String>
</Data>
</Object>

```

LineBlind

Plant::Piping::LineBlind

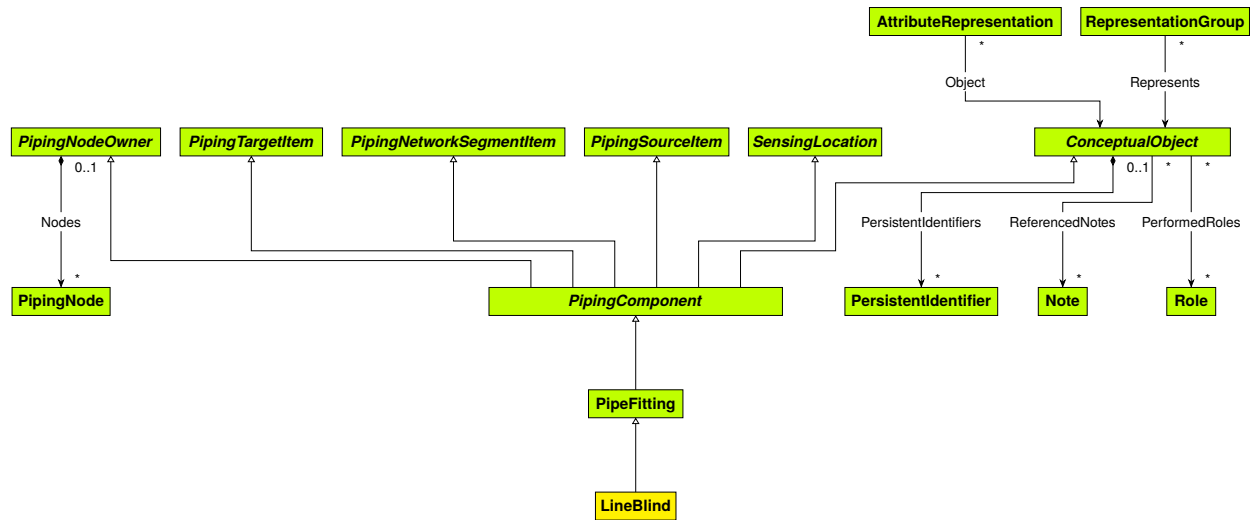
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS280034>

LINE BLIND

A functional unit used to blind off a process stream (from <http://data.posccaesar.org/rdl/RDS280034>).



Supertypes

- *PipeFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.LineBlind"/>
```

MassFlowMeasuringElement

Plant::Piping::MassFlowMeasuringElement

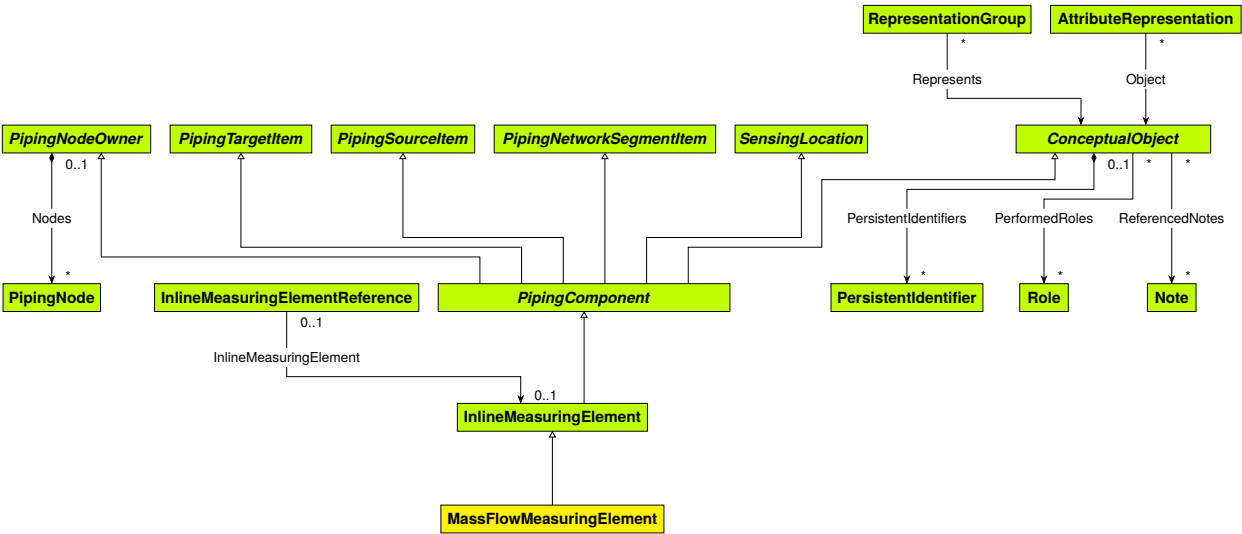
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/MassFlowMeasuringElement>

MASS FLOW MEASURING ELEMENT

A MASS FLOW MEASURING ELEMENT is a FLOW MEASURING ELEMENT that is used to measure MASS FLOW RATE.



Supertypes

- *InlineMeasuringElement*

Example in DEXPI XML

```
<Object type="Plant/Piping.MassFlowMeasuringElement"/>
```

NeedleValve

Plant::Piping::NeedleValve

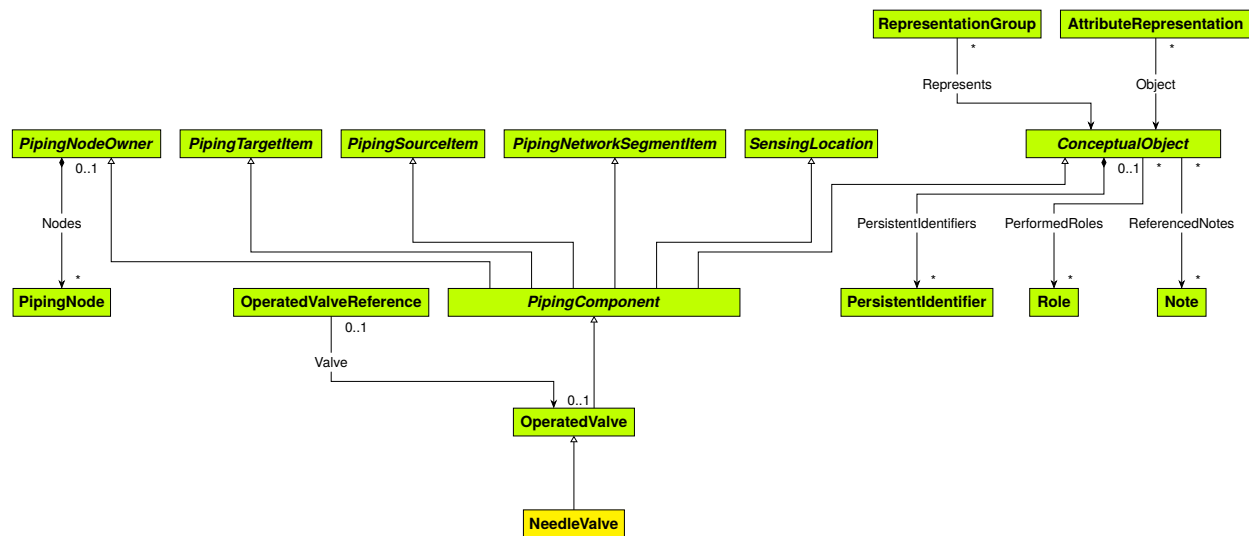
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS421064>

NEEDLE VALVE

A globe valve that has a closure member with the shape of a conical plug (needle) which closes into a small seat (from <http://data.posccaesar.org/rdl/RDS421064>).



Supertypes

- *OperatedValve*

Example in DEXPI XML

```
<Object type="Plant/Piping.NeedleValve"/>
```

OperatedValve

Plant::Piping::OperatedValve

Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS11141590>

OPERATED VALVE

A valve that includes an external means of operation. (E.g. handwheel / lever / actuator.)



- ## Subtypes

- ## Members

Data Properties

Name	Multiplic- ity	Type	Or- dered	Unique
<i>InsulationThickness</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no
<i>InsulationType</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>NumberOfPorts</i>	0..1	<i>NumberOfPortsClassification</i>	no	no
<i>Operation</i>	0..1	<i>OperationClassification</i>	no	no
<i>PipingClassCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>PipingComponent- Name</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>PipingComponent- Number</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

InsulationThickness

Data Property

The insulation thickness of the *OperatedValve*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → LengthUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS4238040>

INSULATION THICKNESS

Example

```
PhysicalQuantity(
    Unit = Millimetre,
    Value = 40.0)
```

InsulationType

Data Property

The identification code for the insulation type related to the *OperatedValve*. So far, DEXPI does not define restrictions for valid values.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/InsulationTypeAssignmentClass>

INSULATION TYPE ASSIGNMENT CLASS

Example

"Q"

NumberOfPorts**Data Property**

A specialization indicating the number of ports of the *OperatedValve*.

Type: *NumberOfPortsClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NumberOfPortsSpecialization>

NUMBER OF PORTS SPECIALIZATION

Example

TwoPortValve

Operation**Data Property**

A specialization indicating the operation of the *OperatedValve*.

Type: *OperationClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/OperationSpecialization>

OPERATION SPECIALIZATION

Example

ContinuousOperation

PipingClassCode**Data Property**

The identification code of the piping class of the *OperatedValve*. So far, DEXPI does not define restrictions for valid values.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PipingClassCodeAssignmentClass>

PIPING CLASS CODE ASSIGNMENT CLASS

Example

"75HB13"

PipingComponentName**Data Property**

A string to classify the *OperatedValve*. DEXPI does not prescribe the classification system. Typically, company or site standards are used.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PipingComponentNameAssignmentClass>

PIPING COMPONENT NAME ASSIGNMENT CLASS

Example

"73KH12"

PipingComponentNumber**Data Property**

An identifier of the *OperatedValve*. DEXPI does not prescribe the scope of the identifier, i.e., whether it should be unique in, e.g., a *PipingNetworkSegment* or a *PipingNetworkSystem*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PipingComponentNumberAssignmentClass>

PIPING COMPONENT NUMBER ASSIGNMENT CLASS

Example

"C2"

Example in DEXPI XML

```
<Object type="Plant/Piping.OperatedValve">
  <Data property="InsulationThickness">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Millimetre"/>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

(continues on next page)

(continued from previous page)

```

    <Data property="Value">
      <Double>40.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="InsulationType">
  <String>Q</String>
</Data>
<Data property="NumberOfPorts">
  <DataReference data="Plant/Enumerations.NumberOfPortsClassification.TwoPortValve"/>
</Data>
<Data property="Operation">
  <DataReference data="Plant/Enumerations.OperationClassification.ContinuousOperation"/
  >
</Data>
<Data property="PipingClassCode">
  <String>75HB13</String>
</Data>
<Data property="PipingComponentName">
  <String>73KH12</String>
</Data>
<Data property="PipingComponentNumber">
  <String>C2</String>
</Data>
</Object>

```

Penetration

Plant::Piping::Penetration

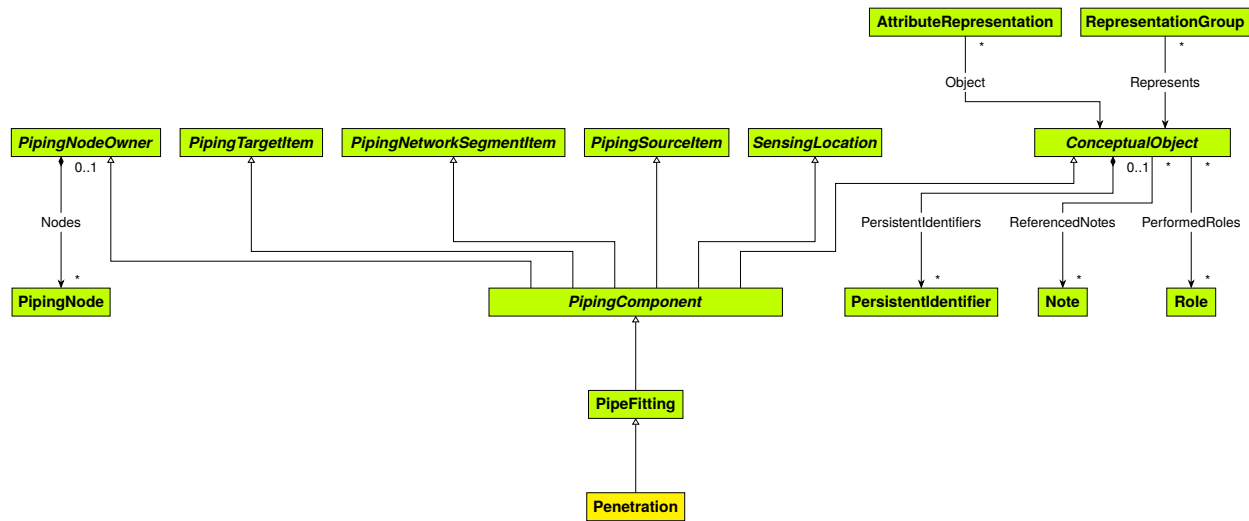
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS13068275>

PENETRATION

A device intended to provide a penetration (from <http://data.posccaesar.org/rdl/RDS13068275>).



Supertypes

- *PipeFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.Penetration"/>
```

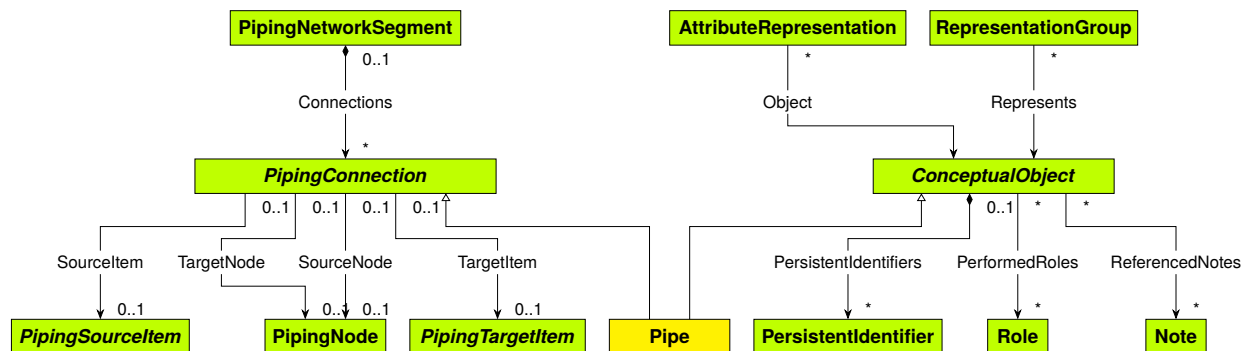
Pipe

Plant::Piping::Pipe

Overview

Concrete Class

An elementary piece of piping, i.e., not interrupted by any item.



Supertypes

- *ConceptualObject*
- *PipingConnection*

Example in DEXPI XML

```
<Object type="Plant/Piping.Pipe"/>
```

PipeCoupling

Plant::Piping::PipeCoupling

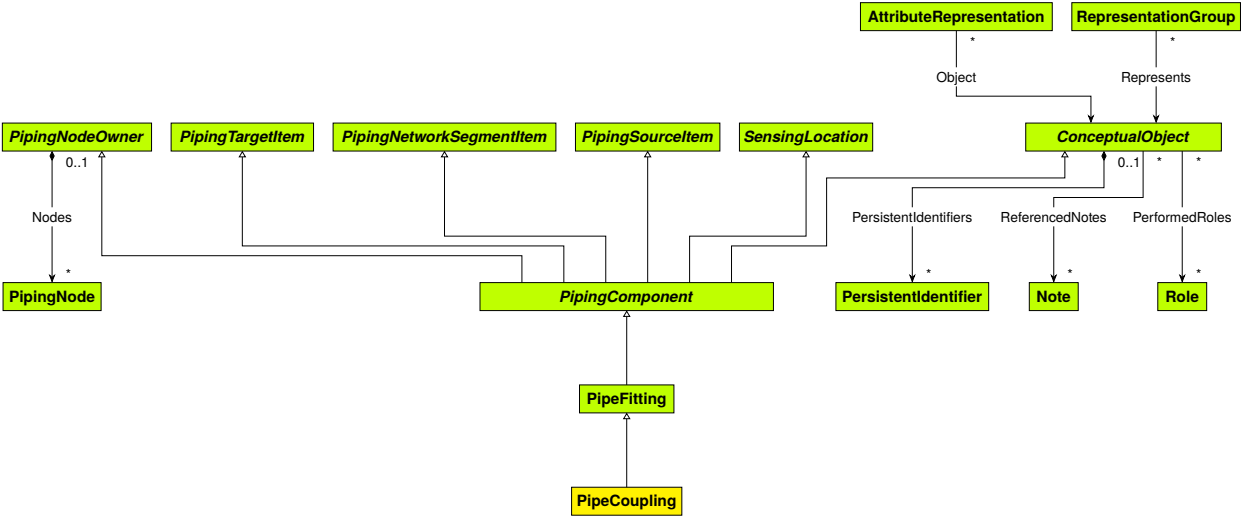
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS415664>

PIPE COUPLING

An ‘artefact’ that is a one-piece cylindrical section intended to join pipes and/or piping components (from <http://data.posccaesar.org/rdl/RDS415664>).



Supertypes

- *PipeFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.PipeCoupling"/>
```

PipeFitting

Plant::Piping::PipeFitting

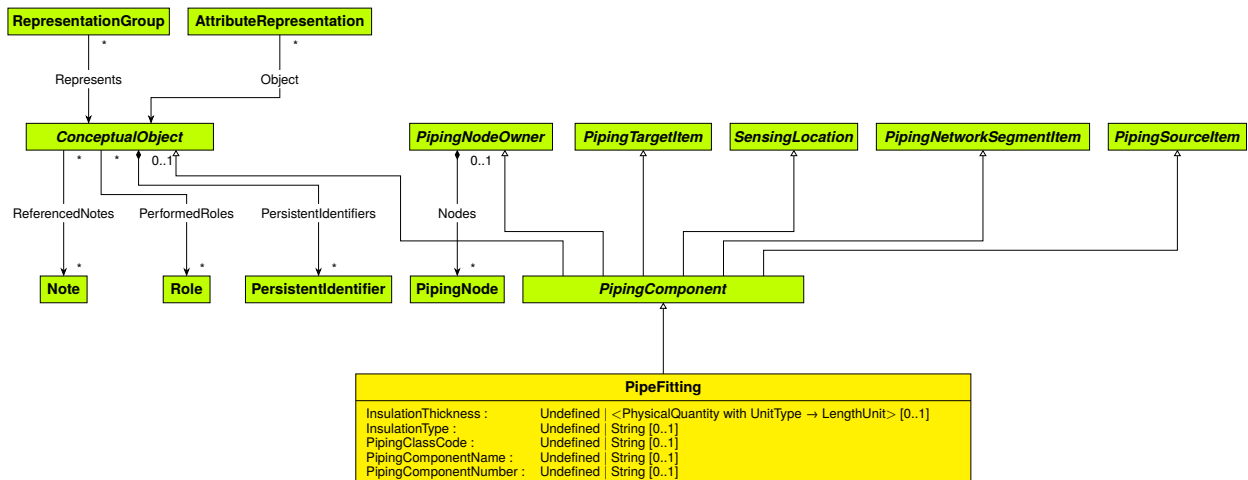
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/PipeFitting>

PIPE FITTING

A pipe fitting.



Supertypes

- *PipingComponent*

Subtypes

- *BlindFlange*
- *ClampedFlangeCoupling*
- *Compensator*
- *ConicalStrainer*
- *Flange*
- *FlangedConnection*
- *Funnel*
- *Hose*
- *IlluminatedSightGlass*
- *InLineMixer*
- *LineBlind*
- *Penetration*
- *PipeCoupling*
- *PipeFlangeSpacer*
- *PipeFlangeSpade*
- *PipeReducer*
- *PipeTee*
- *RestrictionOrifice*
- *Sensorwell*
- *SightGlass*
- *Silencer*
- *SteamTrap*

- *Strainer*
- *VentLine*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>InsulationThickness</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no
<i>InsulationType</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>PipingClassCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>PipingComponent-Name</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>PipingComponent-Number</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

InsulationThickness

Data Property

The insulation thickness of the *PipeFitting*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → LengthUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS4238040>

INSULATION THICKNESS

Example

```
PhysicalQuantity(
  Unit = Millimetre,
  Value = 40.0)
```

InsulationType

Data Property

The identification code for the insulation type related to the *PipeFitting*. So far, DEXPI does not define restrictions for valid values.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/InsulationTypeAssignmentClass>

INSULATION TYPE ASSIGNMENT CLASS

Example

"Q"

PipingClassCode

Data Property

The identification code of the piping class of the *PipeFitting*. So far, DEXPI does not define restrictions for valid values.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PipingClassCodeAssignmentClass>

PIPING CLASS CODE ASSIGNMENT CLASS

Example

"75HB13"

PipingComponentName

Data Property

A string to classify the *PipeFitting*. DEXPI does not prescribe the classification system. Typically, company or site standards are used.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PipingComponentNameAssignmentClass>

PIPING COMPONENT NAME ASSIGNMENT CLASS

Example

"73KH12"

PipingComponentNumber

Data Property

An identifier of the *PipeFitting*. DEXPI does not prescribe the scope of the identifier, i.e., whether it should be unique in, e.g., a *PipingNetworkSegment* or a *PipingNetworkSystem*.

Type: *Undefined* | *String*

Multiplicity: 0..1**Ordered:** no**Unique:** no**RDL:** <http://sandbox.dexpi.org/rdl/PipingComponentNumberAssignmentClass>

PIPING COMPONENT NUMBER ASSIGNMENT CLASS

Example`"C2"`**Example in DEXPI XML**

```

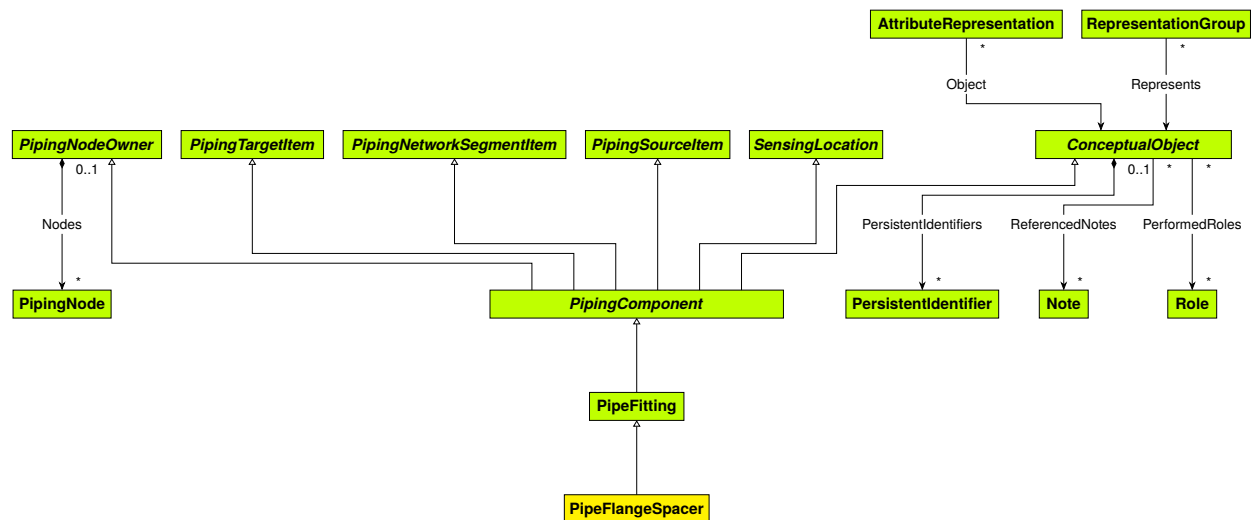
<Object type="Plant/Piping.PipeFitting">
  <Data property="InsulationThickness">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Millimetre"/>
      </Data>
      <Data property="Value">
        <Double>40.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="InsulationType">
    <String>Q</String>
  </Data>
  <Data property="PipingClassCode">
    <String>75HB13</String>
  </Data>
  <Data property="PipingComponentName">
    <String>73KH12</String>
  </Data>
  <Data property="PipingComponentNumber">
    <String>C2</String>
  </Data>
</Object>

```

PipeFlangeSpacer*Plant::Piping::PipeFlangeSpacer***Overview****Concrete Class****RDL:** <http://data.posccaesar.org/rdl/RDS472724>

PIPE FLANGE SPACER

A ‘spacer’ and an ‘artefact’ that is intended to be inserted between two pipe flanged ends to provide the distance between the flanges required to insert a ‘pipe flange spade’ (from <http://data.posccaesar.org/rdl/RDS472724>).



Supertypes

- *PipeFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.PipeFlangeSpacer"/>
```

PipeFlangeSpade

Plant::Piping::PipeFlangeSpade

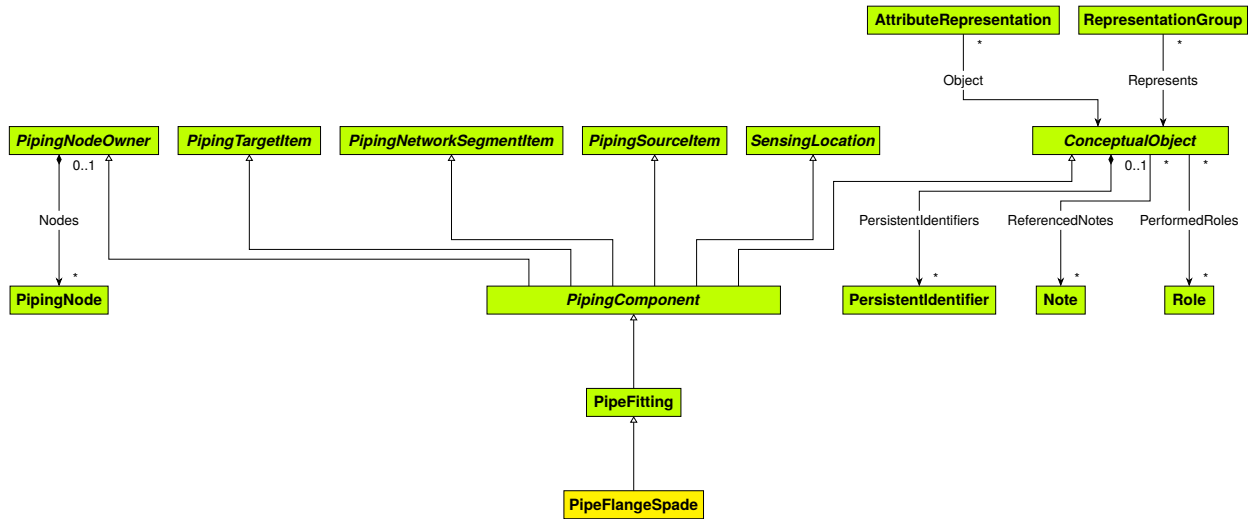
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS472679>

PIPE FLANGE SPADE

A 'line blind' and an 'artefact' that is a circular plate with no central opening and holes to match mating flanged ends. It is also equipped with a handle (from <http://data.posccaesar.org/rdl/RDS472679>).



Supertypes

- *PipeFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.PipeFlangeSpade"/>
```

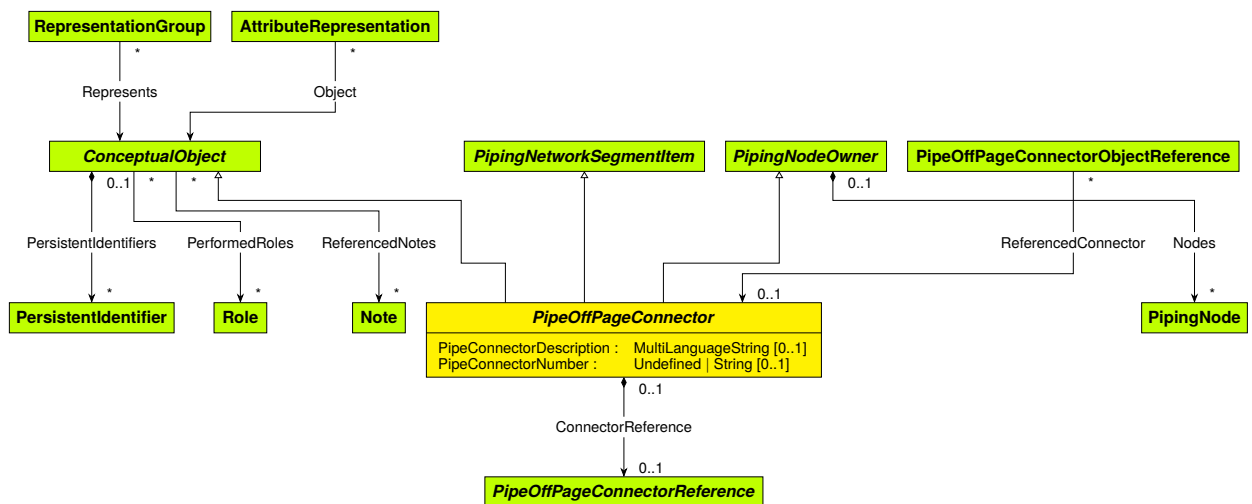
PipeOffPageConnector

Plant::Piping::PipeOffPageConnector

Overview

Abstract Class

A connector that indicates that a piping network segment is continued elsewhere, either on the same PID or on another PID. Graphically, it is usually represented as an arrow.



Supertypes

- *ConceptualObject*
- *PipingNetworkSegmentItem*
- *PipingNodeOwner*

Subtypes

- *FlowInPipeOffPageConnector*
- *FlowOutPipeOffPageConnector*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>ConnectorReference</i>	0..1	<i>PipeOffPageConnectorReference</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>PipeConnectorDescription</i>	0..1	<i>MultiLanguageString</i>	no	no
<i>PipeConnectorNumber</i>	0..1	<i>Undefined String</i>	no	no

ConnectorReference

Composition Property

A reference indicating to which other `Plant.ProcessEquipment.PipeOffPageConnector` this `Plant.ProcessEquipment.PipeOffPageConnector` is connected.

Type: *PipeOffPageConnectorReference*

Multiplicity: 0..1

Ordered: no

PipeConnectorDescription

Data Property

A description of the *PipeOffPageConnector*.

Type: *MultiLanguageString*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PipeConnectorDescriptionAssignmentClass>

PIPE CONNECTOR DESCRIPTION ASSIGNMENT CLASS

Example

```
" (('en', 'from reaction unit'), ('de', 'von Reaktionseinheit')) "
```

PipeConnectorNumber**Data Property**

The number of the *PipeOffPageConnector*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PipeConnectorNumberAssignmentClass>

PIPE CONNECTOR NUMBER ASSIGNMENT CLASS

Example

```
"123"
```

PipeOffPageConnectorObjectReference

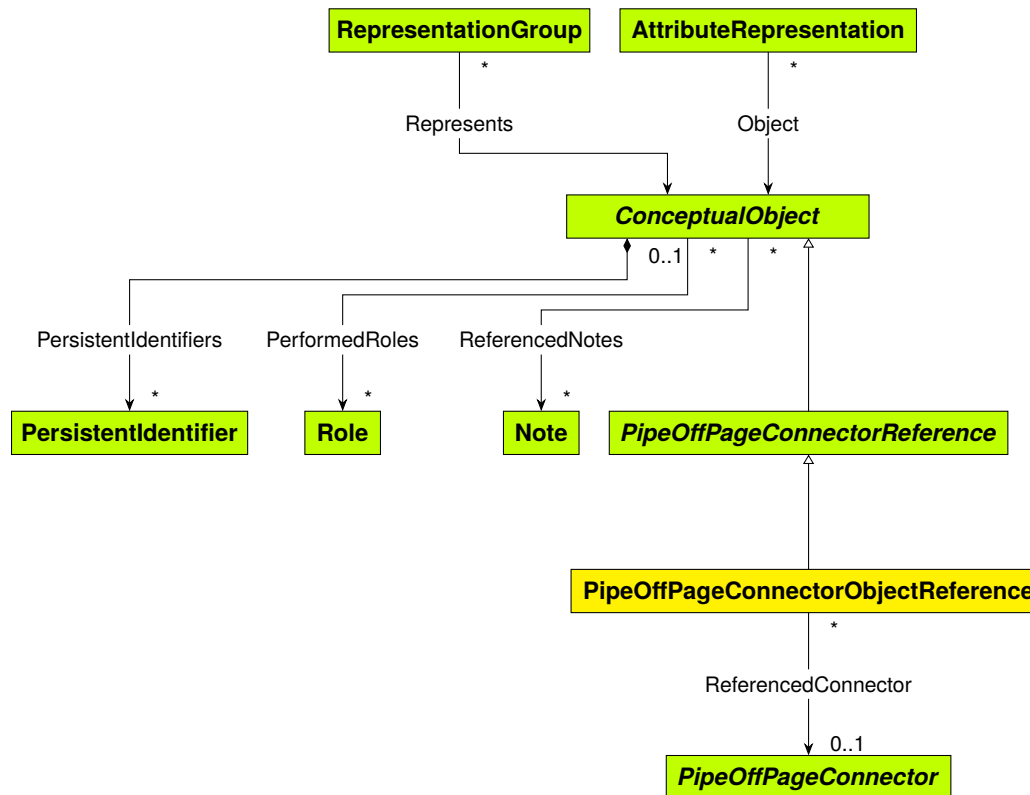
Plant::Piping::PipeOffPageConnectorObjectReference

Overview**Concrete Class**

RDL: <http://sandbox.dexpi.org/rdl/PipeOffPageConnectorObjectReference>

PIPE OFF PAGE CONNECTOR OBJECT REFERENCE

A reference to a *PipeOffPageConnector* by an association.



Supertypes

- *PipeOffPageConnectorReference*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>ReferencedConnector</i>	0..1	<i>PipeOffPageConnector</i>	no	no	*

ReferencedConnector

Reference Property

The Plant.ProcessEquipment.PipeOffPageConnector referenced.

Type: *PipeOffPageConnector*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

Example in DEXPI XML

```
<Object type="Plant/Piping.PipeOffPageConnectorObjectReference"/>
```

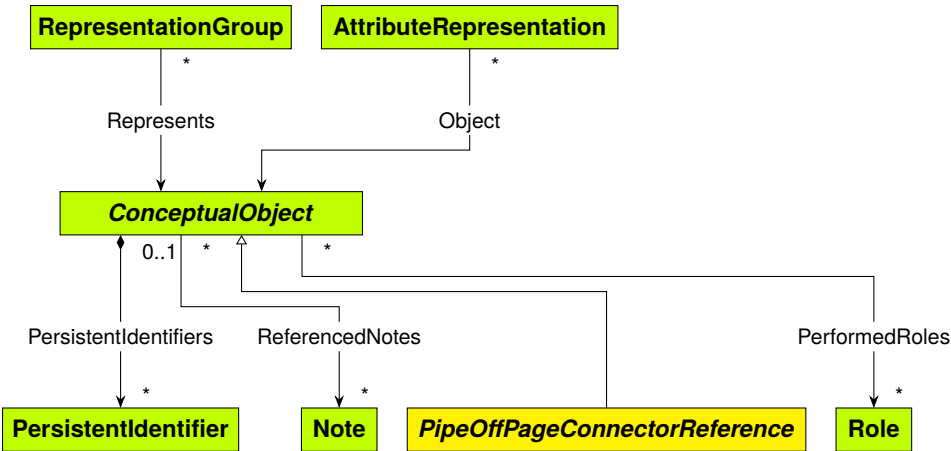
PipeOffPageConnectorReference

Plant::Piping::PipeOffPageConnectorReference

Overview

Abstract Class

A reference to a *PipeOffPageConnector*.



Supertypes

- *ConceptualObject*

Subtypes

- *PipeOffPageConnectorObjectReference*
- *PipeOffPageConnectorReferenceByNumber*

PipeOffPageConnectorReferenceByNumber

Plant::Piping::PipeOffPageConnectorReferenceByNumber

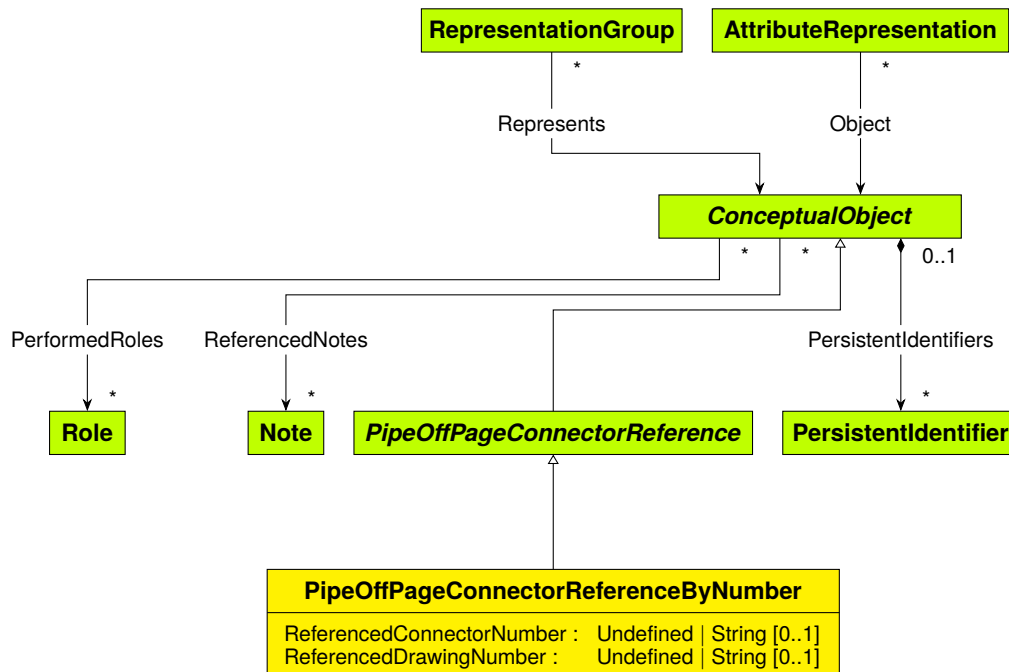
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/PipeOffPageConnectorReferenceByNumber>

PIPE OFF PAGE CONNECTOR REFERENCE BY NUMBER

A reference to a *PipeOffPageConnector* by drawing and connector number.



Supertypes

- *PipeOffPageConnectorReference*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>ReferencedConnectorNumber</i>	0..1	<i>Undefined String</i>	no	no
<i>ReferencedDrawingNumber</i>	0..1	<i>Undefined String</i>	no	no

ReferencedConnectorNumber

Data Property

The connector number of the referenced connector.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ReferencedConnectorNumberAssignmentClass>

REFERENCED CONNECTOR NUMBER ASSIGNMENT CLASS

Example

"97"

ReferencedDrawingNumber**Data Property**

The Core.Diagram.Metadata.DrawingNumberAssignmentClass of the PID that contains the referenced connector.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ReferencedDrawingNumberAssignmentClass>

REFERENCED DRAWING NUMBER ASSIGNMENT CLASS

Example

"123/A93"

Example in DEXPI XML

```

<Object type="Plant/Piping.PipeOffPageConnectorReferenceByNumber">
  <Data property="ReferencedConnectorNumber">
    <String>97</String>
  </Data>
  <Data property="ReferencedDrawingNumber">
    <String>123/A93</String>
  </Data>
</Object>

```

PipeReducer

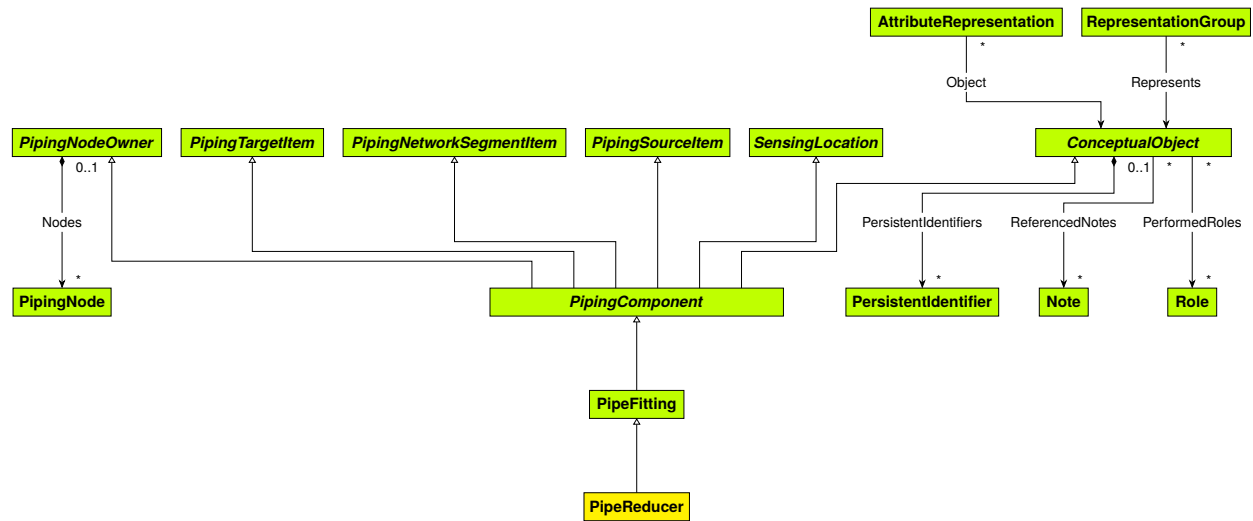
Plant::Piping::PipeReducer

Overview**Concrete Class**

RDL: <http://data.posccaesar.org/rdl/RDS416294>

PIPE REDUCER

An ‘artefact’ that has different nominal pipe size at the two ends, intended to connect pipes or piping components (from <http://data.posccaesar.org/rdl/RDS416294>).



Supertypes

- *PipeFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.PipeReducer"/>
```

PipeTee

Plant::Piping::PipeTee

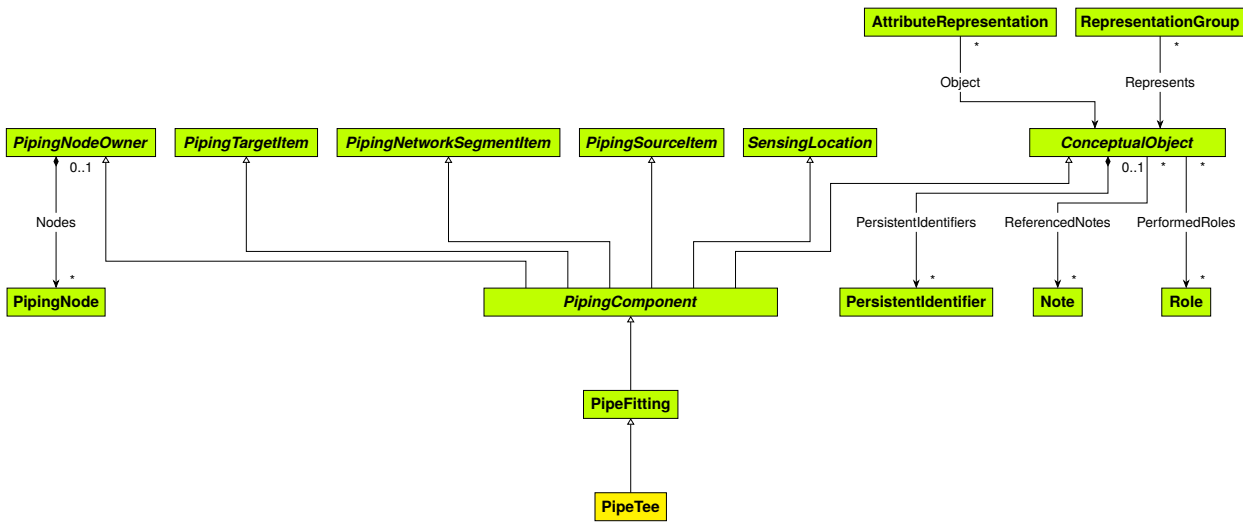
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS427724>

PIPE TEE

An ‘artefact’ that has three piping ends in T-shape, including a branch at 90 degrees (from <http://data.posccaesar.org/rdl/RDS427724>).



Supertypes

- *PipeFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.PipeTee"/>
```

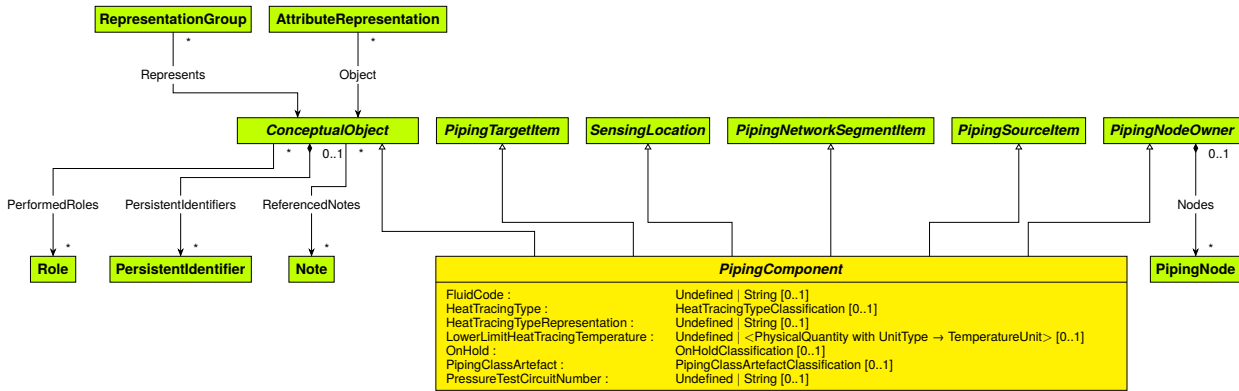
PipingComponent

Plant::Piping::PipingComponent

Overview

Abstract Class

A piping component



Supertypes

- *ConceptualObject*
- *PipingNetworkSegmentItem*
- *PipingNodeOwner*

- *PipingSourceItem*
- *PipingTargetItem*
- *SensingLocation*

Subtypes

- *CheckValve*
- *InlineMeasuringElement*
- *OperatedValve*
- *PipeFitting*
- *SafetyValveOrFitting*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>FluidCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>HeatTracingType</i>	0..1	<i>HeatTracingTypeClassification</i>	no	no
<i>HeatTracingTypeRepresentation</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>LowerLimitHeatTracingTemperature</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → TemperatureUnit></i>	no	no
<i>OnHold</i>	0..1	<i>OnHoldClassification</i>	no	no
<i>PipingClassArtefact</i>	0..1	<i>PipingClassArtefactClassification</i>	no	no
<i>PressureTestCircuitNumber</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

FluidCode

Data Property

The identification code of the fluid related to the *PipingComponent*. So far, DEXPI does not define restrictions for valid values.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/FluidCodeAssignmentClass>

FLUID CODE ASSIGNMENT CLASS

Example

"MNB"

HeatTracingType

Data Property

A specialization indicating the heat tracing type related to the *PipingComponent*.

Type: *HeatTracingTypeClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/HeatTracingTypeSpecialization>

HEAT TRACING TYPE SPECIALIZATION

Example

ElectricalHeatTracingSystem

HeatTracingTypeRepresentation

Data Property

The heat tracing type related to the *PipingComponent*, represented as a string.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/HeatTracingTypeRepresentationAssignmentClass>

HEAT TRACING TYPE REPRESENTATION ASSIGNMENT CLASS

Example

"E"

LowerLimitHeatTracingTemperature

Data Property

The lower limit for the temperature that a heat tracing system must ensure for the *PipingComponent*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with } \text{UnitType} \rightarrow \text{TemperatureUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LowerLimitHeatTracingTemperature>

LOWER LIMIT HEAT TRACING TEMPERATURE

Example

PhysicalQuantity(
 Unit = *DegreeCelsius*,
 Value = 100.0)

OnHold**Data Property**

A specialization indicating if the *PipingComponent* is on hold or not.

Type: *OnHoldClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/OnHoldSpecialization>

ON HOLD SPECIALIZATION

Example

OnHold

PipingClassArtefact**Data Property**

A specialization indicating if the *PipingComponent* is an artefact that is described by a piping class.

Type: *PipingClassArtefactClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PipingClassArtefactSpecialization>

PIPING CLASS ARTEFACT SPECIALIZATION

Example

PipingClassArtefact

PressureTestCircuitNumber**Data Property**

The number of the pressure test circuit of the *PipingComponent*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PressureTestCircuitNumberAssignmentClass>

PRESSURE TEST CIRCUIT NUMBER ASSIGNMENT CLASS

Example

"TC123"

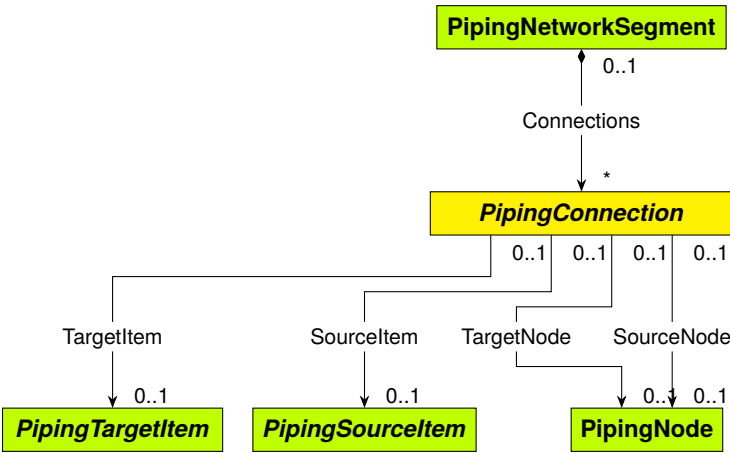
PipingConnection

Plant::Piping::PipingConnection

Overview

Abstract Class

An elementary connection between two piping items.



Subtypes

- *DirectPipingConnection*
- *Pipe*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>SourceItem</i>	0..1	<i>PipingSourceItem</i>	no	no	0..1
<i>SourceNode</i>	0..1	<i>PipingNode</i>	no	no	0..1
<i>TargetItem</i>	0..1	<i>PipingTargetItem</i>	no	no	0..1
<i>TargetNode</i>	0..1	<i>PipingNode</i>	no	no	0..1

SourceItem

Reference Property

The *PipingSourceItem* at which the PipingConnection starts.

Type: *PipingSourceItem*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

SourceNode

Reference Property

The *PipingNode* at which the PipingConnection starts. The SourceNode must belong to the *SourceItem*.

Type: *PipingNode*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

TargetItem

Reference Property

The *PipingTargetItem* at which the PipingConnection starts.

Type: *PipingTargetItem*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

TargetNode

Reference Property

The *PipingNode* at which the PipingConnection ends. The TargetNode must belong to the *TargetItem*.

Type: *PipingNode*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

PipingNetworkSegment

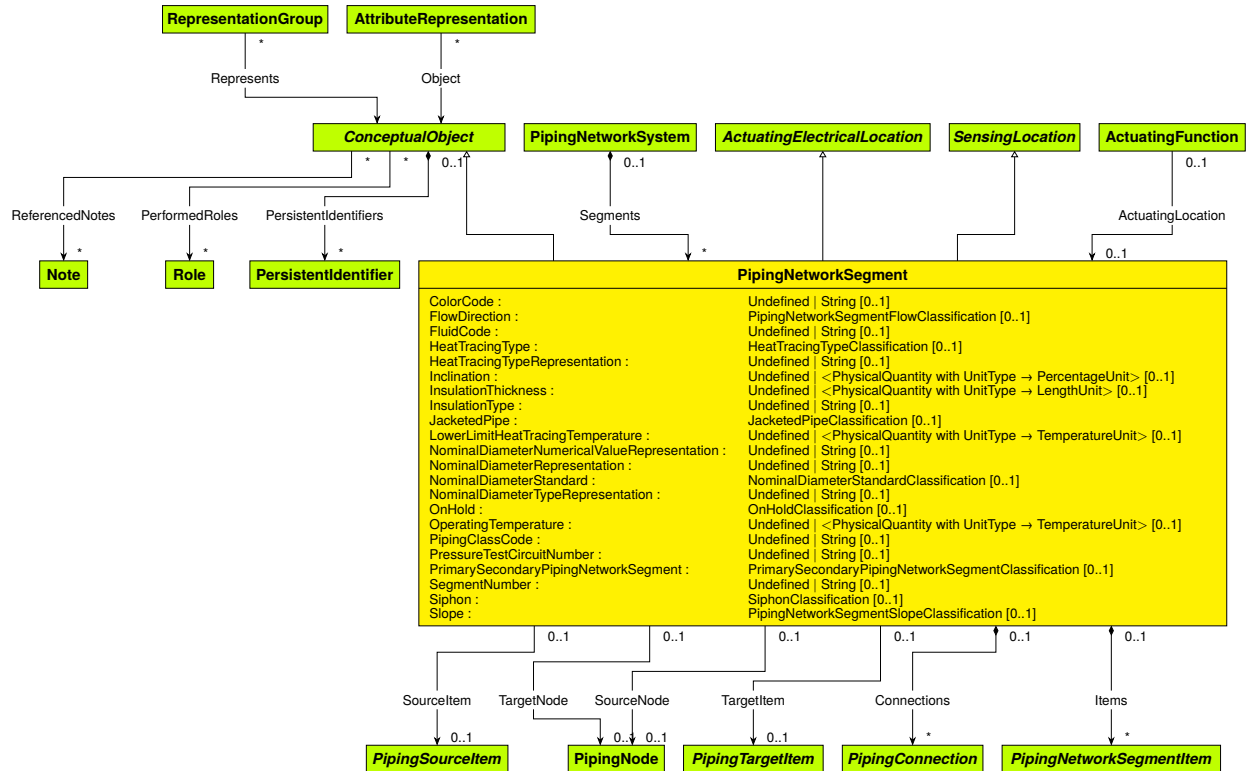
Plant::Piping::PipingNetworkSegment

Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS267704>

PIPING NETWORK SEGMENT



Supertypes

- *ActuatingElectricalLocation*
- *ConceptualObject*
- *SensingLocation*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Connections</i>	*	<i>PipingConnection</i>	no
<i>Items</i>	*	<i>PipingNetworkSegmentItem</i>	no

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>SourceItem</i>	0..1	<i>PipingSourceItem</i>	no	no	0..1
<i>SourceNode</i>	0..1	<i>PipingNode</i>	no	no	0..1
<i>TargetItem</i>	0..1	<i>PipingTargetItem</i>	no	no	0..1
<i>TargetNode</i>	0..1	<i>PipingNode</i>	no	no	0..1

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>ColorCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>FlowDirection</i>	0..1	<i>PipingNetworkSegmentFlowClassification</i>	no	no
<i>FluidCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>HeatTracingType</i>	0..1	<i>HeatTracingTypeClassification</i>	no	no
<i>HeatTracingTypeRepresentation</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>Inclination</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PercentageUnit></i>	no	no
<i>InsulationThickness</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no
<i>InsulationType</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>JacketedPipe</i>	0..1	<i>JacketedPipeClassification</i>	no	no
<i>LowerLimitHeatTracingTemperature</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → TemperatureUnit></i>	no	no
<i>NominalDiameterNumericalValueRepresentation</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>NominalDiameterRepresentation</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>NominalDiameterStandard</i>	0..1	<i>NominalDiameterStandardClassification</i>	no	no
<i>NominalDiameterTypeRepresentation</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>OnHold</i>	0..1	<i>OnHoldClassification</i>	no	no
<i>OperatingTemperature</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → TemperatureUnit></i>	no	no
<i>PipingClassCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>PressureTestCircuitNumber</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>PrimarySecondaryPipingNetworkSegment</i>	0..1	<i>PrimarySecondaryPipingNetworkSegmentClassification</i>	no	no
<i>SegmentNumber</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>Siphon</i>	0..1	<i>SiphonClassification</i>	no	no
<i>Slope</i>	0..1	<i>PipingNetworkSegmentSlopeClassification</i>	no	no

ColorCode

Data Property

The color code of the *PipingNetworkSegment*, represented as a string.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ColorCodeAssignmentClass>

COLOR CODE ASSIGNMENT CLASS

Example

"C321"

Connections

Composition Property

The connections of the *PipingNetworkSegment*.

Type: *PipingConnection*

Multiplicity: *

Ordered: no

FlowDirection

Data Property

A specialization indicating if the *PipingNetworkSegment* enables dual flow or not.

Type: *PipingNetworkSegmentFlowClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/FlowDirectionSpecialization>

FLOW DIRECTION SPECIALIZATION

Example

DualFlowPipingNetworkSegment

FluidCode

Data Property

The identification code of the fluid related to the *PipingNetworkSegment*. So far, DEXPI does not define restrictions for valid values.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/FluidCodeAssignmentClass>

FLUID CODE ASSIGNMENT CLASS

Example

"MNB"

HeatTracingType**Data Property**

A specialization indicating the heat tracing type related to the *PipingNetworkSegment*.

Type: *HeatTracingTypeClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/HeatTracingTypeSpecialization>

HEAT TRACING TYPE SPECIALIZATION

Example

ElectricalHeatTracingSystem

HeatTracingTypeRepresentation**Data Property**

The heat tracing type related to the *PipingNetworkSegment*, represented as a string.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/HeatTracingTypeRepresentationAssignmentClass>

HEAT TRACING TYPE REPRESENTATION ASSIGNMENT CLASS

Example

"E"

Inclination**Data Property**

The inclination (slope) of the *PipingNetworkSegment* in percent.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PercentageUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS17688057>

INCLINATION

Example

PhysicalQuantity(
 Unit = *Percent*,
 Value = 10.0)

InsulationThickness**Data Property**

The insulation thickness of the *PipingNetworkSegment*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → LengthUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS4238040>

INSULATION THICKNESS

Example

PhysicalQuantity(
 Unit = *Millimetre*,
 Value = 40.0)

InsulationType**Data Property**

The identification code for the insulation type related to the *PipingNetworkSegment*. So far, DEXPI does not define restrictions for valid values.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/InsulationTypeAssignmentClass>

INSULATION TYPE ASSIGNMENT CLASS

Example

"Q"

Items**Composition Property**

The items of the *PipingNetworkSegment*.

Type: *PipingNetworkSegmentItem*

Multiplicity: *

Ordered: no

JacketedPipe

Data Property

A specialization indicating whether the *PipingNetworkSegment* is jacketed.

Type: *JacketedPipeClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/JacketedPipeSpecialization>

JACKETED PIPE SPECIALIZATION

Example

JacketedPipe

LowerLimitHeatTracingTemperature

Data Property

The lower limit for the temperature that a heat tracing system must ensure for the *PipingNetworkSegment*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → TemperatureUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LowerLimitHeatTracingTemperature>

LOWER LIMIT HEAT TRACING TEMPERATURE

Example

PhysicalQuantity(

Unit = *DegreeCelsius*,

Value = 100.0)

NominalDiameterNumericalValueRepresentation

Data Property

A readable representation of the numerical value of the nominal diameter of the *PipingNetworkSegment*, without any type or unit of measure.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalDiameterNumericalValueRepresentationAssignmentClass>

NOMINAL DIAMETER NUMERICAL VALUE REPRESENTATION ASSIGNMENT CLASS

Example

"25"

NominalDiameterRepresentation

Data Property

A readable representation of the nominal diameter of the *PipingNetworkSegment*. It normally contains a numerical value and a type or unit of measure.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalDiameterRepresentationAssignmentClass>

NOMINAL DIAMETER REPRESENTATION ASSIGNMENT CLASS

Example

"DN 25"

NominalDiameterStandard

Data Property

The nominal diameter of the *PipingNetworkSegment*, given as a reference to a nominal diameter standard and value.

Type: *NominalDiameterStandardClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalDiameterStandardSpecialization>

NOMINAL DIAMETER STANDARD SPECIALIZATION

Example

Din2448ObjectDn25

NominalDiameterTypeRepresentation

Data Property

A readable representation of the type or unit of measure of the nominal diameter of the *PipingNetworkSegment*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalDiameterTypeRepresentationAssignmentClass>

NOMINAL DIAMETER TYPE REPRESENTATION ASSIGNMENT CLASS

Example

"DN"

OnHold

Data Property

A specialization indicating if the *PipingNetworkSegment* is on hold or not.

Type: *OnHoldClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/OnHoldSpecialization>

ON HOLD SPECIALIZATION

Example

OnHold

OperatingTemperature

Data Property

The operating temperature of the *PipingNetworkSegment*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{TemperatureUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS357119>

OPERATING TEMPERATURE

Example

PhysicalQuantity(
 Unit = *DegreeCelsius*,
 Value = 100.0)

PipingClassCode

Data Property

The identification code of the piping class of the *PipingNetworkSegment*. So far, DEXPI does not define restrictions for valid values.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PipingClassCodeAssignmentClass>

PIPING CLASS CODE ASSIGNMENT CLASS

Example

"75HB13"

PressureTestCircuitNumber

Data Property

The number of the pressure test circuit of the *PipingNetworkSegment*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PressureTestCircuitNumberAssignmentClass>

PRESSURE TEST CIRCUIT NUMBER ASSIGNMENT CLASS

Example

"TC123"

PrimarySecondaryPipingNetworkSegment

Data Property

A specialization indicating whether the *PipingNetworkSegment* is a primary or secondary *PipingNetworkSegment*.

Type: *PrimarySecondaryPipingNetworkSegmentClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PrimarySecondaryPipingNetworkSegmentSpecialization>

PRIMARY SECONDARY PIPING NETWORK SEGMENT SPECIALIZATION

Example

PrimaryPipingNetworkSegment

SegmentNumber

Data Property

The segment number of a *PipingNetworkSegment*. Values are typically (but not necessarily) string representations of numbers with a prefix.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SegmentNumberAssignmentClass>

SEGMENT NUMBER ASSIGNMENT CLASS

Example

"S3"

Siphon

Data Property

A specialization indicating if the *PipingNetworkSegment* is a siphon or not.

Type: *SiphonClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SiphonSpecialization>

SIPHON SPECIALIZATION

Example

Siphon

Slope

Data Property

A specialization indicating if the *PipingNetworkSegment* is sloped or not.

Type: *PipingNetworkSegmentSlopeClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SlopeSpecialization>

SLOPE SPECIALIZATION

Example

SlopedPipingNetworkSegment

SourceItem

Reference Property

The item at which the PipingNetworkSegment starts.

Type: *PipingSourceItem*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

SourceNode

Reference Property

The *PipingNode* at which the PipingNetworkSegment starts.

Type: *PipingNode*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

TargetItem

Reference Property

The item at which the PipingNetworkSegment ends.

Type: *PipingTargetItem*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

TargetNode

Reference Property

The *PipingNode* at which the PipingNetworkSegment ends.

Type: *PipingNode*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

Example in DEXPI XML

```
<Object type="Plant/Piping.PipingNetworkSegment">
  <Data property="ColorCode">
    <String>C321</String>
  </Data>
  <Data property="FlowDirection">
    <DataReference data="Plant/Enumerations.PipingNetworkSegmentFlowClassification.
↪DualFlowPipingNetworkSegment"/>
  </Data>
  <Data property="FluidCode">
    <String>MNb</String>
  </Data>
  <Data property="HeatTracingType">
    <DataReference data="Plant/Enumerations.HeatTracingTypeClassification.
↪ElectricalHeatTracingSystem"/>
  </Data>
  <Data property="HeatTracingTypeRepresentation">
    <String>E</String>
  </Data>
  <Data property="Inclination">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PercentageUnit.Percent"/>
      </Data>
      <Data property="Value">
        <Double>10.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="InsulationThickness">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Millimetre"/>
      </Data>
      <Data property="Value">
        <Double>40.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="InsulationType">
    <String>Q</String>
  </Data>
  <Data property="JacketedPipe">
    <DataReference data="Plant/Enumerations.JacketedPipeClassification.JacketedPipe"/>
  </Data>
  <Data property="LowerLimitHeatTracingTemperature">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
```

(continues on next page)

(continued from previous page)

```

        <DataReference data="Core/PhysicalQuantities.TemperatureUnit.DegreeCelsius"/>
    </Data>
    <Data property="Value">
        <Double>100.0</Double>
    </Data>
</AggregatedDataValue>
</Data>
<Data property="NominalDiameterNumericalValueRepresentation">
    <String>25</String>
</Data>
<Data property="NominalDiameterRepresentation">
    <String>DN 25</String>
</Data>
<Data property="NominalDiameterStandard">
    <DataReference data="Plant/Enumerations.NominalDiameterStandardClassification.
↪Din24480ObjectDn25"/>
</Data>
<Data property="NominalDiameterTypeRepresentation">
    <String>DN</String>
</Data>
<Data property="OnHold">
    <DataReference data="Plant/Enumerations.OnHoldClassification.OnHold"/>
</Data>
<Data property="OperatingTemperature">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.TemperatureUnit.DegreeCelsius"/>
        </Data>
        <Data property="Value">
            <Double>100.0</Double>
        </Data>
    </AggregatedDataValue>
</Data>
<Data property="PipingClassCode">
    <String>75HB13</String>
</Data>
<Data property="PressureTestCircuitNumber">
    <String>TC123</String>
</Data>
<Data property="PrimarySecondaryPipingNetworkSegment">
    <DataReference data="Plant/Enumerations.
↪PrimarySecondaryPipingNetworkSegmentClassification.PrimaryPipingNetworkSegment"/>
</Data>
<Data property="SegmentNumber">
    <String>S3</String>
</Data>
<Data property="Siphon">
    <DataReference data="Plant/Enumerations.SiphonClassification.Siphon"/>
</Data>
<Data property="Slope">
    <DataReference data="Plant/Enumerations.PipingNetworkSegmentSlopeClassification.
↪SlopedPipingNetworkSegment"/>

```

(continues on next page)

(continued from previous page)

```
</Data>
</Object>
```

PipingNetworkSegmentItem

Plant::Piping::PipingNetworkSegmentItem

Overview

Abstract Class

An item that can be part of a *PipingNetworkSegment*.

PipingNetworkSegmentItem

Subtypes

- *PipeOffPageConnector*
- *PipingComponent*
- *PropertyBreak*

PipingNetworkSystem

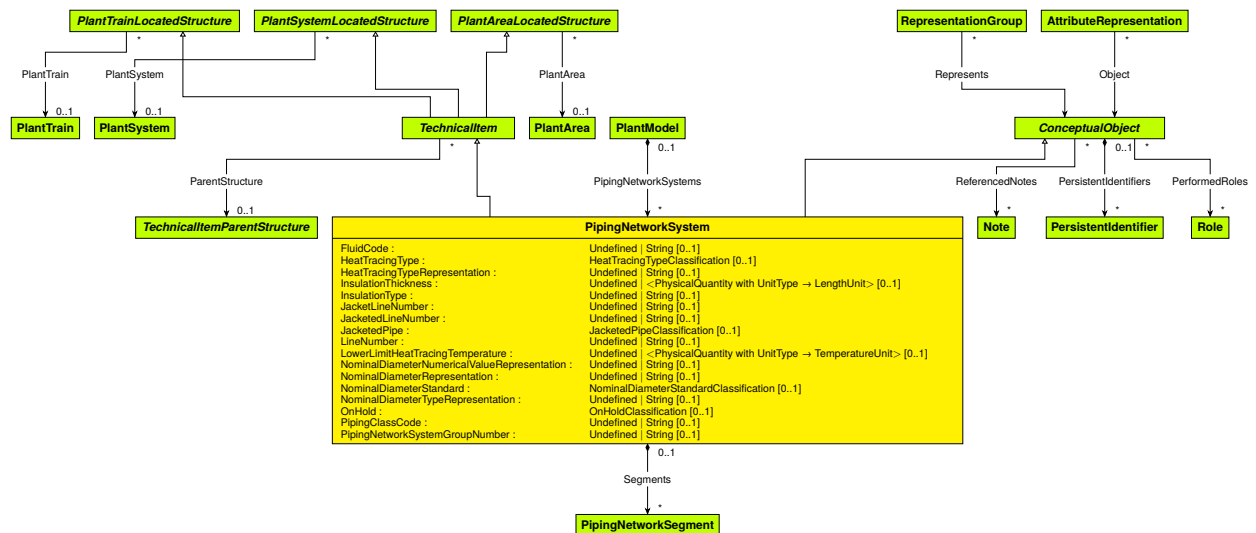
Plant::Piping::PipingNetworkSystem

Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS270359>

PIPING NETWORK SYSTEM



Supertypes

- *ConceptualObject*
- *TechnicalItem*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Segments</i>	*	<i>PipingNetworkSegment</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>FluidCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>HeatTracingType</i>	0..1	<i>HeatTracingTypeClassification</i>	no	no
<i>HeatTracingTypeRepresentation</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>InsulationThickness</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no
<i>InsulationType</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>JacketLineNumber</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>JacketedLineNumber</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>JacketedPipe</i>	0..1	<i>JacketedPipeClassification</i>	no	no
<i>LineNumber</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>LowerLimitHeatTracingTemperature</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → TemperatureUnit></i>	no	no
<i>NominalDiameterNumericalValueRepresentation</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>NominalDiameterRepresentation</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>NominalDiameterStandard</i>	0..1	<i>NominalDiameterStandardClassification</i>	no	no
<i>NominalDiameterTypeRepresentation</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>OnHold</i>	0..1	<i>OnHoldClassification</i>	no	no
<i>PipingClassCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>PipingNetworkSystemGroupNumber</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

FluidCode

Data Property

The identification code of the fluid related to the *PipingNetworkSystem*. So far, DEXPI does not define restrictions for valid values.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/FluidCodeAssignmentClass>

FLUID CODE ASSIGNMENT CLASS

Example

"MNb"

HeatTracingType

Data Property

A specialization indicating the heat tracing type related to the *PipingNetworkSystem*.

Type: *HeatTracingTypeClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/HeatTracingTypeSpecialization>

HEAT TRACING TYPE SPECIALIZATION

Example

ElectricalHeatTracingSystem

HeatTracingTypeRepresentation

Data Property

The heat tracing type related to the *PipingNetworkSystem*, represented as a string.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/HeatTracingTypeRepresentationAssignmentClass>

HEAT TRACING TYPE REPRESENTATION ASSIGNMENT CLASS

Example

"E"

InsulationThickness

Data Property

The insulation thickness of the *PipingNetworkSystem*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with } \text{UnitType} \rightarrow \text{LengthUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS4238040>

INSULATION THICKNESS

Example

PhysicalQuantity(

Unit = *Millimetre*,

Value = 40.0)

InsulationType

Data Property

The identification code for the insulation type related to the *PipingNetworkSystem*. So far, DEXPI does not define restrictions for valid values.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/InsulationTypeAssignmentClass>

INSULATION TYPE ASSIGNMENT CLASS

Example

"Q"

JacketLineNumber

Data Property

The line number of the PipingNetworkSystem that is the jacket of this *PipingNetworkSystem*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/JacketLineNumberAssignmentClass>

JACKET LINE NUMBER ASSIGNMENT CLASS

Example

"47126J"

JacketedLineNumber

Data Property

The line number of the *PipingNetworkSystem* for which this *PipingNetworkSystem* is the jacket.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/JacketedLineNumberAssignmentClass>

JACKETED LINE NUMBER ASSIGNMENT CLASS

Example

"47126"

JacketedPipe

Data Property

A specialization indicating whether the *PipingNetworkSystem* is jacketed.

Type: *JacketedPipeClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/JacketedPipeSpecialization>

JACKETED PIPE SPECIALIZATION

Example

JacketedPipe

LineNumber

Data Property

The line number of a *PipingNetworkSystem*. Values are typically (but not necessarily) string representations of numbers.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LineNumberAssignmentClass>

LINE NUMBER ASSIGNMENT CLASS

Example

"47126"

LowerLimitHeatTracingTemperature**Data Property**

The lower limit for the temperature that a heat tracing system must ensure for the *PipingNetworkSystem*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → TemperatureUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LowerLimitHeatTracingTemperature>

LOWER LIMIT HEAT TRACING TEMPERATURE

Example

```
PhysicalQuantity(
  Unit = DegreeCelsius,
  Value = 100.0)
```

NominalDiameterNumericalValueRepresentation**Data Property**

A readable representation of the numerical value of the nominal diameter of the *PipingNetworkSystem*, without any type or unit of measure.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalDiameterNumericalValueRepresentationAssignmentClass>

NOMINAL DIAMETER NUMERICAL VALUE REPRESENTATION ASSIGNMENT CLASS

Example

"25"

NominalDiameterRepresentation**Data Property**

A readable representation of the nominal diameter of the *PipingNetworkSystem*. It normally contains a numerical value and a type or unit of measure.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalDiameterRepresentationAssignmentClass>

NOMINAL DIAMETER REPRESENTATION ASSIGNMENT CLASS

Example

"DN 25"

NominalDiameterStandard

Data Property

The nominal diameter of the *PipingNetworkSystem*, given as a reference to a nominal diameter standard and value.

Type: *NominalDiameterStandardClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalDiameterStandardSpecialization>

NOMINAL DIAMETER STANDARD SPECIALIZATION

Example

Din2448ObjectDn25

NominalDiameterTypeRepresentation

Data Property

A readable representation of the type or unit of measure of the nominal diameter of the *PipingNetworkSystem*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalDiameterTypeRepresentationAssignmentClass>

NOMINAL DIAMETER TYPE REPRESENTATION ASSIGNMENT CLASS

Example

"DN"

OnHold

Data Property

A specialization indicating if the *PipingNetworkSystem* is on hold or not.

Type: *OnHoldClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/OnHoldSpecialization>

ON HOLD SPECIALIZATION

Example

OnHold

PipingClassCode

Data Property

The identification code of the piping class of the *PipingNetworkSystem*. So far, DEXPI does not define restrictions for valid values.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PipingClassCodeAssignmentClass>

PIPING CLASS CODE ASSIGNMENT CLASS

Example

"75HB13"

PipingNetworkSystemGroupNumber

Data Property

The number of the piping network system group of the *PipingNetworkSystem*, represented as a string.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PipingNetworkSystemGroupNumberAssignmentClass>

PIPING NETWORK SYSTEM GROUP NUMBER ASSIGNMENT CLASS

Example

"G3"

Segments

Composition Property

The segments of the PipingNetworkSystem.

Type: *PipingNetworkSegment*

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/Piping.PipingNetworkSystem">
  <Data property="FluidCode">
    <String>MNb</String>
  </Data>
  <Data property="HeatTracingType">
    <DataReference data="Plant/Enumerations.HeatTracingTypeClassification.
    ↪ElectricalHeatTracingSystem"/>
  </Data>
  <Data property="HeatTracingTypeRepresentation">
    <String>E</String>
  </Data>
  <Data property="InsulationThickness">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Millimetre"/>
      </Data>
      <Data property="Value">
        <Double>40.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="InsulationType">
    <String>Q</String>
  </Data>
  <Data property="JacketLineNumber">
    <String>47126J</String>
  </Data>
  <Data property="JacketedLineNumber">
    <String>47126</String>
  </Data>
  <Data property="JacketedPipe">
    <DataReference data="Plant/Enumerations.JacketedPipeClassification.JacketedPipe"/>
  </Data>
  <Data property="LineNumber">
    <String>47126</String>
  </Data>
  <Data property="LowerLimitHeatTracingTemperature">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.TemperatureUnit.DegreeCelsius"/>
      </Data>
```

(continues on next page)

(continued from previous page)

```
<Data property="Value">
  <Double>100.0</Double>
</Data>
</AggregatedDataValue>
</Data>
<Data property="NominalDiameterNumericalValueRepresentation">
  <String>25</String>
</Data>
<Data property="NominalDiameterRepresentation">
  <String>DN 25</String>
</Data>
<Data property="NominalDiameterStandard">
  <DataReference data="Plant/Enumerations.NominalDiameterStandardClassification.
↪Din2448ObjectDn25"/>
</Data>
<Data property="NominalDiameterTypeRepresentation">
  <String>DN</String>
</Data>
<Data property="OnHold">
  <DataReference data="Plant/Enumerations.OnHoldClassification.OnHold"/>
</Data>
<Data property="PipingClassCode">
  <String>75HB13</String>
</Data>
<Data property="PipingNetworkSystemGroupNumber">
  <String>G3</String>
</Data>
</Object>
```

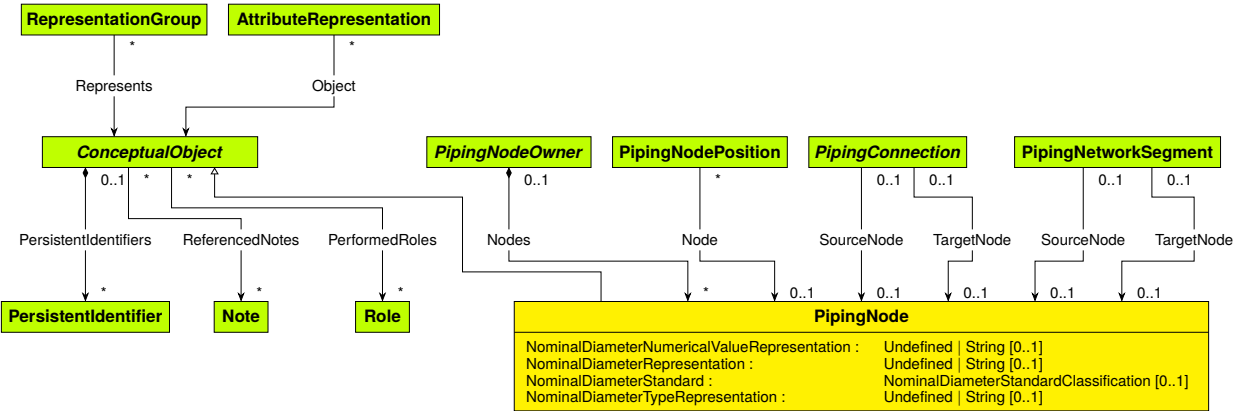
PipingNode

Plant::Piping::PipingNode

Overview

Concrete Class

A possible connection point for a *PipingConnection*.



Supertypes

- *ConceptualObject*

Members

Data Properties

Name	Multiplic- ity	Type	Or- dered	Unique
<i>NominalDiameterNumericalValueRepresentation</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>NominalDiameterRepresentation</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>NominalDiameterStandard</i>	0..1	<i>NominalDiameterStandardClassi- fication</i>	no	no
<i>NominalDiameterTypeRepresentation</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

NominalDiameterNumericalValueRepresentation

Data Property

A readable representation of the numerical value of the nominal diameter of the *PipingNode*, without any type or unit of measure.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalDiameterNumericalValueRepresentationAssignmentClass>

NOMINAL DIAMETER NUMERICAL VALUE REPRESENTATION ASSIGNMENT CLASS

Example

"25"

NominalDiameterRepresentation

Data Property

A readable representation of the nominal diameter of the *PipingNode*. It normally contains a numerical value and a type or unit of measure.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalDiameterRepresentationAssignmentClass>

NOMINAL DIAMETER REPRESENTATION ASSIGNMENT CLASS

Example

"DN 25"

NominalDiameterStandard**Data Property**

The nominal diameter of the *PipingNode*, given as a reference to a nominal diameter standard and value.

Type: *NominalDiameterStandardClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalDiameterStandardSpecialization>

NOMINAL DIAMETER STANDARD SPECIALIZATION

Example

Din2448ObjectDn25

NominalDiameterTypeRepresentation**Data Property**

A readable representation of the type or unit of measure of the nominal diameter of the *PipingNode*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalDiameterTypeRepresentationAssignmentClass>

NOMINAL DIAMETER TYPE REPRESENTATION ASSIGNMENT CLASS

Example

"DN"

Example in DEXPI XML

```

<Object type="Plant/Piping.PipingNode">
  <Data property="NominalDiameterNumericalValueRepresentation">
    <String>25</String>
  </Data>
  <Data property="NominalDiameterRepresentation">
    <String>DN 25</String>
  </Data>
  <Data property="NominalDiameterStandard">
    <DataReference data="Plant/Enumerations.NominalDiameterStandardClassification.
    ↪Din2448ObjectDn25"/>

```

(continues on next page)

(continued from previous page)

```
</Data>
<Data property="NominalDiameterTypeRepresentation">
  <String>DN</String>
</Data>
</Object>
```

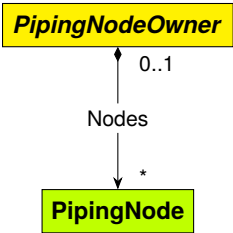
PipingNodeOwner

Plant::Piping::PipingNodeOwner

Overview

Abstract Class

An object that can have *PipingNodes*.



Subtypes

- *Nozzle*
- *PipeOffPageConnector*
- *PipingComponent*
- *PropertyBreak*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Nodes</i>	*	<i>PipingNode</i>	no

Nodes

Composition Property

The *PipingNodes* of the PipingNodeOwner.

Type: *PipingNode*

Multiplicity: *

Ordered: no

PipingSourceItem

Plant::Piping::PipingSourceItem

Overview

Abstract Class

An item that can be the source of a *PipingConnection* (attribute *SourceItem*) or a *PipingNetworkSegment* (attribute *SourceItem*).

PipingSourceItem

Subtypes

- *FlowInPipeOffPageConnector*
- *Nozzle*
- *PipingComponent*
- *PropertyBreak*

PipingTargetItem

Plant::Piping::PipingTargetItem

Overview

Abstract Class

An item that can be the target of a *PipingConnection* (attribute *TargetItem*) or a *PipingNetworkSegment* (attribute *TargetItem*).

PipingTargetItem

Subtypes

- *FlowOutPipeOffPageConnector*
- *Nozzle*
- *PipingComponent*
- *PropertyBreak*

PlugValve

Plant::Piping::PlugValve

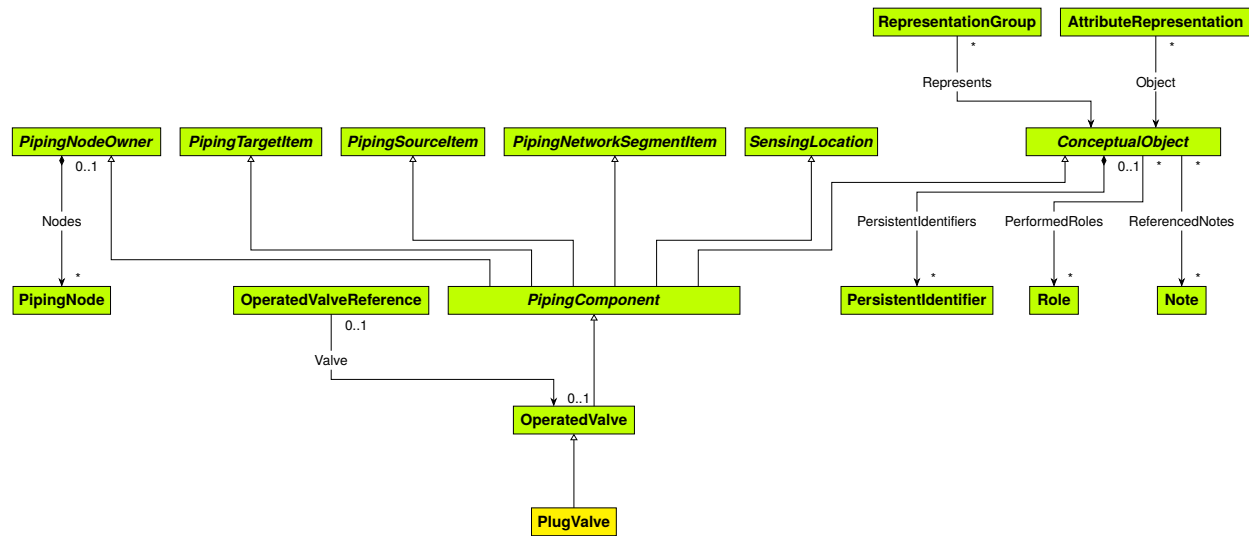
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS421109>

PLUG VALVE

A rotary valve that has a quarter turn action in which the closure member is a cylindrical or tapered plug which operates by rotating on its axis and sealing against a downstream seat (from <http://data.posccaesar.org/rdl/RDS421109>).



Supertypes

- *OperatedValve*

Example in DEXPI XML

```
<Object type="Plant/Piping.PlugValve"/>
```

PositiveDisplacementFlowMeter

Plant::Piping::PositiveDisplacementFlowMeter

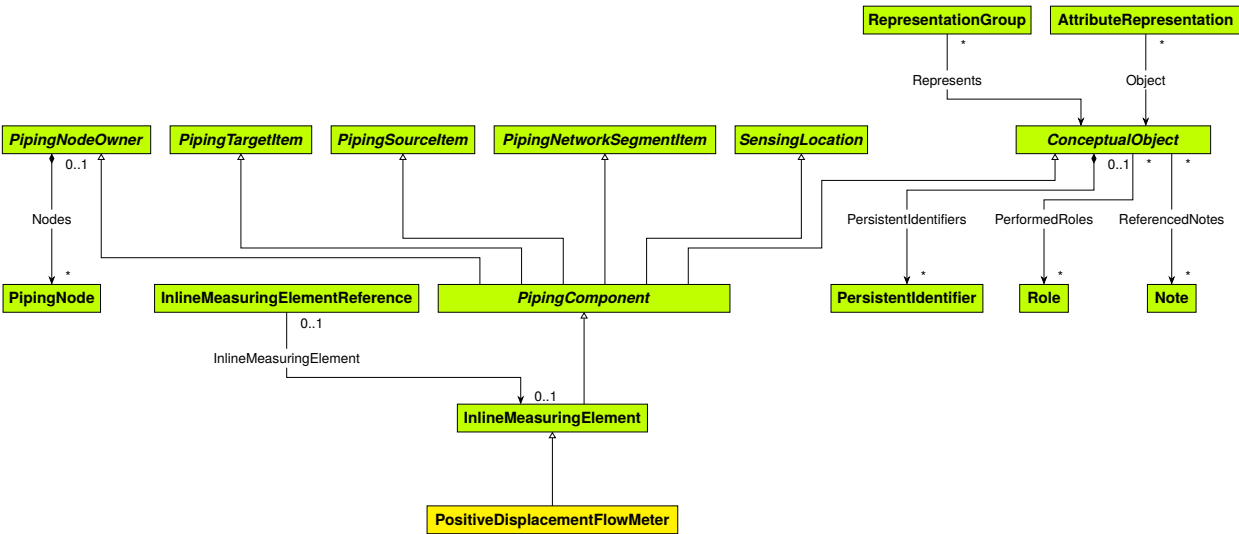
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS418094>

POSITIVE DISPLACEMENT FLOW METER

A flow meter that measures the volumetric flow rate of a liquid or gas by separating the flow stream into known volumes and counting them over time (from <http://data.posccaesar.org/rdl/RDS418094>).



Supertypes

- *InlineMeasuringElement*

Example in DEXPI XML

```
<Object type="Plant/Piping.PositiveDisplacementFlowMeter"/>
```

PropertyBreak

Plant::Piping::PropertyBreak

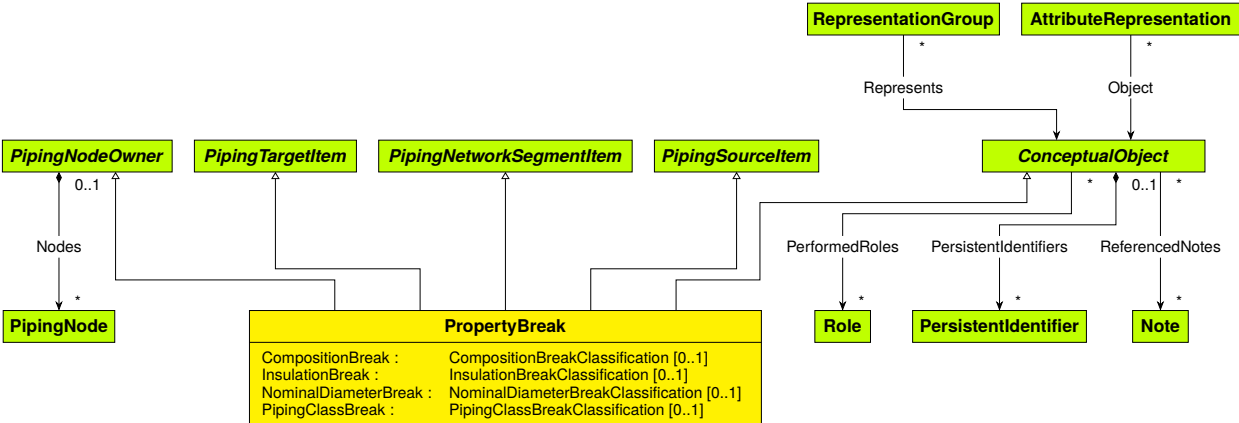
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/PropertyBreak>

PROPERTY BREAK

An explicit indication of a change of piping properties.



Supertypes

- *ConceptualObject*
- *PipingNetworkSegmentItem*
- *PipingNodeOwner*
- *PipingSourceItem*
- *PipingTargetItem*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>CompositionBreak</i>	0..1	<i>CompositionBreakClassification</i>	no	no
<i>InsulationBreak</i>	0..1	<i>InsulationBreakClassification</i>	no	no
<i>NominalDiameterBreak</i>	0..1	<i>NominalDiameterBreakClassification</i>	no	no
<i>PipingClassBreak</i>	0..1	<i>PipingClassBreakClassification</i>	no	no

This specification does not prescribe the use of PropertyBreaks whenever piping properties change. For example, consider a *PipeReducer* where, by definition, the nominal diameter is changed. According to the Proteus Specification, a new *PipingNetworkSegment* must begin immediately after the *PipeReducer*. This *PipingNetworkSegment* will have a different nominal diameter than the one before. Thus, the property break information is available without an explicit PropertyBreak. A PropertyBreak shall be used to emphasize such a change. In a *Diagram*, a PropertyBreak is typically visualized by a *Core.Diagram.PropertyBreakShape*.

A single PropertyBreak can indicate the change of *several* properties e.g., the nominal diameter *and* the piping class. It is recommended to avoid such PropertyBreaks that concern several properties. Instead, for each changed property a separate PropertyBreak should be used. These PropertyBreaks should then be connected with *DirectPipingConnections*. In a *Diagram*, each PropertyBreak can then be visualized by a separate *Core.Diagram.PropertyBreakShape* symbol. In case a diagram contains dedicated symbols for certain combinations of changed properties, it is still admissible to use a single PropertyBreak for these combinations.

CompositionBreak

Data Property

A specialization indicating if the *PropertyBreak* is a composition break or not.

Type: *CompositionBreakClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/CompositionBreakSpecialization>

COMPOSITION BREAK SPECIALIZATION

Example

NoCompositionBreak

InsulationBreak**Data Property**

A specialization indicating if the *PropertyBreak* is an insulation break or not.

Type: *InsulationBreakClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/InsulationBreakSpecialization>

INSULATION BREAK SPECIALIZATION

Example

InsulationBreak

NominalDiameterBreak**Data Property**

A specialization indicating if the *PropertyBreak* is a nominal diameter break or not.

Type: *NominalDiameterBreakClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalDiameterBreakSpecialization>

NOMINAL DIAMETER BREAK SPECIALIZATION

Example

NoNominalDiameterBreak

PipingClassBreak**Data Property**

A specialization indicating if the *PropertyBreak* is a piping class break or not.

Type: *PipingClassBreakClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PipingClassBreakSpecialization>

PIPING CLASS BREAK SPECIALIZATION

Example

PipingClassBreak

Example in DEXPI XML

```
<Object type="Plant/Piping.PropertyBreak">
  <Data property="CompositionBreak">
    <DataReference data="Plant/Enumerations.CompositionBreakClassification.
    ↪NoCompositionBreak"/>
  </Data>
  <Data property="InsulationBreak">
    <DataReference data="Plant/Enumerations.InsulationBreakClassification.InsulationBreak
    ↪"/>
  </Data>
  <Data property="NominalDiameterBreak">
    <DataReference data="Plant/Enumerations.NominalDiameterBreakClassification.
    ↪NoNominalDiameterBreak"/>
  </Data>
  <Data property="PipingClassBreak">
    <DataReference data="Plant/Enumerations.PipingClassBreakClassification.
    ↪PipingClassBreak"/>
  </Data>
</Object>
```

RestrictionOrifice

Plant::Piping::RestrictionOrifice

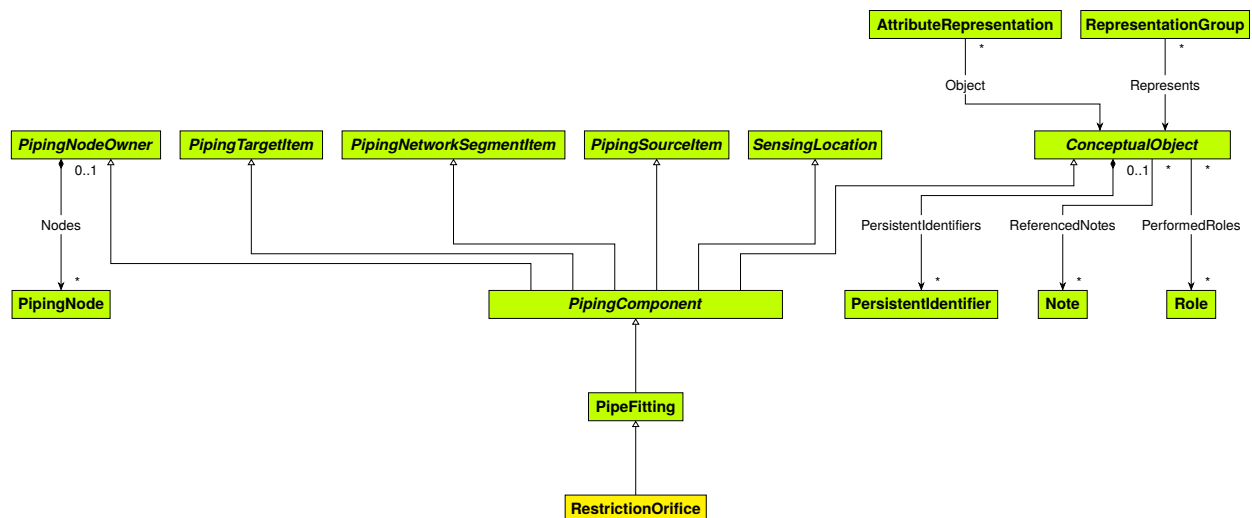
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/RestrictionOrifice>

RESTRICTION ORIFICE

A RESTRICTION ORIFICE is an ORIFICE PLATE that is intended for use as a restrictor.



Supertypes

- *PipeFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.RestrictionOrifice"/>
```

RuptureDisc

Plant::Piping::RuptureDisc

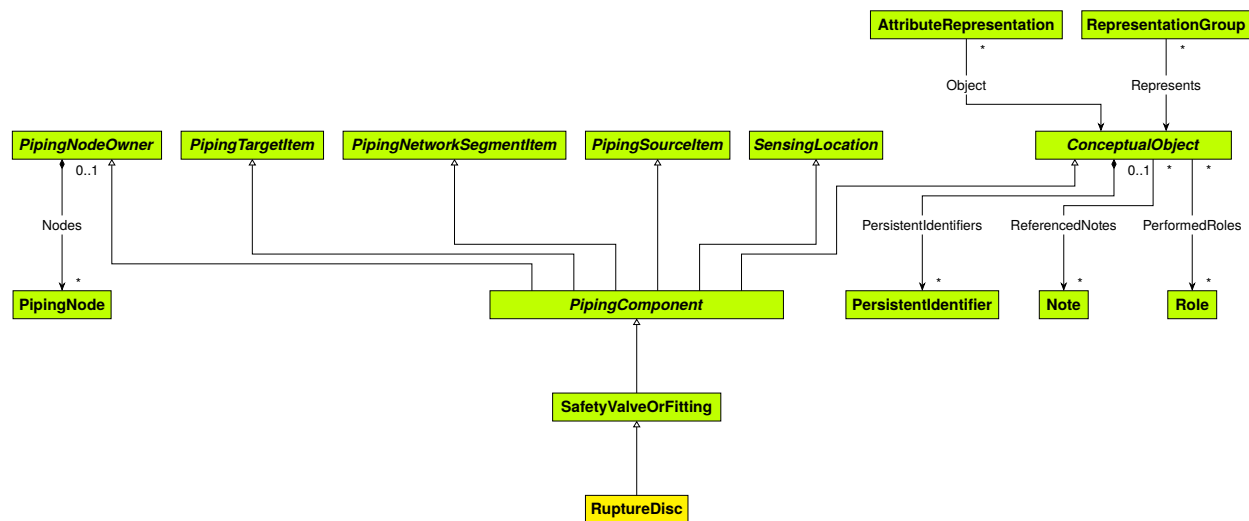
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS8372601>

RUPTURE DISC

A physical object that is designed to burst at a certain excess pressure. It is part of a rupture disc assembly (from <http://data.posccaesar.org/rdl/RDS8372601>).



Supertypes

- *SafetyValveOrFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.RuptureDisc"/>
```

SafetyValveOrFitting

Plant::Piping::SafetyValveOrFitting

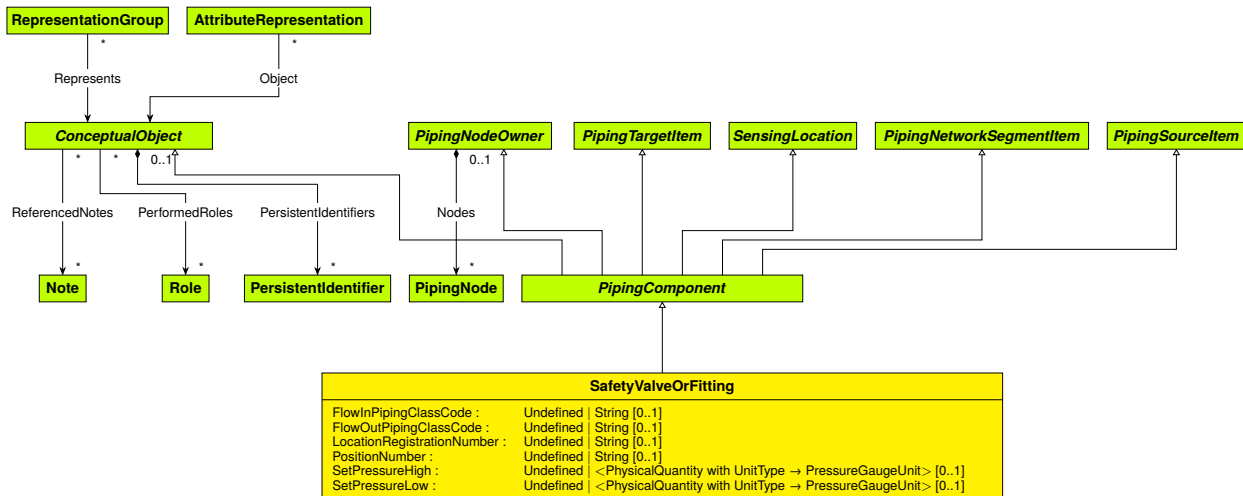
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/SafetyValveOrFitting>

SAFETY VALVE OR FITTING

A safety valve or fitting.



Supertypes

- *PipingComponent*

Subtypes

- *BreatherValve*
- *FlameArrestor*
- *RuptureDisc*
- *SpringLoadedAngleGlobeSafetyValve*
- *SpringLoadedGlobeSafetyValve*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>FlowInPipingClassCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>FlowOutPipingClassCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>LocationRegistrationNumber</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>PositionNumber</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>SetPressureHigh</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PressureGaugeUnit></i>	no	no
<i>SetPressureLow</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PressureGaugeUnit></i>	no	no

FlowInPipingClassCode

Data Property

The code of the piping class at the flow in side of *SafetyValveOrFitting*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/FlowInPipingClassCodeAssignmentClass>

FLOW IN PIPING CLASS CODE ASSIGNMENT CLASS

Example

"75HB13"

FlowOutPipingClassCode

Data Property

The code of the piping class at the flow out side of *SafetyValveOrFitting*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/FlowOutPipingClassCodeAssignmentClass>

FLOW OUT PIPING CLASS CODE ASSIGNMENT CLASS

Example

"75HB13"

LocationRegistrationNumber**Data Property**

The location registration number of the *SafetyValveOrFitting*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LocationRegistrationNumberAssignmentClass>

LOCATION REGISTRATION NUMBER ASSIGNMENT CLASS

Example

"L-N123"

PositionNumber**Data Property**

The position number of the *SafetyValveOrFitting*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PositionNumberAssignmentClass>

POSITION NUMBER ASSIGNMENT CLASS

Example

"SV 104.01"

SetPressureHigh**Data Property**

The high pressure at which the *SafetyValveOrFitting* is activated.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PressureGaugeUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SetPressureHigh>

SET PRESSURE HIGH

Example

```
PhysicalQuantity(
  Unit = Bar,
  Value = 30.0)
```

SetPressureLow**Data Property**

The low pressure at which the *SafetyValveOrFitting* is activated.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PressureGaugeUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SetPressureLow>

SET PRESSURE LOW

Example

```
PhysicalQuantity(
  Unit = Bar,
  Value = 0.0)
```

Example in DEXPI XML

```
<Object type="Plant/Piping.SafetyValveOrFitting">
  <Data property="FlowInPipingClassCode">
    <String>75HB13</String>
  </Data>
  <Data property="FlowOutPipingClassCode">
    <String>75HB13</String>
  </Data>
  <Data property="LocationRegistrationNumber">
    <String>L-N123</String>
  </Data>
  <Data property="PositionNumber">
    <String>SV 104.01</String>
  </Data>
  <Data property="SetPressureHigh">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PressureGaugeUnit.Bar"/>
      </Data>
      <Data property="Value">
        <Double>30.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
```

(continues on next page)

(continued from previous page)

```

<Data property="SetPressureLow">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.PressureGaugeUnit.Bar"/>
    </Data>
    <Data property="Value">
      <Double>0.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>

```

Sensorwell

Plant::Piping::Sensorwell

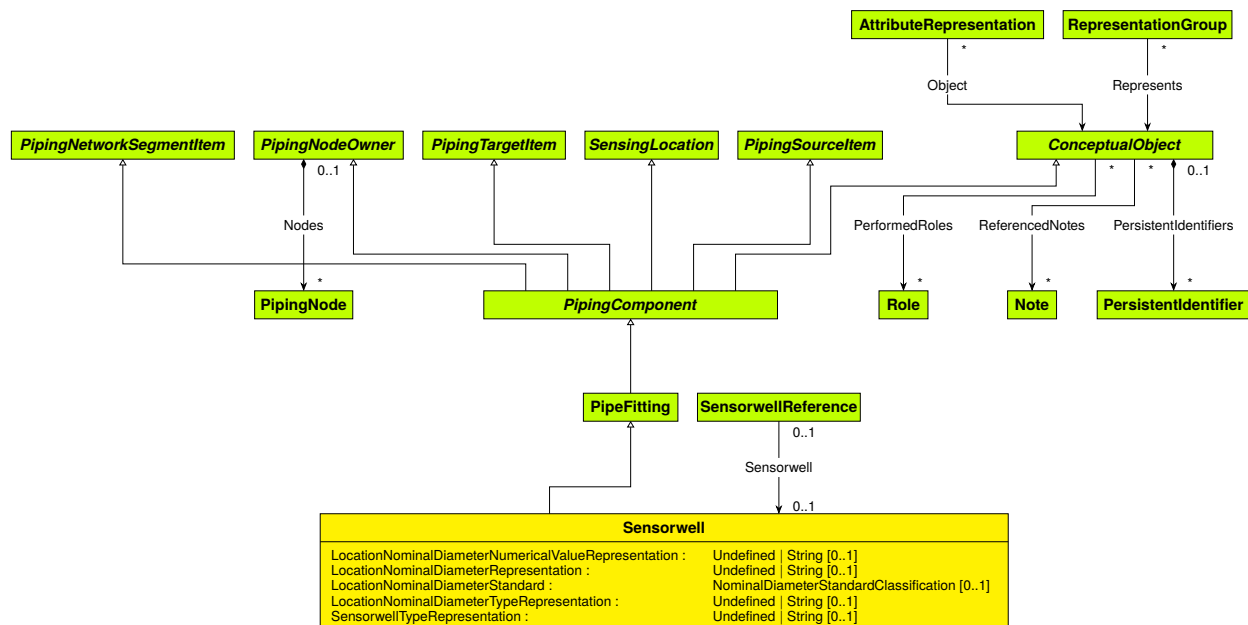
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/Sensorwell>

SENSORWELL

A Sensorwell is a pressure-tight receptacle adapted to receive a sensing element, provided with external threads or other means for pressure-tight attachment to a vessel or pipe.



Supertypes

- *PipeFitting*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>LocationNominalDiameterNumericalValueRepresentation</i>	0..1	<i>Undefined String</i>	no	no
<i>LocationNominalDiameterRepresentation</i>	0..1	<i>Undefined String</i>	no	no
<i>LocationNominalDiameterStandard</i>	0..1	<i>NominalDiameterStandardClassification</i>	no	no
<i>LocationNominalDiameterTypeRepresentation</i>	0..1	<i>Undefined String</i>	no	no
<i>SensorwellTypeRepresentation</i>	0..1	<i>Undefined String</i>	no	no

LocationNominalDiameterNumericalValueRepresentation

Data Property

A readable representation of the numerical value of the nominal diameter at the location of the *Sensorwell*. The purpose of this value is to give a textual representation of the nominal diameter to be used in the graphics of a PID.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LocationNominalDiameterNumericalValueRepresentationAssignmentClass>

LOCATION NOMINAL DIAMETER NUMERICAL VALUE REPRESENTATION ASSIGNMENT CLASS

Example

"25"

LocationNominalDiameterRepresentation

Data Property

A readable representation of the nominal diameter at the location of the *Sensorwell*. The purpose of this value is to give a textual representation of the nominal diameter to be used in the graphics of a PID.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LocationNominalDiameterRepresentationAssignmentClass>

LOCATION NOMINAL DIAMETER REPRESENTATION ASSIGNMENT CLASS

Example

"DN 25"

LocationNominalDiameterStandard

Data Property

The nominal diameter of the location of the *Sensorwell*, given as a reference to a nominal diameter standard and value.

Type: *NominalDiameterStandardClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LocationNominalDiameterStandardSpecialization>

LOCATION NOMINAL DIAMETER STANDARD SPECIALIZATION

Example

Din2448ObjectDn25

LocationNominalDiameterTypeRepresentation

Data Property

A readable representation of the type of the nominal diameter at the location of the *Sensorwell*. The purpose of this value is to give a textual representation of the nominal diameter to be used in the graphics of a PID.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LocationNominalDiameterTypeRepresentationAssignmentClass>

LOCATION NOMINAL DIAMETER TYPE REPRESENTATION ASSIGNMENT CLASS

Example

"DN"

SensorwellTypeRepresentation

Data Property

A textual representation of the type of the *Sensorwell*.

In principle, this attribute can have arbitrary string values to describe the type. However, for certain types, the exact string is prescribed:

- If the sensorwell is a thermowell, the type representation shall be *Thermowell*.
- If the sensorwell is a levelwell, the type representation shall be *Levelwell*.
- If the sensorwell is an analysiswell, the type representation shall be *Analysiswell*.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SensorwellTypeRepresentationAssignmentClass>

SENSORWELL TYPE REPRESENTATION ASSIGNMENT CLASS

Example

"Thermowell"

Example in DEXPI XML

```
<Object type="Plant/Piping.Sensorwell">
  <Data property="LocationNominalDiameterNumericalValueRepresentation">
    <String>25</String>
  </Data>
  <Data property="LocationNominalDiameterRepresentation">
    <String>DN 25</String>
  </Data>
  <Data property="LocationNominalDiameterStandard">
    <DataReference data="Plant/Enumerations.NominalDiameterStandardClassification.
    ↪Din24480objectDn25"/>
  </Data>
  <Data property="LocationNominalDiameterTypeRepresentation">
    <String>DN</String>
  </Data>
  <Data property="SensorwellTypeRepresentation">
    <String>Thermowell</String>
  </Data>
</Object>
```

SightGlass

Plant::Piping::SightGlass

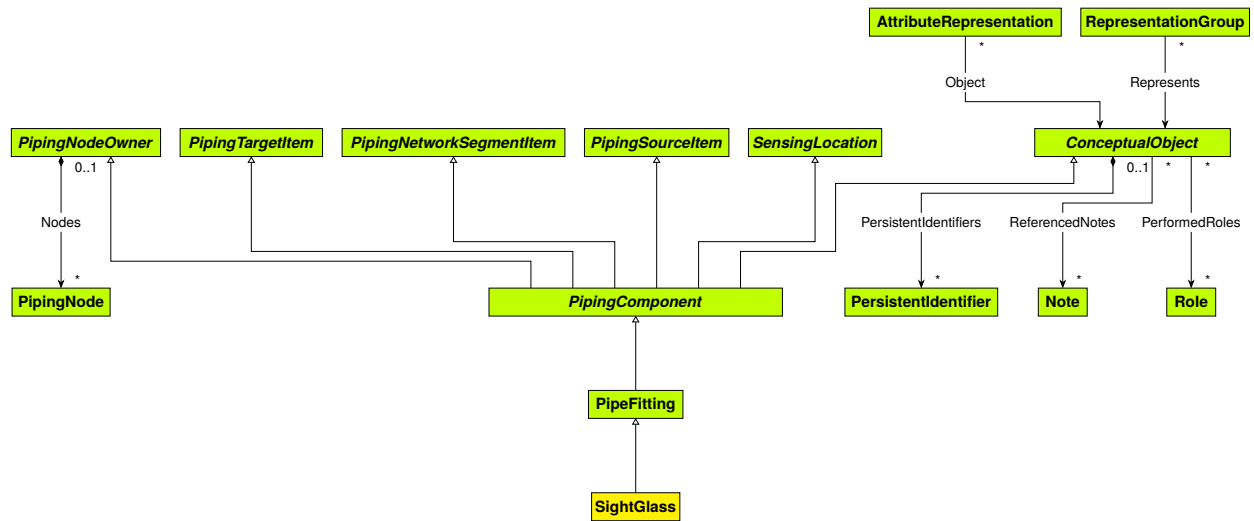
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS648674>

SIGHT GLASS

A physical object that is transparent and intended for viewing a vessel or piping system interior (from <http://data.posccaesar.org/rdl/RDS648674>).



Supertypes

- *PipeFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.SightGlass"/>
```

Silencer

Plant::Piping::Silencer

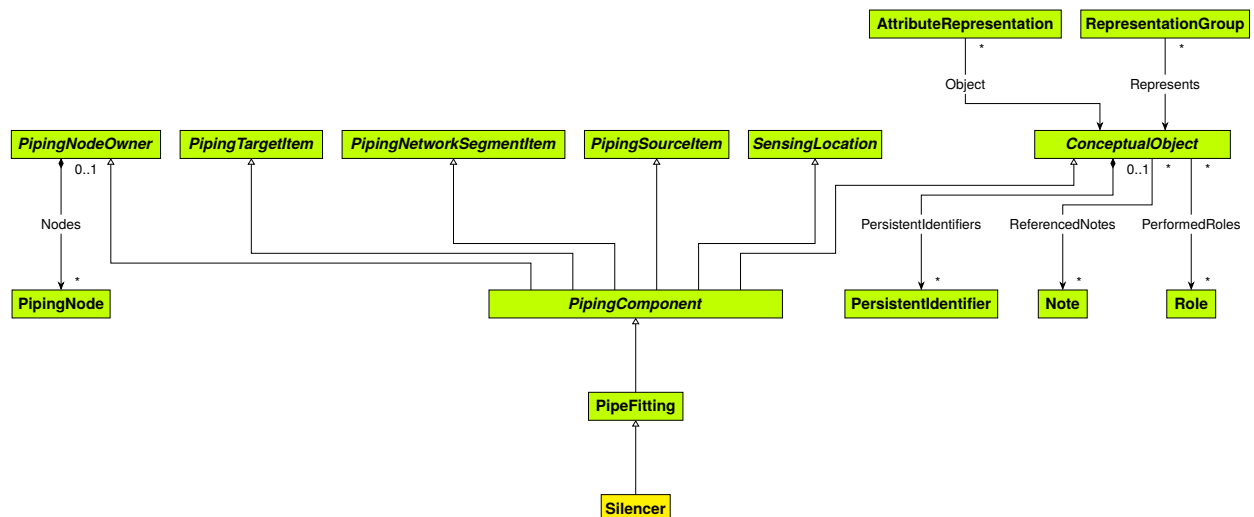
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS1049368591>

SILENCER

A device intended to reduce a noise level (from <http://data.posccaesar.org/rdl/RDS1049368591>).



Supertypes

- *PipeFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.Silencer"/>
```

SpringLoadedAngleGlobeSafetyValve

Plant::Piping::SpringLoadedAngleGlobeSafetyValve

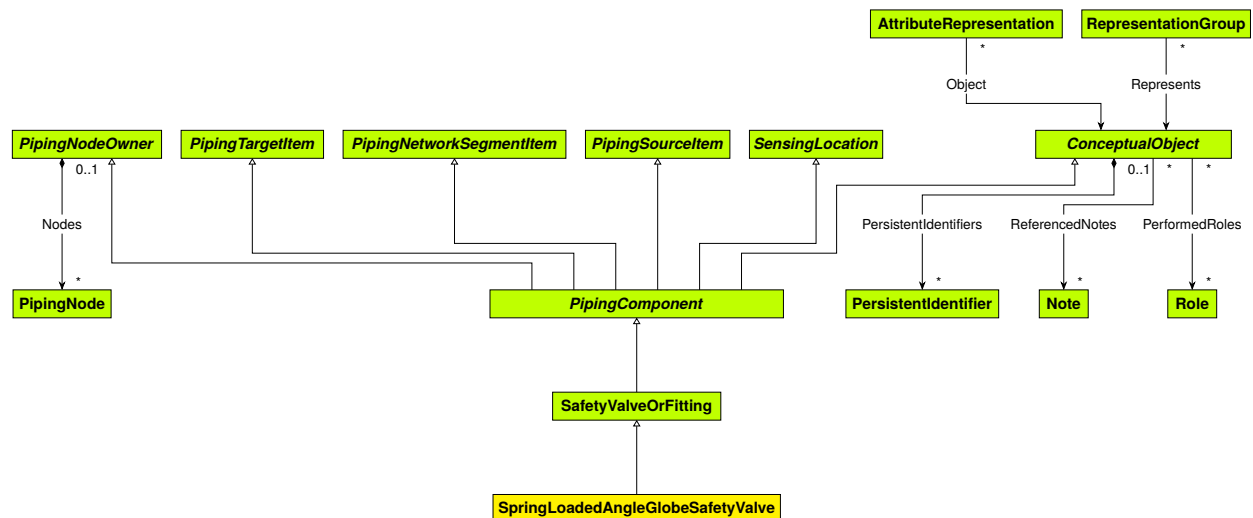
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/SpringLoadedAngleGlobeSafetyValve>

SPRING LOADED ANGLE GLOBE SAFETY VALVE

A spring-loaded angle globe safety valve.



Supertypes

- *SafetyValveOrFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.SpringLoadedAngleGlobeSafetyValve"/>
```

SpringLoadedGlobeSafetyValve

Plant::Piping::SpringLoadedGlobeSafetyValve

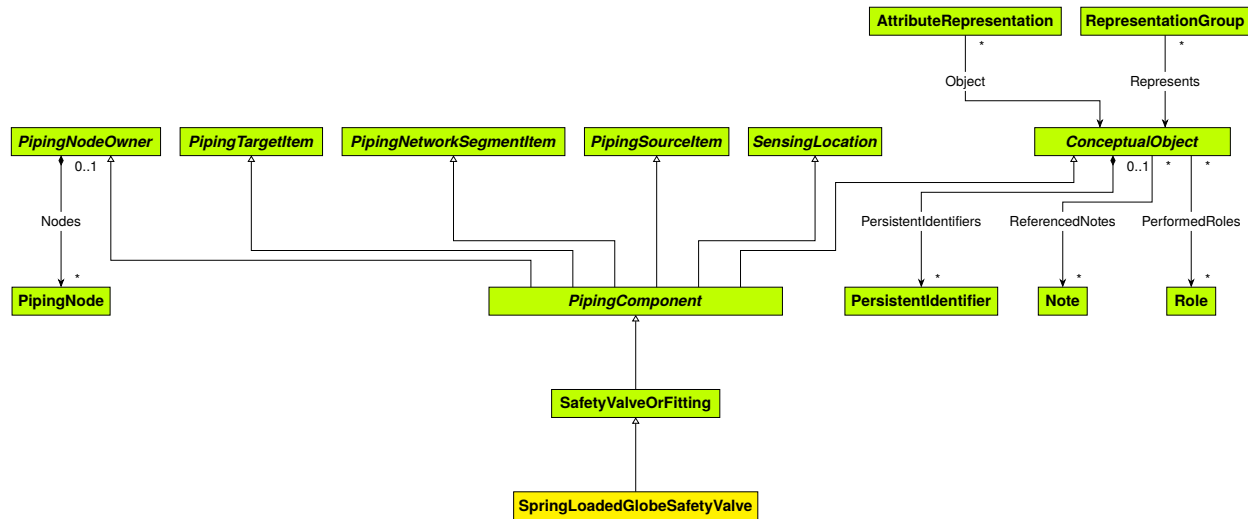
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/SpringLoadedGlobeSafetyValve>

SPRING LOADED GLOBE SAFETY VALVE

A spring-loaded globe safety valve.



Supertypes

- *SafetyValveOrFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.SpringLoadedGlobeSafetyValve"/>
```

SteamTrap

Plant::Piping::SteamTrap

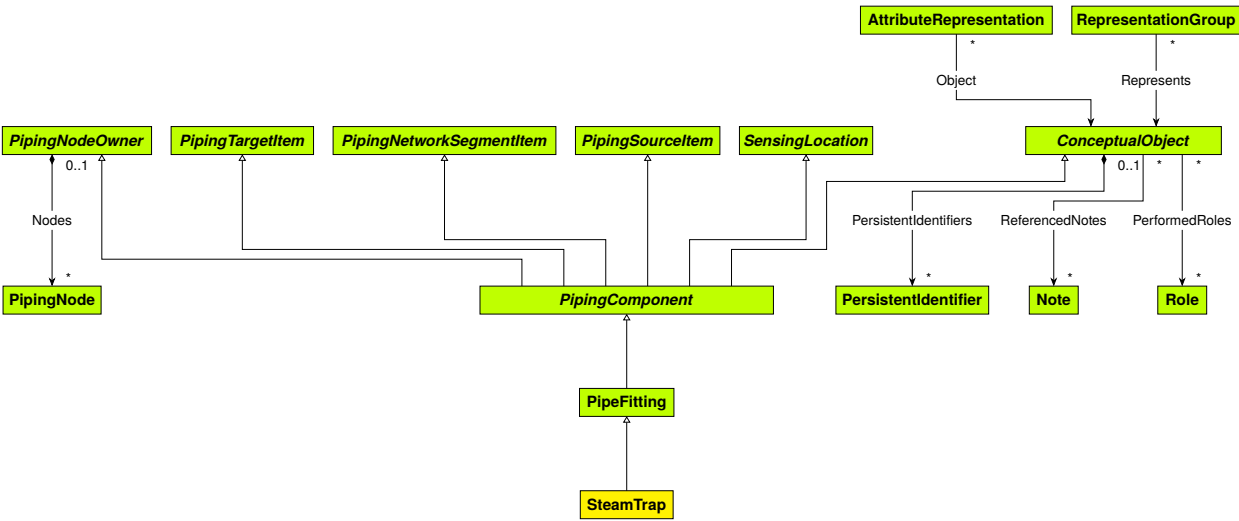
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS5782388>

STEAM TRAP

A trap that consists of a chamber into which condensed steam from steam pipes etc. is allowed to drain, and which automatically ejects it without permitting the escape of steam (from <http://data.posccaesar.org/rdl/RDS5782388>).



Supertypes

- *PipeFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.SteamTrap"/>
```

StraightwayValve

Plant::Piping::StraightwayValve

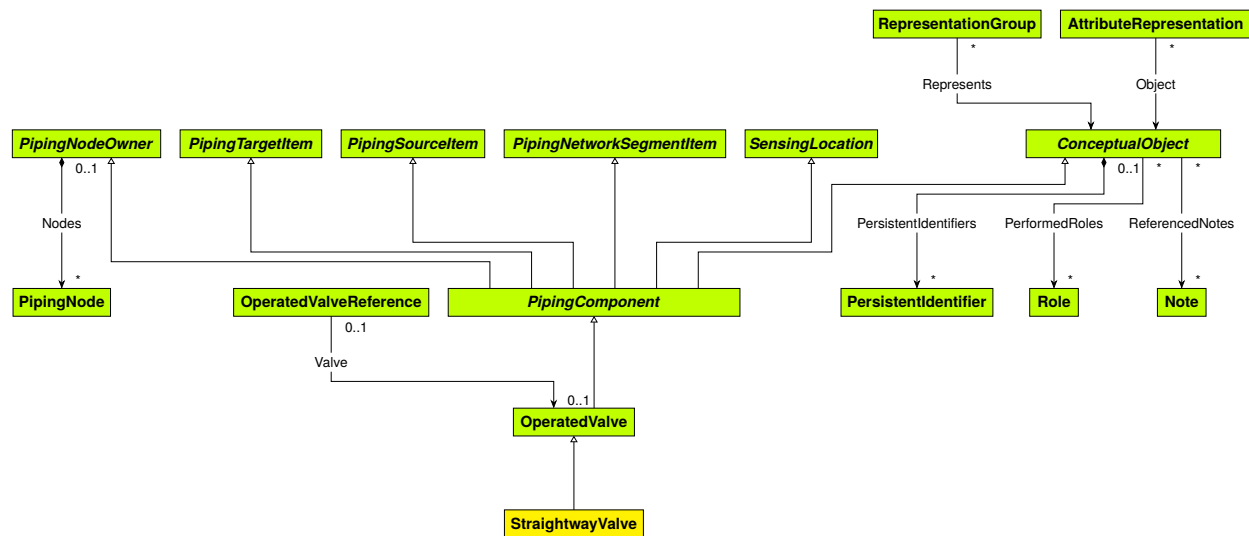
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS9390905>

STRAIGHTWAY VALVE

A valve that is straight, i.e. the centerlines perpendicular to the ends are in-line with no offset (from <http://data.posccaesar.org/rdl/RDS9390905>).



Supertypes

- *OperatedValve*

Example in DEXPI XML

```
<Object type="Plant/Piping.StraightwayValve"/>
```

Strainer

Plant::Piping::Strainer

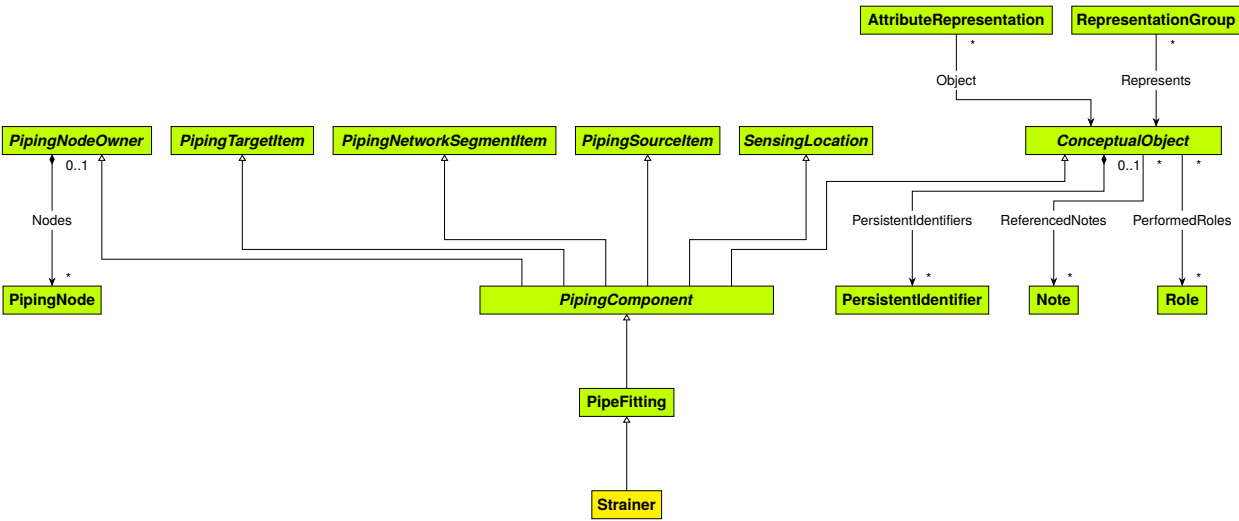
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS422504>

STRAINER

A mechanical separator that is separating solid particles from a fluid by passing the fluid through a wire mesh, screen or metal plates containing perforations or slits (from <http://data.posccaesar.org/rdl/RDS422504>).



Supertypes

- *PipeFitting*

Example in DEXPI XML

```
<Object type="Plant/Piping.Strainer"/>
```

SwingCheckValve

Plant::Piping::SwingCheckValve

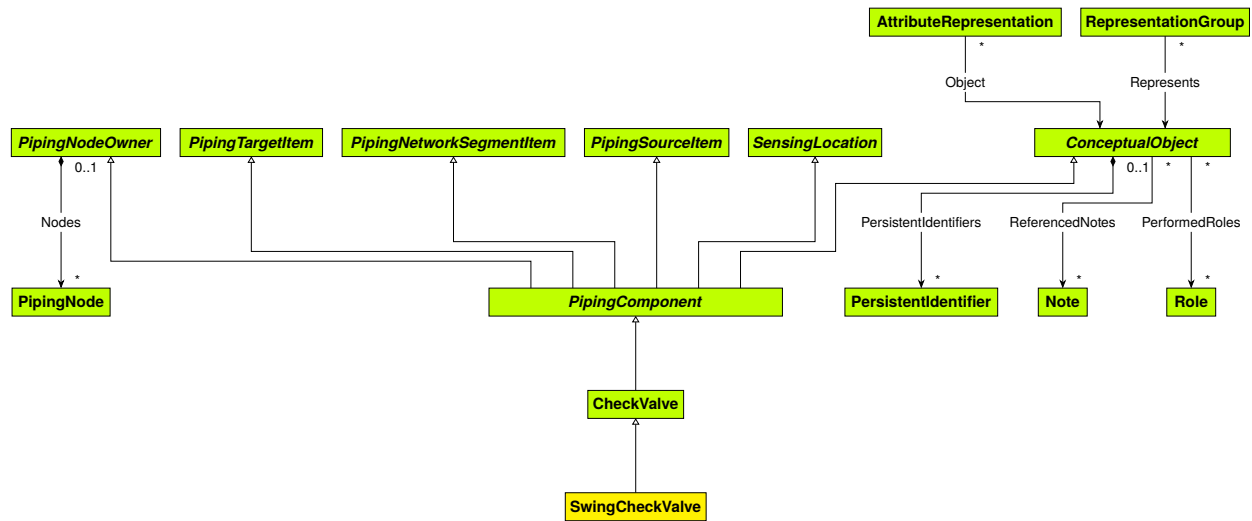
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS610424>

SWING CHECK VALVE

A check valve that is a check valve where the closure member is a disc which swings freely on a hinge and which opens automatically when flow is established and closes automatically when flow ceases or is reversed (from <http://data.posccaesar.org/rdl/RDS610424>).



Supertypes

- *CheckValve*

Example in DEXPI XML

```
<Object type="Plant/Piping.SwingCheckValve"/>
```

TurbineFlowMeter

Plant::Piping::TurbineFlowMeter

Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS417914>

TURBINE FLOW METER

A velocity flow meter that uses a multi bladed rotor to measure fluid flow rate in units of volumetric flow through a closed conduit (from <http://data.posccaesar.org/rdl/RDS417914>).



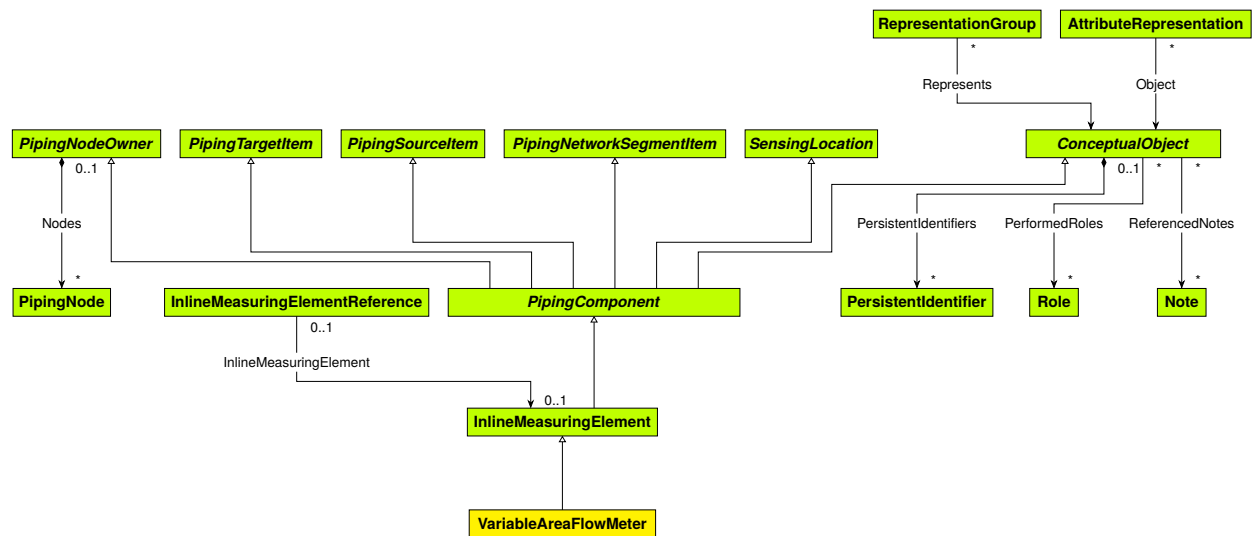
- ### Example in DEXPI XML

VariableAreaFlowMeter

Overview

RDL: <http://data.posccaesar.org/rdl/RDS418229>

A flow meter consisting of a vertical tube with a conically shaped bore which widens to the top in which a solid body (float) is supported by the force exerted by the fluid stream (from <http://data.posccaesar.org/rd/1/RDS418229>).



Supertypes

- *InlineMeasuringElement*

Example in DEXPI XML

```
<Object type="Plant/Piping.VariableAreaFlowMeter"/>
```

VentLine

Plant::Piping::VentLine

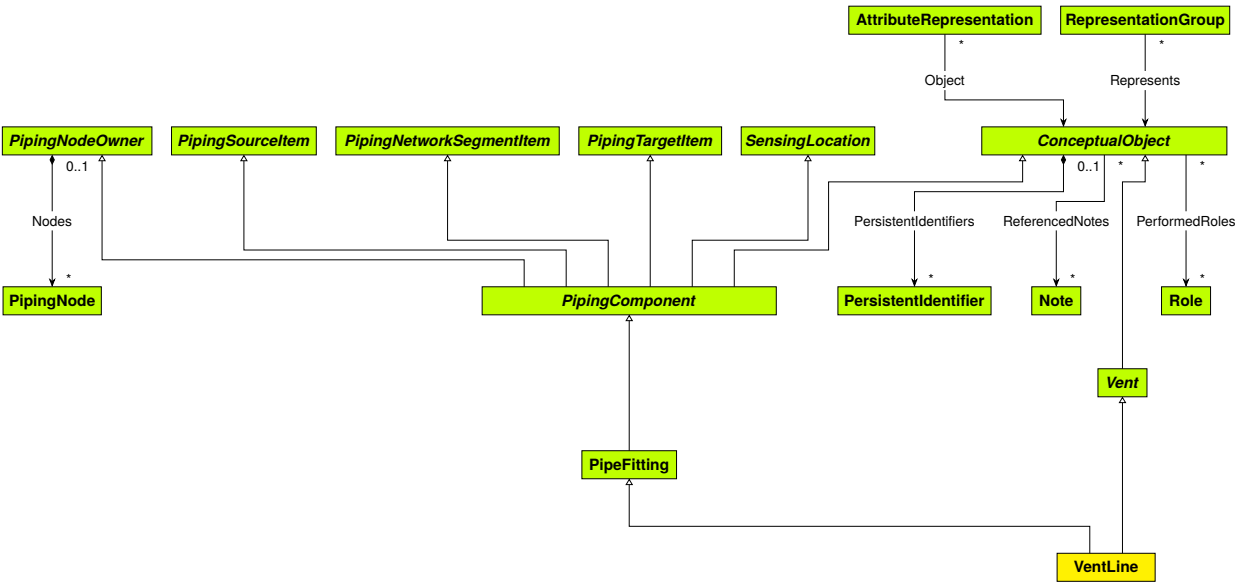
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS905841131>

VENT LINE

A *Vent* that is intended for transport of vented gas from a fluid container, e.g. a vessel or a pipeline, to a vent stack or to a safe area.



Supertypes

- *PipeFitting*
- *Vent*

Example in DEXPI XML

```
<Object type="Plant/Piping.VentLine"/>
```

VenturiTube

Plant::Piping::VenturiTube

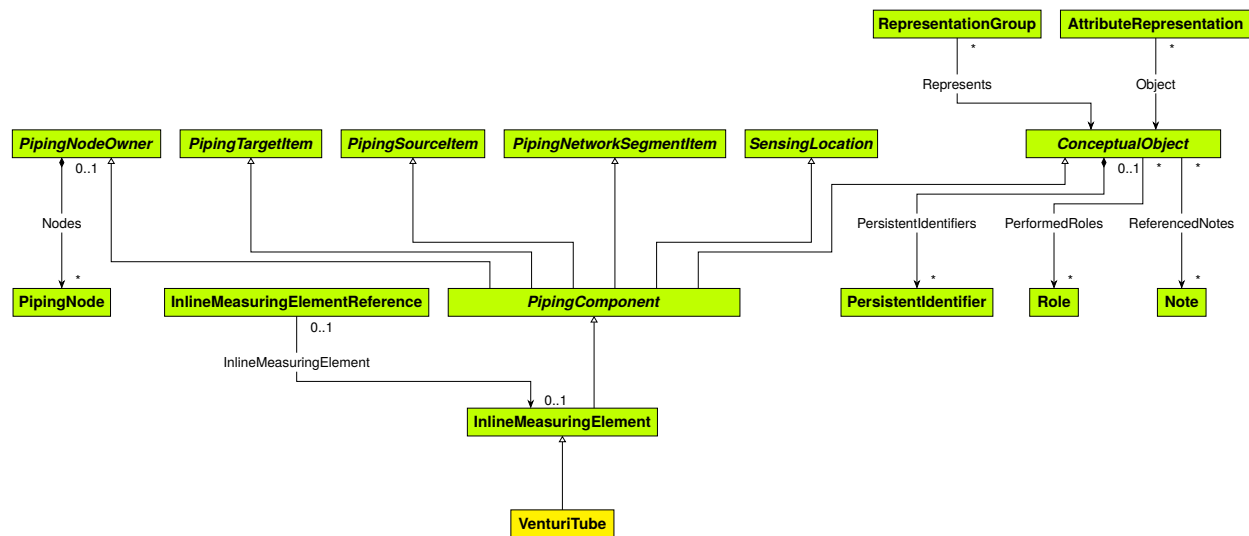
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS648044>

VENTURI TUBE

A ‘measuring device’ that has a constriction with a relative long passage with a smooth coned entry and exit (from <http://data.posccaesar.org/rdl/RDS648044>).



Supertypes

- *InlineMeasuringElement*

Example in DEXPI XML

```
<Object type="Plant/Piping.VenturiTube"/>
```

VolumeFlowMeasuringElement

Plant::Piping::VolumeFlowMeasuringElement

Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/VolumeFlowMeasuringElement>

VOLUME FLOW MEASURING ELEMENT

A VOLUME FLOW MEASURING ELEMENT is a FLOW MEASURING ELEMENT that is used to measure VOLUME FLOW RATE.



- ### Example in DEXPI XML

Plant::Piping

Package

Classes

continues on next page

Table 8 – continued from previous page

Name	Type
<i>FlangedConnection</i>	Concrete Class
<i>FlowInPipeOffPageConnector</i>	Concrete Class
<i>FlowMeasuringElement</i>	Concrete Class
<i>FlowNozzle</i>	Concrete Class
<i>FlowOutPipeOffPageConnector</i>	Concrete Class
<i>Funnel</i>	Concrete Class
<i>GateValve</i>	Concrete Class
<i>GlobeCheckValve</i>	Concrete Class
<i>GlobeValve</i>	Concrete Class
<i>Hose</i>	Concrete Class
<i>IlluminatedSightGlass</i>	Concrete Class
<i>InLineMixer</i>	Concrete Class
<i>InlineMeasuringElement</i>	Concrete Class
<i>LineBlind</i>	Concrete Class
<i>MassFlowMeasuringElement</i>	Concrete Class
<i>NeedleValve</i>	Concrete Class
<i>OperatedValve</i>	Concrete Class
<i>Penetration</i>	Concrete Class
<i>Pipe</i>	Concrete Class
<i>PipeCoupling</i>	Concrete Class
<i>PipeFitting</i>	Concrete Class
<i>PipeFlangeSpacer</i>	Concrete Class
<i>PipeFlangeSpade</i>	Concrete Class
<i>PipeOffPageConnector</i>	Abstract Class
<i>PipeOffPageConnectorObjectReference</i>	Concrete Class
<i>PipeOffPageConnectorReference</i>	Abstract Class
<i>PipeOffPageConnectorReferenceByNumber</i>	Concrete Class
<i>PipeReducer</i>	Concrete Class
<i>PipeTee</i>	Concrete Class
<i>PipingComponent</i>	Abstract Class
<i>PipingConnection</i>	Abstract Class
<i>PipingNetworkSegment</i>	Concrete Class
<i>PipingNetworkSegmentItem</i>	Abstract Class
<i>PipingNetworkSystem</i>	Concrete Class
<i>PipingNode</i>	Concrete Class
<i>PipingNodeOwner</i>	Abstract Class
<i>PipingSourceItem</i>	Abstract Class
<i>PipingTargetItem</i>	Abstract Class
<i>PlugValve</i>	Concrete Class
<i>PositiveDisplacementFlowMeter</i>	Concrete Class
<i>PropertyBreak</i>	Concrete Class
<i>RestrictionOrifice</i>	Concrete Class
<i>RuptureDisc</i>	Concrete Class
<i>SafetyValveOrFitting</i>	Concrete Class
<i>Sensorwell</i>	Concrete Class
<i>SightGlass</i>	Concrete Class
<i>Silencer</i>	Concrete Class
<i>SpringLoadedAngleGlobeSafetyValve</i>	Concrete Class
<i>SpringLoadedGlobeSafetyValve</i>	Concrete Class
<i>SteamTrap</i>	Concrete Class

continues on next page

Table 8 – continued from previous page

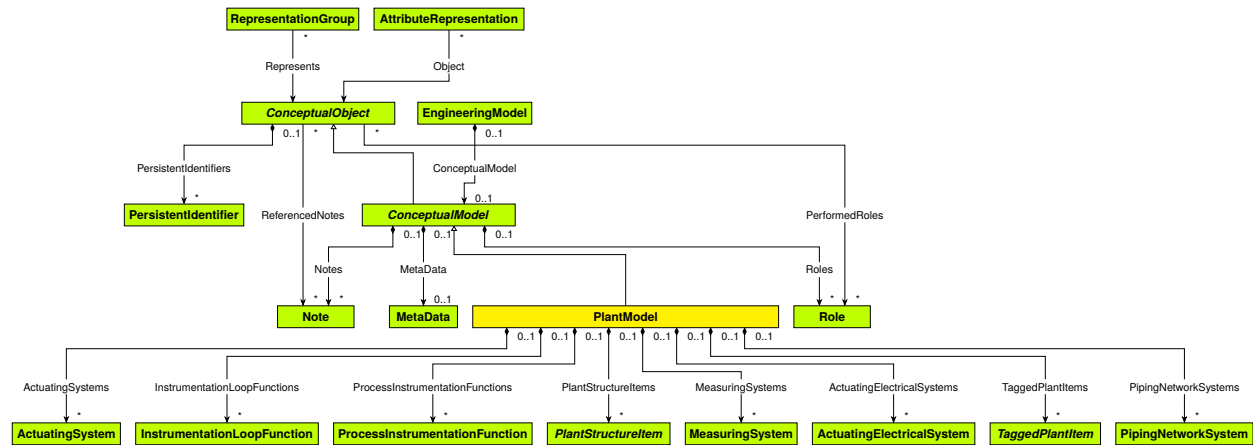
Name	Type
<i>StraightwayValve</i>	Concrete Class
<i>Strainer</i>	Concrete Class
<i>SwingCheckValve</i>	Concrete Class
<i>TurbineFlowMeter</i>	Concrete Class
<i>VariableAreaFlowMeter</i>	Concrete Class
<i>VentLine</i>	Concrete Class
<i>VenturiTube</i>	Concrete Class
<i>VolumeFlowMeasuringElement</i>	Concrete Class

3.2.5 PlantModel

Plant::PlantModel

Overview

Concrete Class



Supertypes

- *ConceptualModel*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>ActuatingElectricalSystems</i>	*	<i>ActuatingElectricalSystem</i>	no
<i>ActuatingSystems</i>	*	<i>ActuatingSystem</i>	no
<i>InstrumentationLoopFunctions</i>	*	<i>InstrumentationLoopFunction</i>	no
<i>MeasuringSystems</i>	*	<i>MeasuringSystem</i>	no
<i>PipingNetworkSystems</i>	*	<i>PipingNetworkSystem</i>	no
<i>PlantStructureItems</i>	*	<i>PlantStructureItem</i>	no
<i>ProcessInstrumentationFunctions</i>	*	<i>ProcessInstrumentationFunction</i>	no
<i>TaggedPlantItems</i>	*	<i>TaggedPlantItem</i>	no

ActuatingElectricalSystems

Composition Property

The *ActuatingElectricalSystems* of the *PlantModel*.

Type: *ActuatingElectricalSystem*

Multiplicity: *

Ordered: no

ActuatingSystems

Composition Property

The *ActuatingSystems* of the *PlantModel*.

Type: *ActuatingSystem*

Multiplicity: *

Ordered: no

InstrumentationLoopFunctions

Composition Property

The *InstrumentationLoopFunctions* of the *PlantModel*.

Type: *InstrumentationLoopFunction*

Multiplicity: *

Ordered: no

MeasuringSystems

Composition Property

The *MeasuringSystems* of the *PlantModel*.

Type: *MeasuringSystem*

Multiplicity: *

Ordered: no

PipingNetworkSystems

Composition Property

The *PipingNetworkSystems* of the *PlantModel*.

Type: *PipingNetworkSystem*

Multiplicity: *

Ordered: no

PlantStructureItems

Composition Property

The *PlantStructureItems* of the *PlantModel*.

Type: *PlantStructureItem*

Multiplicity: *

Ordered: no

ProcessInstrumentationFunctions

Composition Property

The *ProcessInstrumentationFunctions* of the *PlantModel*.

Type: *ProcessInstrumentationFunction*

Multiplicity: *

Ordered: no

TaggedPlantItems

Composition Property

The *TaggedPlantItems* of the *PlantModel*.

Type: *TaggedPlantItem*

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/PlantModel"/>
```

3.2.6 PlantStructure

Enterprise

Plant::PlantStructure::Enterprise

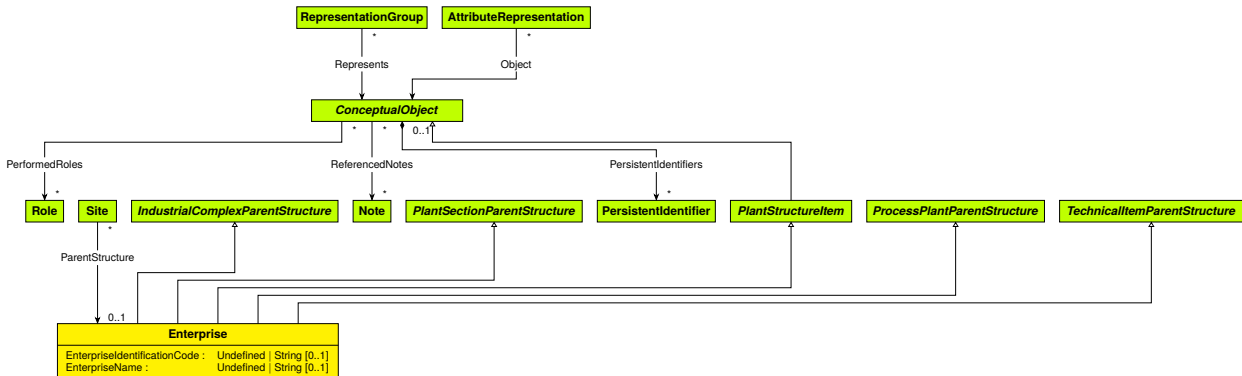
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS10418236543>

ISA95 ENTERPRISE

An enterprise as defined by ISA 95.



Supertypes

- *IndustrialComplexParentStructure*
- *PlantSectionParentStructure*
- *PlantStructureItem*
- *ProcessPlantParentStructure*
- *TechnicalItemParentStructure*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>EnterpriseIdentificationCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>EnterpriseName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

EnterpriseIdentificationCode

Data Property

The identification code of the enterprise.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/EnterpriseIdentificationCodeAssignmentClass>

ENTERPRISE IDENTIFICATION CODE ASSIGNMENT CLASS

Example

"C1248"

EnterpriseName

Data Property

The name of the enterprise.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/EnterpriseNameAssignmentClass>

ENTERPRISE NAME ASSIGNMENT CLASS

Example

"CompAny Ltd."

Example in DEXPI XML

```
<Object type="Plant/PlantStructure.Enterprise">
  <Data property="EnterpriseIdentificationCode">
    <String>C1248</String>
  </Data>
  <Data property="EnterpriseName">
    <String>CompAny Ltd.</String>
  </Data>
</Object>
```

IndustrialComplex

Plant::PlantStructure::IndustrialComplex

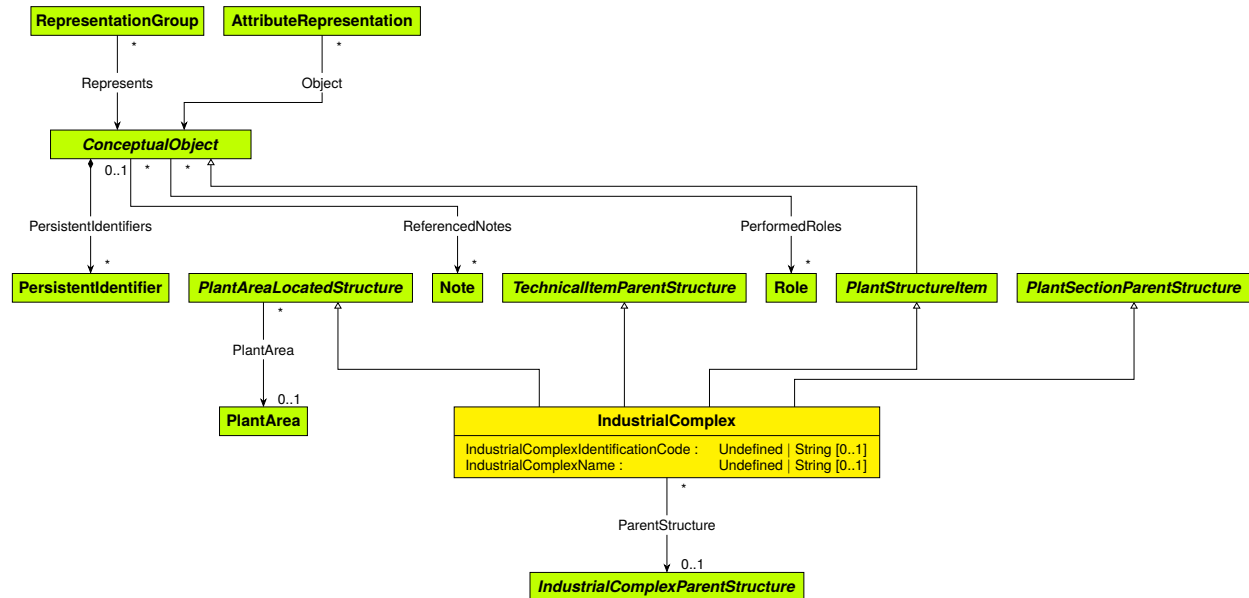
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/IndustrialComplexIso102092012>

INDUSTRIAL COMPLEX ISO10209 2012

An industrial complex as defined by ISO 10209:2012.



Supertypes

- *PlantAreaLocatedStructure*
- *PlantSectionParentStructure*
- *PlantStructureItem*
- *TechnicalItemParentStructure*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite	Multiplicity
<i>ParentStructure</i>	0..1	<i>IndustrialComplexParentStructure</i>	no	no	*	

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>IndustrialComplexIdentificationCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>IndustrialComplexName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

IndustrialComplexIdentificationCode

Data Property

The identification code of the industrial complex.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/IndustrialComplexIdentificationCodeAssignmentClass>

INDUSTRIAL COMPLEX IDENTIFICATION CODE ASSIGNMENT CLASS

Example

"I-Chain"

IndustrialComplexName

Data Property

The name of the industrial complex.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/IndustrialComplexNameAssignmentClass>

INDUSTRIAL COMPLEX NAME ASSIGNMENT CLASS

Example

"Isophorone Chain"

ParentStructure**Reference Property**

A superordinate structure of which the IndustrialComplex is a part.

Type: *IndustrialComplexParentStructure*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

Example in DEXPI XML

```
<Object type="Plant/PlantStructure.IndustrialComplex">
  <Data property="IndustrialComplexIdentificationCode">
    <String>I-Chain</String>
  </Data>
  <Data property="IndustrialComplexName">
    <String>Isophorone Chain</String>
  </Data>
</Object>
```

IndustrialComplexParentStructure

Plant::PlantStructure::IndustrialComplexParentStructure

Overview**Abstract Class**

A *PlantStructureItem* that is a suitable *ParentStructure* of an *IndustrialComplex*.

IndustrialComplexParentStructure

Subtypes

- *Enterprise*
- *Site*

PlantArea

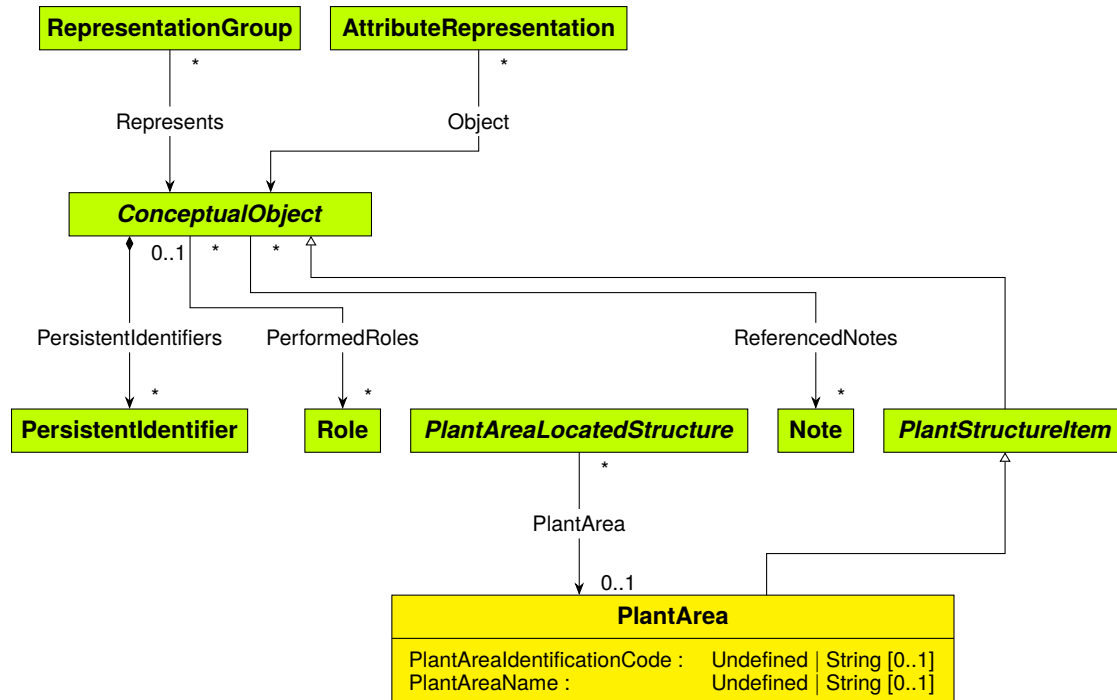
Plant::PlantStructure::PlantArea

Overview**Concrete Class**

RDL: <http://data.posccaesar.org/rdl/RDS10418236534>

AREA ISA95

An area as defined by ISA 95.



Supertypes

- *PlantStructureItem*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>PlantAreaIdentificationCode</i>	0..1	<i>Undefined String</i>	no	no
<i>PlantAreaName</i>	0..1	<i>Undefined String</i>	no	no

PlantAreaIdentificationCode

Data Property

The identification code of the plant area according to ISA-95.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PlantAreaIdentificationCodeAssignmentClass>

PLANT AREA IDENTIFICATION CODE ASSIGNMENT CLASS

Example

"F4"

PlantAreaName

Data Property

The name of the plant area according to ISA-95.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/AreaIsa95NameAssignmentClass>

AREA ISA95 NAME ASSIGNMENT CLASS

Example

"Area F4"

Example in DEXPI XML

```
<Object type="Plant/PlantStructure.PlantArea">
  <Data property="PlantAreaIdentificationCode">
    <String>F4</String>
  </Data>
  <Data property="PlantAreaName">
    <String>Area F4</String>
  </Data>
</Object>
```

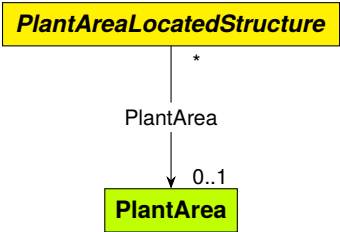
PlantAreaLocatedStructure

Plant::PlantStructure::PlantAreaLocatedStructure

Overview

Abstract Class

A structure that can be located in a *PlantArea*.



Subtypes

- *IndustrialComplex*
- *PlantSection*
- *ProcessPlant*
- *TechnicalItem*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>PlantArea</i>	0..1	<i>PlantArea</i>	no	no	*

PlantArea

Reference Property

The *PlantArea* in which the PlantAreaLocatedStructure is located.

Type: *PlantArea*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

PlantSection

Plant::PlantStructure::PlantSection

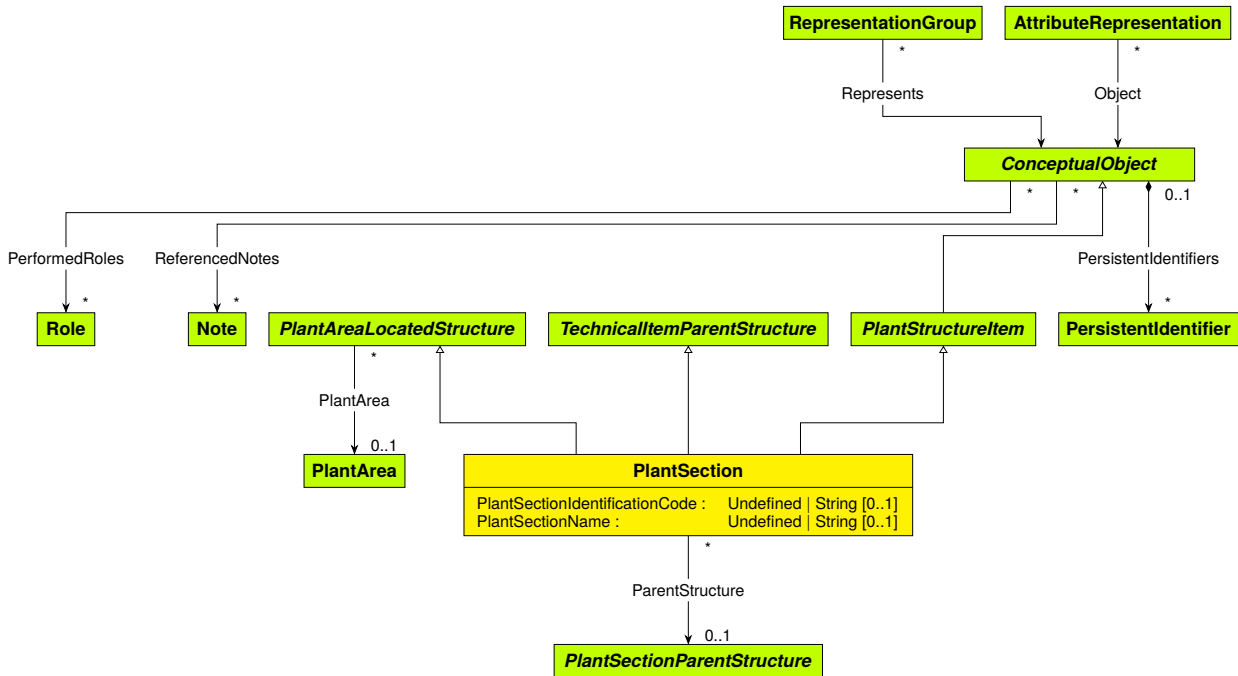
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/PlantSectionIso102092012>

PLANT SECTION ISO10209 2012

A plant section as defined by ISO 10209:2012.



Supertypes

- *PlantAreaLocatedStructure*
- *PlantStructureItem*
- *TechnicalItemParentStructure*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>ParentStructure</i>	0..1	<i>PlantSectionParentStructure</i>	no	no	*

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>PlantSectionIdentificationCode</i>	0..1	<i>Undefined String</i>	no	no
<i>PlantSectionName</i>	0..1	<i>Undefined String</i>	no	no

ParentStructure

Reference Property

A superordinate structure of which the PlantSection is a part.

Type: *PlantSectionParentStructure*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

PlantSectionIdentificationCode

Data Property

The identification code of the plant section.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PlantSectionIdentificationCodeAssignmentClass>

PLANT SECTION IDENTIFICATION CODE ASSIGNMENT CLASS

Example

"10"

PlantSectionName

Data Property

The name of the plant section.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PlantSectionNameAssignmentClass>

PLANT SECTION NAME ASSIGNMENT CLASS

Example

"Utilities"

Example in DEXPI XML

```
<Object type="Plant/PlantStructure.PlantSection">
  <Data property="PlantSectionIdentificationCode">
    <String>10</String>
  </Data>
  <Data property="PlantSectionName">
    <String>Utilities</String>
  </Data>
</Object>
```

PlantSectionParentStructure

Plant::PlantStructure::PlantSectionParentStructure

Overview

Abstract Class

A *PlantStructureItem* that is a suitable *ParentStructure* of a *PlantSection*.

PlantSectionParentStructure

Subtypes

- *Enterprise*
- *IndustrialComplex*
- *ProcessPlant*
- *Site*

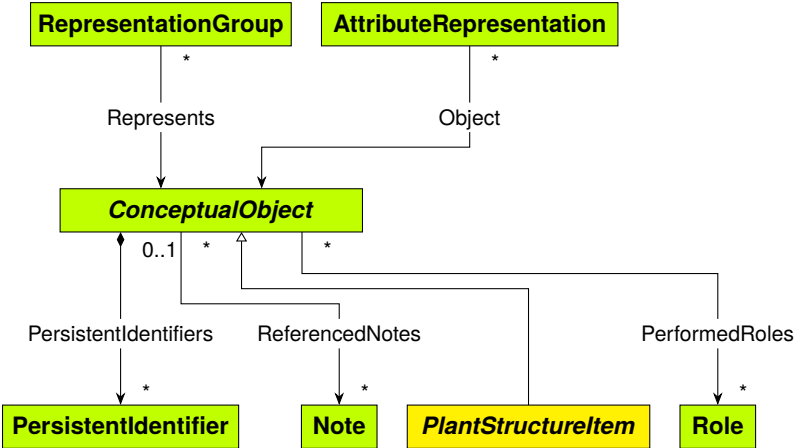
PlantStructureItem

Plant::PlantStructure::PlantStructureItem

Overview

Abstract Class

Item of the plant break down structure.



Supertypes

- *ConceptualObject*

Subtypes

- *Enterprise*
- *IndustrialComplex*
- *PlantArea*
- *PlantSection*

- *PlantSystem*
- *PlantTrain*
- *ProcessPlant*
- *Site*

PlantSystem

Plant::PlantStructure::PlantSystem

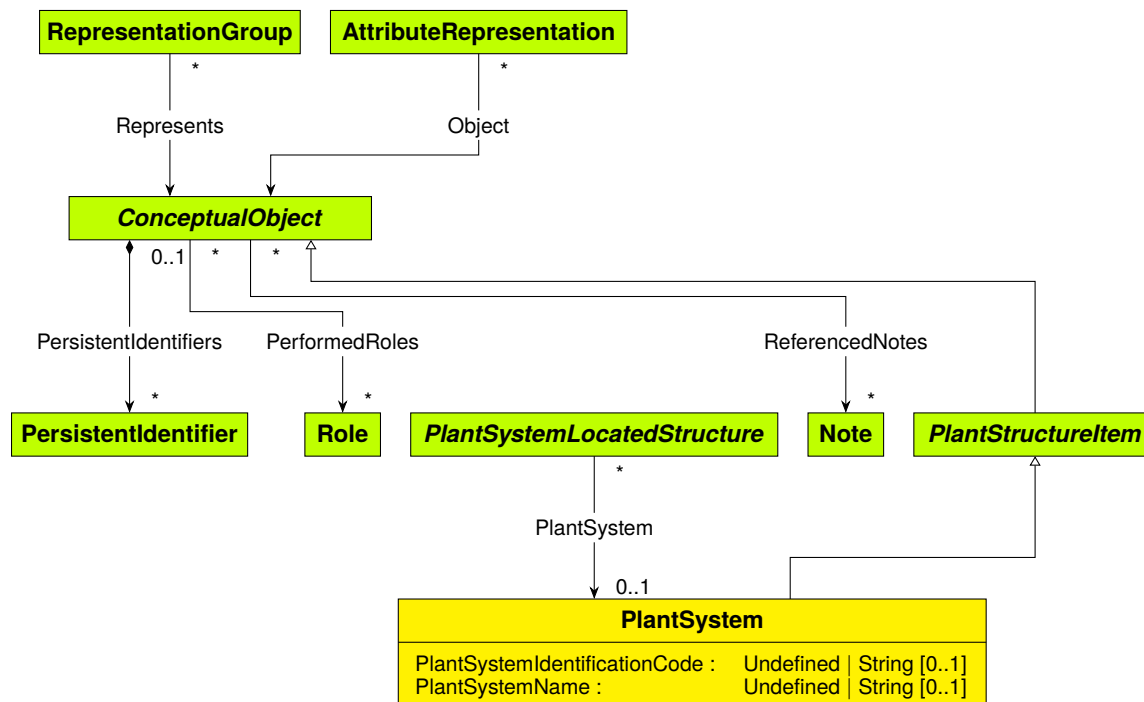
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/PlantSystem>

PLANT SYSTEM

A plant system.



Supertypes

- *PlantStructureItem*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>PlantSystemIdentificationCode</i>	0..1	<i>Undefined String</i>	no	no
<i>PlantSystemName</i>	0..1	<i>Undefined String</i>	no	no

PlantSystemIdentificationCode

Data Property

The identification code of the plant system.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PlantSystemIdentificationCodeAssignmentClass>

PLANT SYSTEM IDENTIFICATION CODE ASSIGNMENT CLASS

Example

"X123"

PlantSystemName

Data Property

The name of the plant system.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PlantSystemNameAssignmentClass>

PLANT SYSTEM NAME ASSIGNMENT CLASS

Example

"System X123"

Example in DEXPI XML

```

<Object type="Plant/PlantStructure.PlantSystem">
  <Data property="PlantSystemIdentificationCode">
    <String>X123</String>
  </Data>
  <Data property="PlantSystemName">
    <String>System X123</String>
  </Data>
</Object>

```

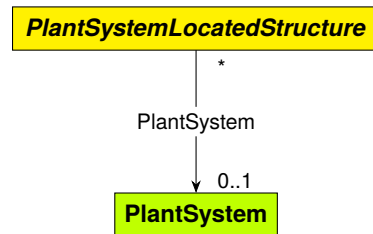
PlantSystemLocatedStructure

Plant::PlantStructure::PlantSystemLocatedStructure

Overview

Abstract Class

A structure that can be located in a *PlantSystem*.



Subtypes

- *TechnicalItem*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>PlantSystem</i>	0..1	<i>PlantSystem</i>	no	no	*

PlantSystem

Reference Property

The *PlantSystem* in which the PlantSystemLocatedStructure is located.

Type: *PlantSystem*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

PlantTrain

Plant::PlantStructure::PlantTrain

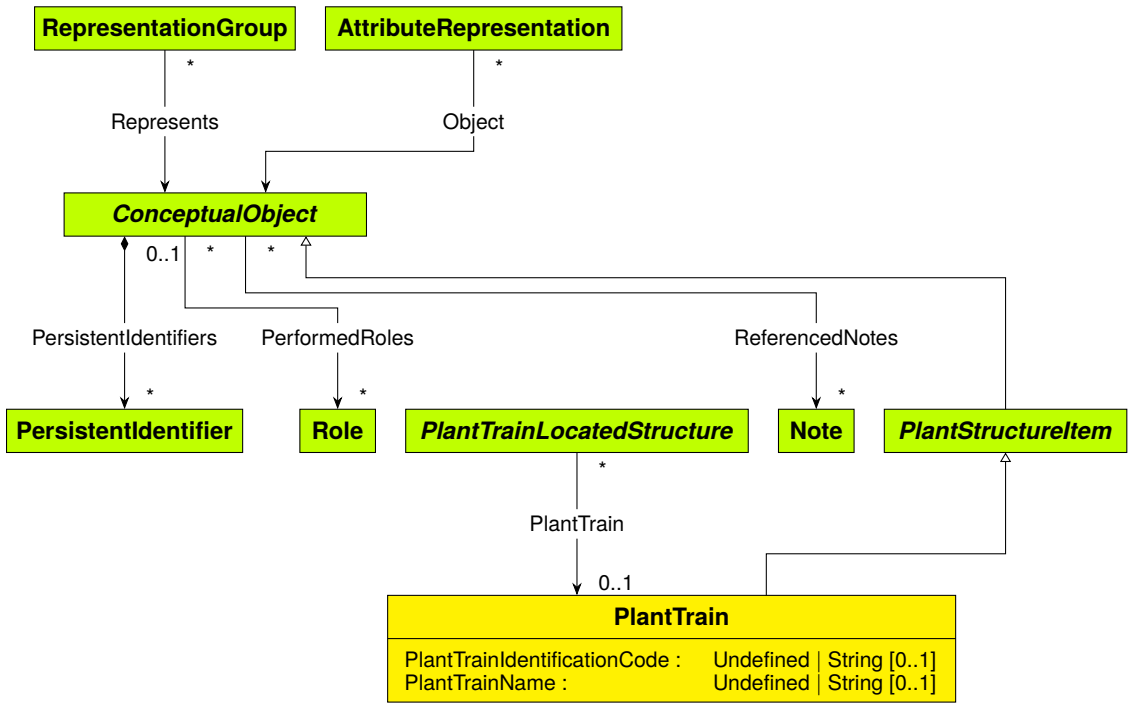
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/PlantTrain>

PLANT TRAIN

A plant train.



Supertypes

- *PlantStructureItem*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>PlantTrainIdentificationCode</i>	0..1	<i>Undefined String</i>	no	no
<i>PlantTrainName</i>	0..1	<i>Undefined String</i>	no	no

PlantTrainIdentificationCode

Data Property

The identification code of the plant train.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PlantTrainIdentificationCodeAssignmentClass>

PLANT TRAIN IDENTIFICATION CODE ASSIGNMENT CLASS

Example

"T456"

PlantTrainName

Data Property

The name of the plant train.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PlantTrainNameAssignmentClass>

PLANT TRAIN NAME ASSIGNMENT CLASS

Example

"Train T456"

Example in DEXPI XML

```
<Object type="Plant/PlantStructure.PlantTrain">
  <Data property="PlantTrainIdentificationCode">
    <String>T456</String>
  </Data>
  <Data property="PlantTrainName">
    <String>Train T456</String>
  </Data>
</Object>
```

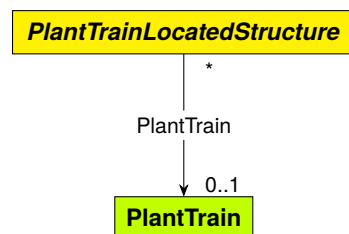
PlantTrainLocatedStructure

Plant::PlantStructure::PlantTrainLocatedStructure

Overview

Abstract Class

A structure that can be located in a *PlantTrain*.



Subtypes

- *TechnicalItem*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>PlantTrain</i>	0..1	<i>PlantTrain</i>	no	no	*

PlantTrain

Reference Property

The *PlantTrain* in which the PlantTrainLocatedStructure is located.

Type: *PlantTrain*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

ProcessPlant

Plant::PlantStructure::ProcessPlant

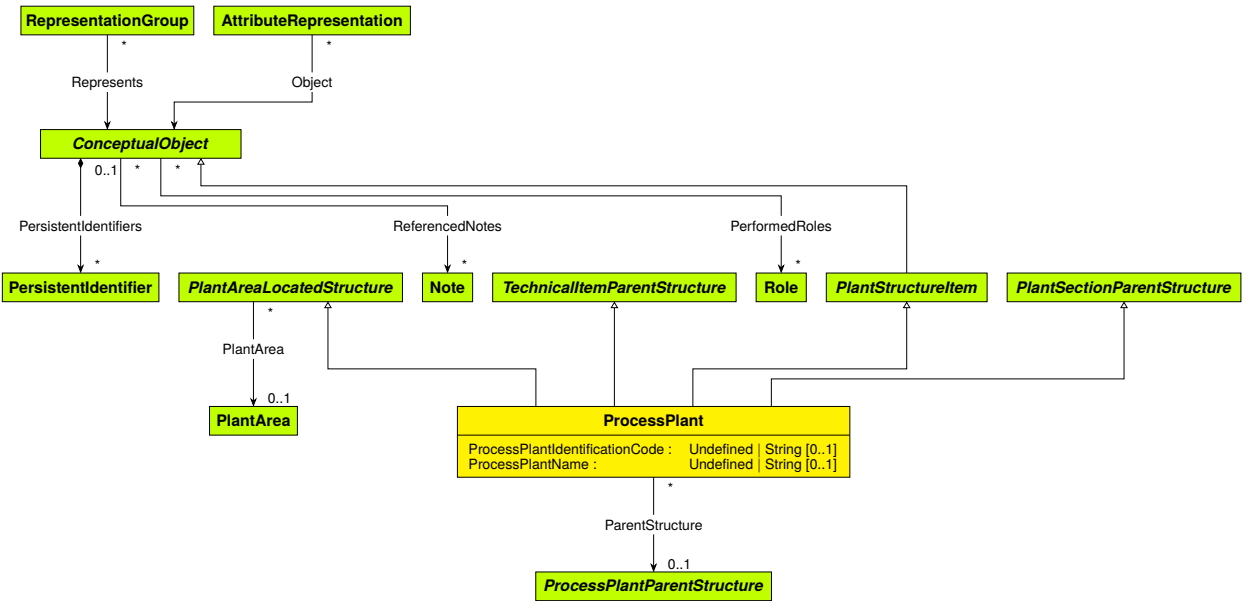
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS7151859>

PROCESS PLANT

A plant employed in carrying out chemical processes, including the required supporting processes (from <http://data.posccaesar.org/rdl/RDS7151859>).



Supertypes

- *PlantAreaLocatedStructure*
- *PlantSectionParentStructure*
- *PlantStructureItem*
- *TechnicalItemParentStructure*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>ParentStructure</i>	0..1	<i>ProcessPlantParentStructure</i>	no	no	*

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>ProcessPlantIdentificationCode</i>	0..1	<i>Undefined String</i>	no	no
<i>ProcessPlantName</i>	0..1	<i>Undefined String</i>	no	no

ParentStructure

Reference Property

A superordinate structure of which the ProcessPlant is a part.

Type: *ProcessPlantParentStructure*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

ProcessPlantIdentificationCode

Data Property

The identification code of the process plant.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ProcessPlantIdentificationCodeAssignmentClass>

PROCESS PLANT IDENTIFICATION CODE ASSIGNMENT CLASS

Example

"ABC"

ProcessPlantName

Data Property

The name of the process plant.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ProcessPlantNameAssignmentClass>

PROCESS PLANT NAME ASSIGNMENT CLASS

Example

"ABC Plant"

Example in DEXPI XML

```

<Object type="Plant/PlantStructure.ProcessPlant">
  <Data property="ProcessPlantIdentificationCode">
    <String>ABC</String>
  </Data>
  <Data property="ProcessPlantName">
    <String>ABC Plant</String>
  </Data>
</Object>

```

ProcessPlantParentStructure

Plant::PlantStructure::ProcessPlantParentStructure

Overview

Abstract Class

A *PlantStructureItem* that is a suitable *ParentStructure* of a *ProcessPlant*.

ProcessPlantParentStructure

Subtypes

- *Enterprise*
- *Site*

Site

Plant::PlantStructure::Site

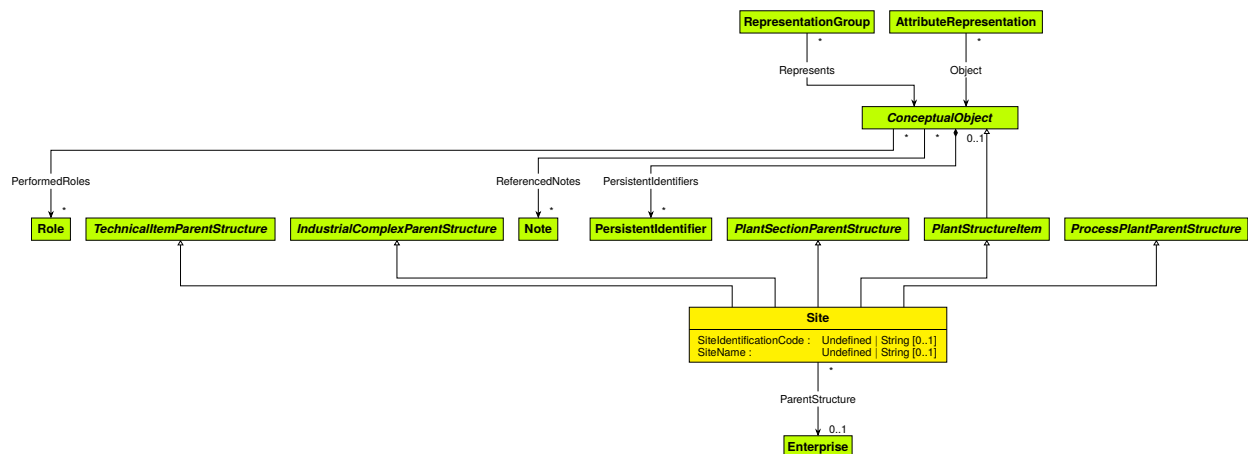
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS10418236632>

SITE ISA95

A site as defined by ISA 95.



Supertypes

- *IndustrialComplexParentStructure*
- *PlantSectionParentStructure*
- *PlantStructureItem*
- *ProcessPlantParentStructure*
- *TechnicalItemParentStructure*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>ParentStructure</i>	0..1	<i>Enterprise</i>	no	no	*

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>SiteIdentificationCode</i>	0..1	<i>Undefined String</i>	no	no
<i>SiteName</i>	0..1	<i>Undefined String</i>	no	no

ParentStructure

Reference Property

A superordinate structure of which the Site is a part.

Type: *Enterprise*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

SiteIdentificationCode

Data Property

The identification code of the site.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SiteIdentificationCodeAssignmentClass>

SITE IDENTIFICATION CODE ASSIGNMENT CLASS

Example

"DC"

SiteName

Data Property

The name of the site.

Type: *Undefined | String*

Multiplicity: 0..1**Ordered:** no**Unique:** no**RDL:** <http://sandbox.dexpi.org/rdl/SiteNameAssignmentClass>

SITE NAME ASSIGNMENT CLASS

Example`"Dexpi City"`**Example in DEXPI XML**

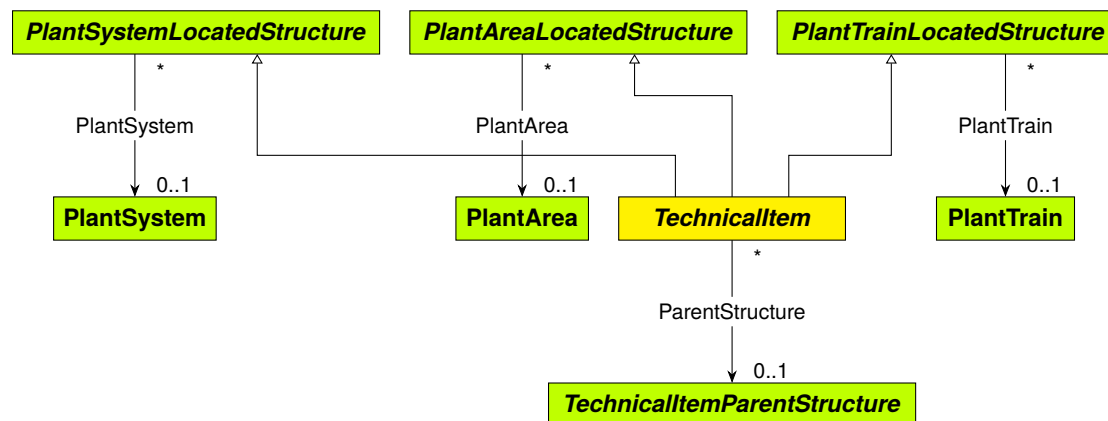
```

<Object type="Plant/PlantStructure.Site">
  <Data property="SiteIdentificationCode">
    <String>DC</String>
  </Data>
  <Data property="SiteName">
    <String>Dexpi City</String>
  </Data>
</Object>

```

TechnicalItem*Plant::PlantStructure::TechnicalItem***Overview****Abstract Class**

An item at the lowest level of the plant structure.

**Supertypes**

- *PlantAreaLocatedStructure*
- *PlantSystemLocatedStructure*
- *PlantTrainLocatedStructure*

Subtypes

- *ActuatingElectricalFunction*
- *ActuatingElectricalSystem*
- *ActuatingFunction*
- *ActuatingSystem*
- *InstrumentationLoopFunction*
- *MeasuringSystem*
- *PipingNetworkSystem*
- *ProcessInstrumentationFunction*
- *ProcessSignalGeneratingFunction*
- *TaggedPlantItem*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>ParentStructure</i>	0..1	<i>TechnicalItemParentStructure</i>	no	no	*

ParentStructure

Reference Property

A superordinate structure of which the *TechnicalItem* is a part.

Type: *TechnicalItemParentStructure*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

TechnicalItemParentStructure

Plant::PlantStructure::TechnicalItemParentStructure

Overview

Abstract Class

A *PlantStructureItem* that is a suitable *ParentStructure* of a *TechnicalItem*.

TechnicalItemParentStructure

Subtypes

- *Enterprise*
- *IndustrialComplex*
- *PlantSection*
- *ProcessPlant*
- *Site*

Plant::PlantStructure

Overview

Package

Members

Classes

Name	Type
<i>Enterprise</i>	Concrete Class
<i>IndustrialComplex</i>	Concrete Class
<i>IndustrialComplexParentStructure</i>	Abstract Class
<i>PlantArea</i>	Concrete Class
<i>PlantAreaLocatedStructure</i>	Abstract Class
<i>PlantSection</i>	Concrete Class
<i>PlantSectionParentStructure</i>	Abstract Class
<i>PlantStructureItem</i>	Abstract Class
<i>PlantSystem</i>	Concrete Class
<i>PlantSystemLocatedStructure</i>	Abstract Class
<i>PlantTrain</i>	Concrete Class
<i>PlantTrainLocatedStructure</i>	Abstract Class
<i>ProcessPlant</i>	Concrete Class
<i>ProcessPlantParentStructure</i>	Abstract Class
<i>Site</i>	Concrete Class
<i>TechnicalItem</i>	Abstract Class
<i>TechnicalItemParentStructure</i>	Abstract Class

3.2.7 ProcessEquipment

AccessNozzle

Plant::ProcessEquipment::AccessNozzle

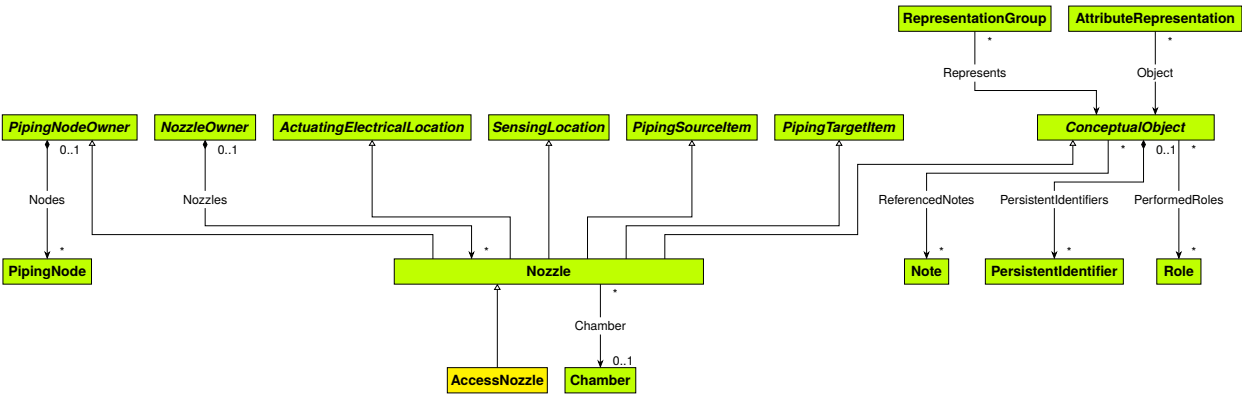
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/AccessNozzle>

ACCESS NOZZLE

A *Nozzle* intended to enable or facilitate access to the interior of its owner. This comprises especially inspection holes like manholes, handholes, and sight holes (for equipment); see *SightGlass* for piping components.



Supertypes

- *Nozzle*

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.AccessNozzle"/>
```

Agglomerator

Plant::ProcessEquipment::Agglomerator

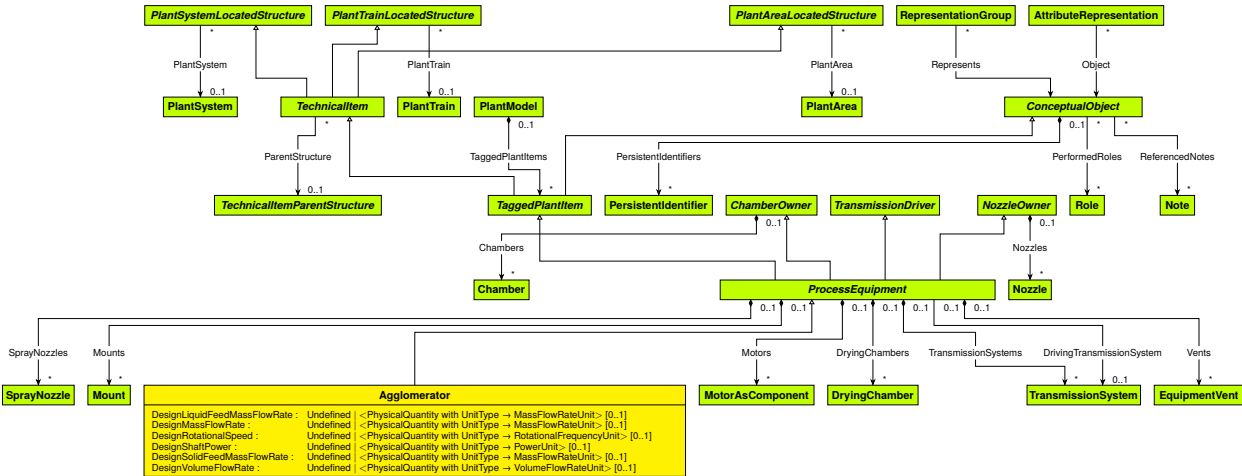
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/Agglomerator>

AGGLOMERATOR

A machine that is capable of agglomerating. It is usually vertically aligned.



Supertypes

- *ProcessEquipment*

Subtypes

- *ReciprocatingPressureAgglomerator*
- *RotatingGrowthAgglomerator*
- *RotatingPressureAgglomerator*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignLiquidFeedMassFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → MassFlowRateUnit></i>	no	no
<i>DesignMassFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → MassFlowRateUnit></i>	no	no
<i>DesignRotationalSpeed</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → RotationalFrequencyUnit></i>	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no
<i>DesignSolidFeedMassFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → MassFlowRateUnit></i>	no	no
<i>DesignVolumeFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VolumeFlowRateUnit></i>	no	no

DesignLiquidFeedMassFlowRate

Data Property

The liquid feed mass flow rate for which the *Agglomerator* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → MassFlowRateUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignLiquidFeedMassFlowRate>

DESIGN LIQUID FEED MASS FLOW RATE

Example

```
PhysicalQuantity(
  Unit = KilogramPerHour,
  Value = 240.0)
```

DesignMassFlowRate

Data Property

The mass flow rate for which the *Agglomerator* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS14286182>

DESIGN MASS FLOW RATE

Example

```
PhysicalQuantity(  
    Unit = KilogramPerSecond,  
    Value = 420.0)
```

DesignRotationalSpeed

Data Property

The rotational speed for which the *Agglomerator* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{RotationalFrequencyUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(  
    Unit = ReciprocalMinute,  
    Value = 180.0)
```

DesignShaftPower

Data Property

The shaft power for which the *Agglomerator* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(  
  Unit = Kilowatt,  
  Value = 400.0)
```

DesignSolidFeedMassFlowRate

Data Property

The solid feed mass flow rate for which the *Agglomerator* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignSolidFeedMassFlowRate>

DESIGN SOLID FEED MASS FLOW RATE

Example

```
PhysicalQuantity(  
  Unit = KilogramPerHour,  
  Value = 220.0)
```

DesignVolumeFlowRate

Data Property

The volume flow rate for which the *Agglomerator* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VolumeFlowRateUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS14286227>

DESIGN VOLUME FLOW RATE

Example

```
PhysicalQuantity(  
  Unit = MetreCubedPerHour,  
  Value = 420.0)
```

Example in DEXPI XML

```

<Object type="Plant/ProcessEquipment.Agglomerator">
  <Data property="DesignLiquidFeedMassFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour"/>
      </Data>
      <Data property="Value">
        <Double>240.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignMassFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerSecond"/>
      </Data>
      <Data property="Value">
        <Double>420.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignRotationalSpeed">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ ReciprocalMinute"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignShaftPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>400.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignSolidFeedMassFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour"/>
      </Data>
      <Data property="Value">
        <Double>220.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>

```

(continues on next page)

(continued from previous page)

```

    </AggregatedDataValue>
  </Data>
  <Data property="DesignVolumeFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour
        ↪"/>
      </Data>
      <Data property="Value">
        <Double>420.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>

```

Agitator

Plant::ProcessEquipment::Agitator

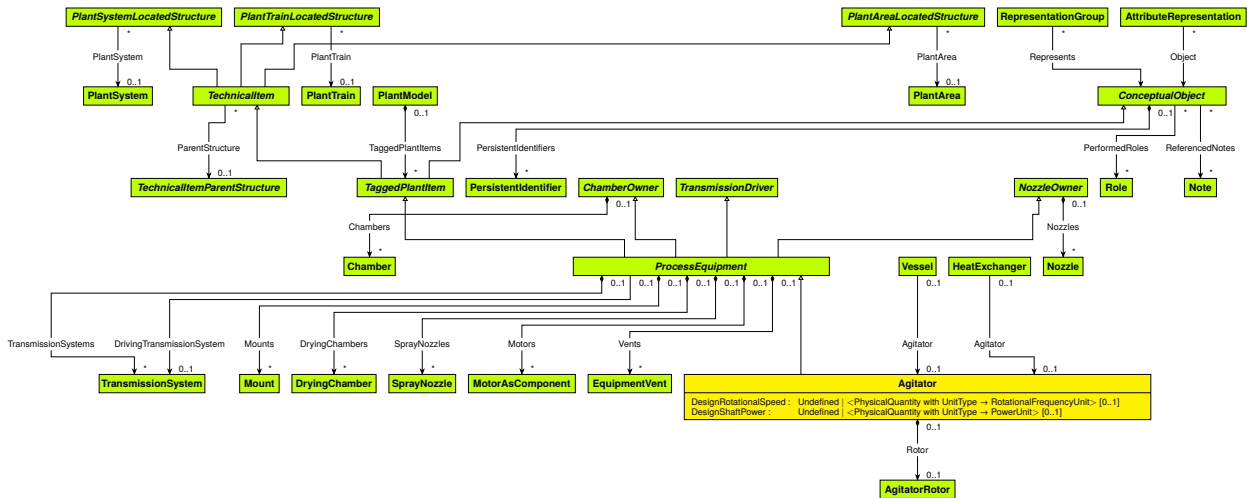
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS16045622>

AGITATOR

An Agitator is a dynamic mixer that stirs or shakes fluids by reaction force from moving vanes.



Supertypes

- *ProcessEquipment*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Rotor</i>	0..1	<i>AgitatorRotor</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignRotationalSpeed</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → RotationalFrequencyUnit></i>	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no

DesignRotationalSpeed

Data Property

The rotational speed for which the *Agitator* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(
  Unit = ReciprocalMinute,
  Value = 180.0)
```

DesignShaftPower

Data Property

The shaft power for which the *Agitator* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PowerUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(  
    Unit = Kilowatt,  
    Value = 400.0)
```

Rotor

Composition Property

The rotor of the Agitator.

Type: *AgitatorRotor*

Multiplicity: 0..1

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Agitator">  
  <Data property="DesignRotationalSpeed">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.  
→ ReciprocalMinute"/>  
      </Data>  
      <Data property="Value">  
        <Double>180.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
  <Data property="DesignShaftPower">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>  
      </Data>  
      <Data property="Value">  
        <Double>400.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
</Object>
```

AgitatorRotor

Plant::ProcessEquipment::AgitatorRotor

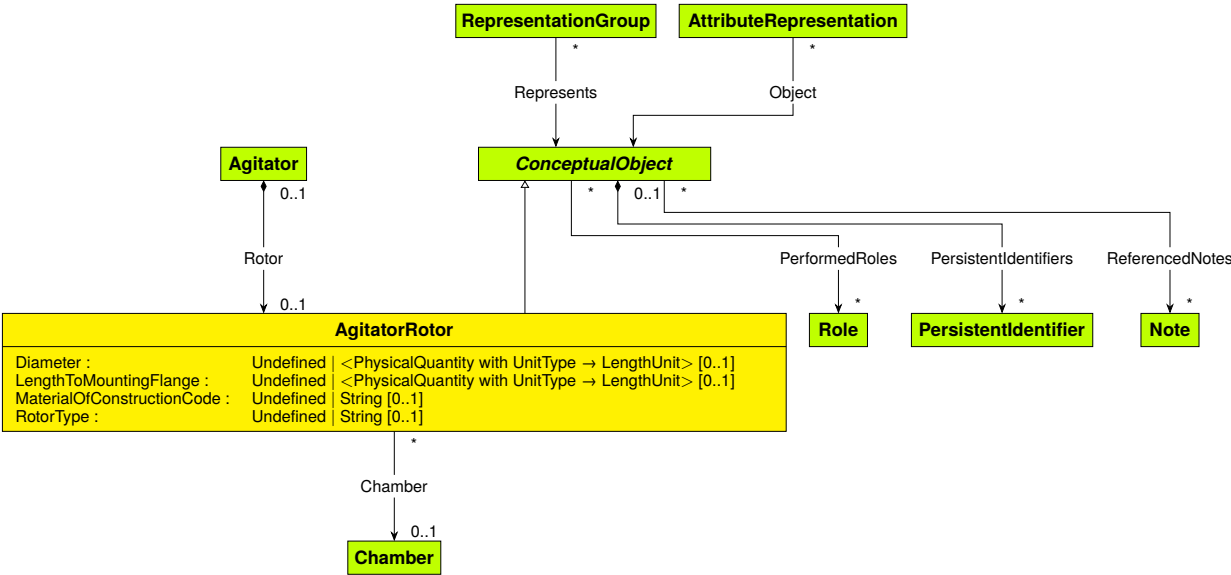
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/AgitatorRotor>

AGITATOR ROTOR

The machine component that is the rotating portion of an *Agitator*.



Supertypes

- *ConceptualObject*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Chamber</i>	0..1	<i>Chamber</i>	no	no	*

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Diameter</i>	0..1	<i>Undefined</i> <PhysicalQuantity with UnitType -> LengthUnit>	no	no
<i>LengthToMountingFlange</i>	0..1	<i>Undefined</i> <PhysicalQuantity with UnitType -> LengthUnit>	no	no
<i>MaterialOfConstructionCode</i>	0..1	<i>Undefined</i> String	no	no
<i>RotorType</i>	0..1	<i>Undefined</i> String	no	no

Chamber

Reference Property

The *Chamber* in which the AgitatorRotor is located, if applicable. The Chamber must be a component of the same object as the AgitatorRotor.

Type: *Chamber*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

Diameter

Data Property

The diameter of the *AgitatorRotor*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with } \text{UnitType} \rightarrow \text{LengthUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS350954>

DIAMETER

Example

```
PhysicalQuantity(  
  Unit = Centimetre,  
  Value = 20.0)
```

LengthToMountingFlange

Data Property

The length to the mounting flange of the *AgitatorRotor*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with } \text{UnitType} \rightarrow \text{LengthUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LengthToMountingFlange>

LENGTH TO MOUNTING FLANGE

Example

```
PhysicalQuantity(  
  Unit = Centimetre,  
  Value = 80.0)
```

MaterialOfConstructionCode

Data Property

A code that gives the material of construction of the *AgitatorRotor*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS1460719741>

MATERIAL OF CONSTRUCTION CODE ASSIGNMENT CLASS

Example

"1.4306"

RotorType

Data Property

The rotor type of the *AgitatorRotor*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/RotorTypeAssignmentClass>

ROTOR TYPE ASSIGNMENT CLASS

Example

"xy1"

Example in DEXPI XML

```

<Object type="Plant/ProcessEquipment.AgitatorRotor">
  <Data property="Diameter">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
      </Data>
      <Data property="Value">
        <Double>20.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="LengthToMountingFlange">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">

```

(continues on next page)

(continued from previous page)

```
<DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
</Data>
<Data property="Value">
  <Double>80.0</Double>
</Data>
</AggregatedDataValue>
</Data>
<Data property="MaterialOfConstructionCode">
  <String>1.4306</String>
</Data>
<Data property="RotorType">
  <String>xy1</String>
</Data>
</Object>
```

AirCoolingSystem

Plant::ProcessEquipment::AirCoolingSystem

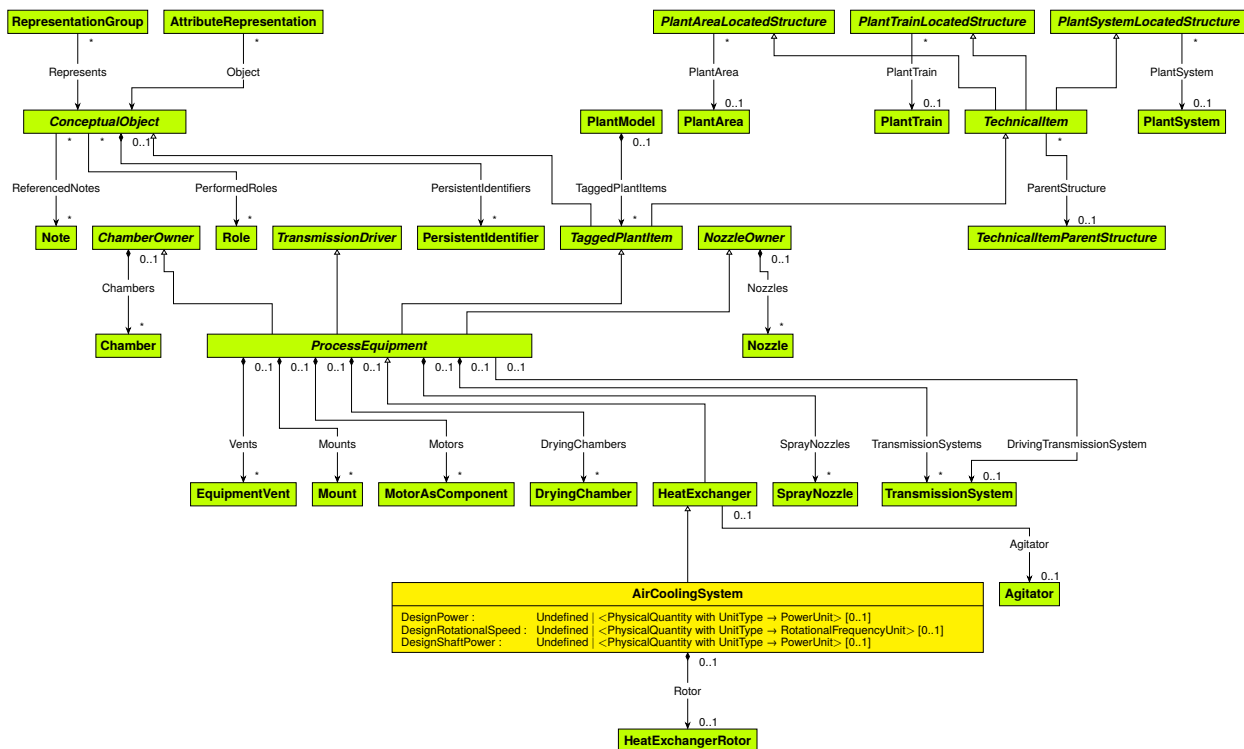
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS277379>

AIR COOLING SYSTEM

A cooling system which uses air as the cooling medium (from <http://data.posccaesar.org/rdl/RDS277379>).



Supertypes

- *HeatExchanger*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Rotor</i>	0..1	<i>HeatExchangerRotor</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no
<i>DesignRotational-Speed</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → Rotational-FrequencyUnit></i>	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no

DesignPower

Data Property

The power for which the *AirCoolingSystem* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PowerUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignPower>

DESIGN POWER

Example

```
PhysicalQuantity(
  Unit = Kilowatt,
  Value = 500.0)
```

DesignRotationalSpeed

Data Property

The rotational speed for which the *AirCoolingSystem* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(  
    Unit = ReciprocalMinute,  
    Value = 180.0)
```

DesignShaftPower

Data Property

The shaft power for which the *AirCoolingSystem* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PowerUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(  
    Unit = Kilowatt,  
    Value = 400.0)
```

Rotor

Composition Property

The rotor of the *AirCoolingSystem*.

Type: *HeatExchangerRotor*

Multiplicity: 0..1

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.AirCoolingSystem">  
  <Data property="DesignPower">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>  
      </Data>  
      <Data property="Value">
```

(continues on next page)

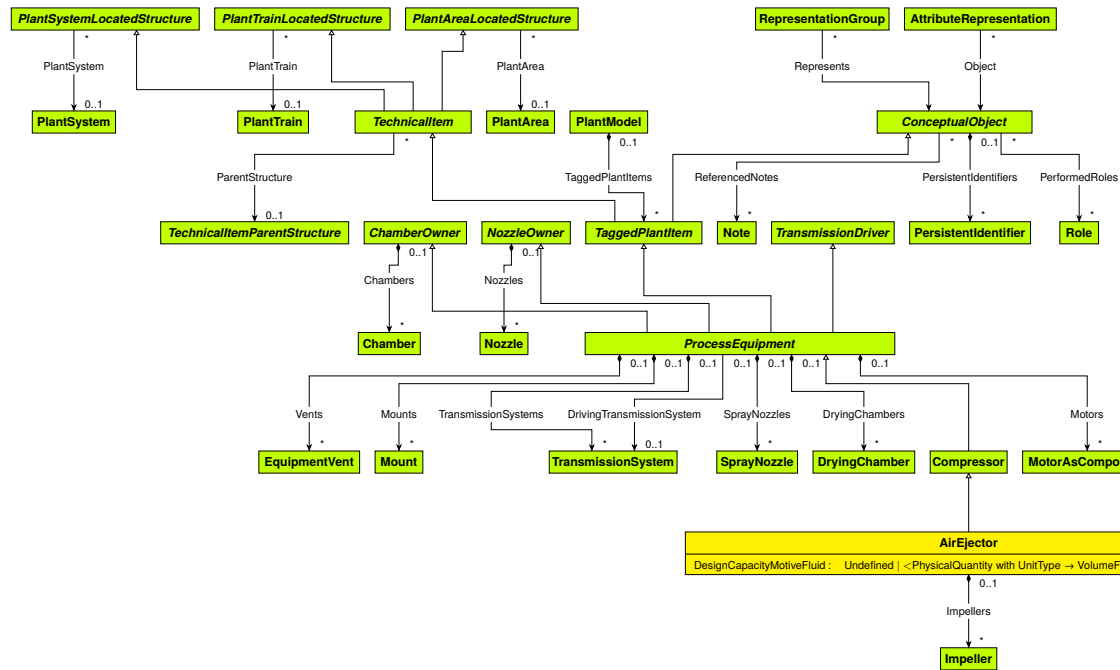
(continued from previous page)

```

        <Double>500.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignRotationalSpeed">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ReciprocalMinute"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignShaftPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>400.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>

```

AirEjector*Plant::ProcessEquipment::AirEjector***Overview****Concrete Class****RDL:** <http://data.posccaesar.org/rdl/RDS5770157>**AIR EJECTOR**An ejector intended to create vacuum using compressed air (from <http://data.posccaesar.org/rdl/RDS5770157>).



Supertypes

- *Compressor*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Impellers</i>	*	<i>Impeller</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignCapacityMotiveFluid</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VolumeFlowRateUnit></i>	no	no

DesignCapacityMotiveFluid

Data Property

The capacity of the volume flow rate for the motive fluid for which the *AirEjector* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → VolumeFlowRateUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignCapacityMotiveFluid>

DESIGN CAPACITY MOTIVE FLUID

Example

```
PhysicalQuantity(
    Unit = MetreCubedPerHour,
    Value = 40.0)
```

Impellers

Composition Property

The impellers of the AirEjector.

Type: *Impeller*

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.AirEjector">
  <Data property="DesignCapacityMotiveFluid">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour
↪"/>
      </Data>
      <Data property="Value">
        <Double>40.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

AlternatingCurrentGenerator

Plant::ProcessEquipment::AlternatingCurrentGenerator

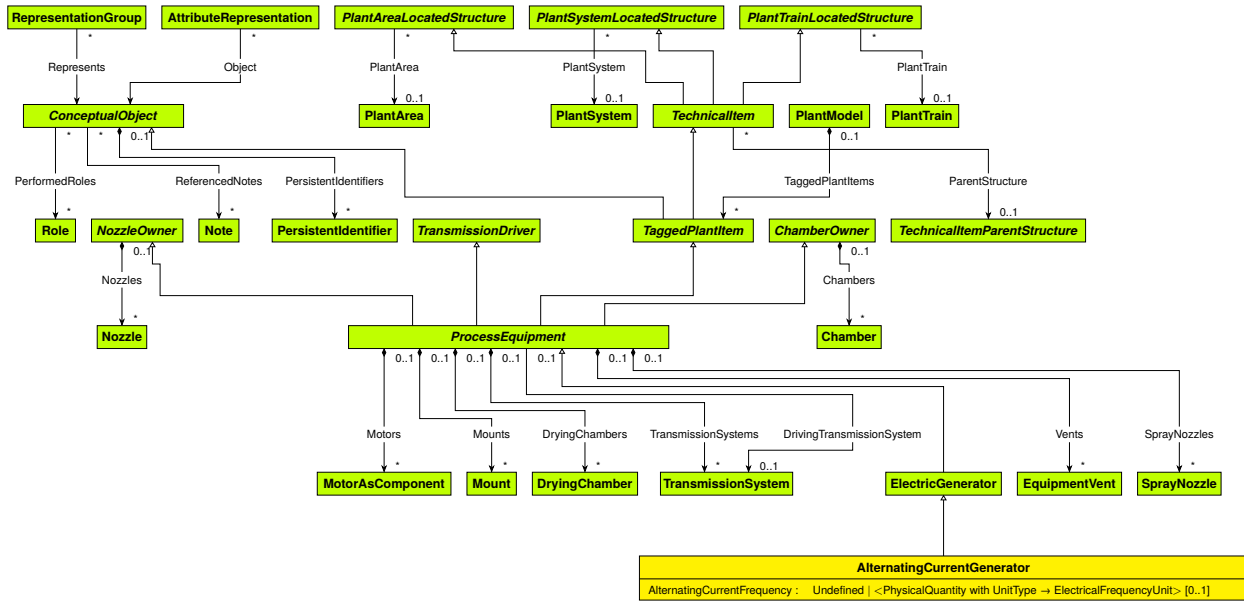
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS873359>

ALTERNATING CURRENT GENERATOR

An electric generator for the production of alternating current and voltage (from <http://data.posccaesar.org/rdl/RDS873359>).



Supertypes

- *ElectricGenerator*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>AlternatingCurrentFrequency</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → ElectricalFrequencyUnit></i>	no	no

AlternatingCurrentFrequency

Data Property

The alternating current frequency of the *AlternatingCurrentGenerator*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → ElectricalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/AlternatingCurrentFrequency>

ALTERNATING CURRENT FREQUENCY

Example

```
PhysicalQuantity(
  Unit = Hertz,
  Value = 180.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.AlternatingCurrentGenerator">
  <Data property="AlternatingCurrentFrequency">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.ElectricalFrequencyUnit.Hertz"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

AlternatingCurrentMotor

Plant::ProcessEquipment::AlternatingCurrentMotor

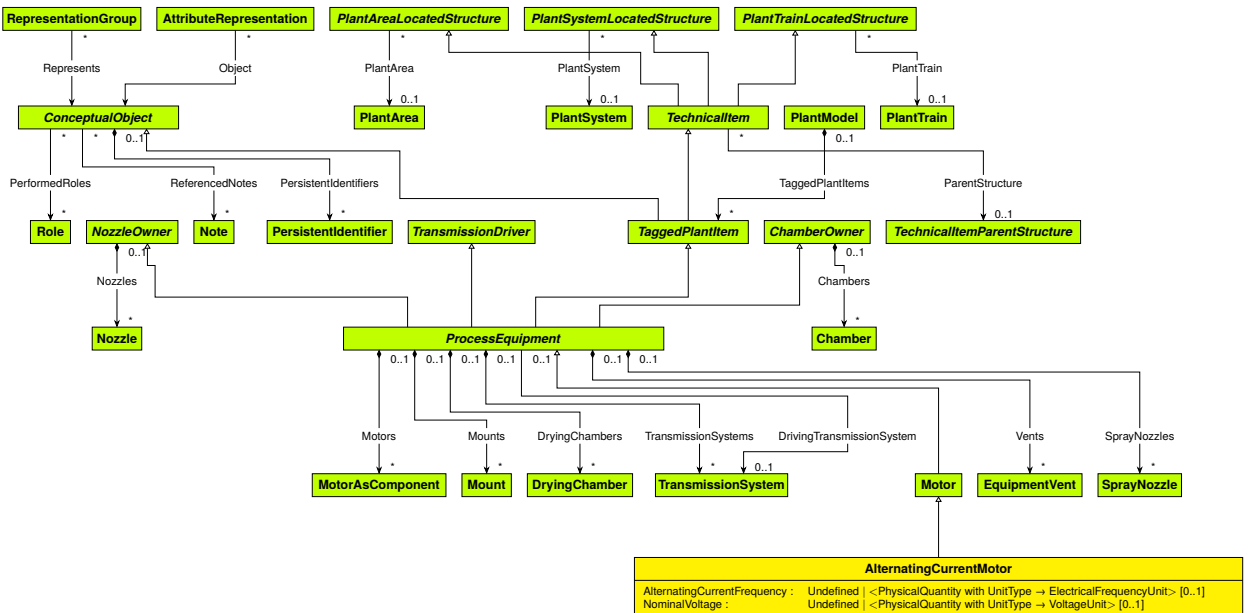
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS472994>

ALTERNATING CURRENT MOTOR

An electric motor driven by alternating electric current



Supertypes

- *Motor*

Members

Data Properties

Name	Multi- plicity	Type	Or- dered	Unique
<i>AlternatingCurrentFrequency</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → ElectricalFrequencyUnit></i>	no	no
<i>NominalVoltage</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VoltageUnit></i>	no	no

AlternatingCurrentFrequency

Data Property

The alternating current frequency of the *AlternatingCurrentMotor*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → ElectricalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/AlternatingCurrentFrequency>

ALTERNATING CURRENT FREQUENCY

Example

```
PhysicalQuantity(  
    Unit = Hertz,  
    Value = 180.0)
```

NominalVoltage

Data Property

The nominal voltage of the *AlternatingCurrentMotor*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → VoltageUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS369449>

NOMINAL VOLTAGE

Example

```
PhysicalQuantity(
  Unit = Volt,
  Value = 230.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.AlternatingCurrentMotor">
  <Data property="AlternatingCurrentFrequency">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.ElectricalFrequencyUnit.Hertz"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="NominalVoltage">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.VoltageUnit.Volt"/>
      </Data>
      <Data property="Value">
        <Double>230.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

AlternatingCurrentMotorAsComponent

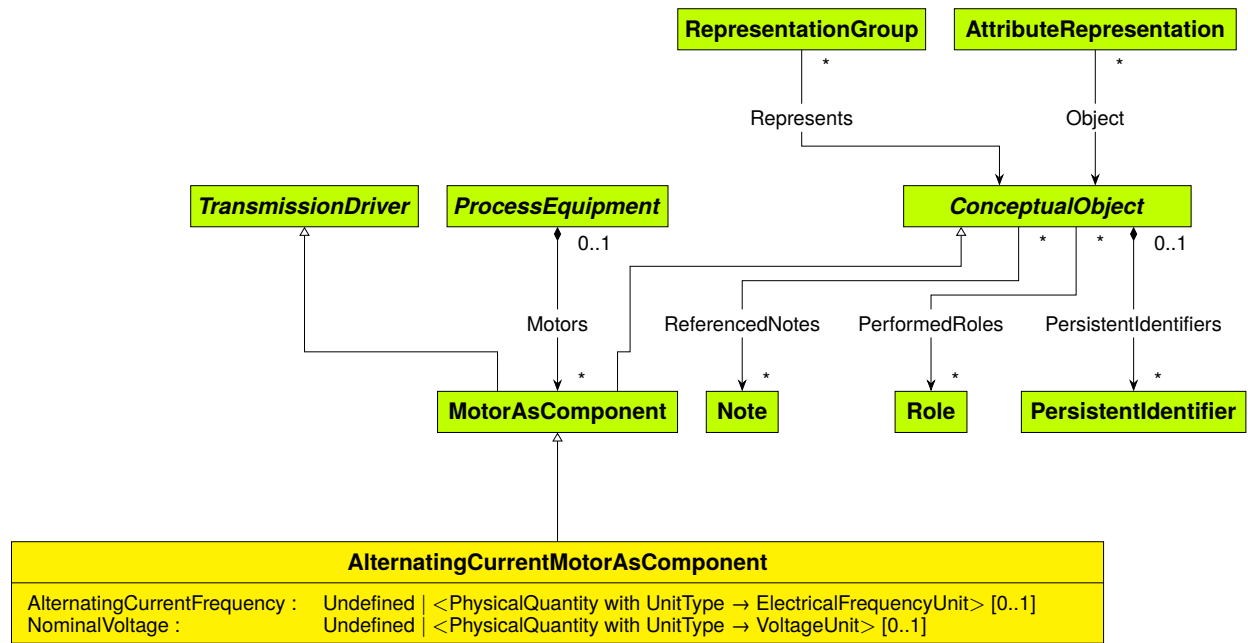
Plant::ProcessEquipment::AlternatingCurrentMotorAsComponent

Overview**Concrete Class**

RDL: <http://sandbox.dexpi.org/rdl/AlternatingCurrentMotorAsComponent>

ALTERNATING CURRENT MOTOR AS COMPONENT

An electric motor driven by alternating electric current that is used as a component of an apparatus or of a machine.



Supertypes

- *MotorAsComponent*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>AlternatingCurrentFrequency</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → ElectricalFrequencyUnit></i>	no	no
<i>NominalVoltage</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VoltageUnit></i>	no	no

AlternatingCurrentFrequency

Data Property

The alternating current frequency of the *AlternatingCurrentMotorAsComponent*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → ElectricalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/AlternatingCurrentFrequency>

ALTERNATING CURRENT FREQUENCY

Example

```
PhysicalQuantity(
  Unit = Hertz,
  Value = 180.0)
```

NominalVoltage**Data Property**

The nominal voltage of the *AlternatingCurrentMotorAsComponent*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VoltageUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS369449>

NOMINAL VOLTAGE

Example

```
PhysicalQuantity(
  Unit = Volt,
  Value = 230.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.AlternatingCurrentMotorAsComponent">
  <Data property="AlternatingCurrentFrequency">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.ElectricalFrequencyUnit.Hertz"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="NominalVoltage">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.VoltageUnit.Volt"/>
      </Data>
      <Data property="Value">
        <Double>230.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```


Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignRotationalSpeed</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → RotationalFrequencyUnit></i>	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no

DesignRotationalSpeed

Data Property

The rotational speed for which the *AxialCompressor* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(  
    Unit = ReciprocalMinute,  
    Value = 180.0)
```

DesignShaftPower

Data Property

The shaft power for which the *AxialCompressor* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PowerUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(  
    Unit = Kilowatt,  
    Value = 400.0)
```

Impellers

Composition Property

The impellers of the AxialCompressor.

Type: *Impeller*

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.AxialCompressor">
  <Data property="DesignRotationalSpeed">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ReciprocalMinute"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignShaftPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>400.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

AxialFan

Plant::ProcessEquipment::AxialFan

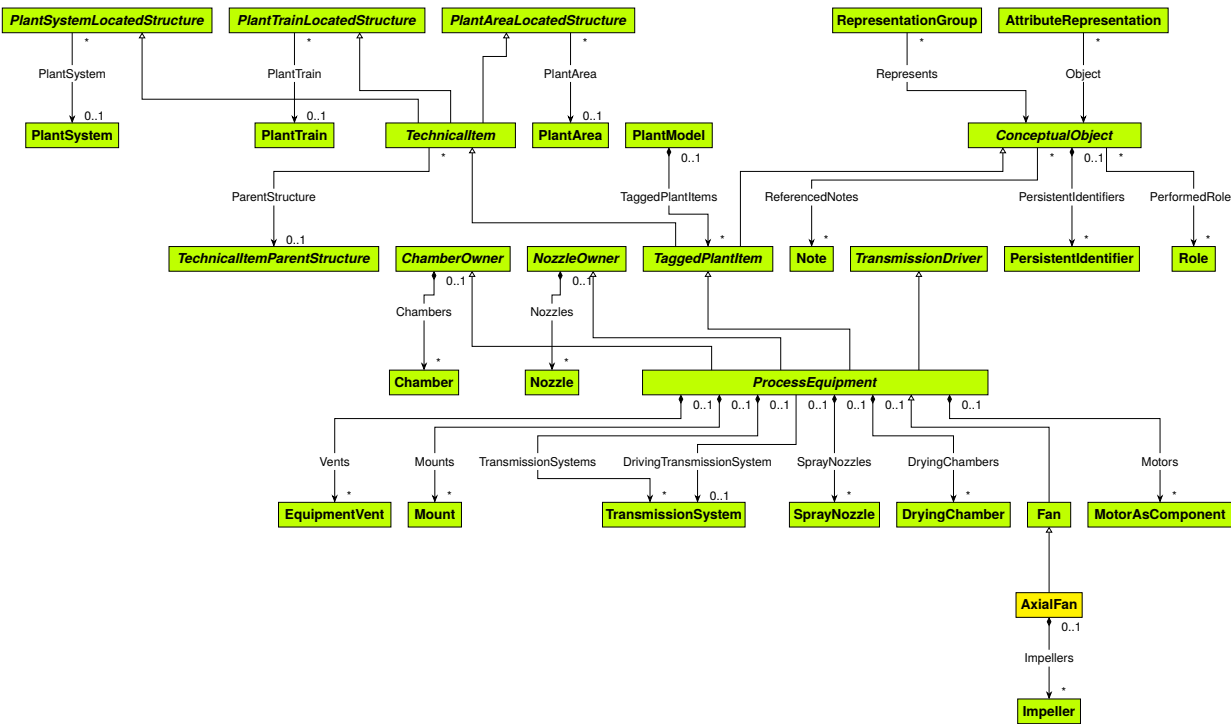
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS414044>

AXIAL FAN

A fan where the flow is along axis of shaft and the pressure ratio is relatively low (from <http://data.posccaesar.org/rdl/RDS414044>).



Supertypes

- *Fan*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Impellers</i>	*	<i>Impeller</i>	no

Impellers

Composition Property

The impellers of the AxialFan.

Type: *Impeller*

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.AxialFan"/>
```

BatchWeigher

Plant::ProcessEquipment::BatchWeigher

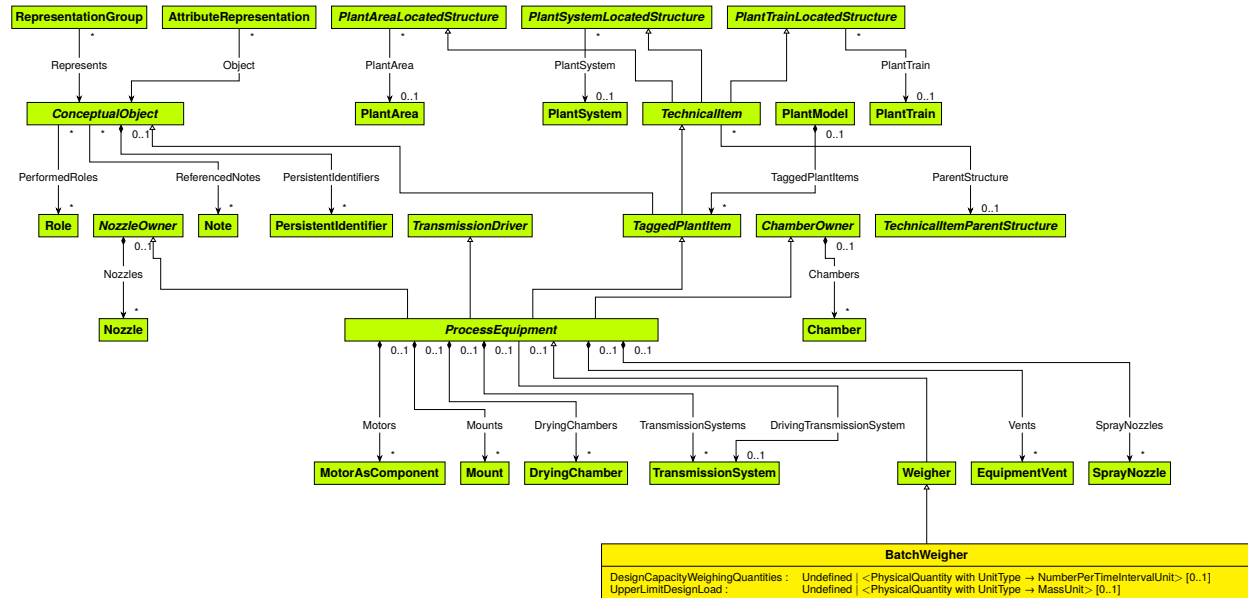
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/BatchWeigher>

BATCH WEIGHER

A *Weigher* that is operating in batch mode.



Supertypes

- *Weigher*

Members

Data Properties

Name	Multi- plicity	Type	Or- dered	Unique
<i>DesignCapacityWeighingQuantities</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → NumberPerTimeIntervalUnit></i>	no	no
<i>UpperLimitDesignLoad</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → MassUnit></i>	no	no

DesignCapacityWeighingQuantities

Data Property

The capacity for the number of weighing quantities per time for which the *BatchWeigher* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → NumberPerTimeIntervalUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignCapacityWeighingQuantities>

DESIGN CAPACITY WEIGHING QUANTITIES

Example

```
PhysicalQuantity(  
    Unit = ReciprocalSecond,  
    Value = 42.0)
```

UpperLimitDesignLoad

Data Property

The upper limit for the load for which the *BatchWeigher* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → MassUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitDesignLoad>

UPPER LIMIT DESIGN LOAD

Example

```
PhysicalQuantity(  
    Unit = Kilogram,  
    Value = 900.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.BatchWeigher">  
  <Data property="DesignCapacityWeighingQuantities">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.NumberPerTimeIntervalUnit.  
↪ReciprocalSecond"/>  
      </Data>  
      <Data property="Value">  
        <Double>42.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
  <Data property="UpperLimitDesignLoad">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
```

(continues on next page)

(continued from previous page)

```

<Data property="Unit">
  <DataReference data="Core/PhysicalQuantities.MassUnit.Kilogram"/>
</Data>
<Data property="Value">
  <Double>900.0</Double>
</Data>
</AggregatedDataValue>
</Data>
</Object>

```

Blower

Plant::ProcessEquipment::Blower

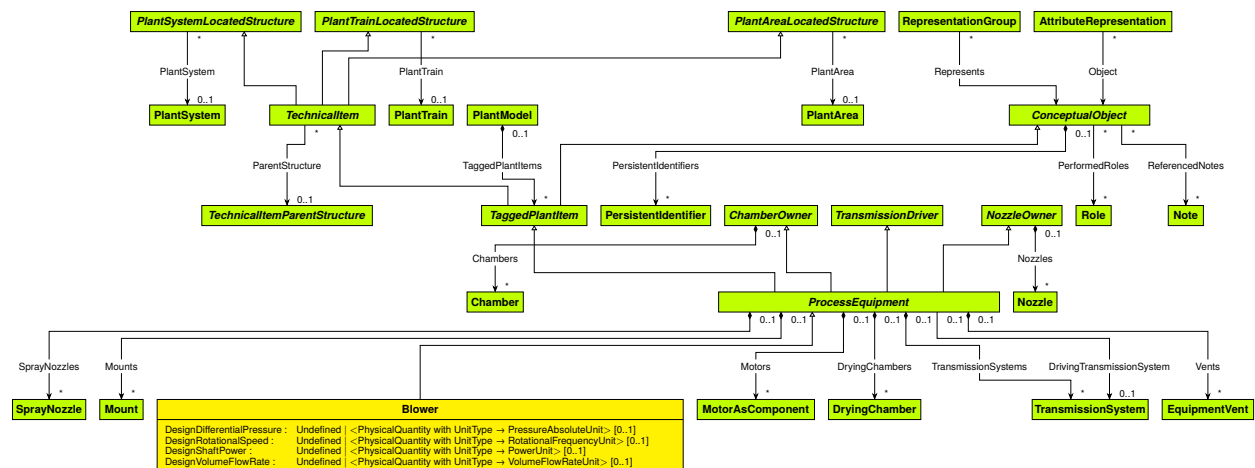
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS417149>

BLOWER

A physical object intended to compress and supply air or gas at medium pressure and medium volume flow (from <http://data.posccaesar.org/rdl/RDS417149>).



Supertypes

- *ProcessEquipment*

Subtypes

- *AxialBlower*
- *CentrifugalBlower*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignDifferentialPressure</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PressureAbsoluteUnit></i>	no	no
<i>DesignRotationalSpeed</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → RotationalFrequencyUnit></i>	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no
<i>DesignVolumeFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VolumeFlowRateUnit></i>	no	no

DesignDifferentialPressure

Data Property

The differential pressure for which the *Blower* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PressureAbsoluteUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignDifferentialPressure>

DESIGN DIFFERENTIAL PRESSURE

Example

```
PhysicalQuantity(  
    Unit = Bar,  
    Value = 4.8)
```

DesignRotationalSpeed

Data Property

The rotational speed for which the *Blower* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(
  Unit = ReciprocalMinute,
  Value = 180.0)
```

DesignShaftPower**Data Property**

The shaft power for which the *Blower* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(
  Unit = Kilowatt,
  Value = 400.0)
```

DesignVolumeFlowRate**Data Property**

The volume flow rate for which the *Blower* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VolumeFlowRateUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS14286227>

DESIGN VOLUME FLOW RATE

Example

```
PhysicalQuantity(
  Unit = MetreCubedPerHour,
  Value = 420.0)
```

Example in DEXPI XML

```

<Object type="Plant/ProcessEquipment.Blower">
  <Data property="DesignDifferentialPressure">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PressureAbsoluteUnit.Bar"/>
      </Data>
      <Data property="Value">
        <Double>4.8</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignRotationalSpeed">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ ReciprocalMinute"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignShaftPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>400.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignVolumeFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour
↪ "/>
      </Data>
      <Data property="Value">
        <Double>420.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>

```

Boiler

Plant::ProcessEquipment::Boiler

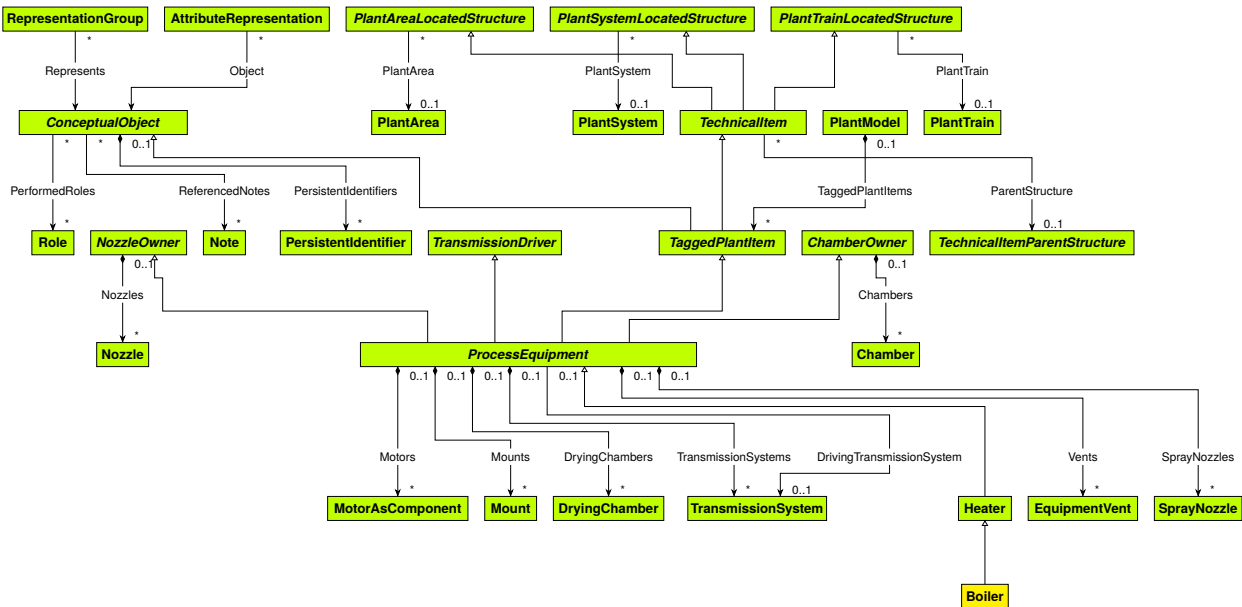
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS14058190>

BOILER

A *Heater* that brings a liquid to its boiling point.



Supertypes

- *Heater*

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Boiler"/>
```

BriquettingRoller

Plant::ProcessEquipment::BriquettingRoller

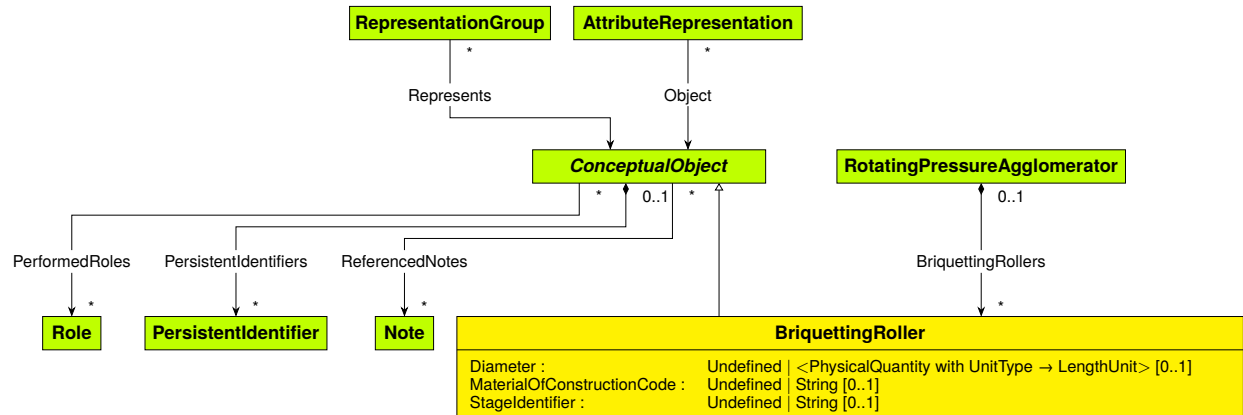
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/BriquettingRoller>

BRIQUETTING ROLLER

An element of an *Agglomerator* that compresses bulk material into briquettes.



Supertypes

- *ConceptualObject*

Members

Data Properties

Name	Multi- plicity	Type	Or- dered	Unique
<i>Diameter</i>	0..1	<i>Undefined</i> <PhysicalQuantity with UnitType → LengthUnit>	no	no
<i>MaterialOfConstruction-Code</i>	0..1	<i>Undefined</i> String	no	no
<i>StageIdentifier</i>	0..1	<i>Undefined</i> String	no	no

Diameter

Data Property

The diameter of the *BriquettingRoller*.

Type: *Undefined* | <PhysicalQuantity with UnitType → LengthUnit>

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS350954>

DIAMETER

Example

```
PhysicalQuantity(
  Unit = Centimetre,
  Value = 20.0)
```

MaterialOfConstructionCode

Data Property

A code that gives the material of construction of the *BriquettingRoller*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS1460719741>

MATERIAL OF CONSTRUCTION CODE ASSIGNMENT CLASS

Example

"1.4306"

StageIdentifier

Data Property

The stage identifier of the *BriquettingRoller*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/StageIdentifierAssignmentClass>

STAGE IDENTIFIER ASSIGNMENT CLASS

Example

"s1"

Example in DEXPI XML

```

<Object type="Plant/ProcessEquipment.BriquettingRoller">
  <Data property="Diameter">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
      </Data>
      <Data property="Value">
        <Double>20.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="MaterialOfConstructionCode">
    <String>1.4306</String>
  </Data>

```

(continues on next page)

(continued from previous page)

```

<Data property="StageIdentifier">
  <String>s1</String>
</Data>
</Object>

```

Burner

Plant::ProcessEquipment::Burner

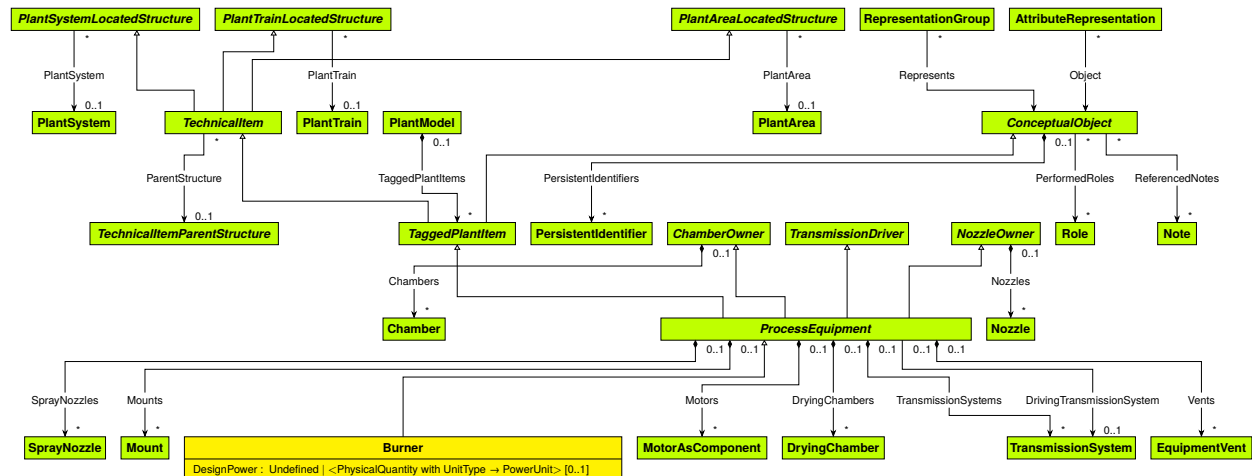
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS284399>

BURNER

A physical object that is intended to release thermal energy by burning a combustible mixture (from <http://data.posccaesar.org/rdl/RDS284399>).



Supertypes

- *ProcessEquipment*

Members

Data Properties

Name	Multiplic-ity	Type	Or-dered	Unique
<i>Design-Power</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no

DesignPower

Data Property

The power for which the *Burner* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignPower>

DESIGN POWER

Example

```
PhysicalQuantity(  
    Unit = Kilowatt,  
    Value = 500.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Burner">  
  <Data property="DesignPower">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>  
      </Data>  
      <Data property="Value">  
        <Double>500.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
</Object>
```

CentrifugalBlower

Plant::ProcessEquipment::CentrifugalBlower

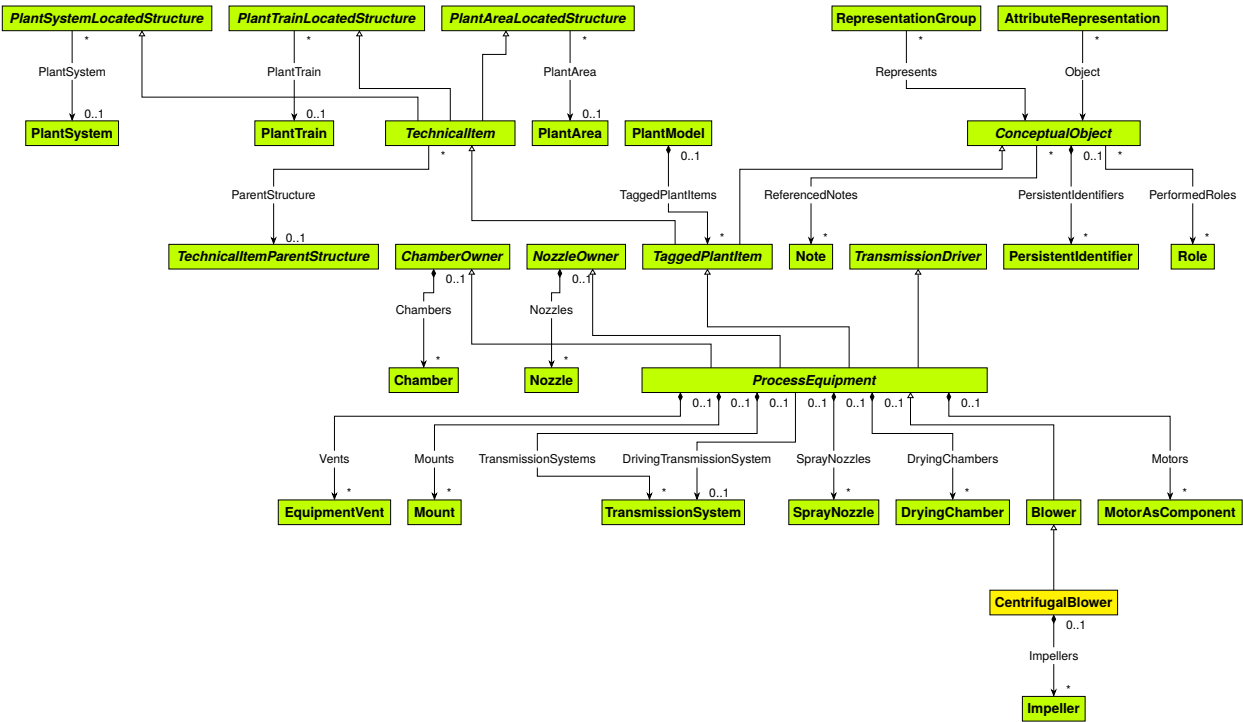
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS16766514>

CENTRIFUGAL BLOWER

A blower in which one ore more impellers accelerate the flow and where the main flow through the impeller is radial.



Supertypes

- *Blower*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Impellers</i>	*	<i>Impeller</i>	no

Impellers

Composition Property

The impellers of the CentrifugalBlower.

Type: *Impeller*

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.CentrifugalBlower"/>
```


CentrifugalCompressor

Plant::ProcessEquipment::CentrifugalCompressor

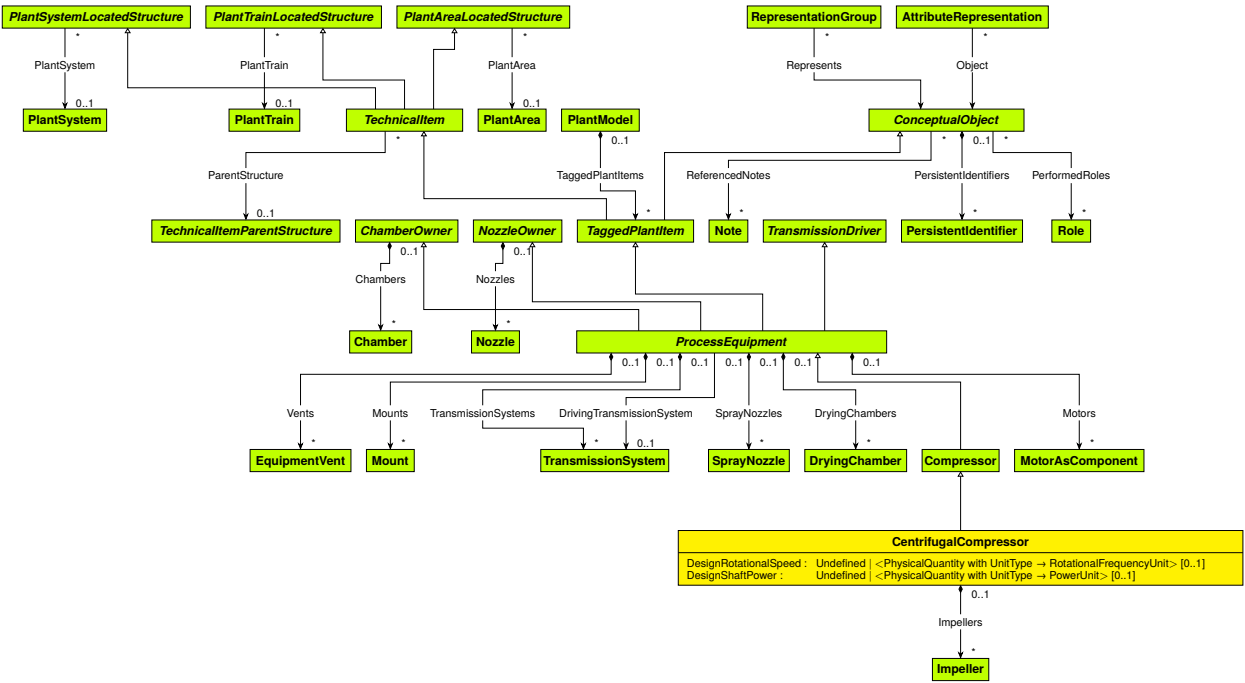
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS417194>

CENTRIFUGAL COMPRESSOR

A dynamic compressor in which one ore more impellers accelerate the gas and where the main flow through the impeller is radial (from <http://data.posccaesar.org/rdl/RDS417194>).



Supertypes

- *Compressor*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Impellers</i>	*	<i>Impeller</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignRotationalSpeed</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → RotationalFrequencyUnit></i>	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no

DesignRotationalSpeed

Data Property

The rotational speed for which the *CentrifugalCompressor* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(  
    Unit = ReciprocalMinute,  
    Value = 1800.0)
```

DesignShaftPower

Data Property

The shaft power for which the *CentrifugalCompressor* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PowerUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(  
    Unit = Kilowatt,  
    Value = 4000.0)
```

Impellers

Composition Property

The impellers of the CentrifugalCompressor.

Type: *Impeller*

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.CentrifugalCompressor">
  <Data property="DesignRotationalSpeed">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ReciprocalMinute"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignShaftPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>400.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

CentrifugalPump

Plant::ProcessEquipment::CentrifugalPump

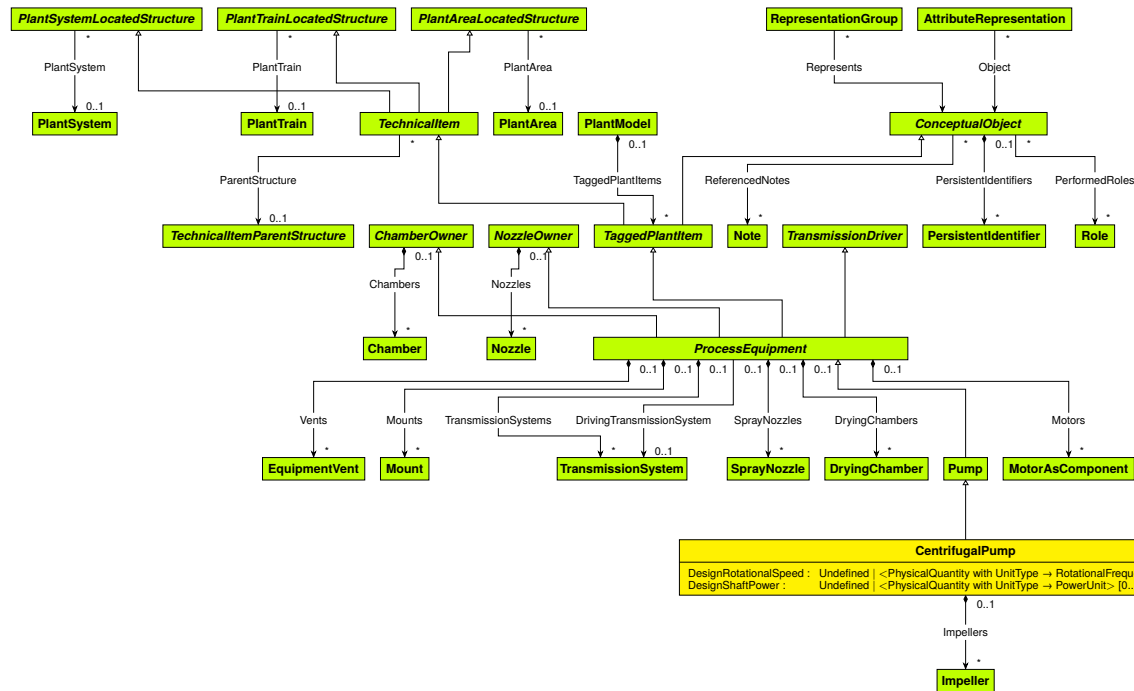
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS416834>

CENTRIFUGAL PUMP

A dynamic pump utilizing impellers provided with vanes generating centrifugal force to achieve the required pressure head (from <http://data.posccaesar.org/rdl/RDS416834>).



Supertypes

- *Pump*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Impellers</i>	*	<i>Impeller</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignRotationalSpeed</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → RotationalFrequencyUnit></i>	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no

DesignRotationalSpeed

Data Property

The rotational speed for which the *CentrifugalPump* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(
  Unit = ReciprocalMinute,
  Value = 180.0)
```

DesignShaftPower

Data Property

The shaft power for which the *CentrifugalPump* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PowerUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(
  Unit = Kilowatt,
  Value = 400.0)
```

Impellers

Composition Property

The impellers of the *CentrifugalPump*.

Type: *Impeller*

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.CentrifugalPump">
  <Data property="DesignRotationalSpeed">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ ReciprocalMinute"/>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

(continues on next page)

(continued from previous page)

```

    <Data property="Value">
      <Double>180.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="DesignShaftPower">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
    </Data>
    <Data property="Value">
      <Double>400.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>

```

Centrifuge

Plant::ProcessEquipment::Centrifuge

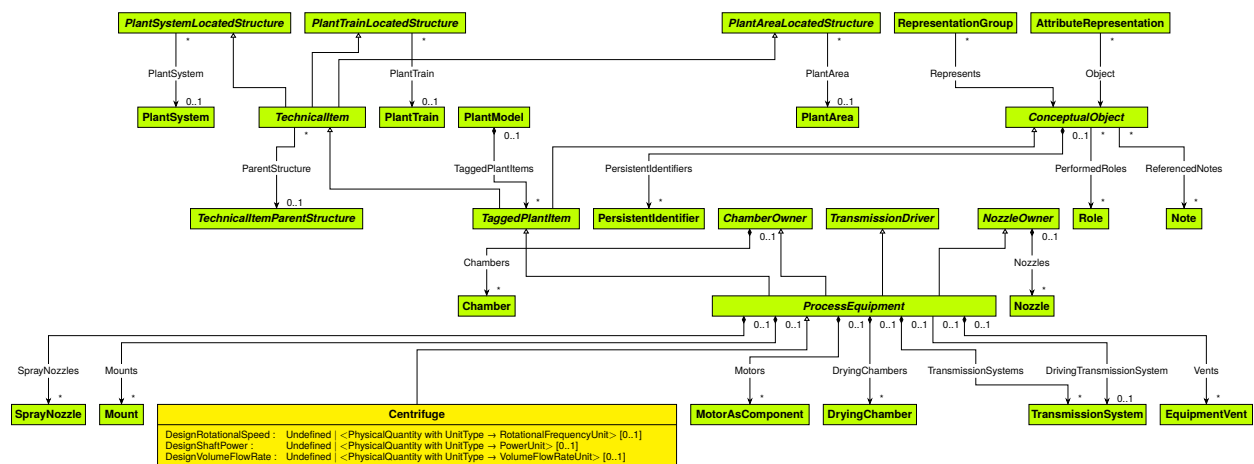
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS420974>

CENTRIFUGE

A 'separator' and 'machine' that uses centrifugal force to separate phases of different densities (from <http://data.posccaesar.org/rdl/RDS420974>).



Supertypes

- *ProcessEquipment*

Subtypes

- *FilteringCentrifuge*
- *SedimentalCentrifuge*

Members

Data Properties

Name	Multi- plicity	Type	Or- dered	Unique
<i>DesignRotational-Speed</i>	0..1	<i>Undefined</i> < <i>PhysicalQuantity</i> with <i>UnitType</i> → <i>Rotational-FrequencyUnit</i> >	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> < <i>PhysicalQuantity</i> with <i>UnitType</i> → <i>PowerUnit</i> >	no	no
<i>DesignVolume-FlowRate</i>	0..1	<i>Undefined</i> < <i>PhysicalQuantity</i> with <i>UnitType</i> → <i>Volume-FlowRateUnit</i> >	no	no

DesignRotationalSpeed

Data Property

The rotational speed for which the *Centrifuge* is designed.

Type: *Undefined* | <*PhysicalQuantity* with *UnitType* → *RotationalFrequencyUnit*>

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(
  Unit = ReciprocalMinute,
  Value = 180.0)
```

DesignShaftPower

Data Property

The shaft power for which the *Centrifuge* is designed.

Type: *Undefined* | <*PhysicalQuantity* with *UnitType* → *PowerUnit*>

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(
  Unit = Kilowatt,
  Value = 400.0)
```

DesignVolumeFlowRate**Data Property**

The volume flow rate for which the *Centrifuge* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VolumeFlowRateUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS14286227>

DESIGN VOLUME FLOW RATE

Example

```
PhysicalQuantity(
  Unit = MetreCubedPerHour,
  Value = 420.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Centrifuge">
  <Data property="DesignRotationalSpeed">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ReciprocalMinute"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignShaftPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>400.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignVolumeFlowRate">
```

(continues on next page)

(continued from previous page)

```
<AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
  <Data property="Unit">
    <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour
    ↪"/>
  </Data>
  <Data property="Value">
    <Double>420.0</Double>
  </Data>
</AggregatedDataValue>
</Data>
</Object>
```

Chamber

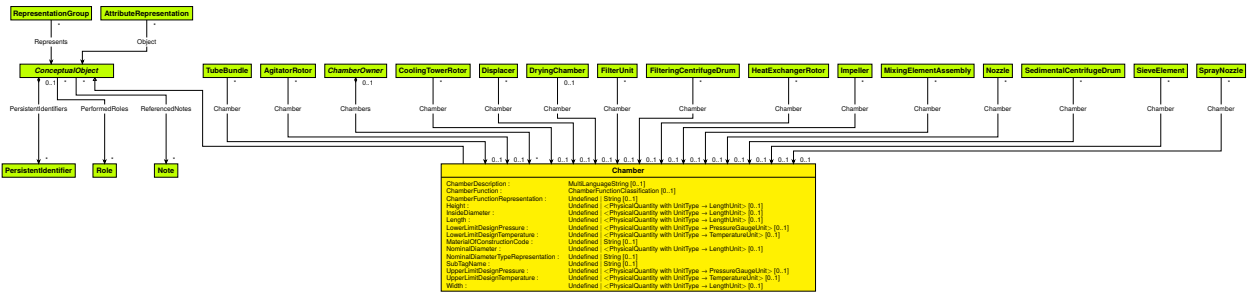
Plant::ProcessEquipment::Chamber

Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS903151421>

CHAMBER



Supertypes

- *ConceptualObject*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>ChamberDescription</i>	0..1	<i>MultiLanguageString</i>	no	no
<i>ChamberFunction</i>	0..1	<i>ChamberFunctionClassification</i>	no	no
<i>ChamberFunctionRepresentation</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>Height</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no
<i>InsideDiameter</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no
<i>Length</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no
<i>LowerLimitDesignPressure</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PressureGaugeUnit></i>	no	no
<i>LowerLimitDesignTemperature</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → TemperatureUnit></i>	no	no
<i>MaterialOfConstructionCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>NominalDiameter</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no
<i>NominalDiameterTypeRepresentation</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>SubTagName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>UpperLimitDesignPressure</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PressureGaugeUnit></i>	no	no
<i>UpperLimitDesignTemperature</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → TemperatureUnit></i>	no	no
<i>Width</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no

ChamberDescription

Data Property

The description of the *Chamber*.

Type: *MultiLanguageString*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ChamberDescriptionAssignmentClass>

CHAMBER DESCRIPTION ASSIGNMENT CLASS

Example

```
MultiLanguageString(
  SingleLanguageStrings = [
    SingleLanguageString(
```

```

    Language = "en",
    Value = "jacket chamber"))))

```

ChamberFunction

Data Property

A specialization indicating the function of the *Chamber*.

Type: *ChamberFunctionClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ChamberFunctionSpecialization>

CHAMBER FUNCTION SPECIALIZATION

Example

Heating

ChamberFunctionRepresentation

Data Property

A short textual description of the function of the *Chamber*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ChamberFunctionAssignmentClass>

CHAMBER FUNCTION ASSIGNMENT CLASS

Example

"cooling"

Height

Data Property

The height of the *Chamber*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with } \text{UnitType} \rightarrow \text{LengthUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS357704>

HEIGHT

Example

```
PhysicalQuantity(  
  Unit = Centimetre,  
  Value = 220.0)
```

InsideDiameter**Data Property**

The inside diameter of the *Chamber*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS357209>

INSIDE DIAMETER

Example

```
PhysicalQuantity(  
  Unit = Centimetre,  
  Value = 60.0)
```

Length**Data Property**

The length of the *Chamber*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS373094>

LENGTH

Example

```
PhysicalQuantity(  
  Unit = Centimetre,  
  Value = 160.0)
```

LowerLimitDesignPressure

Data Property

The lower limit for the pressure for which the *Chamber* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PressureGaugeUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS360674>

LOWER LIMIT DESIGN PRESSURE

Example

PhysicalQuantity(

Unit = *Bar*,

Value = -0.5)

LowerLimitDesignTemperature

Data Property

The lower limit for the temperature for which the *Chamber* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{TemperatureUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS360494>

LOWER LIMIT DESIGN TEMPERATURE

Example

PhysicalQuantity(

Unit = *DegreeCelsius*,

Value = -45.0)

MaterialOfConstructionCode

Data Property

A code that gives the material of construction of the *Chamber*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS1460719741>

MATERIAL OF CONSTRUCTION CODE ASSIGNMENT CLASS

Example

"1.4306"

NominalDiameter

Data Property

The nominal diameter of the *Chamber*, given as a length. See also *NominalDiameterTypeRepresentation*.

Type: *Undefined* | *<PhysicalQuantity* with *UnitType* → *LengthUnit**>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS366794>

NOMINAL DIAMETER

Example

PhysicalQuantity(
 Unit = *Centimetre*,
 Value = 80.0)

NominalDiameterTypeRepresentation

Data Property

A readable representation of the type or unit of measure of the nominal diameter of the *Chamber*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalDiameterTypeRepresentationAssignmentClass>

NOMINAL DIAMETER TYPE REPRESENTATION ASSIGNMENT CLASS

Example

"DN"

SubTagName

Data Property

The sub tag name of the *Chamber*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SubTagNameAssignmentClass>

SUB TAG NAME ASSIGNMENT CLASS

Example

"ST1"

UpperLimitDesignPressure

Data Property

The upper limit for the pressure for which the *Chamber* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PressureGaugeUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS1470835011>

UPPER LIMIT DESIGN PRESSURE

Example

PhysicalQuantity(

Unit = Bar,

Value = 60.0)

UpperLimitDesignTemperature

Data Property

The upper limit for the temperature for which the *Chamber* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → TemperatureUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS360449>

UPPER LIMIT DESIGN TEMPERATURE

Example

PhysicalQuantity(

Unit = DegreeCelsius,

Value = 100.0)

Width

Data Property

The width of the *Chamber*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS361709>

WIDTH

Example

```
PhysicalQuantity(  
  Unit = Centimetre,  
  Value = 180.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Chamber">  
  <Data property="ChamberDescription">  
    <AggregatedDataValue type="Core/DataTypes.MultiLanguageString">  
      <Data property="SingleLanguageStrings">  
        <AggregatedDataValue type="Core/DataTypes.SingleLanguageString">  
          <Data property="Language">  
            <String>en</String>  
          </Data>  
          <Data property="Value">  
            <String>jacket chamber</String>  
          </Data>  
        </AggregatedDataValue>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
  <Data property="ChamberFunction">  
    <DataReference data="Plant/Enumerations.ChamberFunctionClassification.Heating"/>  
  </Data>  
  <Data property="ChamberFunctionRepresentation">  
    <String>cooling</String>  
  </Data>  
  <Data property="Height">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>  
      </Data>  
      <Data property="Value">  
        <Double>220.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
</Object>
```

(continues on next page)

(continued from previous page)

```

</Data>
<Data property="InsideDiameter">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
    </Data>
    <Data property="Value">
      <Double>60.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="Length">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
    </Data>
    <Data property="Value">
      <Double>160.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="LowerLimitDesignPressure">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.PressureGaugeUnit.Bar"/>
    </Data>
    <Data property="Value">
      <Double>-0.5</Double>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="LowerLimitDesignTemperature">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.TemperatureUnit.DegreeCelsius"/>
    </Data>
    <Data property="Value">
      <Double>-45.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="MaterialOfConstructionCode">
  <String>1.4306</String>
</Data>
<Data property="NominalDiameter">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
    </Data>
    <Data property="Value">
      <Double>80.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>

```

(continues on next page)

(continued from previous page)

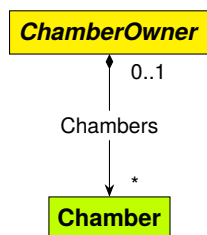
```

    </AggregatedDataValue>
  </Data>
  <Data property="NominalDiameterTypeRepresentation">
    <String>DN</String>
  </Data>
  <Data property="SubTagName">
    <String>ST1</String>
  </Data>
  <Data property="UpperLimitDesignPressure">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PressureGaugeUnit.Bar"/>
      </Data>
      <Data property="Value">
        <Double>60.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="UpperLimitDesignTemperature">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.TemperatureUnit.DegreeCelsius"/>
      </Data>
      <Data property="Value">
        <Double>100.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="Width">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>

```

ChamberOwner*Plant::ProcessEquipment::ChamberOwner***Overview****Abstract Class**

An object that can have chambers.



Subtypes

- *ProcessEquipment*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Chambers</i>	*	<i>Chamber</i>	no

Chambers

Composition Property

The Chambers of the ChamberOwner.

Type: *Chamber*

Multiplicity: *

Ordered: no

Chimney

Plant::ProcessEquipment::Chimney

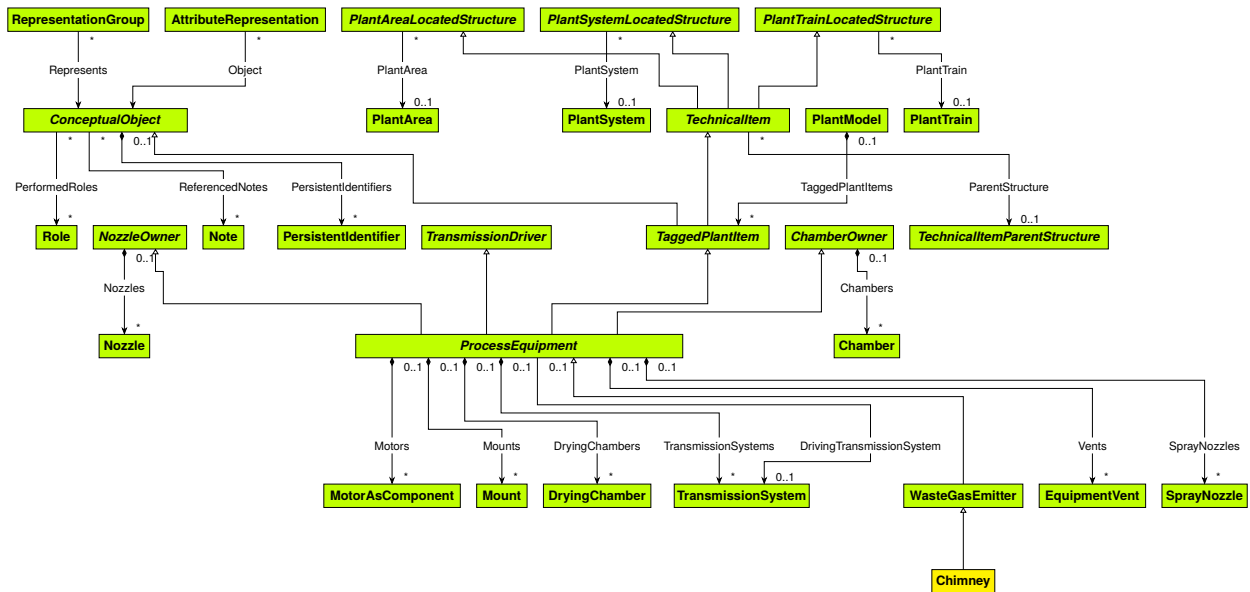
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/Chimney>

CHIMNEY

A *WasteGasEmitter* that is intended to transport waste gas to a high location in the atmosphere.



Supertypes

- *WasteGasEmitter*

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Chimney"/>
```

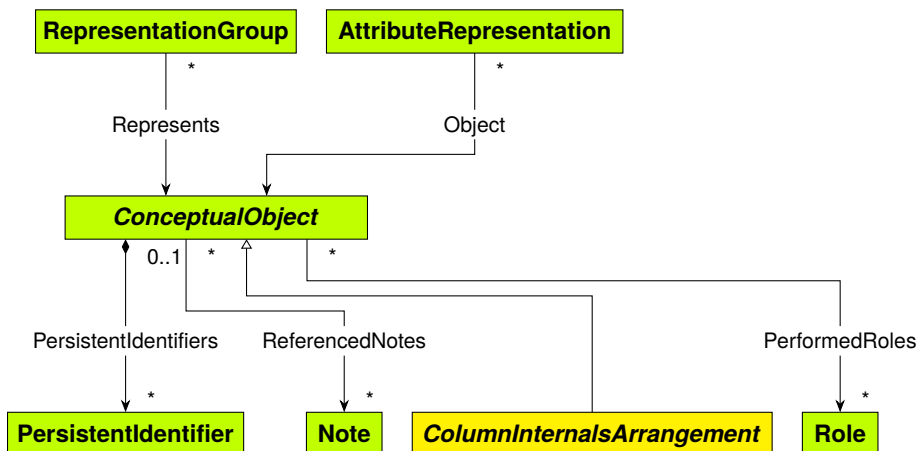
ColumnInternalsArrangement

Plant::ProcessEquipment::ColumnInternalsArrangement

Overview

Abstract Class

The internals of a column, e.g., trays or packings.



Supertypes

- *ConceptualObject*

Subtypes

- *ColumnPackingsArrangement*
- *ColumnTraysArrangement*

ColumnPackingsArrangement

Plant::ProcessEquipment::ColumnPackingsArrangement

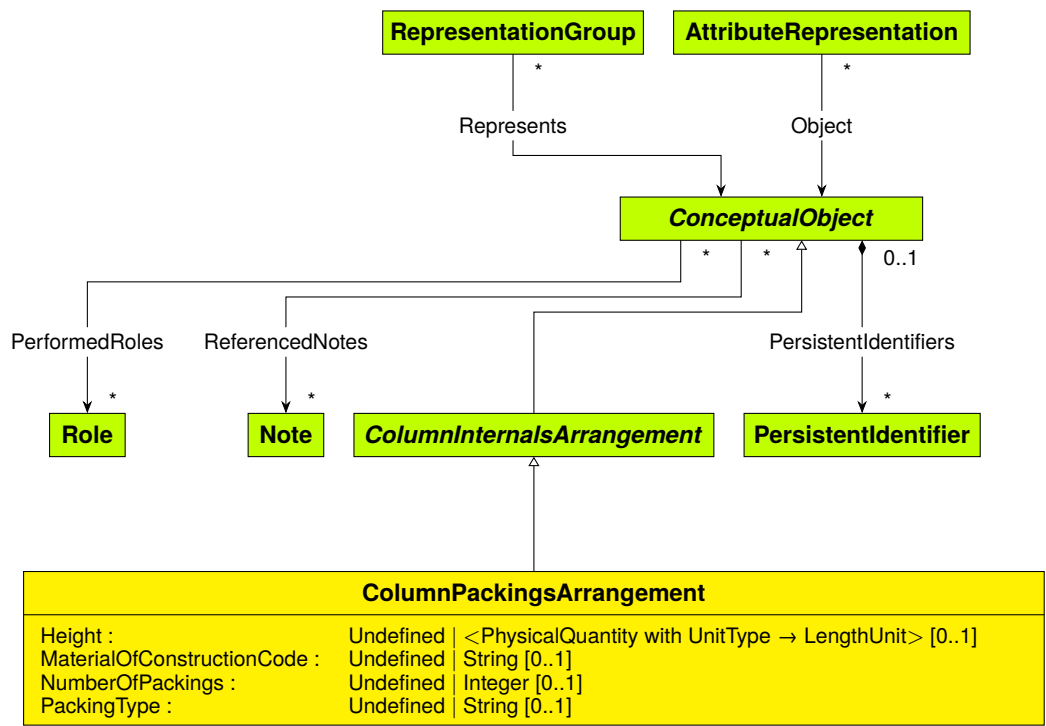
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/ColumnPackingsArrangement>

COLUMN PACKINGS ARRANGEMENT

The packings of a column.



Supertypes

- *ColumnInternalsArrangement*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Height</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no
<i>MaterialOfConstruction-Code</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>NumberOfPackings</i>	0..1	<i>Undefined</i> <i>Integer</i>	no	no
<i>PackingType</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

Height

Data Property

The height of the *ColumnPackingsArrangement*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → LengthUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS357704>

HEIGHT

Example

```
PhysicalQuantity(  
  Unit = Centimetre,  
  Value = 220.0)
```

MaterialOfConstructionCode

Data Property

A code that gives the material of construction of the *ColumnPackingsArrangement*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS1460719741>

MATERIAL OF CONSTRUCTION CODE ASSIGNMENT CLASS

Example

```
"1.4306"
```

NumberOfPackings

Data Property

The number of packings in the *ColumnPackingsArrangement*.

Type: *Undefined* | *Integer*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NumberOfPackings>

NUMBER OF PACKINGS

Example

300

PackingType

Data Property

The type of the packings in the *ColumnPackingsArrangement*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PackingTypeAssignmentClass>

PACKING TYPE ASSIGNMENT CLASS

Example

"rings"

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.ColumnPackingsArrangement">
  <Data property="Height">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
      </Data>
      <Data property="Value">
        <Double>220.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="MaterialOfConstructionCode">
    <String>1.4306</String>
  </Data>
```

(continues on next page)

(continued from previous page)

```

<Data property="NumberOfPackings">
  <Integer>300</Integer>
</Data>
<Data property="PackingType">
  <String>rings</String>
</Data>
</Object>

```

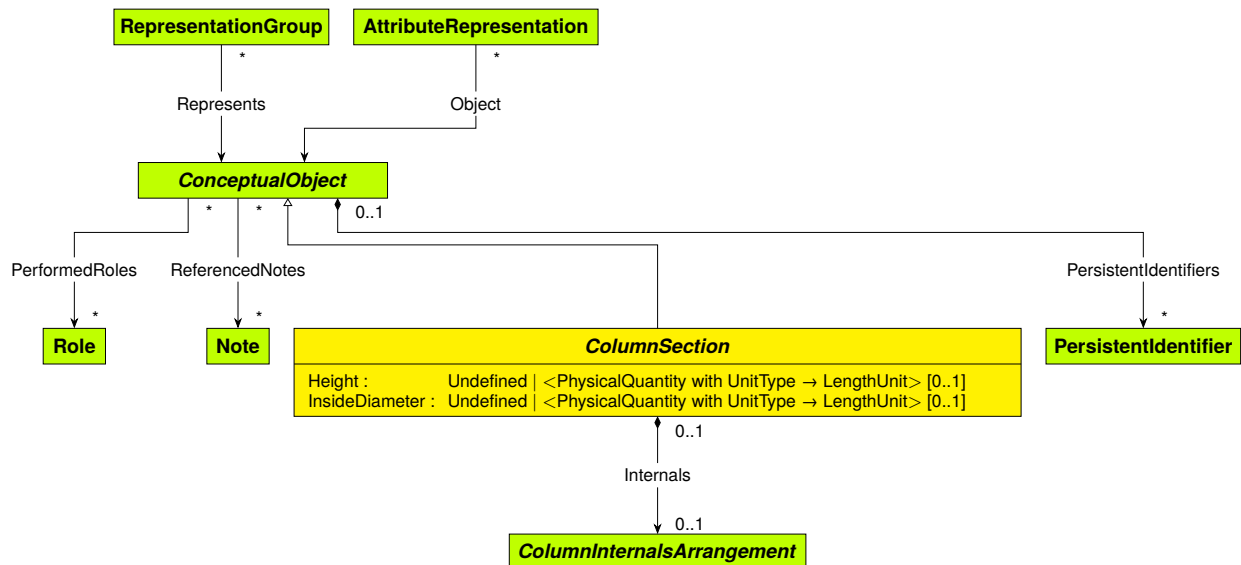
ColumnSection

Plant::ProcessEquipment::ColumnSection

Overview

Abstract Class

A column section.



Supertypes

- *ConceptualObject*

Subtypes

- *SubTaggedColumnSection*
- *TaggedColumnSection*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Internals</i>	0..1	<i>ColumnInternalsArrangement</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Height</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no
<i>InsideDiameter</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no

Height

Data Property

The height of the *ColumnSection*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → LengthUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS357704>

HEIGHT

Example

```
PhysicalQuantity(
  Unit = Centimetre,
  Value = 220.0)
```

InsideDiameter

Data Property

The inside diameter of the *ColumnSection*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → LengthUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS357209>

INSIDE DIAMETER

Example

```
PhysicalQuantity(
  Unit = Centimetre,
  Value = 60.0)
```

Internals

Composition Property

The *ColumnInternalsArrangement* of the ColumnSection.

Type: *ColumnInternalsArrangement*

Multiplicity: 0..1

Ordered: no

ColumnTraysArrangement

Plant::ProcessEquipment::ColumnTraysArrangement

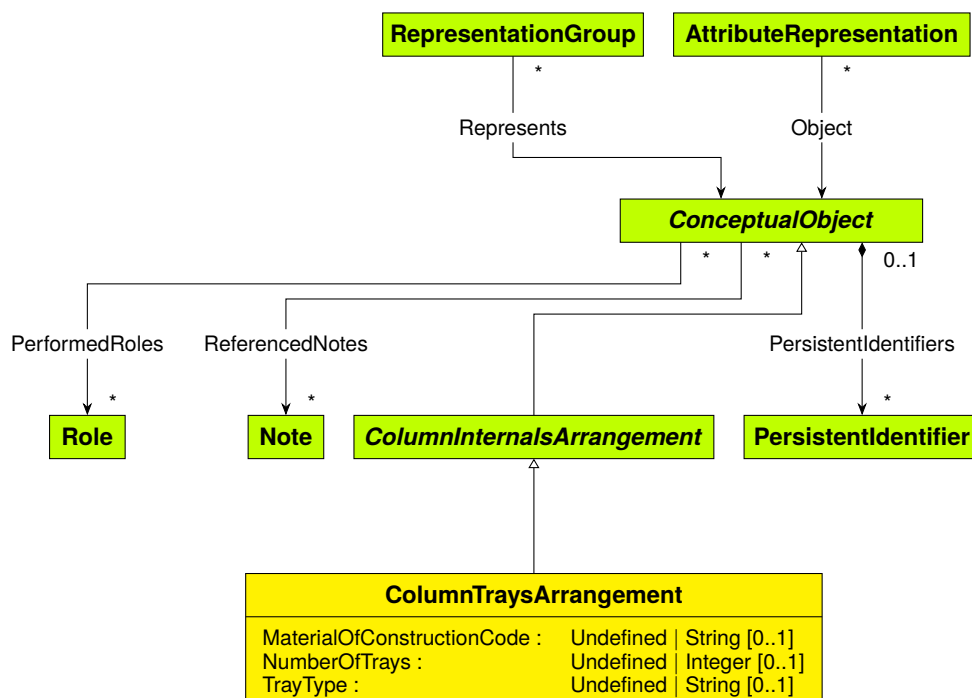
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/ColumnTraysArrangement>

COLUMN TRAYS ARRANGEMENT

The trays of a column.



Supertypes

- *ColumnInternalsArrangement*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>MaterialOfConstructionCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>NumberOfTrays</i>	0..1	<i>Undefined</i> <i>Integer</i>	no	no
<i>TrayType</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

MaterialOfConstructionCode

Data Property

A code that gives the material of construction of the *ColumnTraysArrangement*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS1460719741>

MATERIAL OF CONSTRUCTION CODE ASSIGNMENT CLASS

Example

"1.4306"

NumberOfTrays

Data Property

The number of trays in the *ColumnTraysArrangement*.

Type: *Undefined* | *Integer*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NumberOfTrays>

NUMBER OF TRAYS

Example

16

TrayType

Data Property

The type of the trays in the *ColumnTraysArrangement*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/TrayTypeAssignmentClass>

TRAY TYPE ASSIGNMENT CLASS

Example

"sieve trays"

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.ColumnTraysArrangement">
  <Data property="MaterialOfConstructionCode">
    <String>1.4306</String>
  </Data>
  <Data property="NumberOfTrays">
    <Integer>16</Integer>
  </Data>
  <Data property="TrayType">
    <String>sieve trays</String>
  </Data>
</Object>
```

CombustionEngine

Plant::ProcessEquipment::CombustionEngine

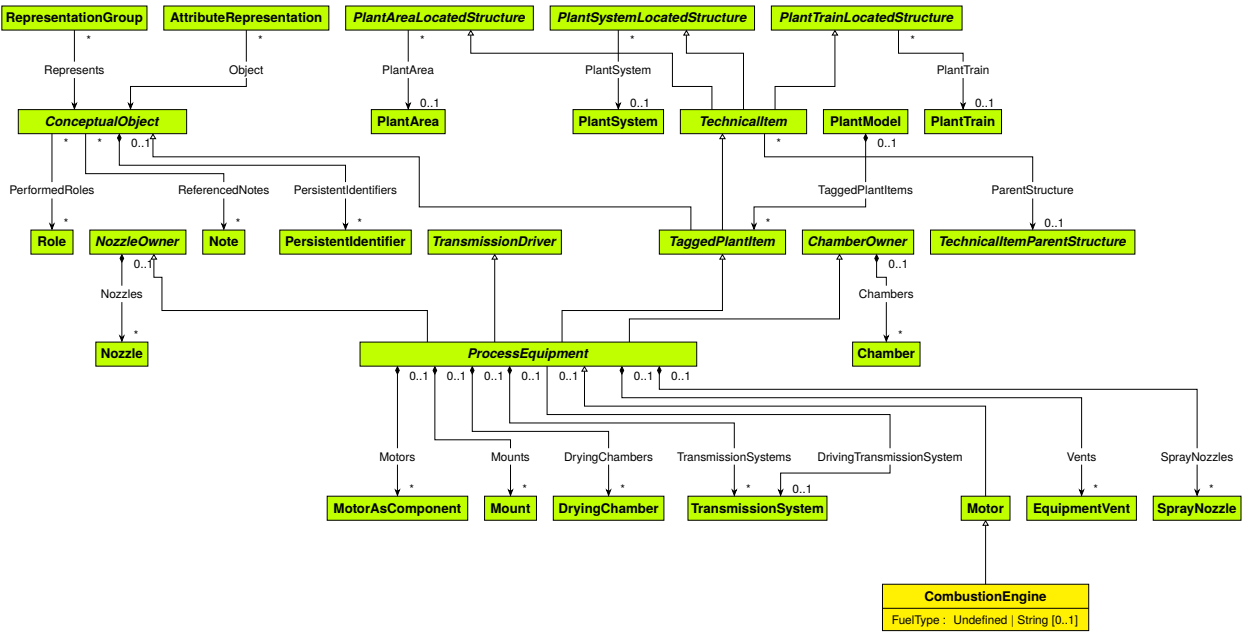
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS1083734>

COMBUSTION ENGINE

An engine intended to deliver power by means of burning fuels (from <http://data.posccaesar.org/rdl/RDS1083734>).



Supertypes

- *Motor*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>FuelType</i>	0..1	<i>Undefined String</i>	no	no

FuelType

Data Property

The fuel type of the *CombustionEngine*.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/FuelType>

FUEL TYPE

Example

"Diesel fuel"

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.CombustionEngine">
  <Data property="FuelType">
    <String>Diesel fuel</String>
  </Data>
</Object>
```

CombustionEngineAsComponent

Plant::ProcessEquipment::CombustionEngineAsComponent

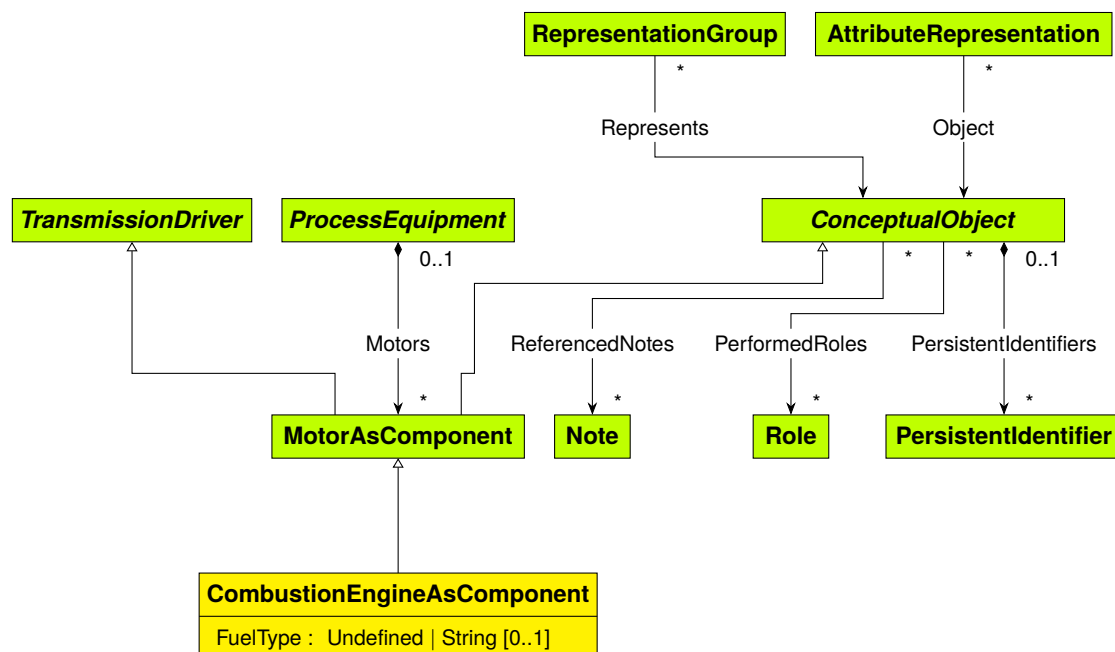
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/CombustionEngineAsComponent>

COMBUSTION ENGINE AS COMPONENT

An engine intended to deliver power by means of burning fuels that is used as component of an apparatus or of a machine.



Supertypes

- *MotorAsComponent*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>FuelType</i>	0..1	<i>Undefined String</i>	no	no

FuelType

Data Property

The fuel type of the *CombustionEngineAsComponent*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/FuelType>

FUEL TYPE

Example

"Diesel fuel"

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.CombustionEngineAsComponent">
  <Data property="FuelType">
    <String>Diesel fuel</String>
  </Data>
</Object>
```

Compressor

Plant::ProcessEquipment::Compressor

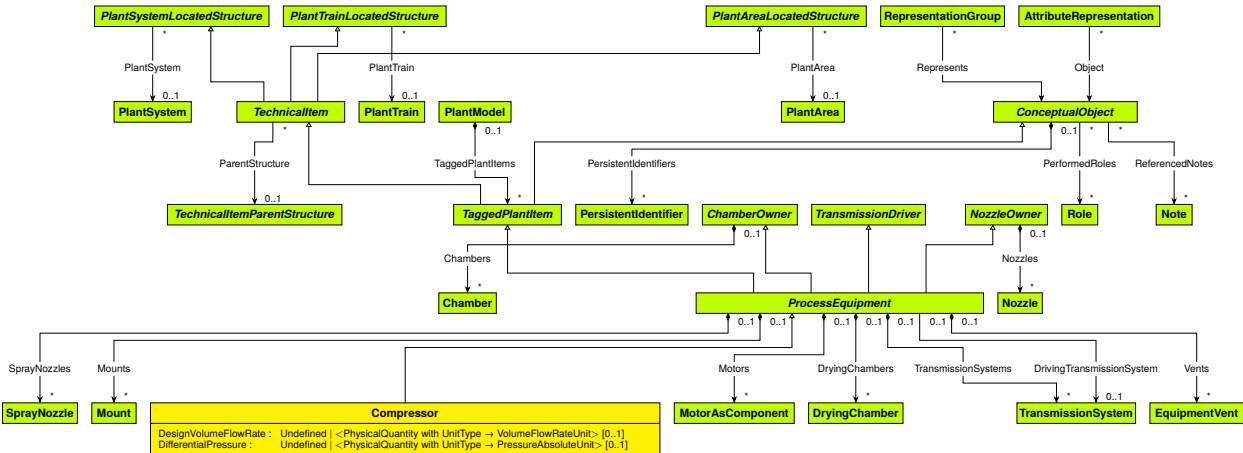
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS14286497>

COMPRESSOR

A machine that has the capability of compressing a gas.



Supertypes

- *ProcessEquipment*

Subtypes

- *AirEjector*
- *AxialCompressor*
- *CentrifugalCompressor*
- *ReciprocatingCompressor*
- *RotaryCompressor*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignVolumeFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VolumeFlowRateUnit></i>	no	no
<i>DifferentialPressure</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PressureAbsoluteUnit></i>	no	no

DesignVolumeFlowRate

Data Property

The volume flow rate for which the *Compressor* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → VolumeFlowRateUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS14286227>

DESIGN VOLUME FLOW RATE

Example

```
PhysicalQuantity(  
    Unit = MetreCubedPerHour,  
    Value = 420.0)
```

DifferentialPressure

Data Property

The differential pressure of the *Compressor*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PressureAbsoluteUnit>*

Multiplicity: 0..1**Ordered:** no**Unique:** no**RDL:** <http://data.posccaesar.org/rdl/RDS361574>

DIFFERENTIAL PRESSURE

Example*PhysicalQuantity*(*Unit* = Bar,*Value* = 4.8)**Example in DEXPI XML**

```

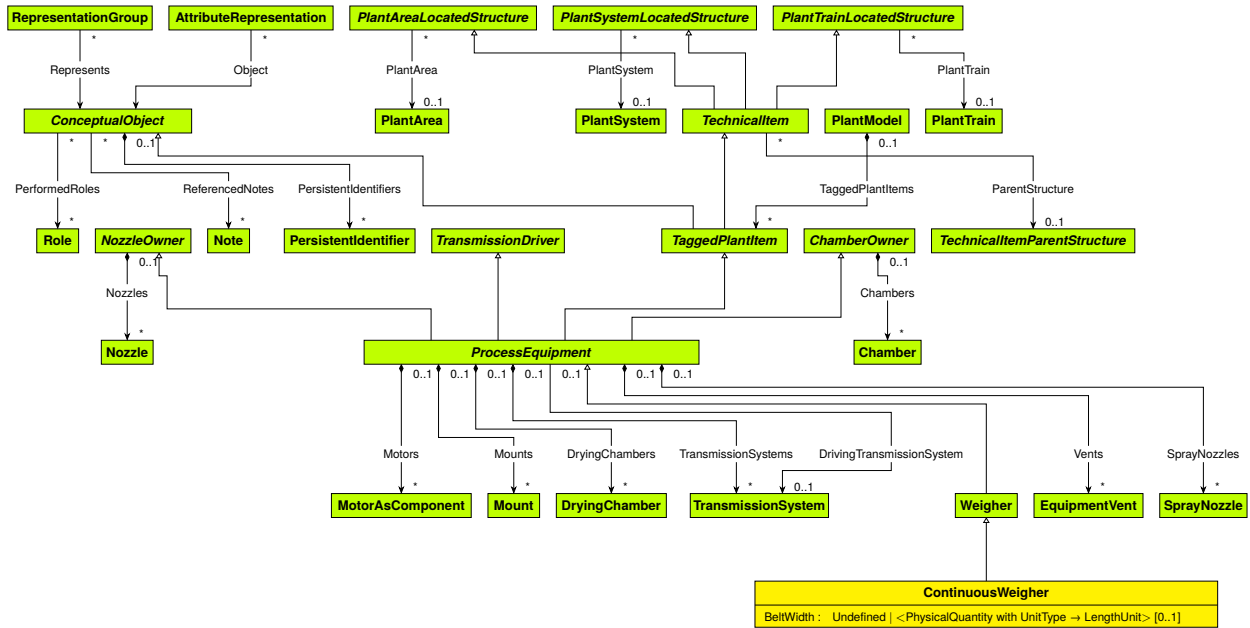
<Object type="Plant/ProcessEquipment.Compressor">
  <Data property="DesignVolumeFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour
→"/>
      </Data>
      <Data property="Value">
        <Double>420.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DifferentialPressure">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PressureAbsoluteUnit.Bar"/>
      </Data>
      <Data property="Value">
        <Double>4.8</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>

```

ContinuousWeigher*Plant::ProcessEquipment::ContinuousWeigher***Overview****Concrete Class****RDL:** <http://sandbox.dexpi.org/rdl/ContinuousWeigher>

CONTINUOUS WEIGHER

A *Weigher* that weighs a mass flow rate in continuous mode.



Supertypes

- *Weigher*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>BeltWidth</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no

BeltWidth

Data Property

The belt width of the *ContinuousWeigher*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → LengthUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/BeltWidth>

BELT WIDTH

Example

```
PhysicalQuantity(
  Unit = Centimetre,
  Value = 180.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.ContinuousWeigher">
  <Data property="BeltWidth">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

ConvectionDryer

Plant::ProcessEquipment::ConvectionDryer

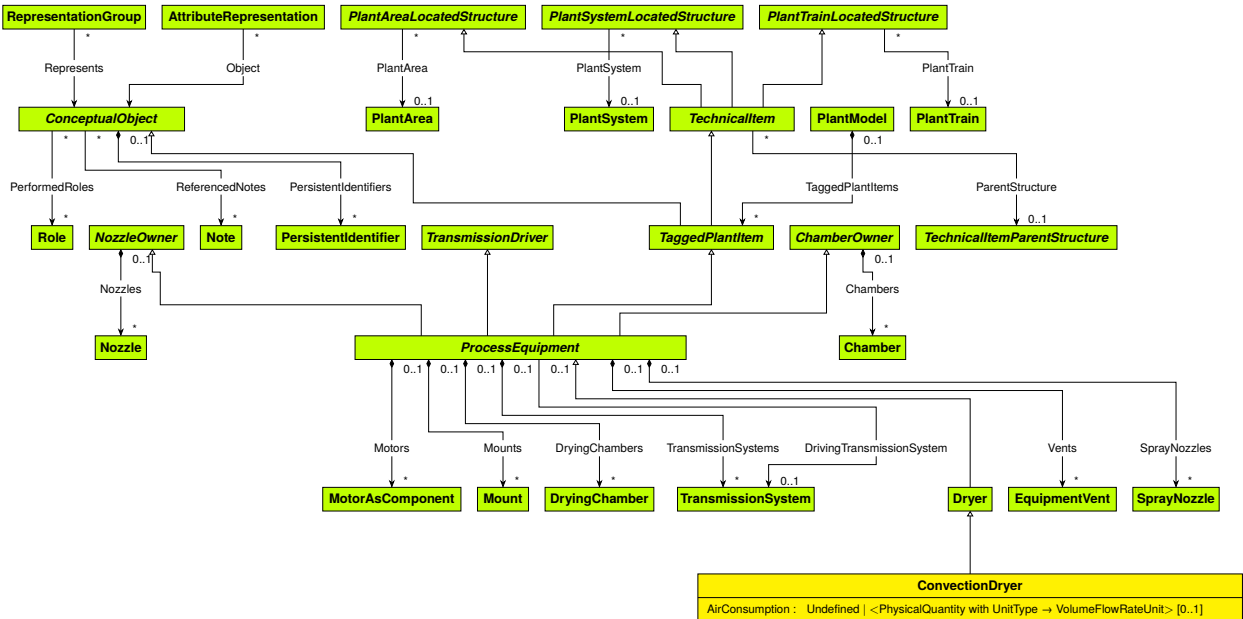
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/ConvectionDryer>

CONVECTION DRYER

A *Dryer* that dries a material by bringing it in contact with a drying gas.



Supertypes

- *Dryer*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>AirConsumption</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VolumeFlowRateUnit></i>	no	no

AirConsumption

Data Property

The consumed air flow of the *ConvectionDryer*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → VolumeFlowRateUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS5875300>

AIR CONSUMPTION

Example

```
PhysicalQuantity(
    Unit = MetreCubedPerHour,
    Value = 40.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.ConvectionDryer">
  <Data property="AirConsumption">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour
↪"/>
      </Data>
      <Data property="Value">
        <Double>40.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

Conveyor

Plant::ProcessEquipment::Conveyor

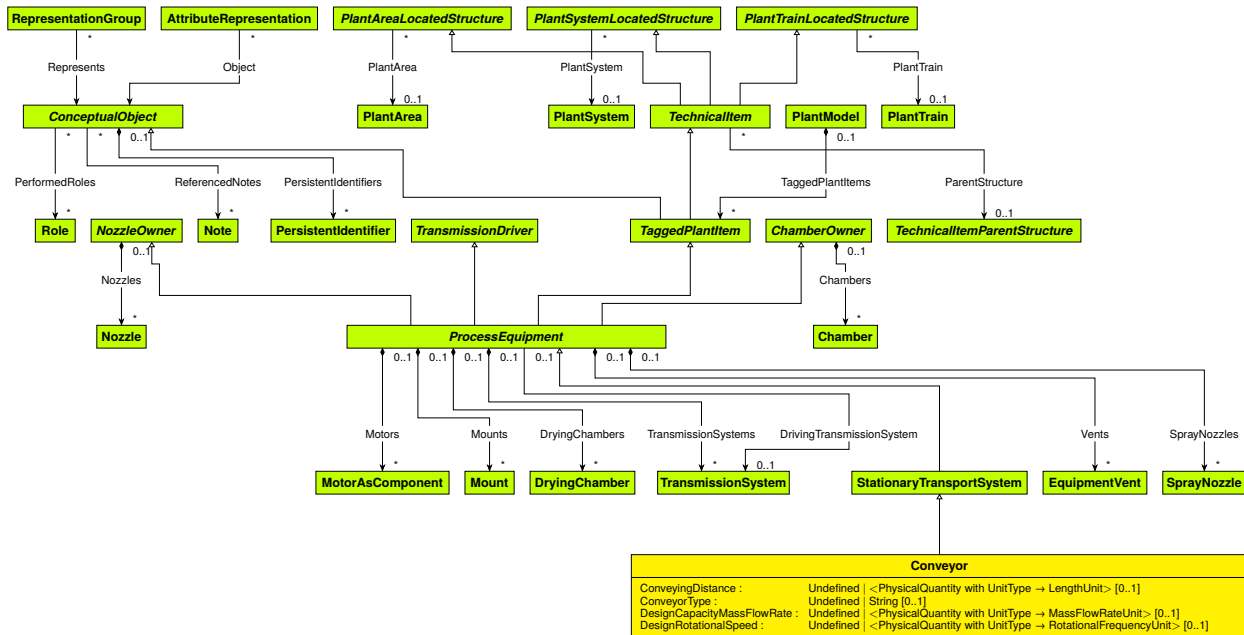
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS11589895>

CONVEYOR

A machine that is capable of conveying material.



Supertypes

- *StationaryTransportSystem*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>ConveyingDistance</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no
<i>ConveyorType</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>DesignCapacityMassFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → MassFlowRateUnit></i>	no	no
<i>DesignRotationalSpeed</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → RotationalFrequencyUnit></i>	no	no

ConveyingDistance

Data Property

The conveying distance of the *Conveyor*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → LengthUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ConveyingDistance>

CONVEYING DISTANCE

Example

PhysicalQuantity(

Unit = Metre,

Value = 12.0)

ConveyorType

Data Property

The type of the conveyor.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ConveyorTypeAssignmentClass>

CONVEYOR TYPE ASSIGNMENT CLASS

Example

"Chain Conveyor"

DesignCapacityMassFlowRate

Data Property

The capacity for the mass flow rate for which the *Conveyor* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → MassFlowRateUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignCapacityMassFlowRate>

DESIGN CAPACITY MASS FLOW RATE

Example

PhysicalQuantity(

Unit = KilogramPerMinute,

Value = 40.0)

DesignRotationalSpeed

Data Property

The rotational speed for which the *Conveyor* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{RotationalFrequencyUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(
    Unit = ReciprocalMinute,
    Value = 180.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Conveyor">
  <Data property="ConveyingDistance">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Metre"/>
      </Data>
      <Data property="Value">
        <Double>12.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="ConveyorType">
    <String>Chain Conveyor</String>
  </Data>
  <Data property="DesignCapacityMassFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerMinute"/>
      </Data>
      <Data property="Value">
        <Double>40.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignRotationalSpeed">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
ReciprocalMinute"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

(continues on next page)

(continued from previous page)

```

</Data>
<Data property="Value">
  <Double>180.0</Double>
</Data>
</AggregatedDataValue>
</Data>
</Object>

```

CoolingTower

Plant::ProcessEquipment::CoolingTower

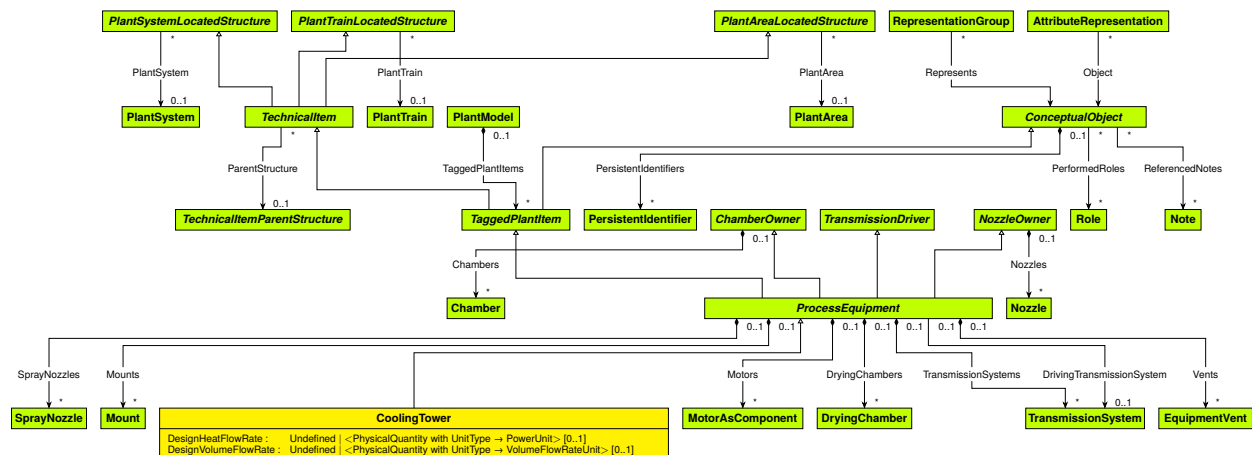
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS14072341>

COOLING TOWER

A cooler and an air cooled heat exchanger that is a tall structure through which air circulates by convection



Supertypes

- *ProcessEquipment*

Subtypes

- *DryCoolingTower*
- *SprayCooler*
- *WetCoolingTower*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignHeatFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no
<i>DesignVolumeFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VolumeFlowRateUnit></i>	no	no

DesignHeatFlowRate

Data Property

The heat flow rate for which the *CoolingTower* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PowerUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignHeatFlowRate>

DESIGN HEAT FLOW RATE

Example

```
PhysicalQuantity(
  Unit = Kilowatt,
  Value = 313.0)
```

DesignVolumeFlowRate

Data Property

The volume flow rate for which the *CoolingTower* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → VolumeFlowRateUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS14286227>

DESIGN VOLUME FLOW RATE

Example

```
PhysicalQuantity(
  Unit = MetreCubedPerHour,
  Value = 420.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.CoolingTower">
  <Data property="DesignHeatFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>313.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignVolumeFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour
↪"/>
      </Data>
      <Data property="Value">
        <Double>420.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

CoolingTowerRotor

Plant::ProcessEquipment::CoolingTowerRotor

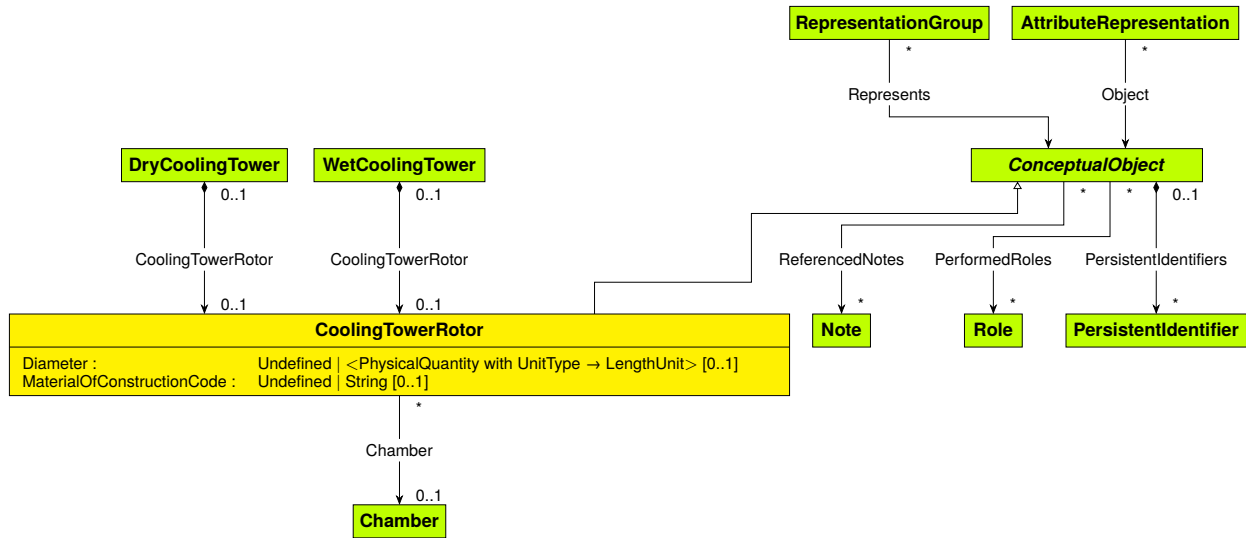
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/CoolingTowerRotor>

COOLING TOWER ROTOR

A rotor of a cooling tower.



Supertypes

- *ConceptualObject*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Chamber</i>	0..1	<i>Chamber</i>	no	no	*

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Diameter</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no
<i>MaterialOfConstruction-Code</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

Chamber

Reference Property

The *Chamber* in which the **CoolingTowerRotor** is located, if applicable. The Chamber must be a component of the same object as the **CoolingTowerRotor**.

Type: *Chamber*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

Diameter

Data Property

The diameter of the *CoolingTowerRotor*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS350954>

DIAMETER

Example

```
PhysicalQuantity(  
    Unit = Centimetre,  
    Value = 20.0)
```

MaterialOfConstructionCode

Data Property

A code that gives the material of construction of the *CoolingTowerRotor*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS1460719741>

MATERIAL OF CONSTRUCTION CODE ASSIGNMENT CLASS

Example

"1.4306"

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.CoolingTowerRotor">  
  <Data property="Diameter">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>  
      </Data>  
      <Data property="Value">  
        <Double>20.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>
```

(continues on next page)

(continued from previous page)

```
<Data property="MaterialOfConstructionCode">
  <String>1.4306</String>
</Data>
</Object>
```

Crusher

Plant::ProcessEquipment::Crusher

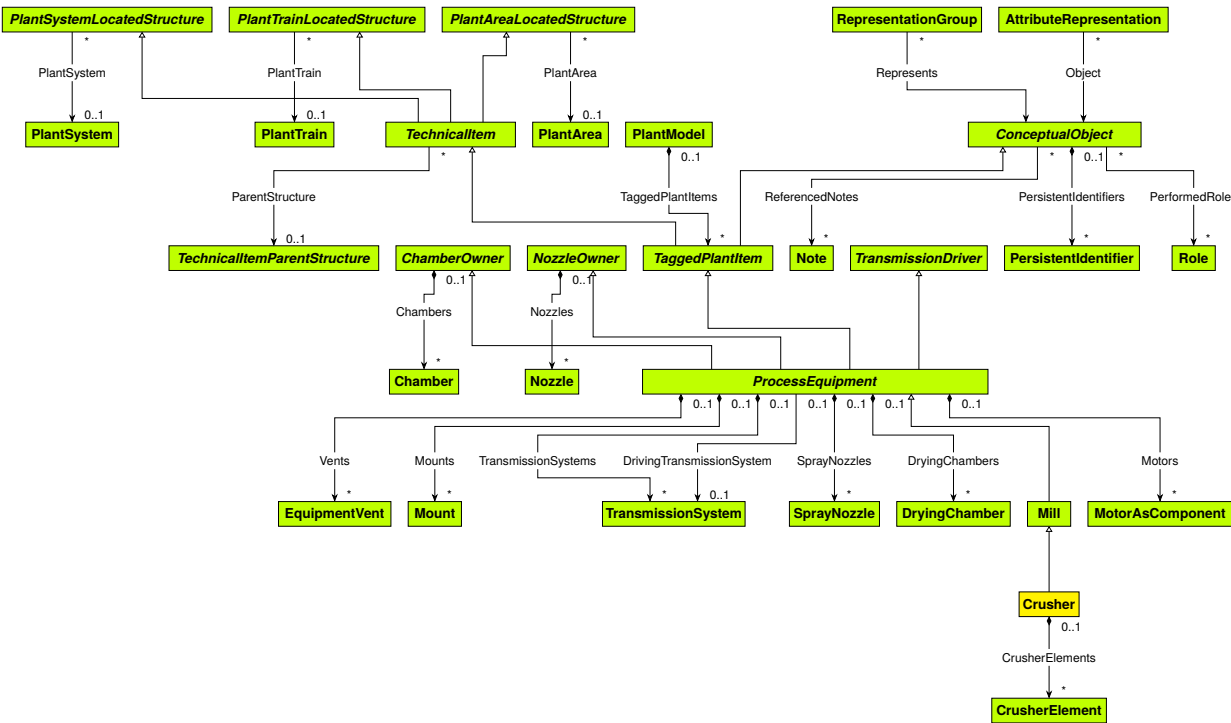
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS11589940>

CRUSHER

A mill that uses pressure or impact to reduce the particle size of solid materials (from <http://data.posccaesar.org/rdl/RDS11589940>).



Supertypes

- *Mill*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>CrusherElements</i>	*	<i>CrusherElement</i>	no

CrusherElements

Composition Property

The crusher elements of the Crusher.

Type: *CrusherElement*

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Crusher"/>
```

CrusherElement

Plant::ProcessEquipment::CrusherElement

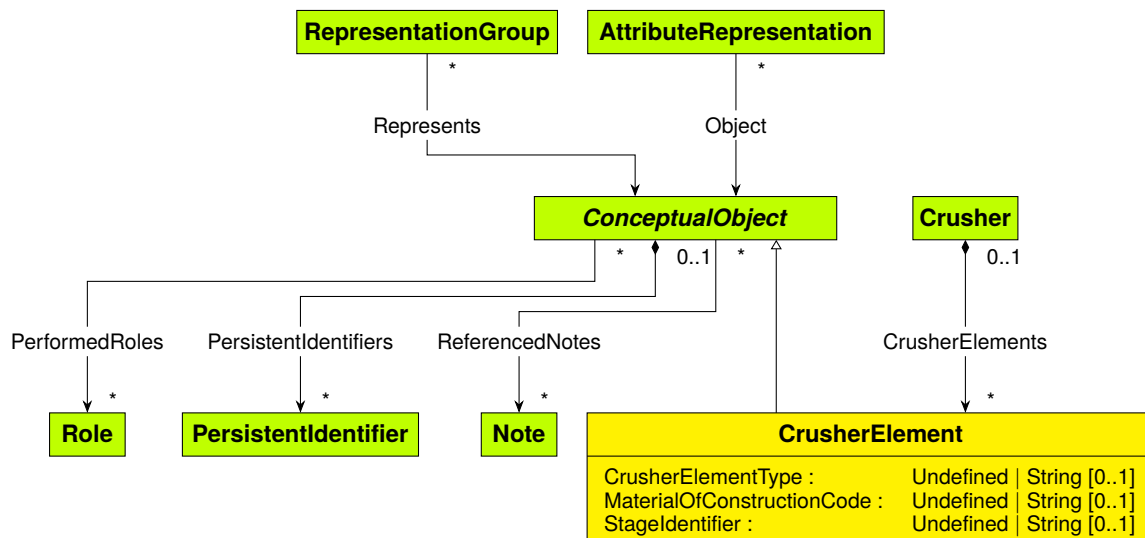
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/CrusherUnit>

CRUSHER UNIT

A functional component of a *Crusher*.



Supertypes

- *ConceptualObject*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>CrusherElementType</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>MaterialOfConstructionCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>StageIdentifier</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

CrusherElementType

Data Property

The type of the *CrusherElement*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/CrusherElementTypeAssignmentClass>

CRUSHER ELEMENT TYPE ASSIGNMENT CLASS

Example

"Cone Crusher"

MaterialOfConstructionCode

Data Property

A code that gives the material of construction of the *CrusherElement*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS1460719741>

MATERIAL OF CONSTRUCTION CODE ASSIGNMENT CLASS

Example

"1.4306"

StageIdentifier

Data Property

The stage identifier of the *CrusherElement*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/StageIdentifierAssignmentClass>

STAGE IDENTIFIER ASSIGNMENT CLASS

Example

"s1"

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.CrusherElement">
  <Data property="CrusherElementType">
    <String>Cone Crusher</String>
  </Data>
  <Data property="MaterialOfConstructionCode">
    <String>1.4306</String>
  </Data>
  <Data property="StageIdentifier">
    <String>s1</String>
  </Data>
</Object>
```

DirectCurrentGenerator

Plant::ProcessEquipment::DirectCurrentGenerator

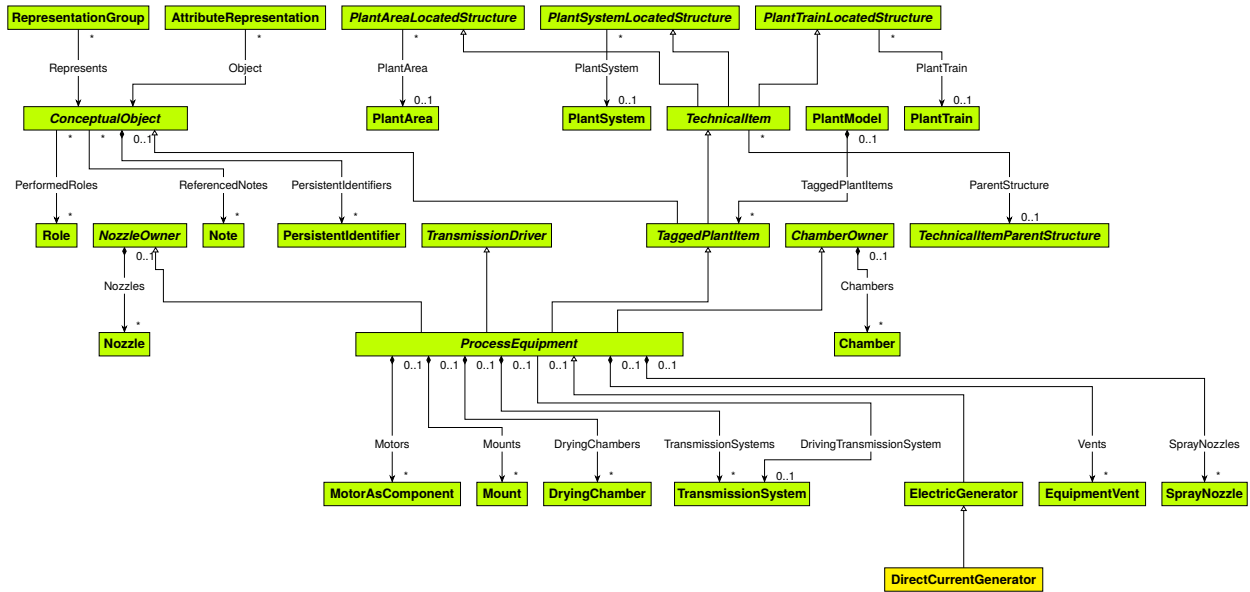
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/DirectCurrentGenerator>

DIRECT CURRENT GENERATOR

An *ElectricGenerator* and current generator for the production of direct current (DC).



Supertypes

- *ElectricGenerator*

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.DirectCurrentGenerator"/>
```

DirectCurrentMotor

Plant::ProcessEquipment::DirectCurrentMotor

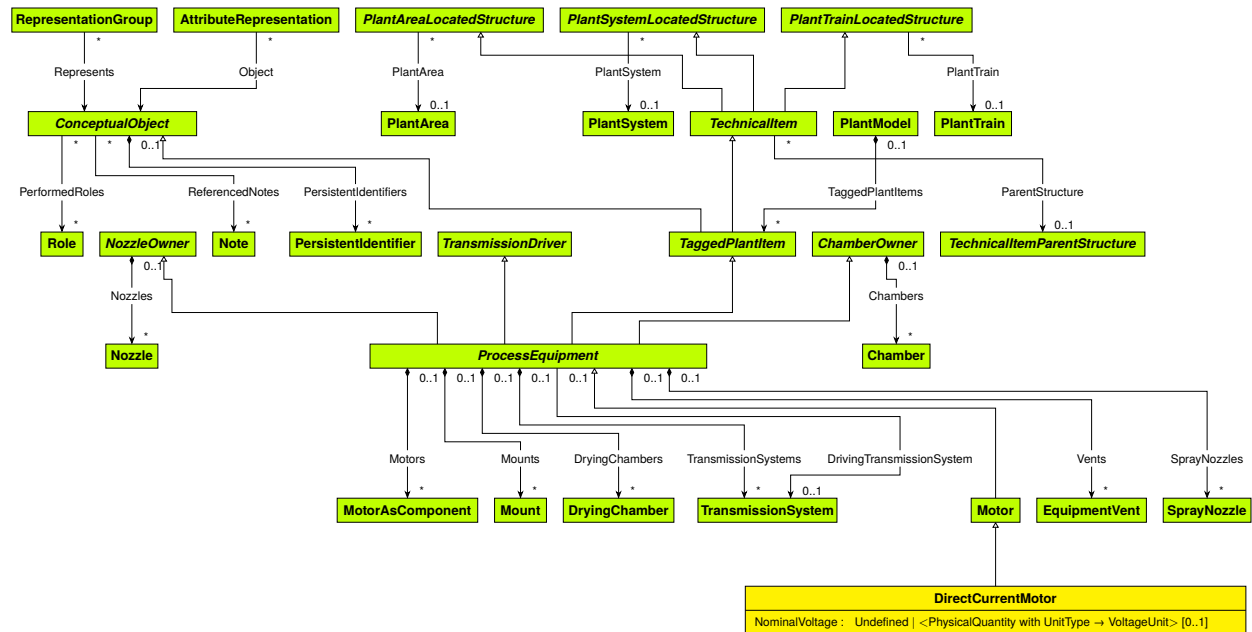
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS472949>

DIRECT CURRENT MOTOR

An electric motor for operation by direct current (from <http://data.posccaesar.org/rdl/RDS472949>).



Supertypes

- *Motor*

Members

Data Properties

Name	Multiplic-ity	Type	Or-dered	Unique
<i>NominalVoltage</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VoltageUnit></i>	no	no

NominalVoltage

Data Property

The nominal voltage of the *DirectCurrentMotor*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → VoltageUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS369449>

NOMINAL VOLTAGE

Example

```
PhysicalQuantity(
  Unit = Volt,
  Value = 230.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.DirectCurrentMotor">
  <Data property="NominalVoltage">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.VoltageUnit.Volt"/>
      </Data>
      <Data property="Value">
        <Double>230.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

DirectCurrentMotorAsComponent

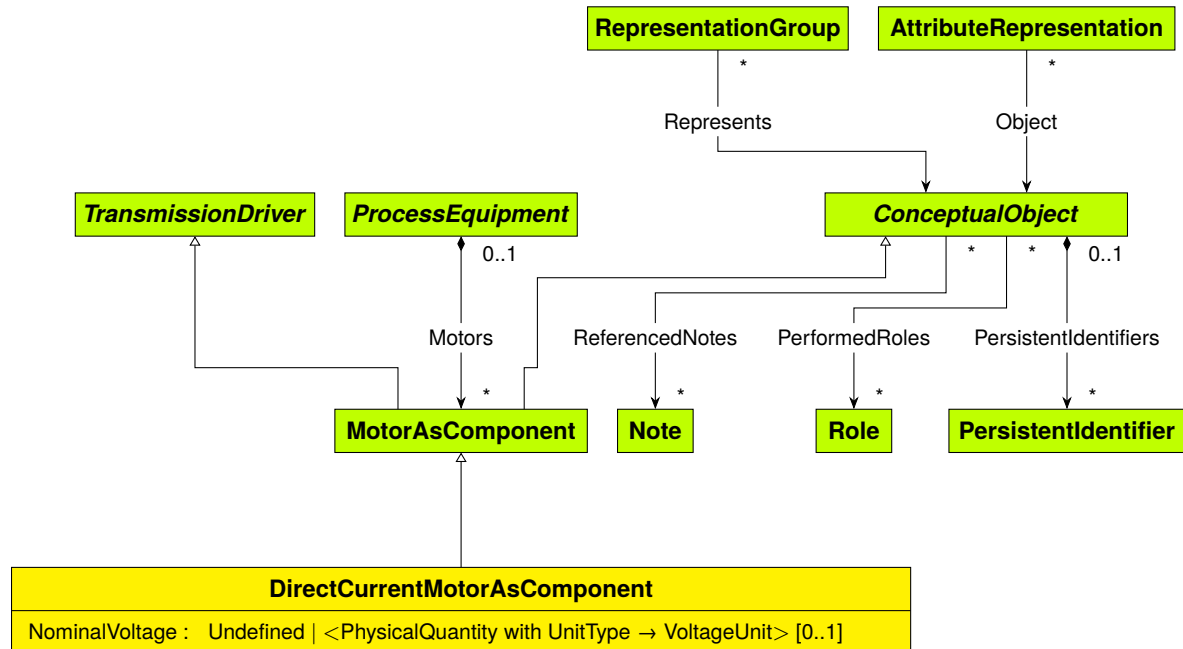
Plant::ProcessEquipment::DirectCurrentMotorAsComponent

Overview**Concrete Class**

RDL: <http://sandbox.dexpi.org/rdl/DirectCurrentMotorAsComponent>

DIRECT CURRENT MOTOR AS COMPONENT

An electric motor for operation by direct current that is used as component of an apparatus or of a machine.



Supertypes

- *MotorAsComponent*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>NominalVoltage</i>	0..1	<i>Undefined</i> <PhysicalQuantity with UnitType → VoltageUnit>	no	no

NominalVoltage

Data Property

The nominal voltage of the *DirectCurrentMotorAsComponent*.

Type: *Undefined* | <PhysicalQuantity with UnitType → VoltageUnit>

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS369449>

NOMINAL VOLTAGE

Example

PhysicalQuantity(
 Unit = *Volt*,
 Value = 230.0)

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.DirectCurrentMotorAsComponent">
  <Data property="NominalVoltage">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.VoltageUnit.Volt"/>
      </Data>
      <Data property="Value">
        <Double>230.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

Displacer

Plant::ProcessEquipment::Displacer

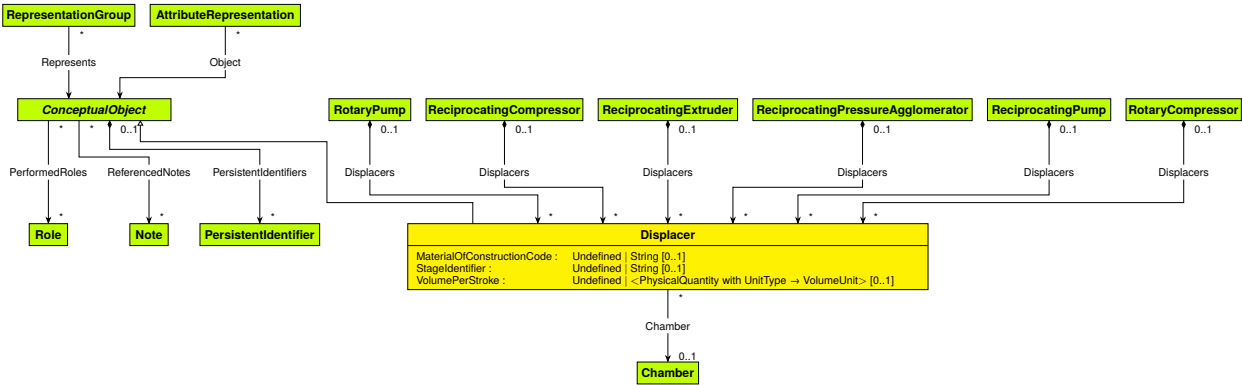
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/Displacer>

DISPLACER

An object that has the purpose of displacing a fluid.



Supertypes

- *ConceptualObject*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Chamber</i>	0..1	<i>Chamber</i>	no	no	*

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>MaterialOfConstructionCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>StageIdentifier</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>VolumePerStroke</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VolumeUnit></i>	no	no

Chamber

Reference Property

The *Chamber* in which the Displacer is located, if applicable. The Chamber must be a component of the same object as the Displacer.

Type: *Chamber*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

MaterialOfConstructionCode

Data Property

A code that gives the material of construction of the *Displacer*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS1460719741>

MATERIAL OF CONSTRUCTION CODE ASSIGNMENT CLASS

Example

"1.4306"

StageIdentifier**Data Property**

The stage identifier of the *Displacer*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/StageIdentifierAssignmentClass>

STAGE IDENTIFIER ASSIGNMENT CLASS

Example

"s1"

VolumePerStroke**Data Property**

The volume per stroke of the *Displacer*.

Type: *Undefined* | $\langle \textit{PhysicalQuantity}$ with *UnitType* $\rightarrow$ *VolumeUnit* $\rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS7503244>

VOLUME PER STROKE

Example

PhysicalQuantity(
 Unit = *CentimetreCubed*,
 Value = 80.0)

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Displacer">
  <Data property="MaterialOfConstructionCode">
    <String>1.4306</String>
  </Data>
  <Data property="StageIdentifier">
    <String>s1</String>
  </Data>
```

(continues on next page)

(continued from previous page)

```

<Data property="VolumePerStroke">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.VolumeUnit.CentimetreCubed"/>
    </Data>
    <Data property="Value">
      <Double>80.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>

```

DryCoolingTower

Plant::ProcessEquipment::DryCoolingTower

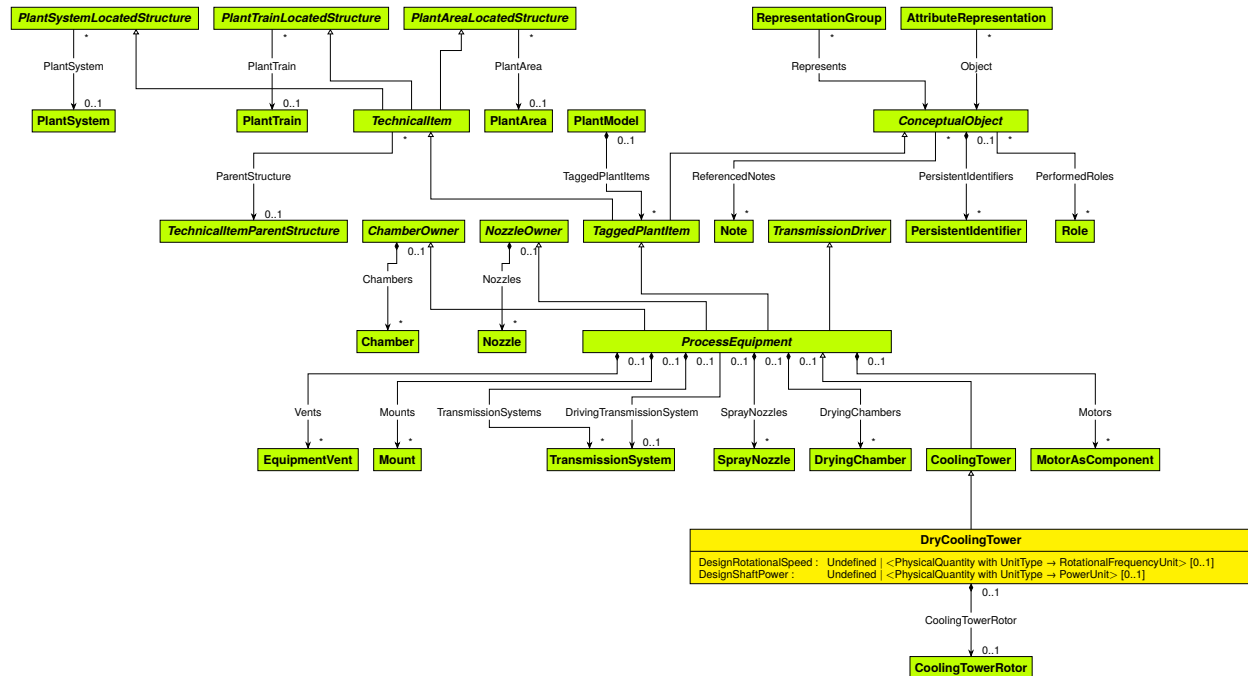
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS14072386>

DRY COOLING TOWER

A *CoolingTower* that is an indirect contact heat exchanger where, by full utilization of dry surface coil sections, no direct contact (and no evaporation) occurs between air and water; hence the water is cooled totally by sensible heat transfer (from <http://data.15926.org/rdl/RDS14072386>).



Supertypes

- *CoolingTower*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>CoolingTowerRotor</i>	0..1	<i>CoolingTowerRotor</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignRotationalSpeed</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → Rotational-FrequencyUnit></i>	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no

CoolingTowerRotor

Composition Property

The cooling tower rotor of the *DryCoolingTower*.

Type: *CoolingTowerRotor*

Multiplicity: 0..1

Ordered: no

DesignRotationalSpeed

Data Property

The rotational speed for which the *DryCoolingTower* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(
  Unit = ReciprocalMinute,
  Value = 180.0)
```

DesignShaftPower

Data Property

The shaft power for which the *DryCoolingTower* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(  
    Unit = Kilowatt,  
    Value = 400.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.DryCoolingTower">  
  <Data property="DesignRotationalSpeed">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.  
↪ReciprocalMinute"/>  
      </Data>  
      <Data property="Value">  
        <Double>180.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
  <Data property="DesignShaftPower">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>  
      </Data>  
      <Data property="Value">  
        <Double>400.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
</Object>
```

Dryer

Plant::ProcessEquipment::Dryer

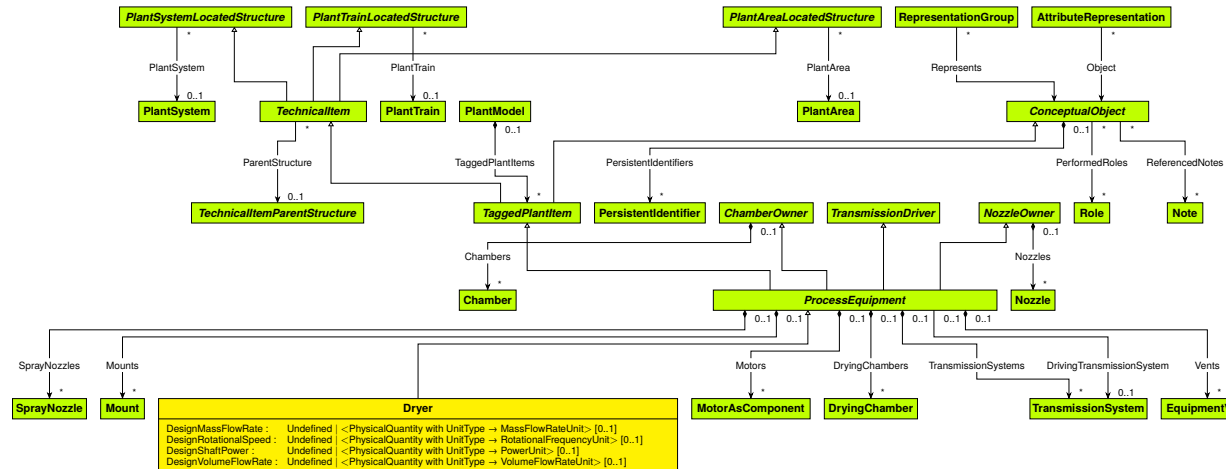
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS1066939451>

DRIER

An object that has the capability of drying (from <http://data.15926.org/rdl/RDS1066939451>).



Supertypes

- *ProcessEquipment*

Subtypes

- *ConvectionDryer*
- *HeatedSurfaceDryer*

Members

Data Properties

Name	Multi- plicity	Type	Or- dered	Unique
<i>DesignMassFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → MassFlowRateUnit></i>	no	no
<i>DesignRotationalSpeed</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → RotationalFrequencyUnit></i>	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no
<i>DesignVolumeFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VolumeFlowRateUnit></i>	no	no

DesignMassFlowRate

Data Property

The mass flow rate for which the *Dryer* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → MassFlowRateUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS14286182>

DESIGN MASS FLOW RATE

Example

```
PhysicalQuantity(  
    Unit = KilogramPerSecond,  
    Value = 420.0)
```

DesignRotationalSpeed

Data Property

The rotational speed for which the *Dryer* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{RotationalFrequencyUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(  
    Unit = ReciprocalMinute,  
    Value = 180.0)
```

DesignShaftPower

Data Property

The shaft power for which the *Dryer* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(
  Unit = Kilowatt,
  Value = 400.0)
```

DesignVolumeFlowRate**Data Property**

The volume flow rate for which the *Dryer* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VolumeFlowRateUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS14286227>

DESIGN VOLUME FLOW RATE

Example

```
PhysicalQuantity(
  Unit = MetreCubedPerHour,
  Value = 420.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Dryer">
  <Data property="DesignMassFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerSecond"/>
      </Data>
      <Data property="Value">
        <Double>420.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignRotationalSpeed">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
ReciprocalMinute"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
```

(continues on next page)

(continued from previous page)

```

<Data property="DesignShaftPower">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
    </Data>
    <Data property="Value">
      <Double>400.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="DesignVolumeFlowRate">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour
↪"/>
    </Data>
    <Data property="Value">
      <Double>420.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>

```

DryingChamber

Plant::ProcessEquipment::DryingChamber

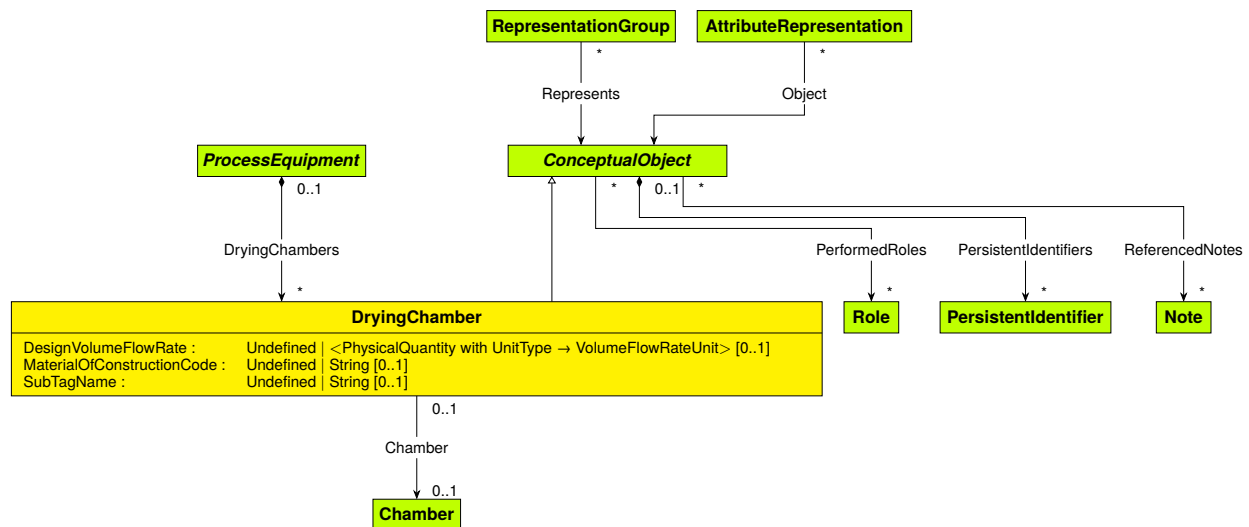
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/DryingChamber>

DRYING CHAMBER

A device that is a chamber, fixed or portable, for drying used as a component of an apparatus or a machine.



Supertypes

- *ConceptualObject*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Chamber</i>	0..1	<i>Chamber</i>	no	no	0..1

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignVolumeFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VolumeFlowRateUnit></i>	no	no
<i>MaterialOfConstructionCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>SubTagName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

Chamber

Reference Property

The *Chamber* in which the *DryingChamber* is located, if applicable. The *Chamber* must be a component of the same object as the *DryingChamber*.

Type: *Chamber*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

DesignVolumeFlowRate

Data Property

The volume flow rate for which the *DryingChamber* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → VolumeFlowRateUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS14286227>

DESIGN VOLUME FLOW RATE

Example

```
PhysicalQuantity(  
    Unit = MetreCubedPerHour,  
    Value = 420.0)
```

MaterialOfConstructionCode

Data Property

A code that gives the material of construction of the *DryingChamber*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS1460719741>

MATERIAL OF CONSTRUCTION CODE ASSIGNMENT CLASS

Example

"1.4306"

SubTagName

Data Property

The sub tag name of the *DryingChamber*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SubTagNameAssignmentClass>

SUB TAG NAME ASSIGNMENT CLASS

Example

"ST1"

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.DryingChamber">  
  <Data property="DesignVolumeFlowRate">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour  
↪"/>  
      </Data>
```

(continues on next page)

(continued from previous page)

```

    <Data property="Value">
      <Double>420.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="MaterialOfConstructionCode">
  <String>1.4306</String>
</Data>
<Data property="SubTagName">
  <String>ST1</String>
</Data>
</Object>

```

EjectorPump

Plant::ProcessEquipment::EjectorPump

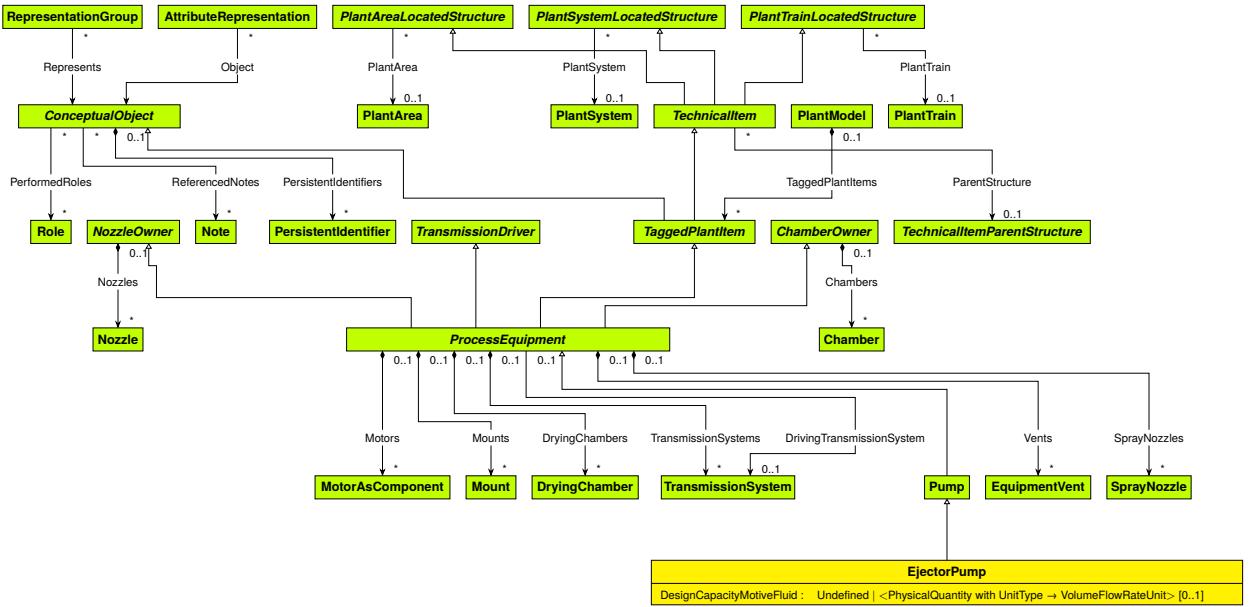
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS860624>

EJECTOR PUMP

A pump which uses pressurized gas or liquid passing through an ejector to transport liquid (from <http://data.posccaesar.org/rdl/RDS860624>).



Supertypes

- *Pump*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignCapacityMotiveFluid</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VolumeFlowRateUnit></i>	no	no

DesignCapacityMotiveFluid

Data Property

The capacity of the volume flow rate for the motive fluid for which the *EjectorPump* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → VolumeFlowRateUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignCapacityMotiveFluid>

DESIGN CAPACITY MOTIVE FLUID

Example

```
PhysicalQuantity(
    Unit = MetreCubedPerHour,
    Value = 40.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.EjectorPump">
  <Data property="DesignCapacityMotiveFluid">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour
↪"/>
      </Data>
      <Data property="Value">
        <Double>40.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

ElectricGenerator

Plant::ProcessEquipment::ElectricGenerator

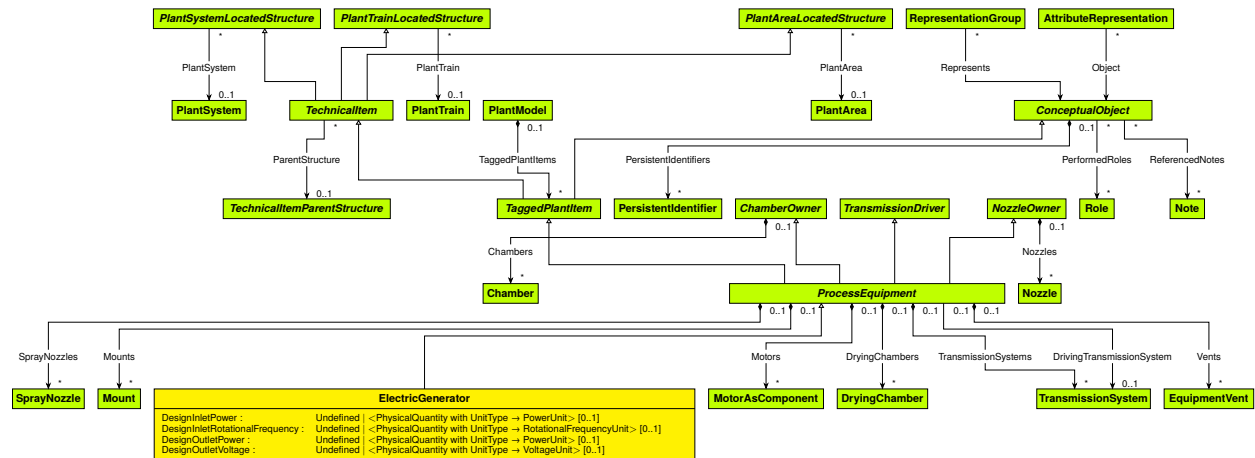
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS415709>

ELECTRIC GENERATOR

An electric rotating machine that transforms non-electric energy into electric energy



Supertypes

- *ProcessEquipment*

Subtypes

- *AlternatingCurrentGenerator*
- *DirectCurrentGenerator*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignInletPower</i>	0..1	Undefined <PhysicalQuantity with UnitType → PowerUnit>	no	no
<i>DesignInletRotational-Frequency</i>	0..1	Undefined <PhysicalQuantity with UnitType → RotationalFrequencyUnit>	no	no
<i>DesignOutletPower</i>	0..1	Undefined <PhysicalQuantity with UnitType → PowerUnit>	no	no
<i>DesignOutletVoltage</i>	0..1	Undefined <PhysicalQuantity with UnitType → VoltageUnit>	no	no

DesignInletPower

Data Property

The inlet power for which the *ElectricGenerator* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignInletPower>

DESIGN INLET POWER

Example

```
PhysicalQuantity(  
  Unit = Kilowatt,  
  Value = 300.0)
```

DesignInletRotationalFrequency

Data Property

The inlet rotational frequency for which the *ElectricGenerator* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{RotationalFrequencyUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignInletRotationalFrequency>

DESIGN INLET ROTATIONAL FREQUENCY

Example

```
PhysicalQuantity(  
  Unit = ReciprocalMinute,  
  Value = 180.0)
```

DesignOutletPower

Data Property

The outlet power for which the *ElectricGenerator* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignOutletPower>

DESIGN OUTLET POWER

Example

```
PhysicalQuantity(
  Unit = Kilowatt,
  Value = 500.0)
```

DesignOutletVoltage

Data Property

The outlet voltage for which the *ElectricGenerator* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VoltageUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignOutletVoltage>

DESIGN OUTLET VOLTAGE

Example

```
PhysicalQuantity(
  Unit = Volt,
  Value = 600.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.ElectricGenerator">
  <Data property="DesignInletPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>300.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignInletRotationalFrequency">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ReciprocalMinute"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

(continues on next page)

(continued from previous page)

```

    </Data>
  </AggregatedDataValue>
</Data>
<Data property="DesignOutletPower">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
    </Data>
    <Data property="Value">
      <Double>500.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="DesignOutletVoltage">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.VoltageUnit.Volt"/>
    </Data>
    <Data property="Value">
      <Double>600.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>

```

ElectricHeater

Plant::ProcessEquipment::ElectricHeater

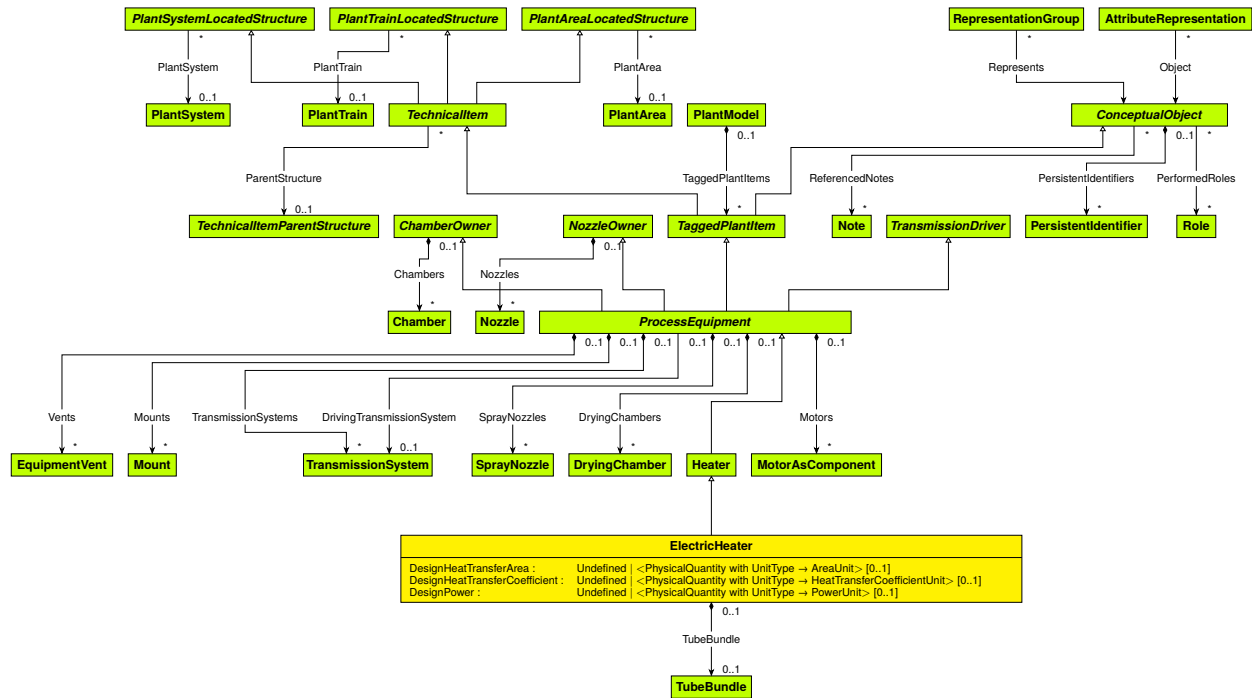
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS14070475>

ELECTRIC HEATER

A heater in which electric energy is converted into heat for useful purposes (from <http://data.posccaesar.org/rdl/RDS14070475>).



Supertypes

- *Heater*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>TubeBundle</i>	0..1	<i>TubeBundle</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignHeatTransferArea</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → AreaUnit></i>	no	no
<i>DesignHeatTransferCoefficient</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → HeatTransferCoefficientUnit></i>	no	no
<i>DesignPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no

DesignHeatTransferArea

Data Property

The heat transfer area for which the *ElectricHeater* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → AreaUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignHeatTransferArea>

DESIGN HEAT TRANSFER AREA

Example

PhysicalQuantity(
 Unit = *MetreSquared*,
 Value = 46.8)

DesignHeatTransferCoefficient

Data Property

The heat transfer coefficient for which the *ElectricHeater* is designed.

Type: *Undefined* | $\langle \textit{PhysicalQuantity}$ with *UnitType* $\rightarrow$ *HeatTransferCoefficientUnit* $\rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignHeatTransferCoefficient>

DESIGN HEAT TRANSFER COEFFICIENT

Example

PhysicalQuantity(
 Unit = *KilowattPerMetreSquaredKelvin*,
 Value = 1.2)

DesignPower

Data Property

The power for which the *ElectricHeater* is designed.

Type: *Undefined* | $\langle \textit{PhysicalQuantity}$ with *UnitType* $\rightarrow$ *PowerUnit* $\rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignPower>

DESIGN POWER

Example

```
PhysicalQuantity(
    Unit = Kilowatt,
    Value = 500.0)
```

TubeBundle**Composition Property**

The tube bundle of the ElectricHeater.

Type: *TubeBundle*

Multiplicity: 0..1

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.ElectricHeater">
  <Data property="DesignHeatTransferArea">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.AreaUnit.MetreSquared"/>
      </Data>
      <Data property="Value">
        <Double>46.8</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignHeatTransferCoefficient">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.HeatTransferCoefficientUnit.
↪KilowattPerMetreSquaredKelvin"/>
      </Data>
      <Data property="Value">
        <Double>1.2</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>500.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

ElectricalSeparator

Plant::ProcessEquipment::ElectricalSeparator

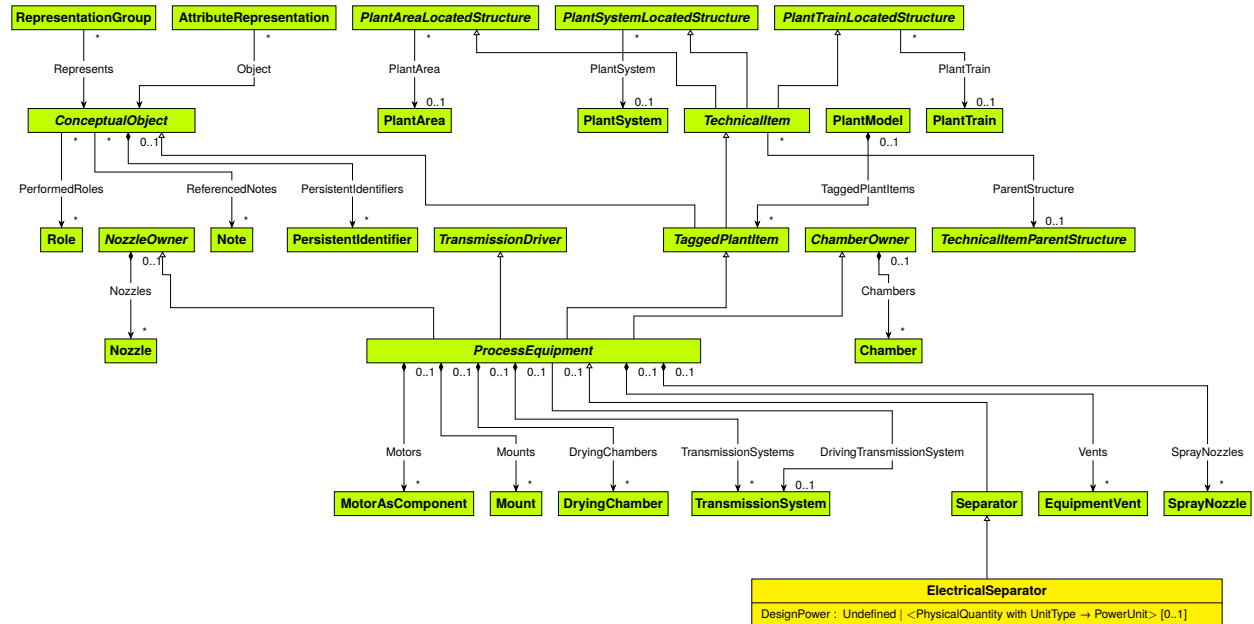
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/ElectricalSeparator>

ELECTRICAL SEPARATOR

A separator that uses electromagnetic, magnetic or electrostatic forces to separate phases.



Supertypes

- *Separator*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignPower</i>	0..1	<i>Undefined</i> <PhysicalQuantity with UnitType → PowerUnit>	no	no

DesignPower

Data Property

The power for which the *ElectricalSeparator* is designed.

Type: *Undefined* | <PhysicalQuantity with UnitType → PowerUnit>

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignPower>

DESIGN POWER

Example

```
PhysicalQuantity(
  Unit = Kilowatt,
  Value = 500.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.ElectricalSeparator">
  <Data property="DesignPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>500.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

EquipmentVent

Plant::ProcessEquipment::EquipmentVent

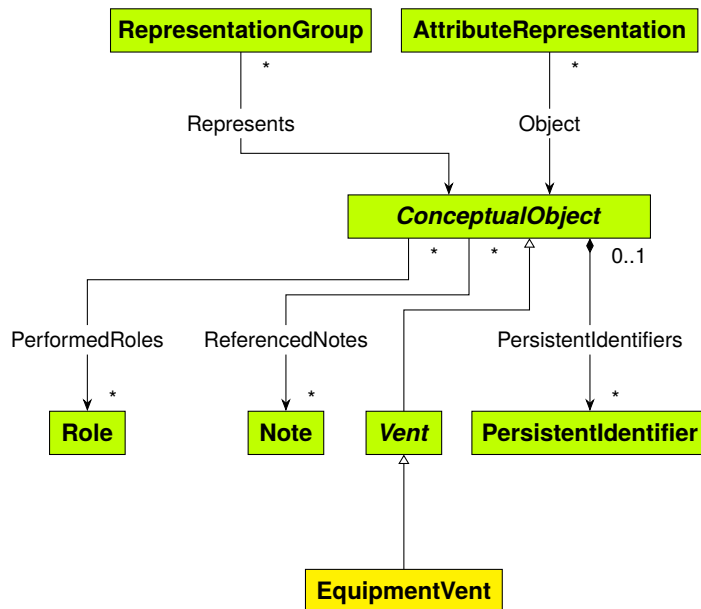
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS16766649>

CASING VENT

A *Vent* that is intended to release the pressure in a casing or to prevent airlocks in a fluid, and also acts as valve to release air trapped in the casing.



Supertypes

- *Vent*

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.EquipmentVent"/>
```

Extruder

Plant::ProcessEquipment::Extruder

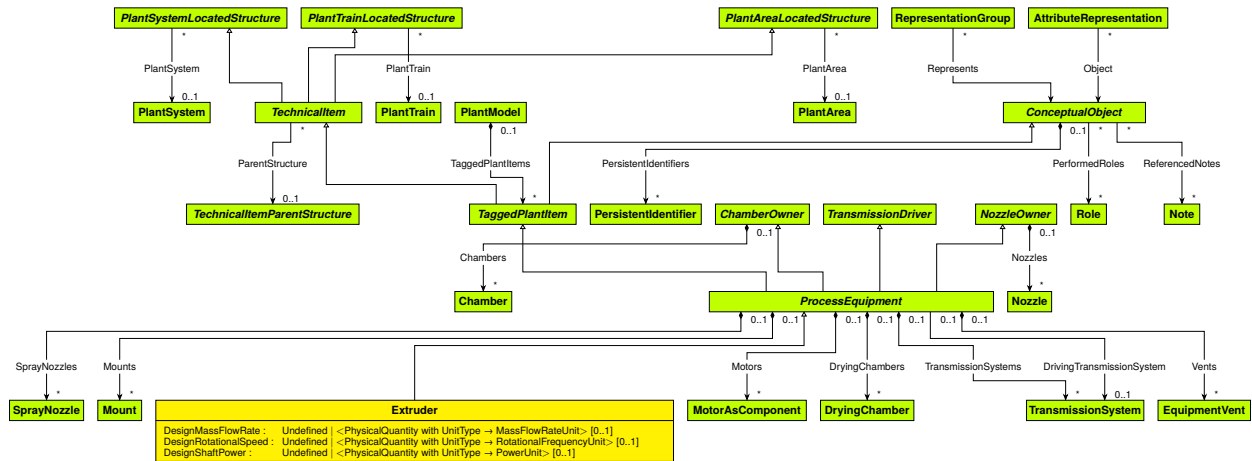
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS394044551>

EXTRUDER

A machine that has the capability of extruding (from <http://data.15926.org/rdl/RDS394044551>).



Supertypes

- *ProcessEquipment*

Subtypes

- *ReciprocatingExtruder*
- *RotatingExtruder*

Members

Data Properties

Name	Multi- plicity	Type	Or- dered	Unique
<i>DesignMassFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → MassFlowRateUnit></i>	no	no
<i>DesignRotationalSpeed</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → RotationalFrequencyUnit></i>	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no

DesignMassFlowRate

Data Property

The mass flow rate for which the *Extruder* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → MassFlowRateUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS14286182>

DESIGN MASS FLOW RATE

Example

```
PhysicalQuantity(  
  Unit = KilogramPerSecond,  
  Value = 420.0)
```

DesignRotationalSpeed**Data Property**

The rotational speed for which the *Extruder* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{RotationalFrequencyUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(  
  Unit = ReciprocalMinute,  
  Value = 180.0)
```

DesignShaftPower**Data Property**

The shaft power for which the *Extruder* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(  
  Unit = Kilowatt,  
  Value = 400.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Extruder">
  <Data property="DesignMassFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerSecond"/>
      </Data>
      <Data property="Value">
        <Double>420.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignRotationalSpeed">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
        ReciprocalMinute"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignShaftPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>400.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

Fan

Plant::ProcessEquipment::Fan

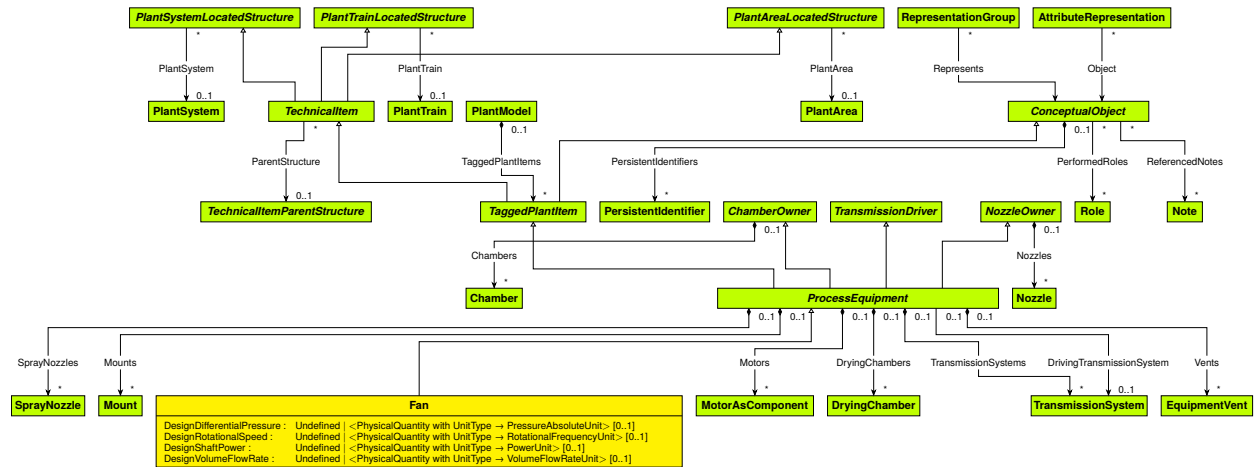
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS415169>

FAN

An ‘artefact’ and a ‘gas pressure increase device’ that is intended to deliver or exhaust volumes of vapour or gas at low differential pressure (from <http://data.posccaesar.org/rdl/RDS415169>).



Supertypes

- *ProcessEquipment*

Subtypes

- *AxialFan*
- *RadialFan*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignDifferentialPressure</i>	0..1	Undefined <PhysicalQuantity with UnitType → PressureAbsoluteUnit>	no	no
<i>DesignRotationalSpeed</i>	0..1	Undefined <PhysicalQuantity with UnitType → RotationalFrequencyUnit>	no	no
<i>DesignShaftPower</i>	0..1	Undefined <PhysicalQuantity with UnitType → PowerUnit>	no	no
<i>DesignVolumeFlowRate</i>	0..1	Undefined <PhysicalQuantity with UnitType → VolumeFlowRateUnit>	no	no

DesignDifferentialPressure

Data Property

The differential pressure for which the *Fan* is designed.

Type: Undefined | <PhysicalQuantity with UnitType → PressureAbsoluteUnit>

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignDifferentialPressure>

DESIGN DIFFERENTIAL PRESSURE

Example

PhysicalQuantity(
 Unit = *Bar*,
 Value = 4.8)

DesignRotationalSpeed**Data Property**

The rotational speed for which the *Fan* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{RotationalFrequencyUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

PhysicalQuantity(
 Unit = *ReciprocalMinute*,
 Value = 180.0)

DesignShaftPower**Data Property**

The shaft power for which the *Fan* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

PhysicalQuantity(
 Unit = *Kilowatt*,
 Value = 400.0)

DesignVolumeFlowRate

Data Property

The volume flow rate for which the *Fan* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VolumeFlowRateUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS14286227>

DESIGN VOLUME FLOW RATE

Example

```
PhysicalQuantity(  
    Unit = MetreCubedPerHour,  
    Value = 420.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Fan">  
  <Data property="DesignDifferentialPressure">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.PressureAbsoluteUnit.Bar"/>  
      </Data>  
      <Data property="Value">  
        <Double>4.8</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
  <Data property="DesignRotationalSpeed">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.  
↪ReciprocalMinute"/>  
      </Data>  
      <Data property="Value">  
        <Double>180.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
  <Data property="DesignShaftPower">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>  
      </Data>  
      <Data property="Value">  
        <Double>400.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
</Object>
```

(continues on next page)

(continued from previous page)

```
</AggregatedDataValue>
</Data>
<Data property="DesignVolumeFlowRate">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour"
      ↪"/>
    </Data>
    <Data property="Value">
      <Double>420.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>
```

Feeder

Plant::ProcessEquipment::Feeder

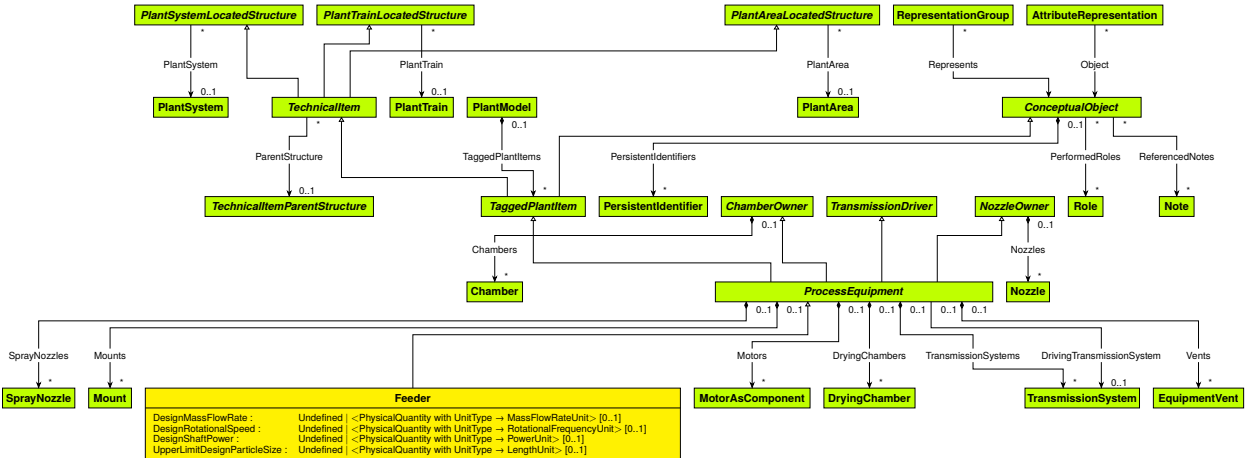
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS300644>

FEEDER

A closed fluid transporter that is a gathering line tied into a trunk line (from <http://data.15926.org/rdl/RDS300644>).



Supertypes

- *ProcessEquipment*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignMassFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → MassFlowRateUnit></i>	no	no
<i>DesignRotationalSpeed</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → RotationalFrequencyUnit></i>	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no
<i>UpperLimitDesignParticleSize</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no

DesignMassFlowRate

Data Property

The mass flow rate for which the *Feeder* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → MassFlowRateUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS14286182>

DESIGN MASS FLOW RATE

Example

```
PhysicalQuantity(  
    Unit = KilogramPerSecond,  
    Value = 420.0)
```

DesignRotationalSpeed

Data Property

The rotational speed for which the *Feeder* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(
  Unit = ReciprocalMinute,
  Value = 180.0)
```

DesignShaftPower**Data Property**

The shaft power for which the *Feeder* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(
  Unit = Kilowatt,
  Value = 400.0)
```

UpperLimitDesignParticleSize**Data Property**

The upper limit for the particle size for which the *Feeder* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitDesignParticleSize>

UPPER LIMIT DESIGN PARTICLE SIZE

Example

```
PhysicalQuantity(
  Unit = Micrometre,
  Value = 47.0)
```

Example in DEXPI XML

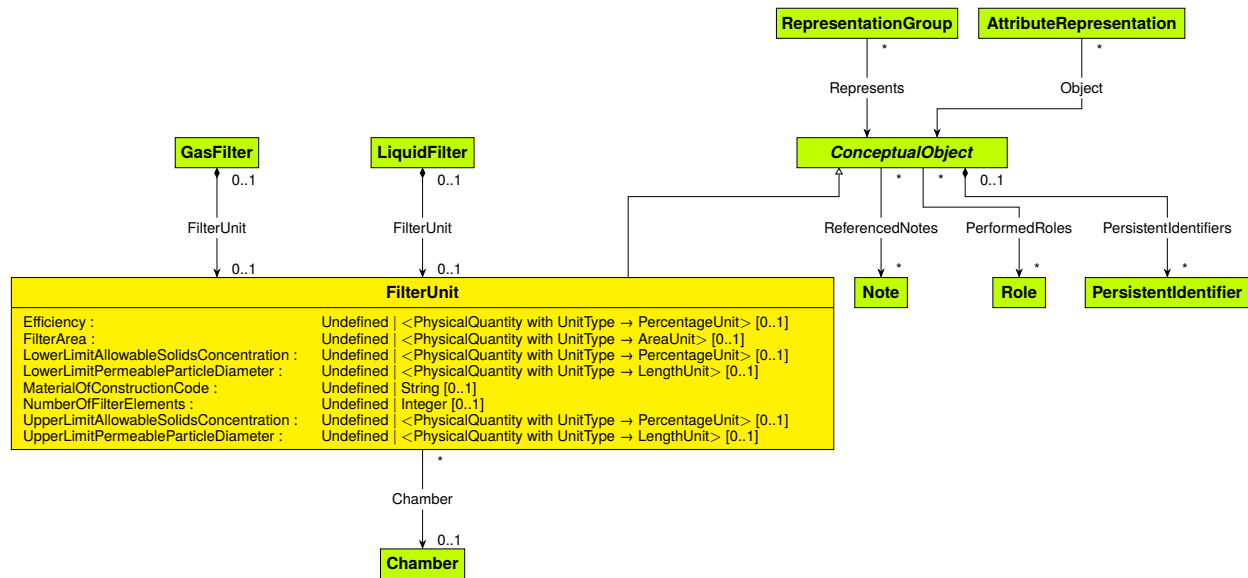
```

<Object type="Plant/ProcessEquipment.Feeder">
  <Data property="DesignMassFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerSecond"/>
      </Data>
      <Data property="Value">
        <Double>420.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignRotationalSpeed">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ ReciprocalMinute"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignShaftPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>400.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="UpperLimitDesignParticleSize">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Micrometre"/>
      </Data>
      <Data property="Value">
        <Double>47.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>

```

Filter

Plant::ProcessEquipment::Filter



Supertypes

- *ConceptualObject*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Chamber</i>	0..1	<i>Chamber</i>	no	no	*

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Efficiency</i>	0..1	<i>Undefined</i> <PhysicalQuantity with UnitType → PercentageUnit>	no	no
<i>FilterArea</i>	0..1	<i>Undefined</i> <PhysicalQuantity with UnitType → AreaUnit>	no	no
<i>LowerLimitAllowableSolidsConcentration</i>	0..1	<i>Undefined</i> <PhysicalQuantity with UnitType → PercentageUnit>	no	no
<i>LowerLimitPermeableParticleDiameter</i>	0..1	<i>Undefined</i> <PhysicalQuantity with UnitType → LengthUnit>	no	no
<i>MaterialOfConstructionCode</i>	0..1	<i>Undefined</i> String	no	no
<i>NumberOfFilterElements</i>	0..1	<i>Undefined</i> Integer	no	no
<i>UpperLimitAllowableSolidsConcentration</i>	0..1	<i>Undefined</i> <PhysicalQuantity with UnitType → PercentageUnit>	no	no
<i>UpperLimitPermeableParticleDiameter</i>	0..1	<i>Undefined</i> <PhysicalQuantity with UnitType → LengthUnit>	no	no

Chamber

Reference Property

The *Chamber* in which the *FilterUnit* is located, if applicable. The Chamber must be a component of the same object as the *FilterUnit*.

Type: *Chamber*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

Efficiency

Data Property

The efficiency of the *FilterUnit*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PercentageUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS362654>

EFFICIENCY

Example

```
PhysicalQuantity(  
  Unit = Percent,  
  Value = 90.0)
```

FilterArea

Data Property

The filter area of the *FilterUnit*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{AreaUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/FilterArea>

FILTER AREA

Example

```
PhysicalQuantity(  
  Unit = MetreSquared,  
  Value = 6.0)
```

LowerLimitAllowableSolidsConcentration**Data Property**

The lower limit for the concentration for solids.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PercentageUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LowerLimitAllowableSolidsConcentration>

LOWER LIMIT ALLOWABLE SOLIDS CONCENTRATION

Example

```
PhysicalQuantity(  
  Unit = Percent,  
  Value = 10.0)
```

LowerLimitPermeableParticleDiameter**Data Property**

The lower limit for the particle size.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LowerLimitPermeableParticleDiameter>

LOWER LIMIT PERMEABLE PARTICLE DIAMETER

Example

```
PhysicalQuantity(  
  Unit = Micrometre,  
  Value = 50.0)
```

MaterialOfConstructionCode**Data Property**

A code that gives the material of construction of the *FilterUnit*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS1460719741>

MATERIAL OF CONSTRUCTION CODE ASSIGNMENT CLASS

Example

"1.4306"

NumberOfFilterElements**Data Property**

The number of filter elements in the *FilterUnit*.

Type: *Undefined* | *Integer*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NumberOfFilterElements>

NUMBER OF FILTER ELEMENTS

Example

36

UpperLimitAllowableSolidsConcentration**Data Property**

The upper limit for the concentration for solids.

Type: *Undefined* | $\langle \textit{PhysicalQuantity}$ with $\textit{UnitType} \rightarrow \textit{PercentageUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitAllowableSolidsConcentration>

UPPER LIMIT ALLOWABLE SOLIDS CONCENTRATION

Example

```
PhysicalQuantity(  
  Unit = Percent,  
  Value = 30.0)
```

UpperLimitPermeableParticleDiameter

Data Property

The maximum of the particle size passing through the *FilterUnit*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitPermeableParticleDiameter>

UPPER LIMIT PERMEABLE PARTICLE DIAMETER

Example

```
PhysicalQuantity(  
  Unit = Micrometre,  
  Value = 400.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.FilterUnit">  
  <Data property="Efficiency">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.PercentageUnit.Percent"/>  
      </Data>  
      <Data property="Value">  
        <Double>90.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
  <Data property="FilterArea">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.AreaUnit.MetreSquared"/>  
      </Data>  
      <Data property="Value">  
        <Double>6.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
  <Data property="LowerLimitAllowableSolidsConcentration">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
```

(continues on next page)

(continued from previous page)

```

    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.PercentageUnit.Percent"/>
    </Data>
    <Data property="Value">
      <Double>10.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="LowerLimitPermeableParticleDiameter">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.LengthUnit.Micrometre"/>
    </Data>
    <Data property="Value">
      <Double>50.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="MaterialOfConstructionCode">
  <String>1.4306</String>
</Data>
<Data property="NumberOfFilterElements">
  <Integer>36</Integer>
</Data>
<Data property="UpperLimitAllowableSolidsConcentration">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.PercentageUnit.Percent"/>
    </Data>
    <Data property="Value">
      <Double>30.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="UpperLimitPermeableParticleDiameter">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.LengthUnit.Micrometre"/>
    </Data>
    <Data property="Value">
      <Double>400.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>

```

FilteringCentrifuge

Plant::ProcessEquipment::FilteringCentrifuge

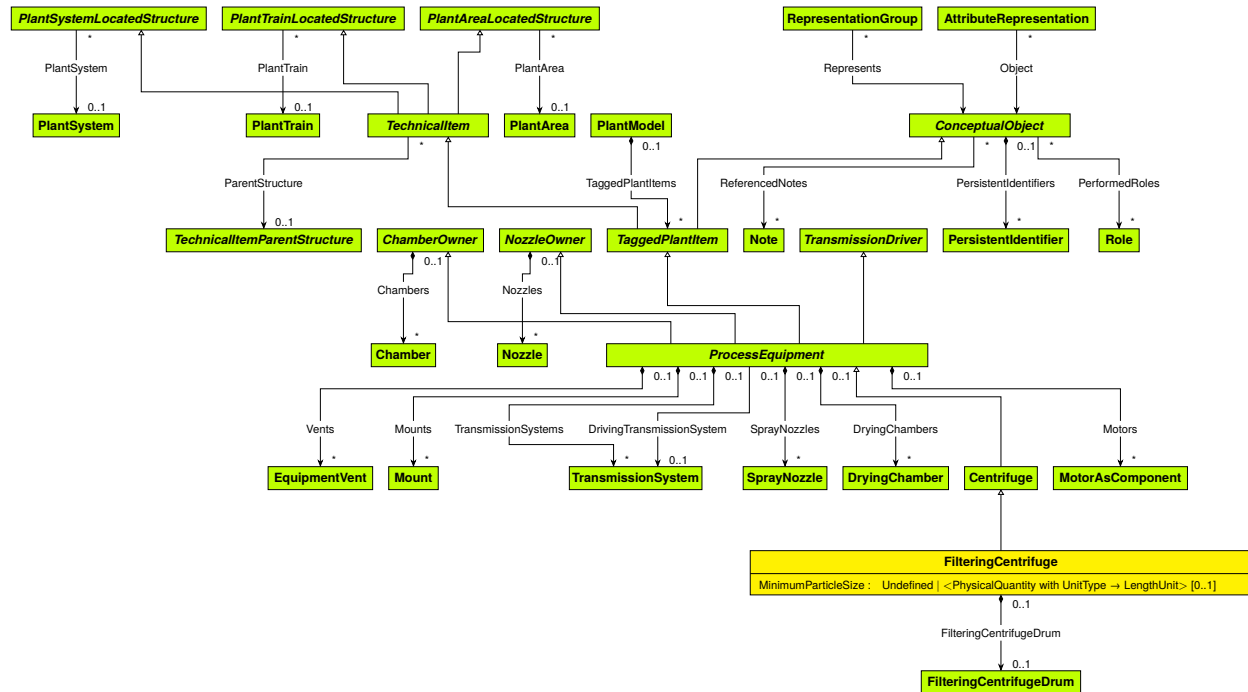
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/FilteringCentrifuge>

FILTERING CENTRIFUGE

A centrifuge intended to separate solids from liquids by centrifugal process based on particle size.



Supertypes

- *Centrifuge*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>FilteringCentrifugeDrum</i>	0..1	<i>FilteringCentrifugeDrum</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>MinimumParticleSize</i>	0..1	<i>Undefined <PhysicalQuantity with UnitType → LengthUnit></i>	no	no

FilteringCentrifugeDrum

Composition Property

The filtering centrifuge drum of the FilteringCentrifuge.

Type: *FilteringCentrifugeDrum*

Multiplicity: 0..1

Ordered: no

MinimumParticleSize

Data Property

The minimum particle size of the *FilteringCentrifuge*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/MinimumParticleSize>

MINIMUM PARTICLE SIZE

Example

```
PhysicalQuantity(  
  Unit = Micrometre,  
  Value = 7.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.FilteringCentrifuge">  
  <Data property="MinimumParticleSize">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Micrometre"/>  
      </Data>  
      <Data property="Value">  
        <Double>7.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
</Object>
```

FilteringCentrifugeDrum

Plant::ProcessEquipment::FilteringCentrifugeDrum

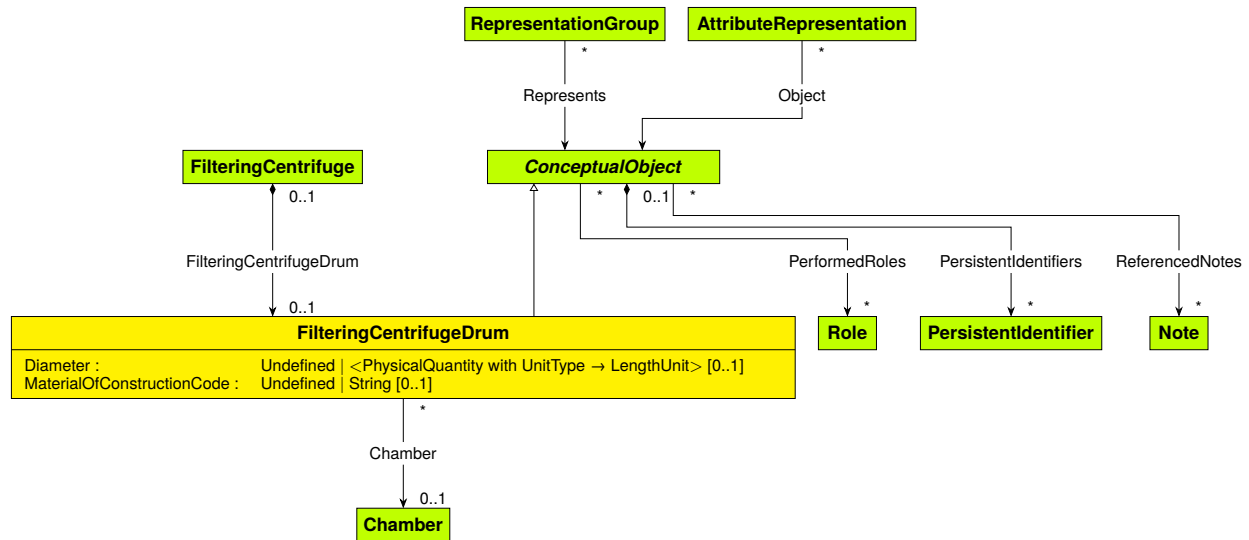
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/FilteringCentrifugeDrum>

FILTERING CENTRIFUGE DRUM

A drum being a component of a FilteringCentrifuge.



Supertypes

- *ConceptualObject*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Chamber</i>	0..1	<i>Chamber</i>	no	no	*

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Diameter</i>	0..1	<i>Undefined</i> < <i>PhysicalQuantity</i> with <i>UnitType</i> → <i>LengthUnit</i> >	no	no
<i>MaterialOfConstructionCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

Chamber

Reference Property

The *Chamber* in which the *FilteringCentrifugeDrum* is located, if applicable. The Chamber must be a component of the same object as the *FilteringCentrifugeDrum*.

Type: *Chamber*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

Diameter

Data Property

The diameter of the *FilteringCentrifugeDrum*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS350954>

DIAMETER

Example

```
PhysicalQuantity(  
  Unit = Centimetre,  
  Value = 20.0)
```

MaterialOfConstructionCode

Data Property

A code that gives the material of construction of the *FilteringCentrifugeDrum*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS1460719741>

MATERIAL OF CONSTRUCTION CODE ASSIGNMENT CLASS

Example

"1.4306"

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.FilteringCentrifugeDrum">
  <Data property="Diameter">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
      </Data>
      <Data property="Value">
        <Double>20.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="MaterialOfConstructionCode">
    <String>1.4306</String>
  </Data>
</Object>
```

Flare

Plant::ProcessEquipment::Flare

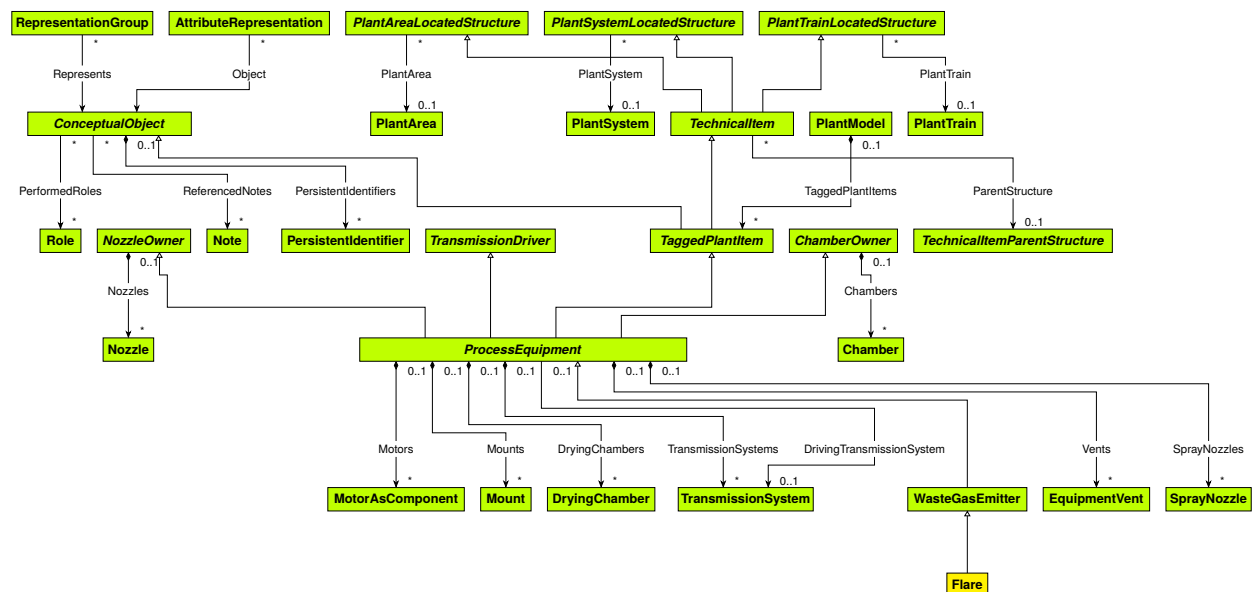
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/Flare>

FLARE

An artefact and waste gas emitter that is intended to burn waste gas in secure distance from the plant or platform.



Supertypes

- WasteGasEmitter

Example in DEXPI XML

<Object type="Plant/ProcessEquipment.Flare"/>

ForkliftTruck

Plant::ProcessEquipment::ForkliftTruck

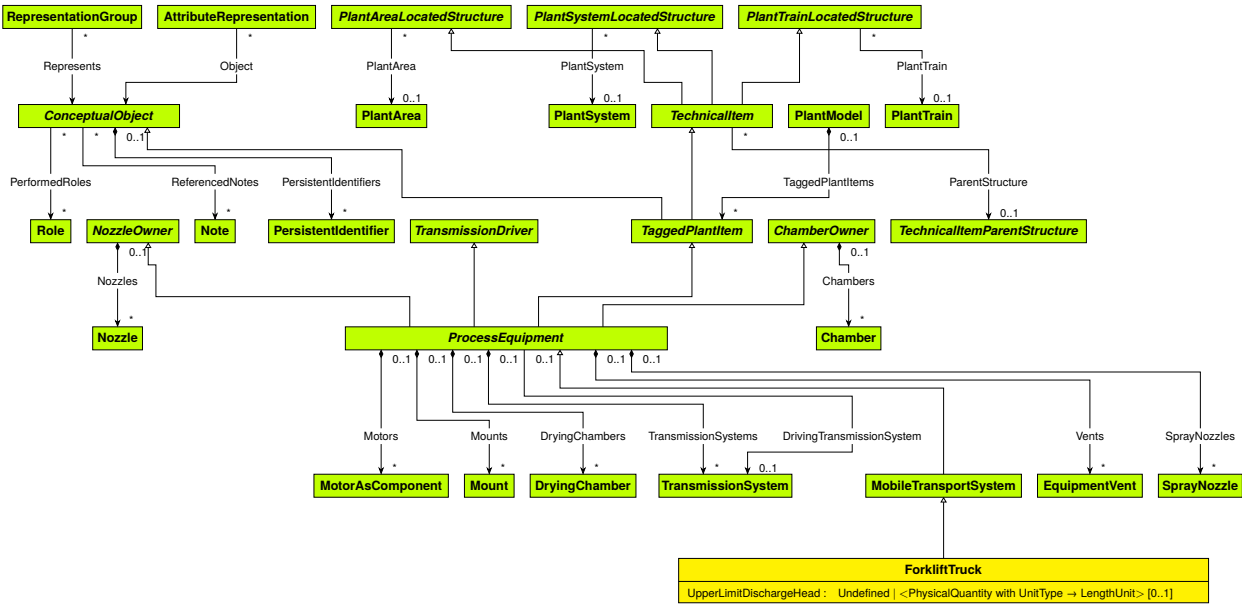
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS11590075>

FORKLIFT TRUCK

A *MobileTransportSystem* and vehicle with power operated prongs that can be raised and lowered by will, for loading, transporting and unloading goods (from <http://data.15926.org/rdl/RDS11590075>).



Supertypes

- MobileTransportSystem

Members

Data Properties

Name	Multiplic-ity	Type	Or-dered	Unique
UpperLimitDischarge-Head	0..1	Undefined <PhysicalQuantity with UnitType → LengthUnit>	no	no

UpperLimitDischargeHead

Data Property

The upper limit for the discharge head of the *ForkliftTruck*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitDischargeHead>

UPPER LIMIT DISCHARGE HEAD

Example

```
PhysicalQuantity(  
    Unit = Centimetre,  
    Value = 37.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.ForkliftTruck">  
  <Data property="UpperLimitDischargeHead">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>  
      </Data>  
      <Data property="Value">  
        <Double>37.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
</Object>
```

Furnace

Plant::ProcessEquipment::Furnace

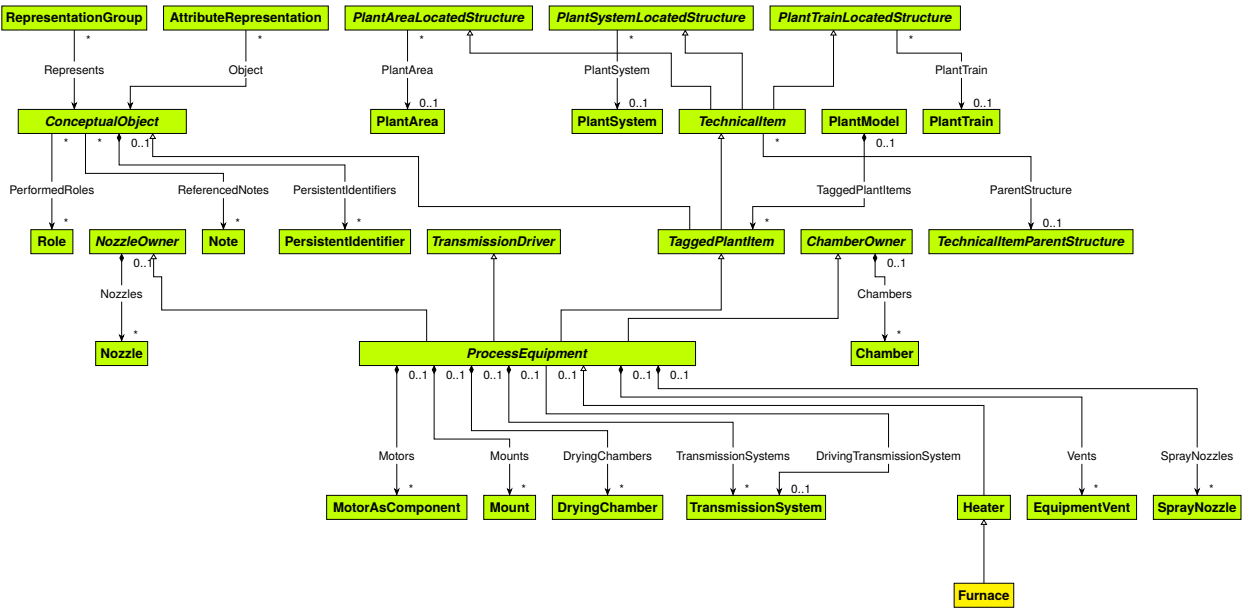
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS441134>

FURNACE

A physical object that is intended to induce a reaction in a process fluid by heating it (from <http://data.posccaesar.org/rdl/RDS441134>).



Supertypes

- *Heater*

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Furnace"/>
```

GasFilter

Plant::ProcessEquipment::GasFilter

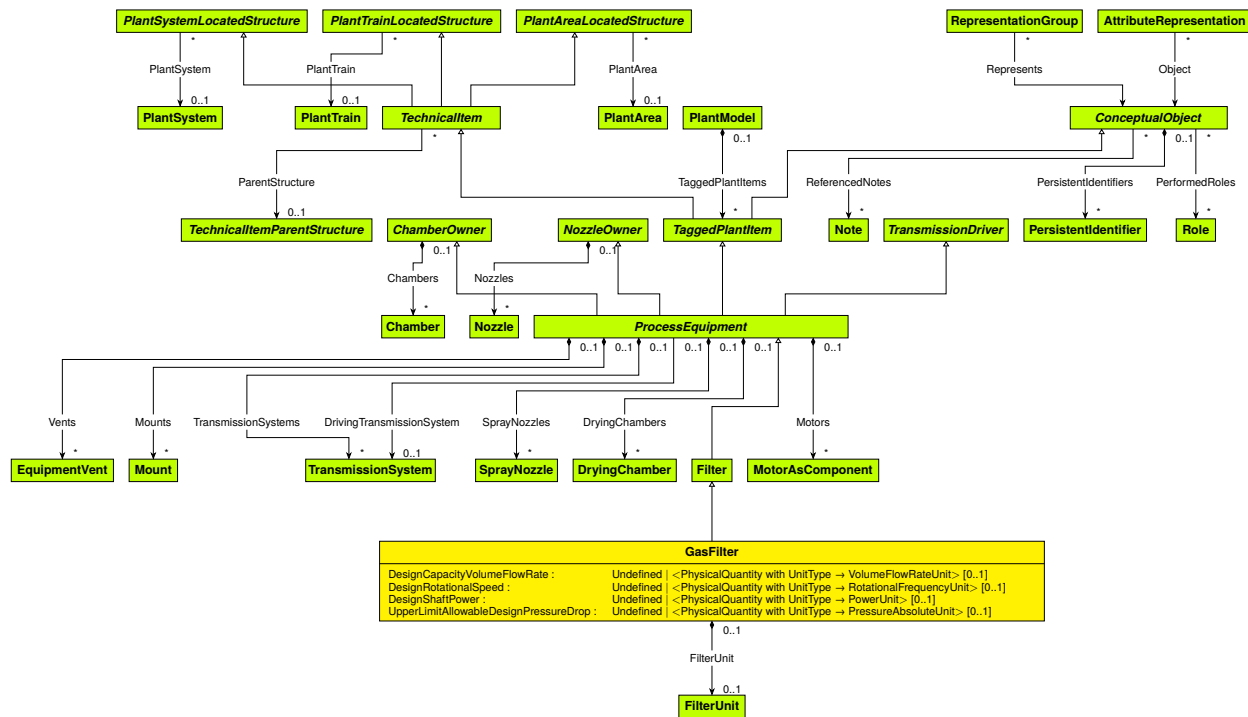
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS4316755843>

GAS FILTER

A filter that is specifically designed to filter a gas.



Supertypes

- *Filter*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>FilterUnit</i>	0..1	<i>FilterUnit</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignCapacityVolumeFlowRate</i>	0..1	<i>Undefined</i> <PhysicalQuantity with UnitType → VolumeFlowRateUnit>	no	no
<i>DesignRotationalSpeed</i>	0..1	<i>Undefined</i> <PhysicalQuantity with UnitType → RotationalFrequencyUnit>	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> <PhysicalQuantity with UnitType → PowerUnit>	no	no
<i>UpperLimitAllowableDesignPressureDrop</i>	0..1	<i>Undefined</i> <PhysicalQuantity with UnitType → PressureAbsoluteUnit>	no	no

DesignCapacityVolumeFlowRate

Data Property

The volume flow rate for which the *GasFilter* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VolumeFlowRateUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignCapacityVolumeFlowRate>

DESIGN CAPACITY VOLUME FLOW RATE

Example

```
PhysicalQuantity(  
    Unit = MetreCubedPerHour,  
    Value = 420.0)
```

DesignRotationalSpeed

Data Property

The rotational speed for which the *GasFilter* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{RotationalFrequencyUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(  
    Unit = ReciprocalMinute,  
    Value = 180.0)
```

DesignShaftPower

Data Property

The shaft power for which the *GasFilter* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(  
    Unit = Kilowatt,  
    Value = 400.0)
```

FilterUnit

Composition Property

The filter unit of the GasFilter.

Type: *FilterUnit*

Multiplicity: 0..1

Ordered: no

UpperLimitAllowableDesignPressureDrop

Data Property

The upper limit for the pressure drop for which the *GasFilter* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PressureAbsoluteUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitAllowableDesignPressureDrop>

UPPER LIMIT ALLOWABLE DESIGN PRESSURE DROP

Example

```
PhysicalQuantity(  
    Unit = Bar,  
    Value = 2.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.GasFilter">  
  <Data property="DesignCapacityVolumeFlowRate">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour  
↪"/>  
      </Data>  
      <Data property="Value">  
        <Double>420.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
</Object>
```

(continues on next page)

(continued from previous page)

```

    </Data>
  </AggregatedDataValue>
</Data>
<Data property="DesignRotationalSpeed">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ReciprocalMinute"/>
    </Data>
    <Data property="Value">
      <Double>180.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="DesignShaftPower">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
    </Data>
    <Data property="Value">
      <Double>400.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="UpperLimitAllowableDesignPressureDrop">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.PressureAbsoluteUnit.Bar"/>
    </Data>
    <Data property="Value">
      <Double>2.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>

```

GasTurbine

Plant::ProcessEquipment::GasTurbine

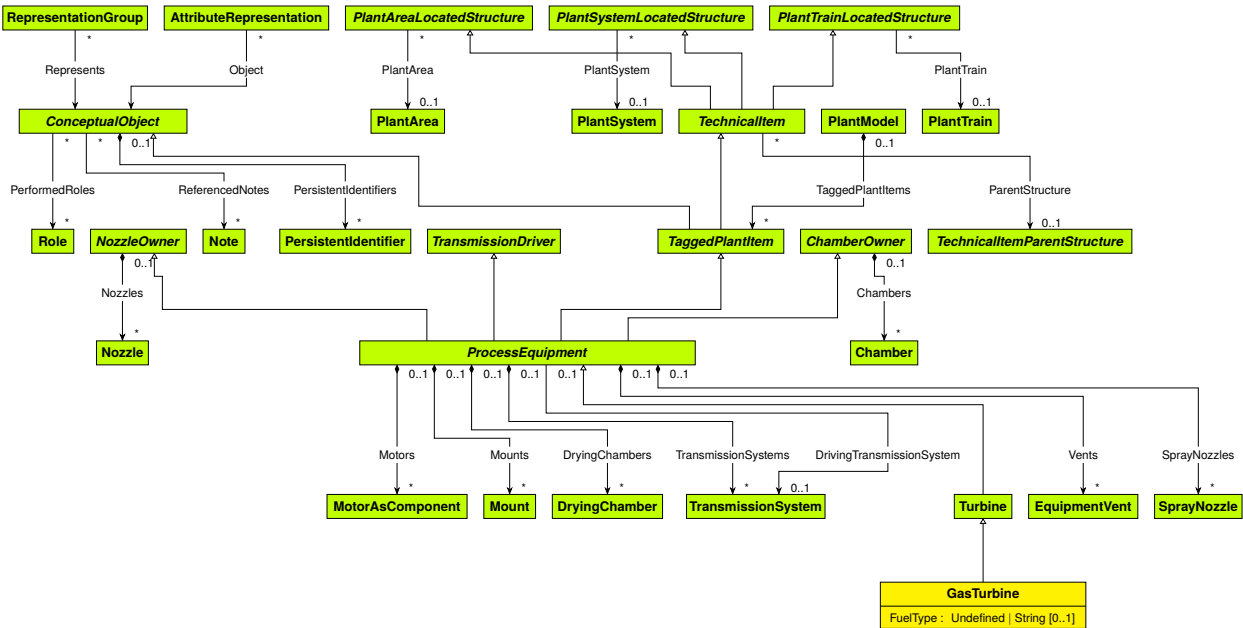
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/GasTurbine>

GAS TURBINE

A machine that is a rotary mechanical device extracting energy from a gas flow and converting it into useful work.



Supertypes

- Turbine

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>FuelType</i>	0..1	<i>Undefined String</i>	no	no

FuelType

Data Property

The fuel type of the *GasTurbine*.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/FuelType>

FUEL TYPE

Example

"Diesel fuel"

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.GasTurbine">
  <Data property="FuelType">
    <String>Diesel fuel</String>
  </Data>
</Object>
```

GearBox

Plant::ProcessEquipment::GearBox

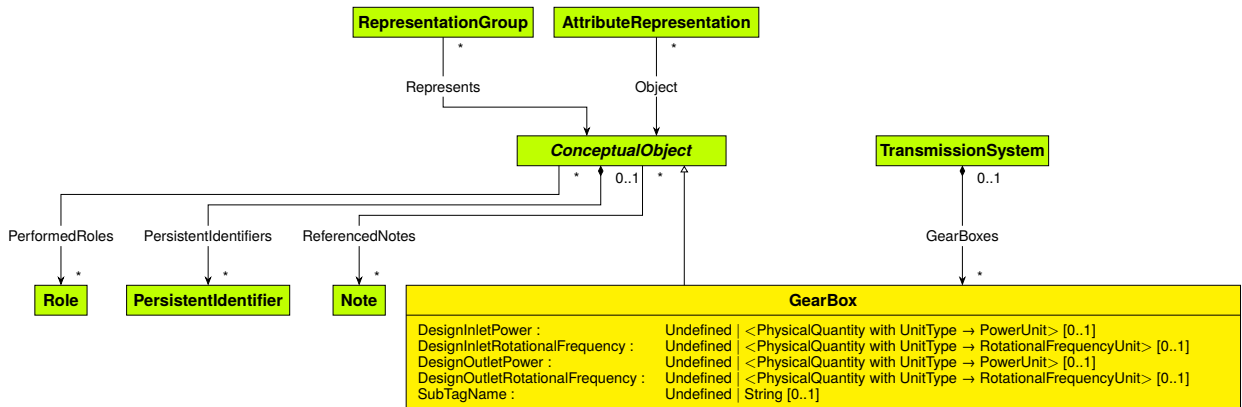
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS889514>

GEARBOX

An artefact that consists of a gear casing with an arrangement of two or more gear-wheels transmitting rotating motion from the input shaft to the output shaft.



Supertypes

- *ConceptualObject*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignInletPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no
<i>DesignInletRotational-Frequency</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → RotationalFrequencyUnit></i>	no	no
<i>DesignOutletPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no
<i>DesignOutletRotational-Frequency</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → RotationalFrequencyUnit></i>	no	no
<i>SubTagName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

DesignInletPower

Data Property

The inlet power for which the *GearBox* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PowerUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignInletPower>

DESIGN INLET POWER

Example

```
PhysicalQuantity(
  Unit = Kilowatt,
  Value = 300.0)
```

DesignInletRotationalFrequency

Data Property

The inlet rotational frequency for which the *GearBox* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignInletRotationalFrequency>

DESIGN INLET ROTATIONAL FREQUENCY

Example

```
PhysicalQuantity(
  Unit = ReciprocalMinute,
  Value = 180.0)
```

DesignOutletPower**Data Property**

The outlet power for which the *GearBox* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignOutletPower>

DESIGN OUTLET POWER

Example

```
PhysicalQuantity(
  Unit = Kilowatt,
  Value = 500.0)
```

DesignOutletRotationalFrequency**Data Property**

The outlet rotational frequency for which the *GearBox* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{RotationalFrequencyUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignOutletRotationalFrequency>

DESIGN OUTLET ROTATIONAL FREQUENCY

Example

```
PhysicalQuantity(
  Unit = ReciprocalMinute,
  Value = 840.0)
```

SubTagName**Data Property**

The sub tag name of the *GearBox*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SubTagNameAssignmentClass>

SUB TAG NAME ASSIGNMENT CLASS

Example

"ST1"

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.GearBox">
  <Data property="DesignInletPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>300.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignInletRotationalFrequency">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ReciprocalMinute"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignOutletPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>500.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignOutletRotationalFrequency">
```

(continues on next page)

(continued from previous page)

```

<AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
  <Data property="Unit">
    <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ReciprocalMinute"/>
  </Data>
  <Data property="Value">
    <Double>840.0</Double>
  </Data>
</AggregatedDataValue>
</Data>
<Data property="SubTagName">
  <String>ST1</String>
</Data>
</Object>

```

GravitationalSeparator

Plant::ProcessEquipment::GravitationalSeparator

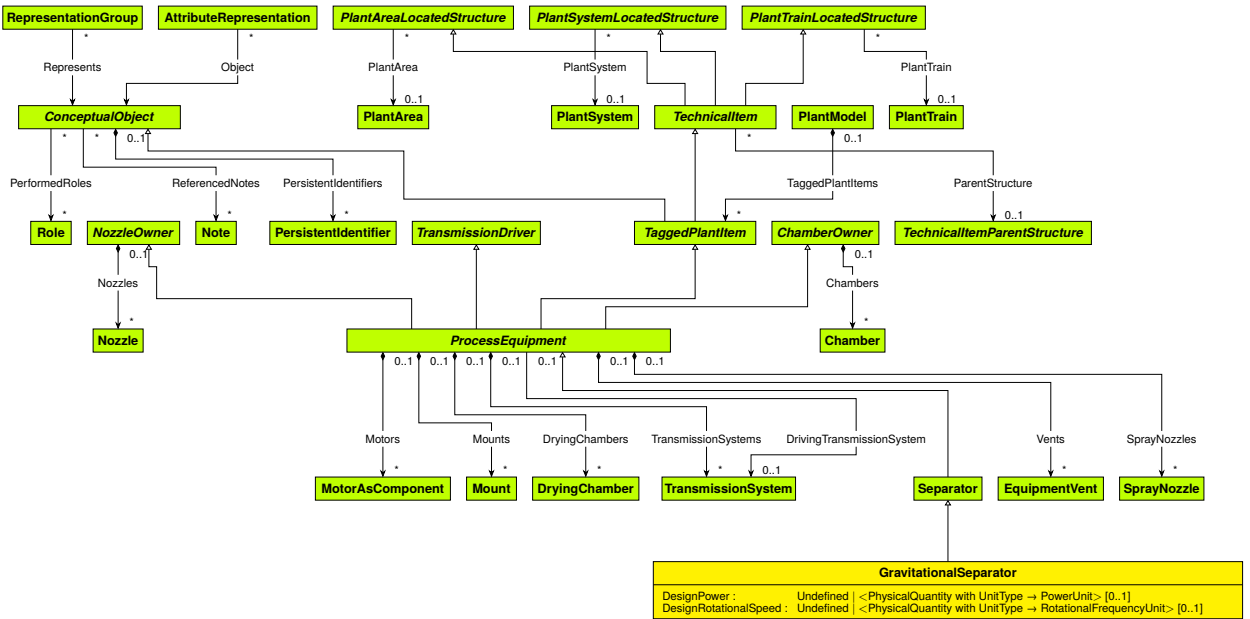
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS16042131>

GRAVITY SEPARATOR

A fluid separator that is based on the difference in specific gravity for the substances to be separated (from <http://data.15926.org/rdl/RDS16042131>).



Supertypes

- *Separator*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no
<i>DesignRotational-Speed</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → Rotational-FrequencyUnit></i>	no	no

DesignPower

Data Property

The power for which the *GravitationalSeparator* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PowerUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignPower>

DESIGN POWER

Example

```
PhysicalQuantity(  
    Unit = Kilowatt,  
    Value = 500.0)
```

DesignRotationalSpeed

Data Property

The rotational speed for which the *GravitationalSeparator* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(
    Unit = ReciprocalMinute,
    Value = 180.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.GravitationalSeparator">
  <Data property="DesignPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>500.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignRotationalSpeed">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ReciprocalMinute"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

Grinder

Plant::ProcessEquipment::Grinder

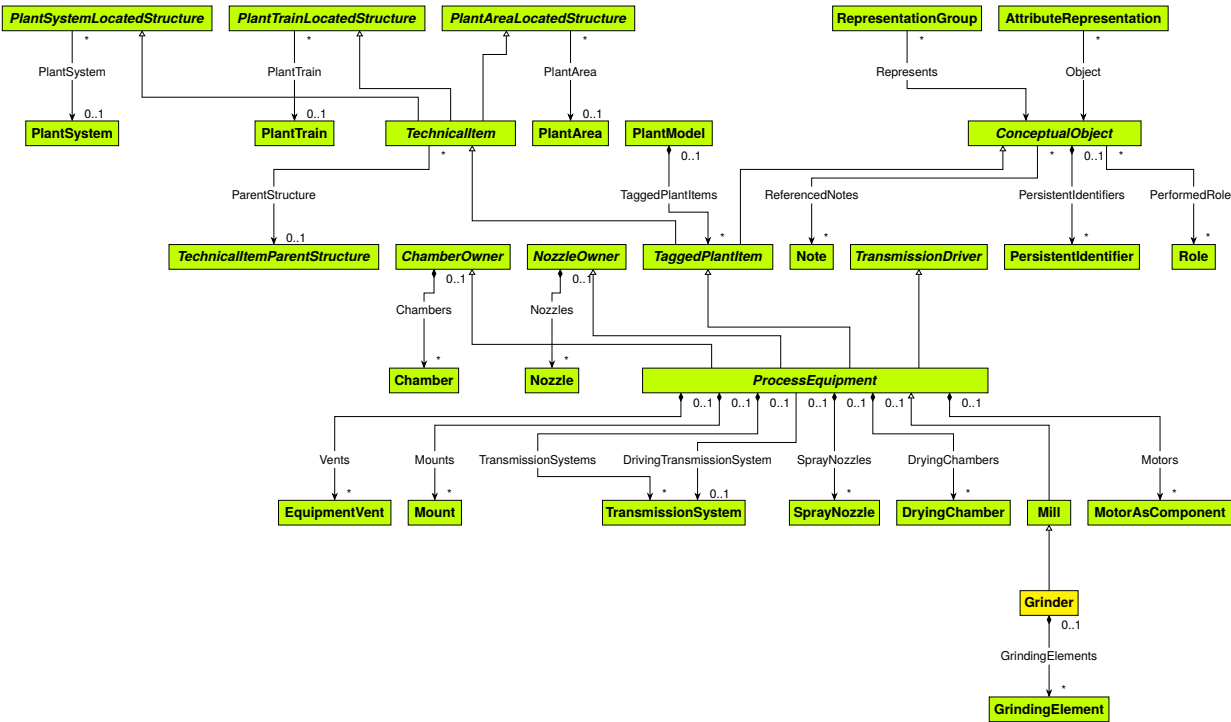
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/Grinder>

GRINDER

A *Mill* that has the capability of grinding.



Supertypes

- *Mill*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>GrindingElements</i>	*	<i>GrindingElement</i>	no

GrindingElements

Composition Property

The grinding elements of the Grinder.

Type: *GrindingElement*

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Grinder"/>
```

GrindingElement

Plant::ProcessEquipment::GrindingElement

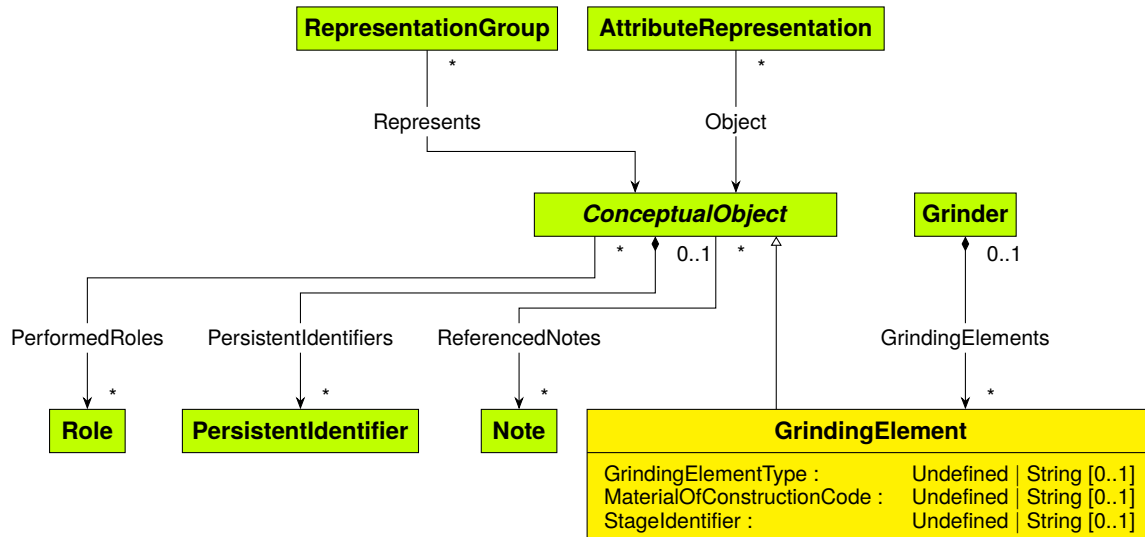
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/GrindingElement>

GRINDING ELEMENT

A functional component of a *Grinder*.



Supertypes

- *ConceptualObject*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>GrindingElementType</i>	0..1	Undefined String	no	no
<i>MaterialOfConstructionCode</i>	0..1	Undefined String	no	no
<i>StageIdentifier</i>	0..1	Undefined String	no	no

GrindingElementType

Data Property

A code that gives the crusher unit type of the *GrindingElement*.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/GrindingElementTypeAssignmentClass>

GRINDING ELEMENT TYPE ASSIGNMENT CLASS

Example

"1.4306"

MaterialOfConstructionCode

Data Property

A code that gives the material of construction of the *GrindingElement*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS1460719741>

MATERIAL OF CONSTRUCTION CODE ASSIGNMENT CLASS

Example

"1.4306"

StageIdentifier

Data Property

The stage identifier of the *GrindingElement*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/StageIdentifierAssignmentClass>

STAGE IDENTIFIER ASSIGNMENT CLASS

Example

"s1"

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.GrindingElement">
  <Data property="GrindingElementType">
    <String>1.4306</String>
  </Data>
  <Data property="MaterialOfConstructionCode">
```

(continues on next page)

(continued from previous page)

```

<String>1.4306</String>
</Data>
<Data property="StageIdentifier">
  <String>s1</String>
</Data>
</Object>

```

HeatExchanger

Plant::ProcessEquipment::HeatExchanger

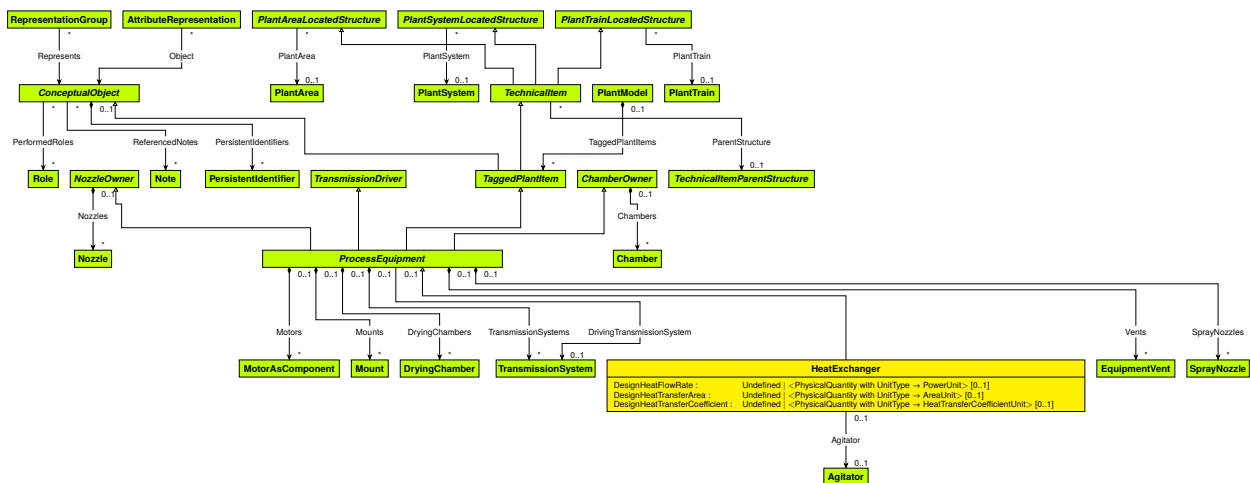
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS304199>

HEAT EXCHANGER

An apparatus or machine that has the capability of heat exchanging (from <http://data.15926.org/rdl/RDS304199>).



Supertypes

- *ProcessEquipment*

Subtypes

- *AirCoolingSystem*
- *PlateHeatExchanger*
- *SpiralHeatExchanger*
- *ThinFilmEvaporator*
- *TubularHeatExchanger*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Agitator</i>	0..1	<i>Agitator</i>	no	no	0..1

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignHeatFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no
<i>DesignHeatTransfer-Area</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → AreaUnit></i>	no	no
<i>DesignHeatTransfer-Coefficient</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → Heat-TransferCoefficientUnit></i>	no	no

Agitator

Reference Property

The *Agitator* of the *HeatExchanger*, if applicable.

Type: *Agitator*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

DesignHeatFlowRate

Data Property

The heat flow rate for which the *HeatExchanger* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PowerUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignHeatFlowRate>

DESIGN HEAT FLOW RATE

Example

PhysicalQuantity(
 Unit = *Kilowatt*,
 Value = 313.0)

DesignHeatTransferArea**Data Property**

The heat transfer area for which the *HeatExchanger* is designed.

Type: *Undefined* | $\langle \textit{PhysicalQuantity}$ with *UnitType* $\rightarrow$ *AreaUnit* $\rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignHeatTransferArea>

DESIGN HEAT TRANSFER AREA

Example

PhysicalQuantity(
 Unit = *MetreSquared*,
 Value = 46.8)

DesignHeatTransferCoefficient**Data Property**

The heat transfer coefficient for which the *HeatExchanger* is designed.

Type: *Undefined* | $\langle \textit{PhysicalQuantity}$ with *UnitType* $\rightarrow$ *HeatTransferCoefficientUnit* $\rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignHeatTransferCoefficient>

DESIGN HEAT TRANSFER COEFFICIENT

Example

PhysicalQuantity(
 Unit = *KilowattPerMetreSquaredKelvin*,
 Value = 1.2)

Example in DEXPI XML

```

<Object type="Plant/ProcessEquipment.HeatExchanger">
  <Data property="DesignHeatFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>313.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignHeatTransferArea">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.AreaUnit.MetreSquared"/>
      </Data>
      <Data property="Value">
        <Double>46.8</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignHeatTransferCoefficient">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.HeatTransferCoefficientUnit.
↪KilowattPerMetreSquaredKelvin"/>
      </Data>
      <Data property="Value">
        <Double>1.2</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>

```

HeatExchangerRotor

Plant::ProcessEquipment::HeatExchangerRotor

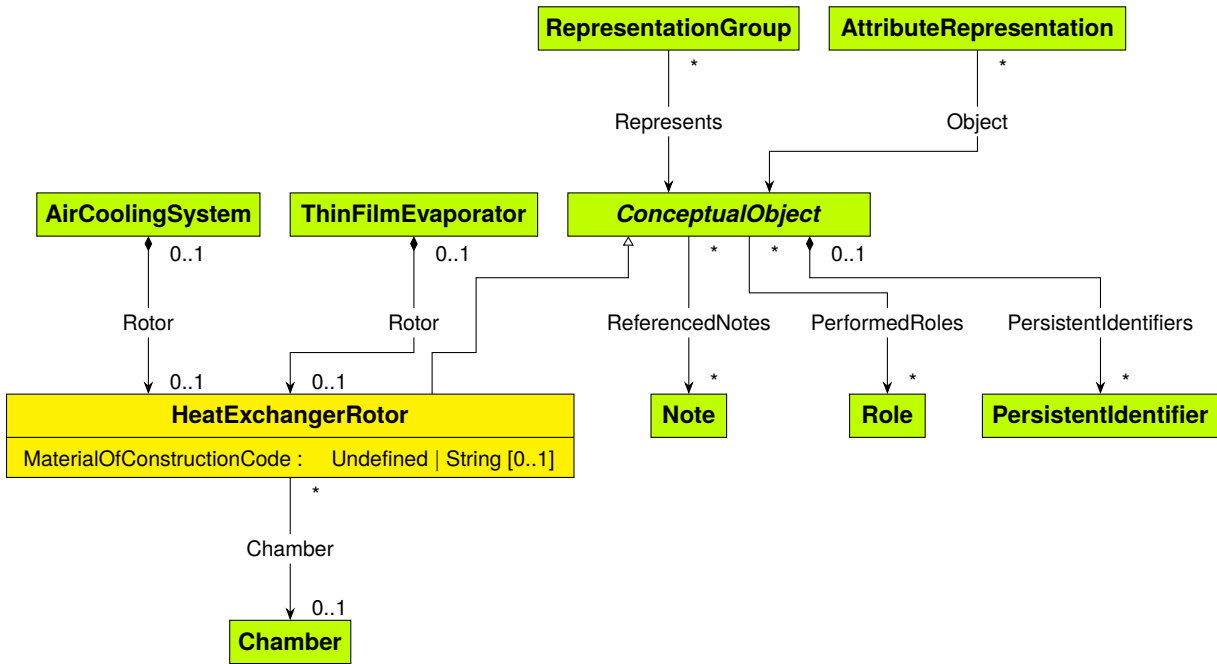
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/HeatExchangerRotor>

HEAT EXCHANGER ROTOR

A rotor as a component of a *HeatExchanger*.



Supertypes

- *ConceptualObject*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Chamber</i>	0..1	<i>Chamber</i>	no	no	*

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>MaterialOfConstructionCode</i>	0..1	<i>Undefined String</i>	no	no

Chamber

Reference Property

The *Chamber* in which the HeatExchangerRotor is located, if applicable. The Chamber must be a component of the same object as the HeatExchangerRotor.

Type: *Chamber*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

MaterialOfConstructionCode

Data Property

A code that gives the material of construction of the *HeatExchangerRotor*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS1460719741>

MATERIAL OF CONSTRUCTION CODE ASSIGNMENT CLASS

Example

"1.4306"

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.HeatExchangerRotor">
  <Data property="MaterialOfConstructionCode">
    <String>1.4306</String>
  </Data>
</Object>
```

HeatedSurfaceDryer

Plant::ProcessEquipment::HeatedSurfaceDryer

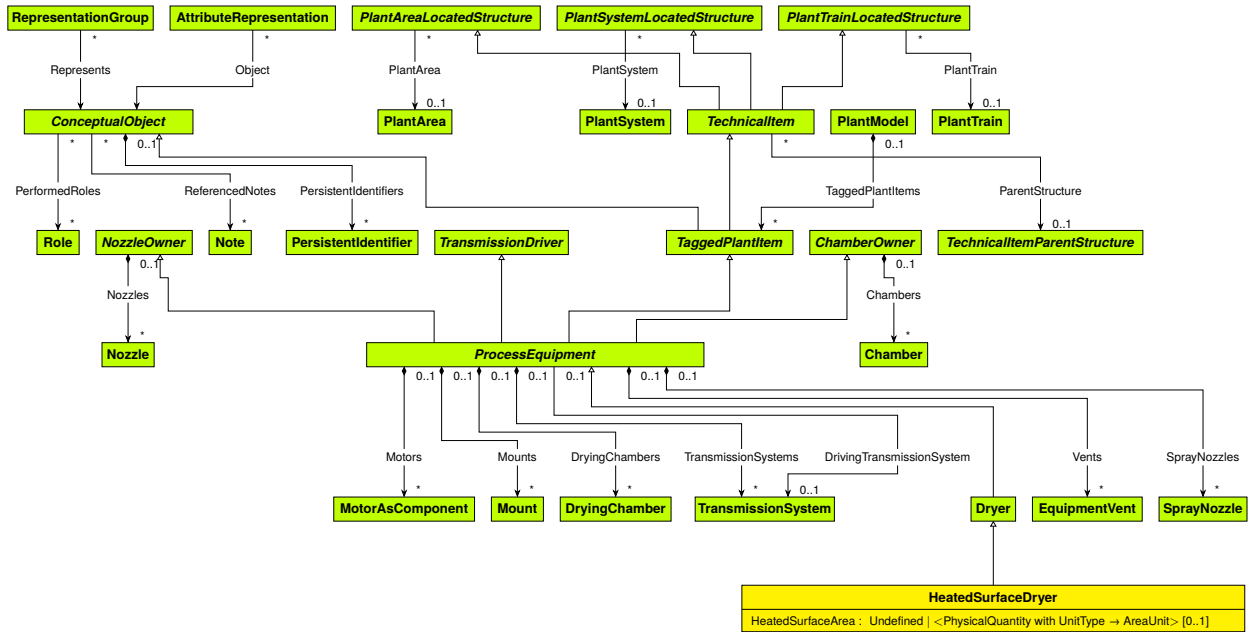
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/HeatedSurfaceDryer>

HEATED SURFACE DRYER

A *Dryer* that dries a material by radiation and/or conduction caused by a heated surface (from <http://data.15926.org/rdl/RDS2228449>).



Supertypes

- *Dryer*

Members

Data Properties

Name	Multiplic-ity	Type	Or-dered	Unique
<i>HeatedSur-faceArea</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → AreaU-nit></i>	no	no

HeatedSurfaceArea

Data Property

The heated surface area of the *HeatedSurfaceDryer*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → AreaUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/HeatedSurfaceArea>

HEATED SURFACE AREA

Example

```
PhysicalQuantity(
    Unit = MetreSquared,
    Value = 6.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.HeatedSurfaceDryer">
  <Data property="HeatedSurfaceArea">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.AreaUnit.MetreSquared"/>
      </Data>
      <Data property="Value">
        <Double>6.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

Heater

Plant::ProcessEquipment::Heater

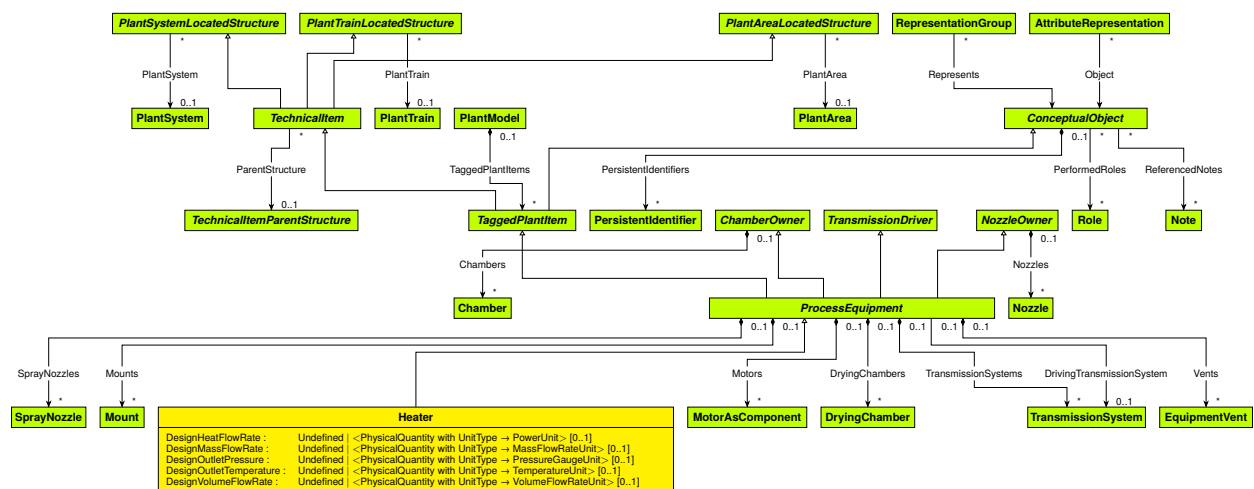
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS13048646>

HEATER

An apparatus or machine that has the capability of heating.



Supertypes

- *ProcessEquipment*

Subtypes

- *Boiler*
- *ElectricHeater*
- *Furnace*
- *SteamGenerator*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignHeatFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no
<i>DesignMassFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → MassFlowRateUnit></i>	no	no
<i>DesignOutletPressure</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PressureGaugeUnit></i>	no	no
<i>DesignOutletTemperature</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → TemperatureUnit></i>	no	no
<i>DesignVolumeFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VolumeFlowRateUnit></i>	no	no

DesignHeatFlowRate

Data Property

The heat flow rate for which the *Heater* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PowerUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignHeatFlowRate>

DESIGN HEAT FLOW RATE

Example

```
PhysicalQuantity(
  Unit = Kilowatt,
  Value = 313.0)
```

DesignMassFlowRate

Data Property

The mass flow rate for which the *Heater* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS14286182>

DESIGN MASS FLOW RATE

Example

```
PhysicalQuantity(  
    Unit = KilogramPerSecond,  
    Value = 420.0)
```

DesignOutletPressure

Data Property

The outlet pressure for which the *Heater* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PressureGaugeUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS7471401>

OUTLET DESIGN PRESSURE

Example

```
PhysicalQuantity(  
    Unit = Bar,  
    Value = -0.5)
```

DesignOutletTemperature

Data Property

The outlet temperature for which the *Heater* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{TemperatureUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS7471243>

OUTLET DESIGN TEMPERATURE

Example

```
PhysicalQuantity(
  Unit = DegreeCelsius,
  Value = -45.0)
```

DesignVolumeFlowRate

Data Property

The volume flow rate for which the *Heater* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VolumeFlowRateUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS14286227>

DESIGN VOLUME FLOW RATE

Example

```
PhysicalQuantity(
  Unit = MetreCubedPerHour,
  Value = 420.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Heater">
  <Data property="DesignHeatFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>313.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignMassFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerSecond"/>
      </Data>
      <Data property="Value">
        <Double>420.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

(continues on next page)

(continued from previous page)

```

    </Data>
  </AggregatedDataValue>
</Data>
<Data property="DesignOutletPressure">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.PressureGaugeUnit.Bar"/>
    </Data>
    <Data property="Value">
      <Double>-0.5</Double>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="DesignOutletTemperature">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.TemperatureUnit.DegreeCelsius"/>
    </Data>
    <Data property="Value">
      <Double>-45.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="DesignVolumeFlowRate">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour
→"/>
    </Data>
    <Data property="Value">
      <Double>420.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>

```

Impeller

Plant::ProcessEquipment::Impeller

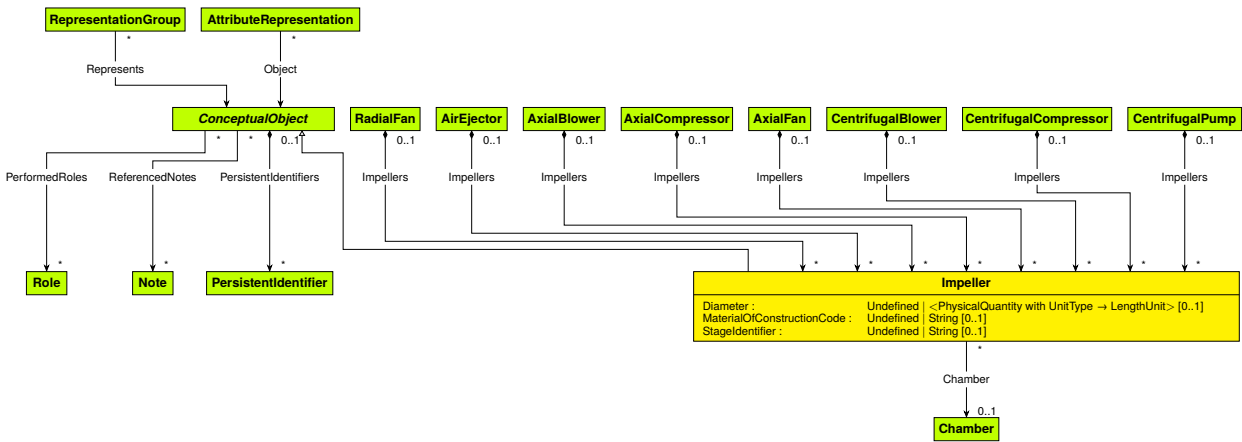
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS414539>

IMPELLER

An energy converter component that is an assembly of rotating vanes within an enclosure which is used to impart energy to or derive energy from a fluid through dynamic force (from <http://data.posccaesar.org/rdl/RDS414539>).



Supertypes

- *ConceptualObject*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Chamber</i>	0..1	<i>Chamber</i>	no	no	*

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Diameter</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no
<i>MaterialOfConstructionCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>StageIdentifier</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

Chamber

Reference Property

The *Chamber* in which the Impeller is located, if applicable. The Chamber must be a component of the same object as the Impeller.

Type: *Chamber*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

Diameter

Data Property

The diameter of the *Impeller*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with } \text{UnitType} \rightarrow \text{LengthUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS350954>

DIAMETER

Example

```
PhysicalQuantity(  
    Unit = Centimetre,  
    Value = 20.0)
```

MaterialOfConstructionCode

Data Property

A code that gives the material of construction of the *Impeller*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS1460719741>

MATERIAL OF CONSTRUCTION CODE ASSIGNMENT CLASS

Example

```
"1.4306"
```

StageIdentifier

Data Property

The stage identifier of the *Impeller*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/StageIdentifierAssignmentClass>

STAGE IDENTIFIER ASSIGNMENT CLASS

Example

"s1"

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Impeller">
  <Data property="Diameter">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
      </Data>
      <Data property="Value">
        <Double>20.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="MaterialOfConstructionCode">
    <String>1.4306</String>
  </Data>
  <Data property="StageIdentifier">
    <String>s1</String>
  </Data>
</Object>
```

InstrumentNozzle

Plant::ProcessEquipment::InstrumentNozzle

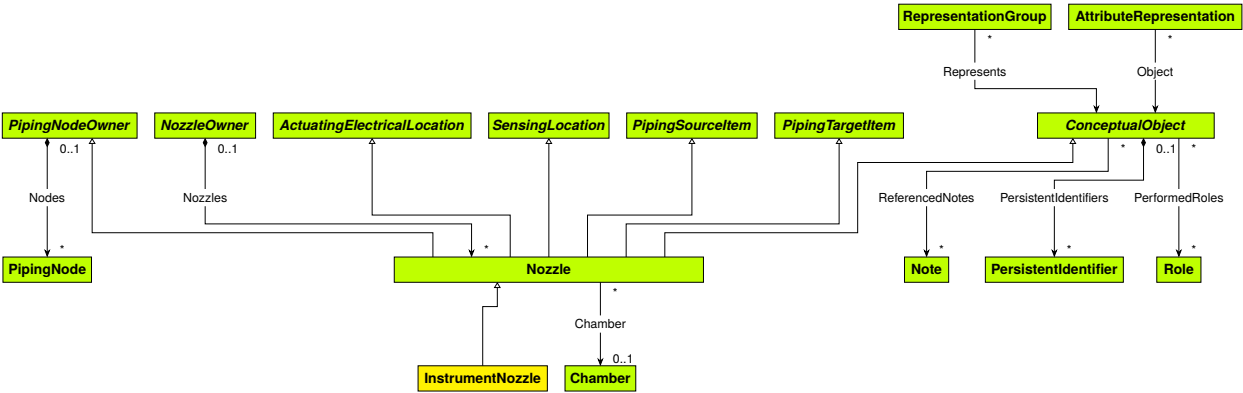
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS431675630>

INSTRUMENT NOZZLE

A *Nozzle* intended for connecting an instrument (from <http://data.posccaesar.org/rdl/RDS431675630>).



Supertypes

- *Nozzle*

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.InstrumentNozzle"/>
```

Kneader

Plant::ProcessEquipment::Kneader

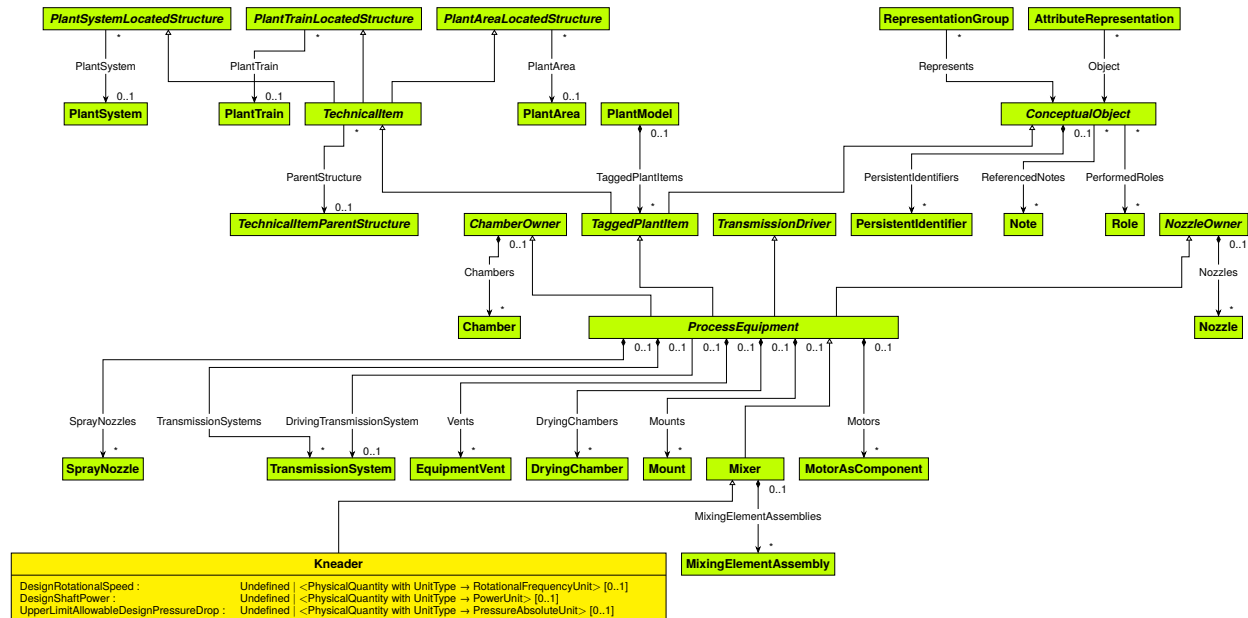
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/Kneader>

KNEADER

A machine that is capable of mixing and working into a uniform mass by, or as if by, folding, pressing, and stretching.



Supertypes

- *Mixer*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignRotationalSpeed</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → RotationalFrequencyUnit></i>	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no
<i>UpperLimitAllowableDesignPressureDrop</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PressureAbsoluteUnit></i>	no	no

DesignRotationalSpeed

Data Property

The rotational speed for which the *Kneader* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(
    Unit = ReciprocalMinute,
    Value = 180.0)
```

DesignShaftPower

Data Property

The shaft power for which the *Kneader* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PowerUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(
    Unit = Kilowatt,
```

Value = 400.0)

UpperLimitAllowableDesignPressureDrop

Data Property

The upper limit for the pressure drop for which the *Kneader* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PressureAbsoluteUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitAllowableDesignPressureDrop>

UPPER LIMIT ALLOWABLE DESIGN PRESSURE DROP

Example

PhysicalQuantity(

Unit = *Bar*,

Value = 2.0)

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Kneader">
  <Data property="DesignRotationalSpeed">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ReciprocalMinute"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignShaftPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>400.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="UpperLimitAllowableDesignPressureDrop">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PressureAbsoluteUnit.Bar"/>
      </Data>
```

(continues on next page)

(continued from previous page)

```
<Data property="Value">
  <Double>2.0</Double>
</Data>
</AggregatedDataValue>
</Data>
</Object>
```

Lift

Plant::ProcessEquipment::Lift

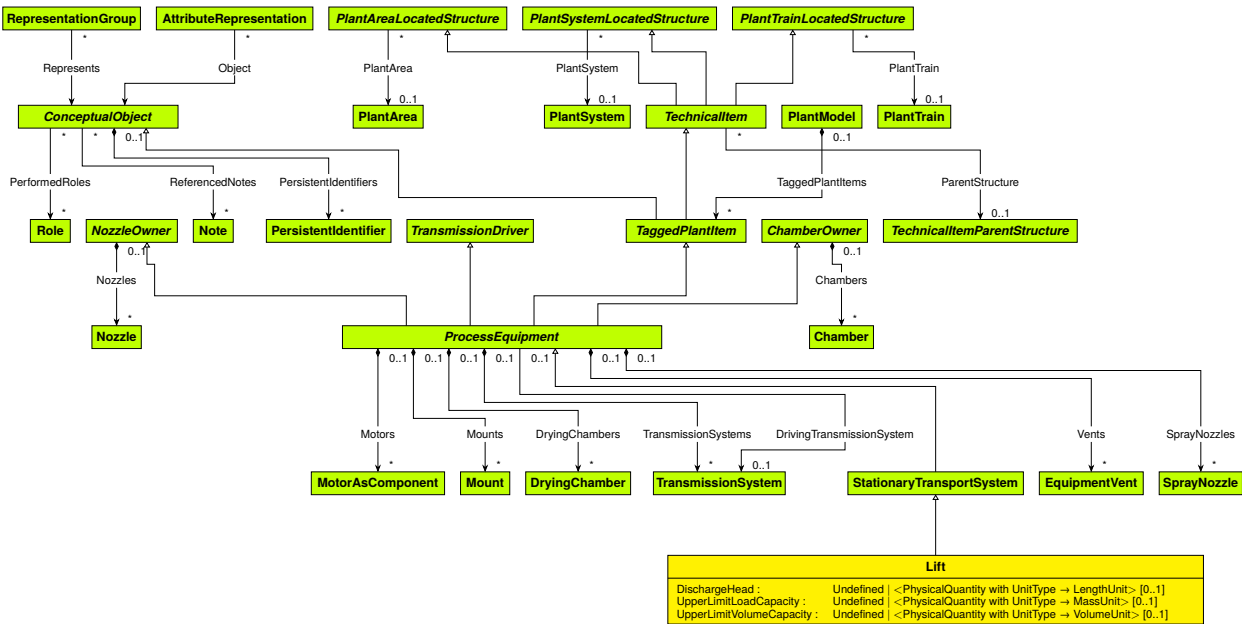
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS13601120>

LIFT

A *StationaryTransportSystem* for transporting persons or things from one level to another (from <http://data.posccaesar.org/rdl/RDS13601120>).



Supertypes

- *StationaryTransportSystem*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DischargeHead</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no
<i>UpperLimitLoadCapacity</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → MassUnit></i>	no	no
<i>UpperLimitVolumeCapacity</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VolumeUnit></i>	no	no

DischargeHead

Data Property

The length of the *Lift*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → LengthUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DischargeHead>

DISCHARGE HEAD

Example

```
PhysicalQuantity(
  Unit = Metre,
  Value = 2.0)
```

UpperLimitLoadCapacity

Data Property

The highest mass to transport for which the *Lift* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → MassUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitLoadCapacity>

UPPER LIMIT LOAD CAPACITY

Example

```
PhysicalQuantity(
  Unit = Kilogram,
```


Value = 900.0)

UpperLimitVolumeCapacity

Data Property

The highest volume to transport for which the *Lift* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VolumeUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitVolumeCapacity>

UPPER LIMIT VOLUME CAPACITY

Example

PhysicalQuantity(
 Unit = *MetreCubed*,
 Value = 4.0)

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Lift">
  <Data property="DischargeHead">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Metre"/>
      </Data>
      <Data property="Value">
        <Double>2.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="UpperLimitLoadCapacity">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.MassUnit.Kilogram"/>
      </Data>
      <Data property="Value">
        <Double>900.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="UpperLimitVolumeCapacity">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.VolumeUnit.MetreCubed"/>
      </Data>
      <Data property="Value">
```

(continues on next page)

(continued from previous page)

```

    <Double>4.0</Double>
  </Data>
</AggregatedDataValue>
</Data>
</Object>

```

LiquidFilter

Plant::ProcessEquipment::LiquidFilter

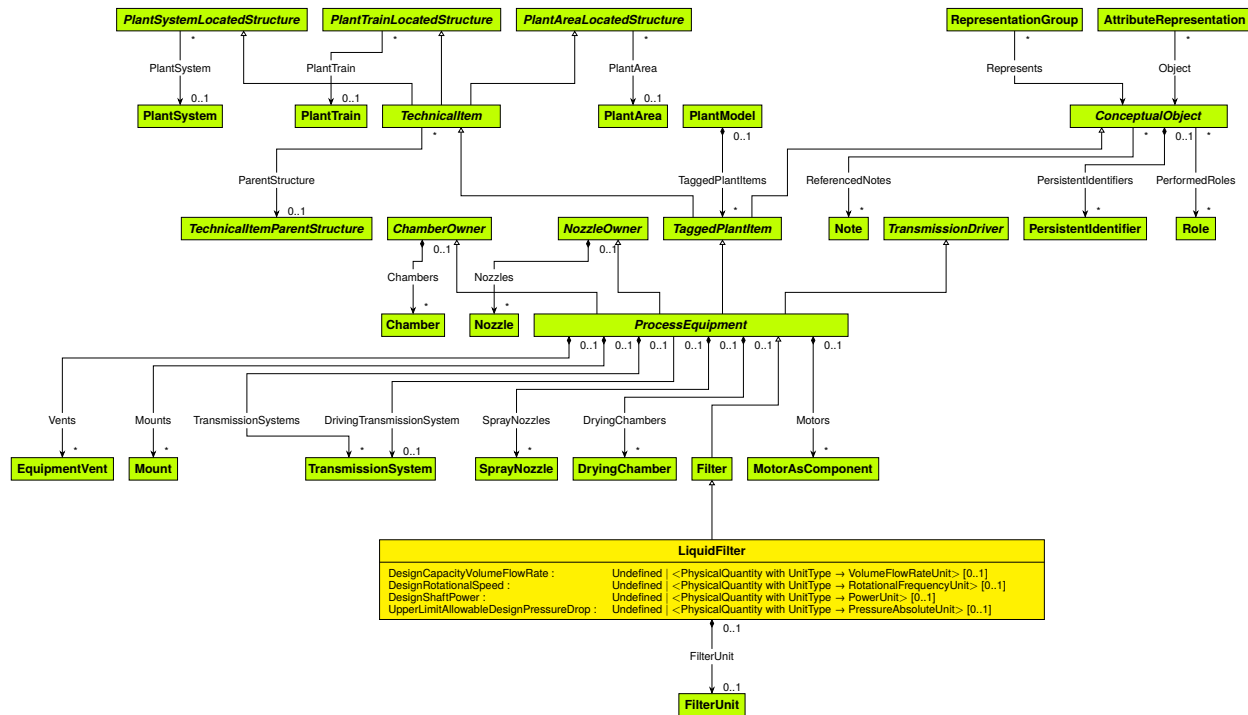
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/LiquidFilter>

LIQUID FILTER

A filter that is specifically designed to filter a liquid.



Supertypes

- *Filter*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>FilterUnit</i>	0..1	<i>FilterUnit</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignCapacityVolumeFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VolumeFlowRateUnit></i>	no	no
<i>DesignRotationalSpeed</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → RotationalFrequencyUnit></i>	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no
<i>UpperLimitAllowableDesignPressureDrop</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PressureAbsoluteUnit></i>	no	no

DesignCapacityVolumeFlowRate

Data Property

The volume flow rate for which the *LiquidFilter* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → VolumeFlowRateUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignCapacityVolumeFlowRate>

DESIGN CAPACITY VOLUME FLOW RATE

Example

```
PhysicalQuantity(
    Unit = MetreCubedPerHour,
    Value = 420.0)
```

DesignRotationalSpeed

Data Property

The rotational speed for which the *LiquidFilter* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

PhysicalQuantity(
 Unit = *ReciprocalMinute*,
 Value = 180.0)

DesignShaftPower**Data Property**

The shaft power for which the *LiquidFilter* is designed.

Type: *Undefined* | $\langle \textit{PhysicalQuantity}$ with *UnitType* $\rightarrow$ *PowerUnit* $\rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

PhysicalQuantity(
 Unit = *Kilowatt*,
 Value = 400.0)

FilterUnit**Composition Property**

The filter unit of the *LiquidFilter*.

Type: *FilterUnit*

Multiplicity: 0..1

Ordered: no

UpperLimitAllowableDesignPressureDrop**Data Property**

The upper limit for the pressure drop for which the *LiquidFilter* is designed.

Type: *Undefined* | $\langle \textit{PhysicalQuantity}$ with *UnitType* $\rightarrow$ *PressureAbsoluteUnit* $\rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitAllowableDesignPressureDrop>

UPPER LIMIT ALLOWABLE DESIGN PRESSURE DROP

Example

PhysicalQuantity(

Unit = *Bar*,

Value = 2.0)

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.LiquidFilter">
  <Data property="DesignCapacityVolumeFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour
↪"/>
      </Data>
      <Data property="Value">
        <Double>420.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignRotationalSpeed">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ReciprocalMinute"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignShaftPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>400.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="UpperLimitAllowableDesignPressureDrop">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PressureAbsoluteUnit.Bar"/>
      </Data>
      <Data property="Value">
        <Double>2.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

LoadingUnloadingSystem

Plant::ProcessEquipment::LoadingUnloadingSystem

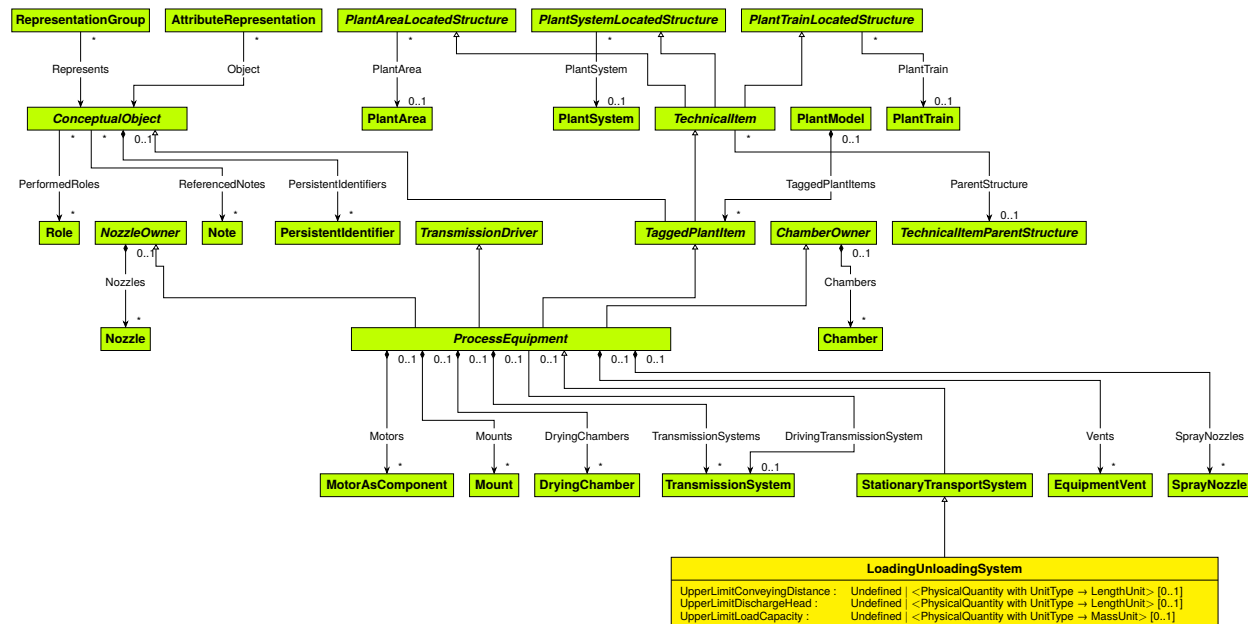
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS11525012>

LOADING - UNLOADING SYSTEM

A transport system that is intended for loading and/or unloading products into/from vehicles, wagons or vessels (from <http://data.posccaesar.org/rdl/RDS11525012>).



Supertypes

- *StationaryTransportSystem*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>UpperLimitConveyingDistance</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no
<i>UpperLimitDischargeHead</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no
<i>UpperLimitLoadCapacity</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → MassUnit></i>	no	no

UpperLimitConveyingDistance

Data Property

The upper limit for the conveying distance of the *LoadingUnloadingSystem*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitConveyingDistance>

UPPER LIMIT CONVEYING DISTANCE

Example

```
PhysicalQuantity(
  Unit = Centimetre,
  Value = 37.0)
```

UpperLimitDischargeHead

Data Property

The upper limit for the discharge head of the *LoadingUnloadingSystem*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitDischargeHead>

UPPER LIMIT DISCHARGE HEAD

Example

```
PhysicalQuantity(
  Unit = Centimetre,
  Value = 37.0)
```

UpperLimitLoadCapacity

Data Property

The highest mass to transport for which the *LoadingUnloadingSystem* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitLoadCapacity>

UPPER LIMIT LOAD CAPACITY

Example

```
PhysicalQuantity(  
    Unit = Kilogram,  
    Value = 900.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.LoadingUnloadingSystem">  
  <Data property="UpperLimitConveyingDistance">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>  
      </Data>  
      <Data property="Value">  
        <Double>37.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
  <Data property="UpperLimitDischargeHead">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>  
      </Data>  
      <Data property="Value">  
        <Double>37.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
  <Data property="UpperLimitLoadCapacity">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.MassUnit.Kilogram"/>  
      </Data>  
      <Data property="Value">  
        <Double>900.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
</Object>
```

MechanicalSeparator

Plant::ProcessEquipment::MechanicalSeparator

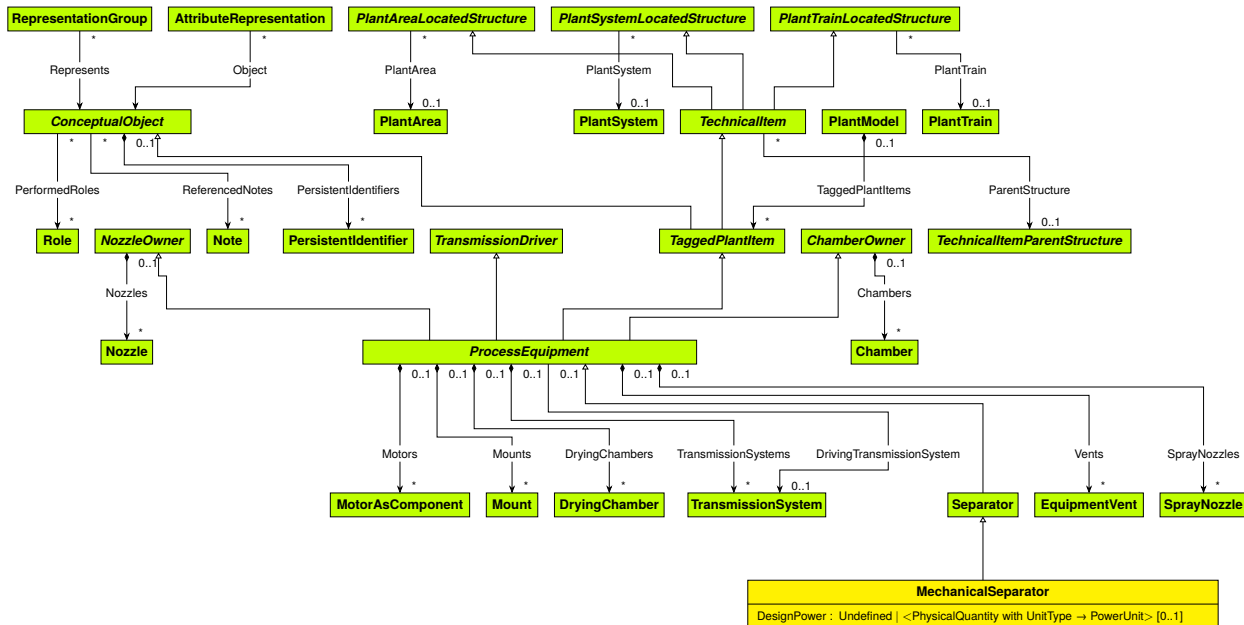
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS279134>

MECHANICAL SEPARATOR

A fluid separator in which mechanical separation of fluids take place (from <http://data.posccaesar.org/rdl/RDS279134>).



Supertypes

- *Separator*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType -> PowerUnit></i>	no	no

DesignPower

Data Property

The power for which the *MechanicalSeparator* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType -> PowerUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignPower>

DESIGN POWER

Example

```
PhysicalQuantity(
  Unit = Kilowatt,
  Value = 500.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.MechanicalSeparator">
  <Data property="DesignPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>500.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

Mill

Plant::ProcessEquipment::Mill

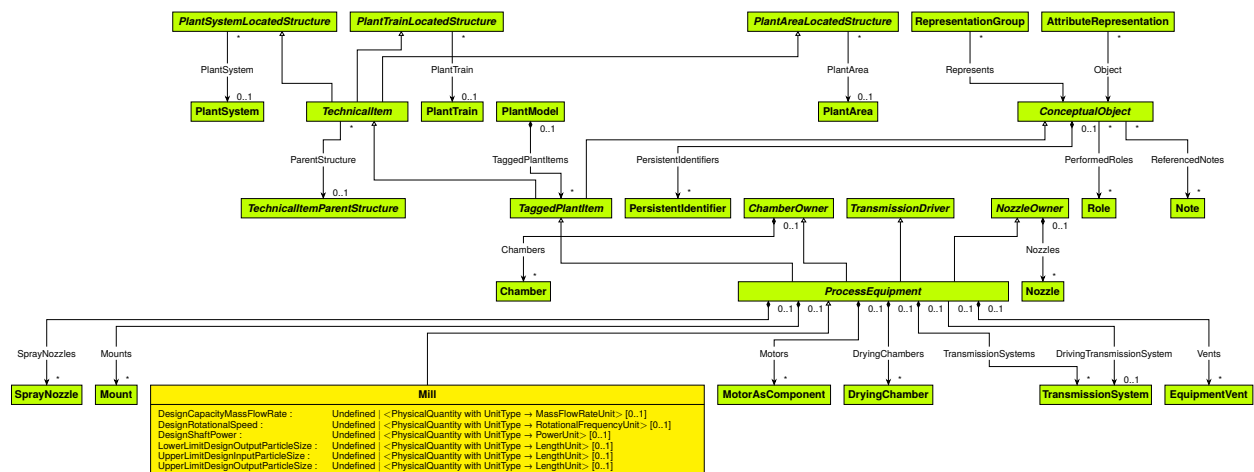
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS11589220>

MILL

A physical object for grinding or pulverizing materials. Also a machine for shaping metal. In general a machine that manufactures by the continuous repetition of some simple action



Supertypes

- *ProcessEquipment*

Subtypes

- *Crusher*
- *Grinder*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignCapacityMassFlowRate</i>	0..1	<i>Undefined</i> < <i>PhysicalQuantity</i> with <i>UnitType</i> → <i>MassFlowRateUnit</i> >	no	no
<i>DesignRotationalSpeed</i>	0..1	<i>Undefined</i> < <i>PhysicalQuantity</i> with <i>UnitType</i> → <i>RotationalFrequencyUnit</i> >	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> < <i>PhysicalQuantity</i> with <i>UnitType</i> → <i>PowerUnit</i> >	no	no
<i>LowerLimitDesignOutputParticleSize</i>	0..1	<i>Undefined</i> < <i>PhysicalQuantity</i> with <i>UnitType</i> → <i>LengthUnit</i> >	no	no
<i>UpperLimitDesignInputParticleSize</i>	0..1	<i>Undefined</i> < <i>PhysicalQuantity</i> with <i>UnitType</i> → <i>LengthUnit</i> >	no	no
<i>UpperLimitDesignOutputParticleSize</i>	0..1	<i>Undefined</i> < <i>PhysicalQuantity</i> with <i>UnitType</i> → <i>LengthUnit</i> >	no	no

DesignCapacityMassFlowRate

Data Property

The capacity for the mass flow rate for which the *Mill* is designed.

Type: *Undefined* | <*PhysicalQuantity* with *UnitType* → *MassFlowRateUnit*>

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignCapacityMassFlowRate>

DESIGN CAPACITY MASS FLOW RATE

Example

PhysicalQuantity(

Unit = *KilogramPerMinute*,

Value = 40.0)

DesignRotationalSpeed

Data Property

The rotational speed for which the *Mill* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{RotationalFrequencyUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(  
    Unit = ReciprocalMinute,  
    Value = 180.0)
```

DesignShaftPower

Data Property

The shaft power for which the *Mill* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(  
    Unit = Kilowatt,  
    Value = 400.0)
```

LowerLimitDesignOutputParticleSize

Data Property

The lower limit for the output particle size for which the *Mill* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LowerLimitDesignOutputParticleSize>

LOWER LIMIT DESIGN OUTPUT PARTICLE SIZE

Example

```
PhysicalQuantity(
  Unit = Micrometre,
  Value = 7.0)
```

UpperLimitDesignInputParticleSize

Data Property

The upper limit for the input particle size for which the *Mill* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitDesignInputParticleSize>

UPPER LIMIT DESIGN INPUT PARTICLE SIZE

Example

```
PhysicalQuantity(
  Unit = Micrometre,
  Value = 47.0)
```

UpperLimitDesignOutputParticleSize

Data Property

The upper limit for the output particle size for which the *Mill* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitDesignOutputParticleSize>

UPPER LIMIT DESIGN OUTPUT PARTICLE SIZE

Example

```
PhysicalQuantity(
  Unit = Micrometre,
  Value = 47.0)
```

Example in DEXPI XML

```

<Object type="Plant/ProcessEquipment.Mill">
  <Data property="DesignCapacityMassFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerMinute"/>
      </Data>
      <Data property="Value">
        <Double>40.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignRotationalSpeed">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ ReciprocalMinute"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignShaftPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>400.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="LowerLimitDesignOutputParticleSize">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Micrometre"/>
      </Data>
      <Data property="Value">
        <Double>7.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="UpperLimitDesignInputParticleSize">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Micrometre"/>
      </Data>
      <Data property="Value">
        <Double>47.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>

```

(continues on next page)

(continued from previous page)

```

    </AggregatedDataValue>
  </Data>
  <Data property="UpperLimitDesignOutputParticleSize">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Micrometre"/>
      </Data>
      <Data property="Value">
        <Double>47.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>

```

Mixer

Plant::ProcessEquipment::Mixer

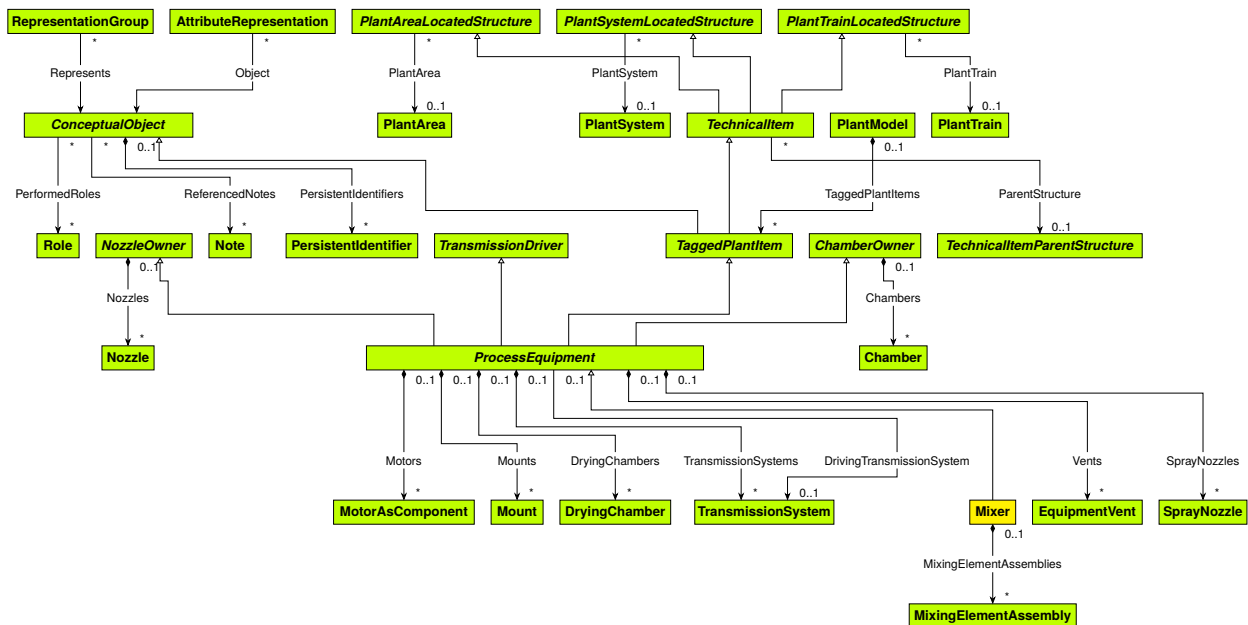
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/Mixer>

MIXER

An apparatus or machine that has the capability of mixing (from <http://data.15926.org/rdl/RDS222370>).



Supertypes

- *ProcessEquipment*

Subtypes

- *Kneader*
- *RotaryMixer*
- *StaticMixer*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>MixingElementAssemblies</i>	*	<i>MixingElementAssembly</i>	no

MixingElementAssemblies

Composition Property

The mixing element assemblies of the Mixer, if applicable.

Type: *MixingElementAssembly*

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Mixer"/>
```

MixingElementAssembly

Plant::ProcessEquipment::MixingElementAssembly

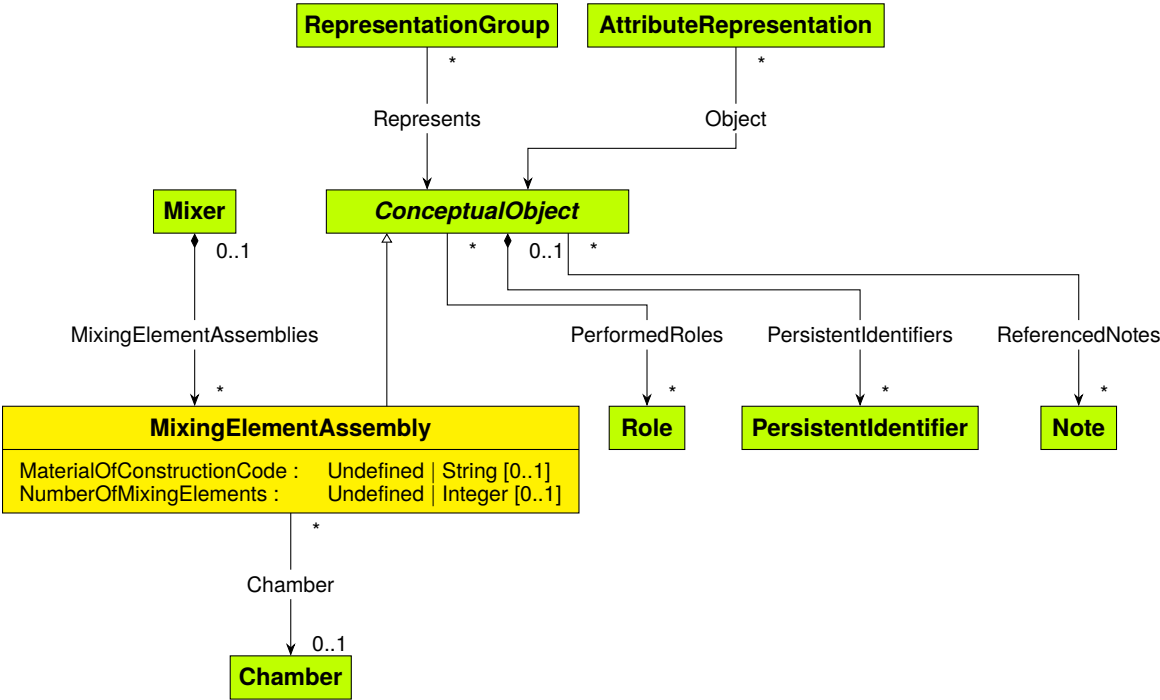
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/MixingElementAssembly>

MIXING ELEMENT ASSEMBLY

Assembly of mixing elements as part of a mixer.



Supertypes

- *ConceptualObject*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Chamber</i>	0..1	<i>Chamber</i>	no	no	*

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>MaterialOfConstructionCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>NumberOfMixingElements</i>	0..1	<i>Undefined</i> <i>Integer</i>	no	no

Chamber

Reference Property

The *Chamber* in which the *MixingElementAssembly* is located, if applicable. The *Chamber* must be a component of the same object as the *MixingElementAssembly*.

Type: *Chamber*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

MaterialOfConstructionCode

Data Property

A code that gives the material of construction of the *MixingElementAssembly*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS1460719741>

MATERIAL OF CONSTRUCTION CODE ASSIGNMENT CLASS

Example

"1.4306"

NumberOfMixingElements

Data Property

The number of mixing elements in the *MixingElementAssembly*.

Type: *Undefined* | *Integer*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NumberOfMixingElements>

NUMBER OF MIXING ELEMENTS

Example

5

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.MixingElementAssembly">
  <Data property="MaterialOfConstructionCode">
    <String>1.4306</String>
  </Data>
  <Data property="NumberOfMixingElements">
    <Integer>5</Integer>
  </Data>
</Object>
```

MobileTransportSystem

Plant::ProcessEquipment::MobileTransportSystem

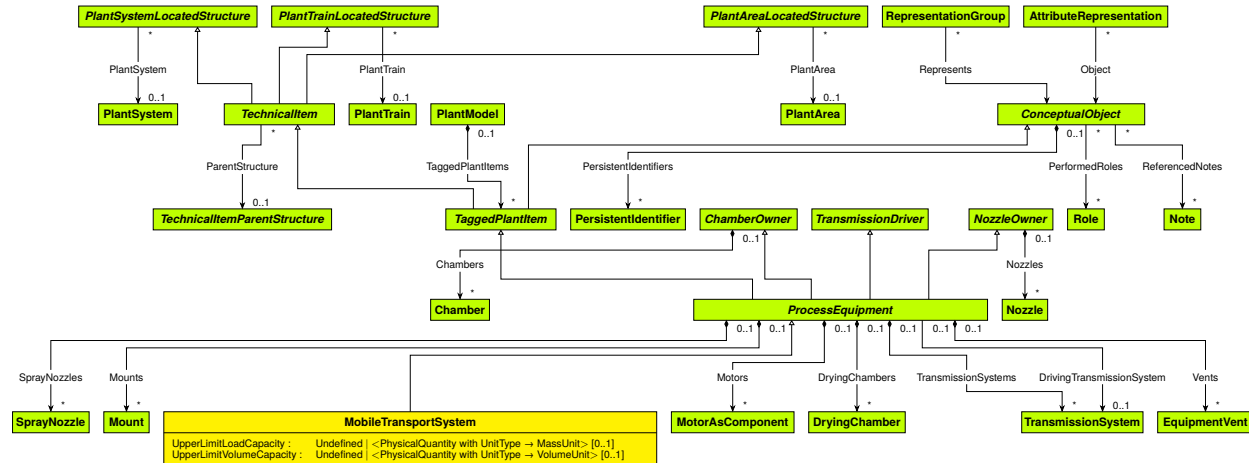
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/MobileTransportSystem>

MOBILE TRANSPORT SYSTEM

A mobile system that is intended to transport, store or load/unload material.



Supertypes

- *ProcessEquipment*

Subtypes

- *ForkliftTruck*
- *RailWaggon*
- *Ship*
- *TransportableContainer*
- *Truck*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>UpperLimitLoadCapacity</i>	0..1	Undefined <PhysicalQuantity with UnitType → MassUnit>	no	no
<i>UpperLimitVolumeCapacity</i>	0..1	Undefined <PhysicalQuantity with UnitType → VolumeUnit>	no	no

UpperLimitLoadCapacity

Data Property

The highest mass to transport for which the *MobileTransportSystem* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with } \text{UnitType} \rightarrow \text{MassUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitLoadCapacity>

UPPER LIMIT LOAD CAPACITY

Example

```
PhysicalQuantity(  
  Unit = Kilogram,  
  Value = 900.0)
```

UpperLimitVolumeCapacity

Data Property

The highest volume to transport for which the *MobileTransportSystem* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with } \text{UnitType} \rightarrow \text{VolumeUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitVolumeCapacity>

UPPER LIMIT VOLUME CAPACITY

Example

```
PhysicalQuantity(  
  Unit = MetreCubed,  
  Value = 4.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.MobileTransportSystem">  
  <Data property="UpperLimitLoadCapacity">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.MassUnit.Kilogram"/>  
      </Data>  
      <Data property="Value">  
        <Double>900.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
</Object>
```

(continues on next page)

(continued from previous page)

```

    </Data>
  </AggregatedDataValue>
</Data>
<Data property="UpperLimitVolumeCapacity">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.VolumeUnit.MetreCubed"/>
    </Data>
    <Data property="Value">
      <Double>4.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>

```

Motor

Plant::ProcessEquipment::Motor

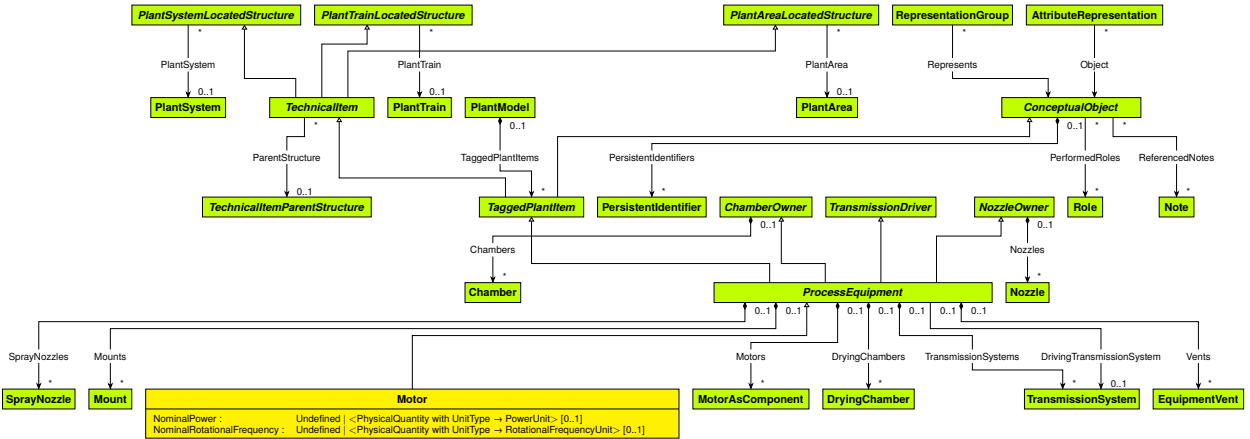
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS7191198>

MOTOR

A driver that is powered by electricity or internal combustion (from <http://data.15926.org/rdl/RDS7191198>).



Supertypes

- *ProcessEquipment*

Subtypes

- *AlternatingCurrentMotor*
- *CombustionEngine*
- *DirectCurrentMotor*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>NominalPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no
<i>NominalRotational-Frequency</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → RotationalFrequencyUnit></i>	no	no

NominalPower

Data Property

The nominal power of the *Motor*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PowerUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalPower>

NOMINAL POWER

Example

```
PhysicalQuantity(  
  Unit = Kilowatt,  
  Value = 400.0)
```

NominalRotationalFrequency

Data Property

The nominal rotational frequency of the *Motor*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalRotationalFrequency>

NOMINAL ROTATIONAL FREQUENCY

Example

```
PhysicalQuantity(  
  Unit = ReciprocalMinute,
```

Value = 240.0)

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Motor">
  <Data property="NominalPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>400.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="NominalRotationalFrequency">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ReciprocalMinute"/>
      </Data>
      <Data property="Value">
        <Double>240.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

MotorAsComponent

Plant::ProcessEquipment::MotorAsComponent

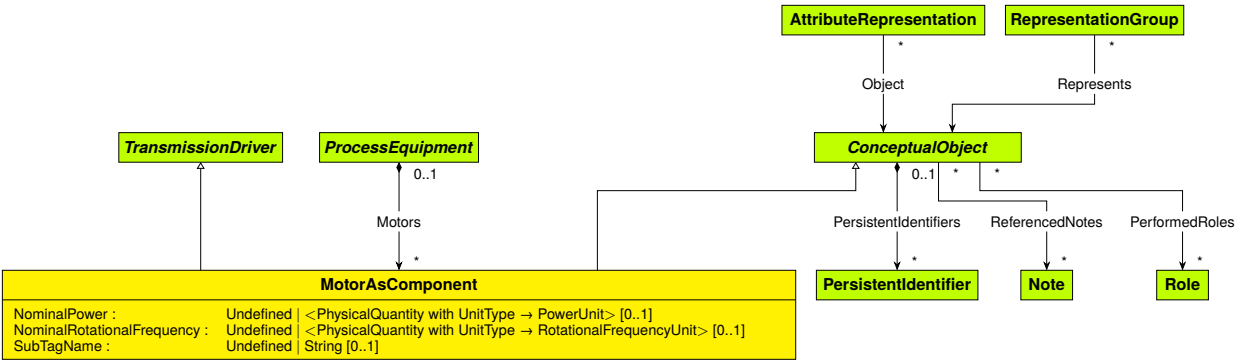
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/MotorAsComponent>

MOTOR AS COMPONENT

A driver that is powered by electricity or internal combustion and is used as component of an apparatus or of a machine.



Supertypes

- *ConceptualObject*
- *TransmissionDriver*

Subtypes

- *AlternatingCurrentMotorAsComponent*
- *CombustionEngineAsComponent*
- *DirectCurrentMotorAsComponent*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>NominalPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no
<i>NominalRotational-Frequency</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → RotationalFrequencyUnit></i>	no	no
<i>SubTagName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

NominalPower

Data Property

The nominal power of the *MotorAsComponent*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PowerUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalPower>

NOMINAL POWER

Example

```
PhysicalQuantity(  
  Unit = Kilowatt,  
  Value = 400.0)
```

NominalRotationalFrequency

Data Property

The nominal rotational frequency of the *MotorAsComponent*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1**Ordered:** no**Unique:** no**RDL:** <http://sandbox.dexpi.org/rdl/NominalRotationalFrequency>

NOMINAL ROTATIONAL FREQUENCY

Example

PhysicalQuantity(
 Unit = *ReciprocalMinute*,
 Value = 240.0)

SubTagName**Data Property**The sub tag name of the *MotorAsComponent*.**Type:** *Undefined* | *String***Multiplicity:** 0..1**Ordered:** no**Unique:** no**RDL:** <http://sandbox.dexpi.org/rdl/SubTagNameAssignmentClass>

SUB TAG NAME ASSIGNMENT CLASS

Example

"ST1"

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.MotorAsComponent">
  <Data property="NominalPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>400.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="NominalRotationalFrequency">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ ReciprocalMinute"/>
      </Data>
```

(continues on next page)

(continued from previous page)

```

<Data property="Value">
  <Double>240.0</Double>
</Data>
</AggregatedDataValue>
</Data>
<Data property="SubTagName">
  <String>ST1</String>
</Data>
</Object>

```

Mount

Plant::ProcessEquipment::Mount

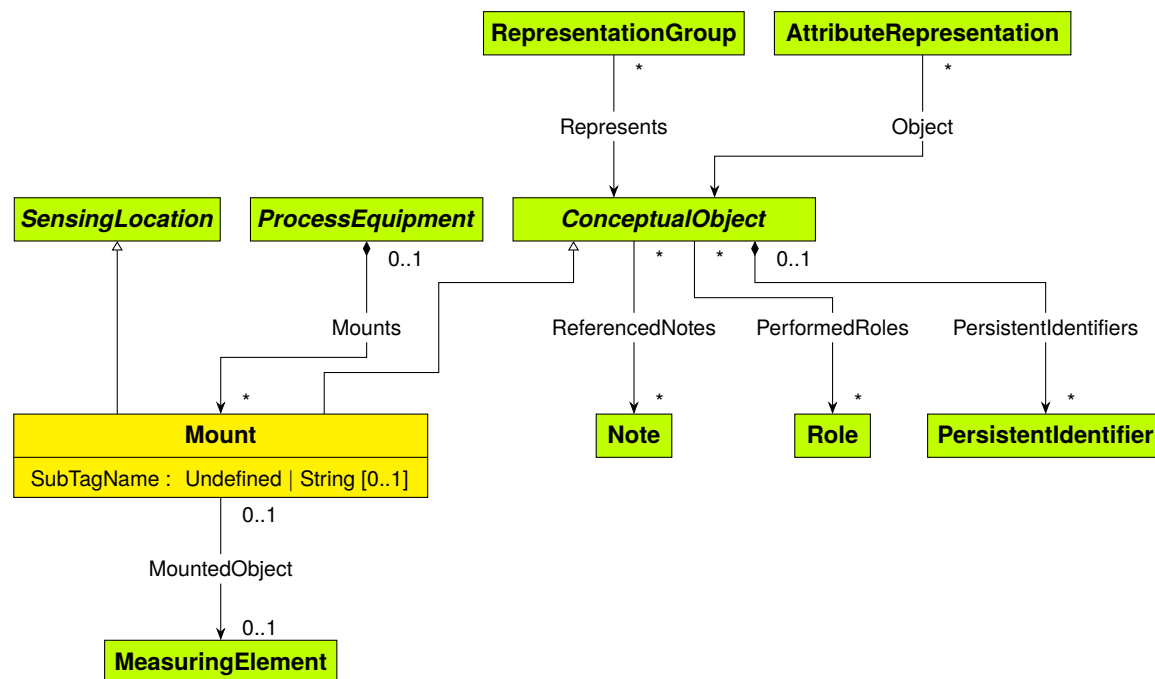
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/Mount>

MOUNT

An object that is capable of supporting something that is mounted on it.



Supertypes

- *ConceptualObject*
- *SensingLocation*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>MountedObject</i>	0..1	<i>MeasuringElement</i>	no	no	0..1

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>SubTagName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

A mount is attached to a piece of Plant.ProcessEquipment.Equipment and can explicitly be used as a *SensingLocation*.

MountedObject

Reference Property

The object that is mounted.

Type: *MeasuringElement*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

SubTagName

Data Property

The sub tag name of the *Mount*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SubTagNameAssignmentClass>

SUB TAG NAME ASSIGNMENT CLASS

Example

"ST1"

Example in DEXPI XML

```

<Object type="Plant/ProcessEquipment.Mount">
  <Data property="SubTagName">
    <String>ST1</String>
  </Data>
</Object>

```

Nozzle

Plant::ProcessEquipment::Nozzle

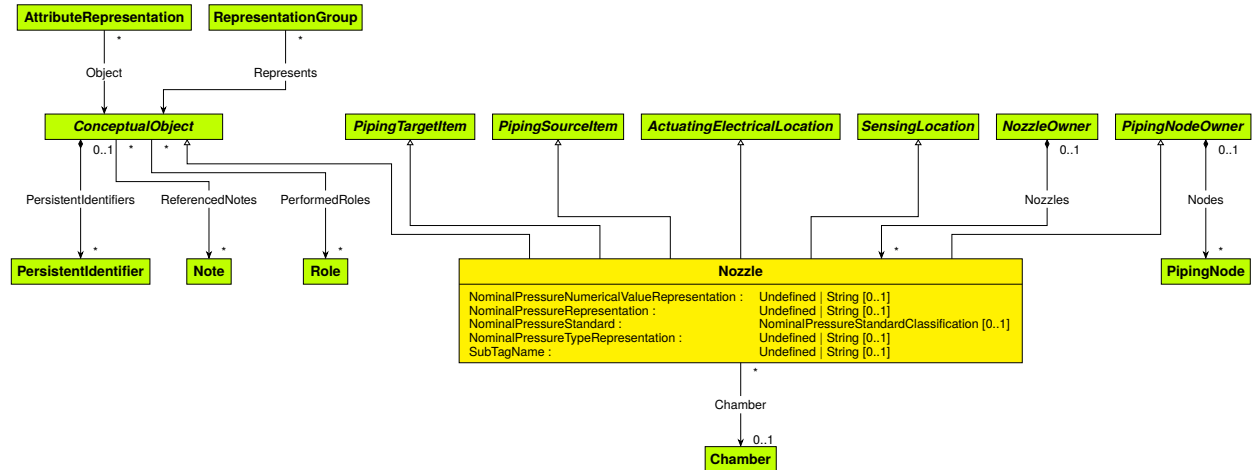
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS415214>

NOZZLE

A projecting short piece of pipe that is attached to another (hollow) body.



Supertypes

- *ActuatingElectricalLocation*
- *ConceptualObject*
- *PipingNodeOwner*
- *PipingSourceItem*
- *PipingTargetItem*
- *SensingLocation*

Subtypes

- *AccessNozzle*
- *InstrumentNozzle*
- *ProcessNozzle*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Chamber</i>	0..1	<i>Chamber</i>	no	no	*

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>NominalPressureNumericalValueRepresentation</i>	0..1	<i>Undefined String</i>	no	no
<i>NominalPressureRepresentation</i>	0..1	<i>Undefined String</i>	no	no
<i>NominalPressureStandard</i>	0..1	<i>NominalPressureStandardClassification</i>	no	no
<i>NominalPressureTypeRepresentation</i>	0..1	<i>Undefined String</i>	no	no
<i>SubTagName</i>	0..1	<i>Undefined String</i>	no	no

Chamber

Reference Property

The *Chamber* at which the Nozzle is located, if applicable. The Chamber must be a component of the same object as the Nozzle.

Type: *Chamber*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

NominalPressureNumericalValueRepresentation

Data Property

A readable representation of the numerical value of the nominal pressure of the *Nozzle*, without any type or unit of measure.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalPressureNumericalValueRepresentationAssignmentClass>

NOMINAL PRESSURE NUMERICAL VALUE REPRESENTATION ASSIGNMENT CLASS

Example

"40"

NominalPressureRepresentation

Data Property

A readable representation of the nominal pressure of the *Nozzle*. It normally contains a numerical value and a type or unit of measure.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalPressureRepresentationAssignmentClass>

NOMINAL PRESSURE REPRESENTATION ASSIGNMENT CLASS

Example

"PN 40"

NominalPressureStandard

Data Property

The nominal pressure of the *Nozzle*, given as a reference to a nominal pressure standard and value.

Type: *NominalPressureStandardClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalPressureStandardSpecialization>

NOMINAL PRESSURE STANDARD SPECIALIZATION

Example

En1333Pn40Artefact

NominalPressureTypeRepresentation

Data Property

A readable representation of the type or unit of measure of the nominal pressure of the *Nozzle*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalPressureTypeRepresentationAssignmentClass>

NOMINAL PRESSURE TYPE REPRESENTATION ASSIGNMENT CLASS

Example

"PN"

SubTagName**Data Property**

The sub tag name of the *Nozzle*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SubTagNameAssignmentClass>

SUB TAG NAME ASSIGNMENT CLASS

Example

"ST1"

Example in DEXPI XML

```

<Object type="Plant/ProcessEquipment.Nozzle">
  <Data property="NominalPressureNumericalValueRepresentation">
    <String>40</String>
  </Data>
  <Data property="NominalPressureRepresentation">
    <String>PN 40</String>
  </Data>
  <Data property="NominalPressureStandard">
    <DataReference data="Plant/Enumerations.NominalPressureStandardClassification.
→ En1333Pn40Artefact"/>
  </Data>
  <Data property="NominalPressureTypeRepresentation">
    <String>PN</String>
  </Data>
  <Data property="SubTagName">
    <String>ST1</String>
  </Data>
</Object>

```

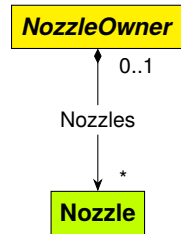
NozzleOwner

Plant::ProcessEquipment::NozzleOwner

Overview

Abstract Class

An object that can have nozzles.



Subtypes

- *ProcessEquipment*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Nozzles</i>	*	<i>Nozzle</i>	no

Nozzles

Composition Property

The nozzles of the NozzleOwner.

Type: *Nozzle*

Multiplicity: *

Ordered: no

PackagingSystem

Plant::ProcessEquipment::PackagingSystem

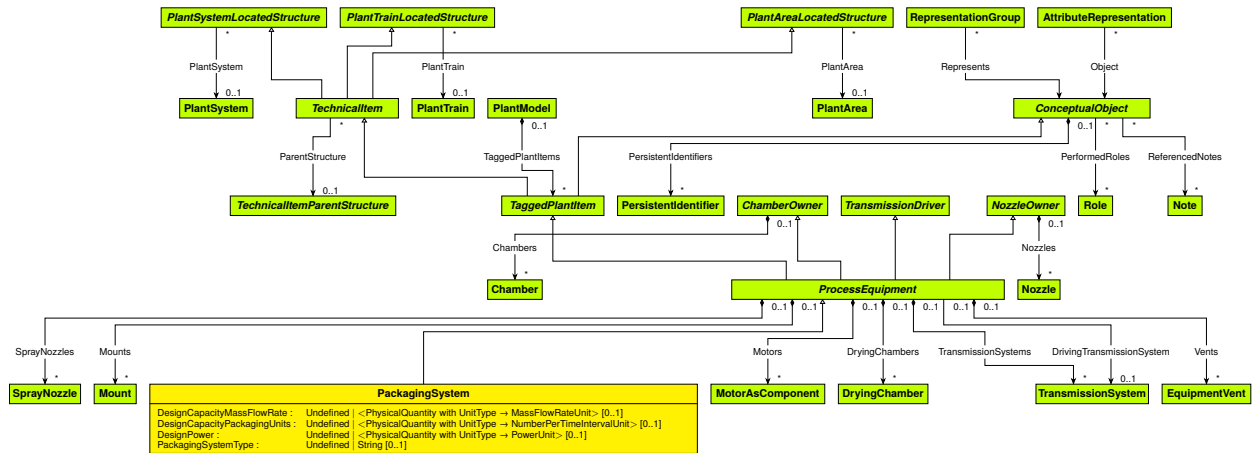
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/PackagingSystem>

PACKAGING SYSTEM

A system that is intended for the preparation of goods for transport, warehousing, logistics, sale, and end use (from <http://data.15926.org/rdl/RDS2228725>).



Supertypes

- *ProcessEquipment*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignCapacityMassFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → MassFlowRateUnit></i>	no	no
<i>DesignCapacityPackagingUnits</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → NumberPerTimeIntervalUnit></i>	no	no
<i>DesignPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no
<i>PackagingSystemType</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

DesignCapacityMassFlowRate

Data Property

The capacity for the mass flow rate for which the *PackagingSystem* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → MassFlowRateUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignCapacityMassFlowRate>

DESIGN CAPACITY MASS FLOW RATE

Example

```
PhysicalQuantity(  
  Unit = KilogramPerMinute,  
  Value = 40.0)
```

DesignCapacityPackagingUnits**Data Property**

The capacity for the number of packaging units per time for which the *PackagingSystem* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{NumberPerTimeIntervalUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignCapacityPackagingUnits>

DESIGN CAPACITY PACKAGING UNITS

Example

```
PhysicalQuantity(  
  Unit = ReciprocalSecond,  
  Value = 140.0)
```

DesignPower**Data Property**

The power for which the *PackagingSystem* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignPower>

DESIGN POWER

Example

```
PhysicalQuantity(  
  Unit = Kilowatt,  
  Value = 500.0)
```

PackagingSystemType

Data Property

The packaging system type of the *PackagingSystem*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PackagingSystemTypeAssignmentClass>

PACKAGING SYSTEM TYPE ASSIGNMENT CLASS

Example

"Automated Packaging."

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.PackagingSystem">
  <Data property="DesignCapacityMassFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerMinute"/>
      </Data>
      <Data property="Value">
        <Double>40.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignCapacityPackagingUnits">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.NumberPerTimeIntervalUnit.
↵ReciprocalSecond"/>
      </Data>
      <Data property="Value">
        <Double>140.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>500.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
```

(continues on next page)

(continued from previous page)

```

<Data property="PackagingSystemType">
  <String>Automated Packaging.</String>
</Data>
</Object>

```

PelletizerDisc

Plant::ProcessEquipment::PelletizerDisc

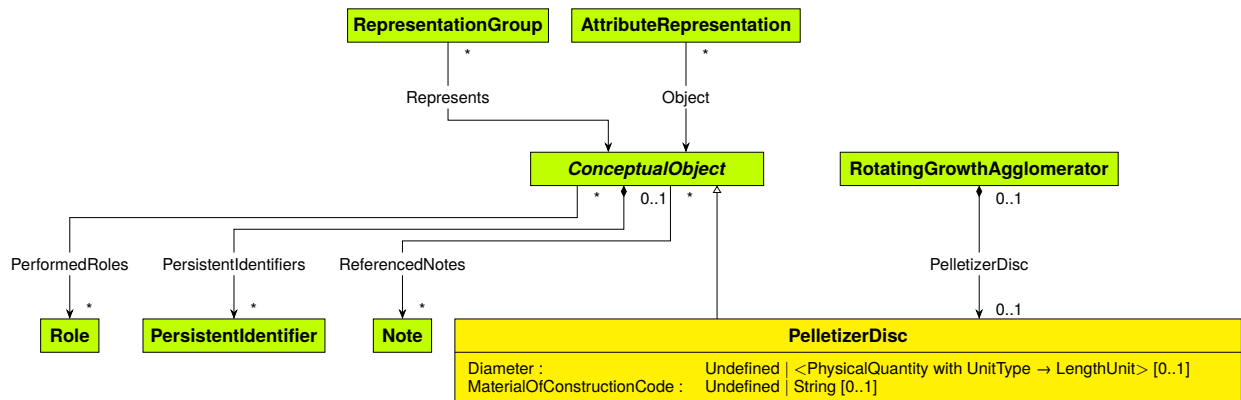
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/PelletingDisc>

PELLETING DISC

A rotating disc as a component of an *Agglomerator*.



Supertypes

- *ConceptualObject*

Members

Data Properties

Name	Multi- plicity	Type	Or- dered	Unique
<i>Diameter</i>	0..1	<i>Undefined</i> < <i>PhysicalQuantity</i> with <i>UnitType</i> → <i>LengthUnit</i> >	no	no
<i>MaterialOfConstruction-Code</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

Diameter

Data Property

The diameter of the *PelletizerDisc*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → LengthUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS350954>

DIAMETER

Example

```
PhysicalQuantity(
  Unit = Centimetre,
  Value = 20.0)
```

MaterialOfConstructionCode

Data Property

A code that gives the material of construction of the *PelletizerDisc*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS1460719741>

MATERIAL OF CONSTRUCTION CODE ASSIGNMENT CLASS

Example

"1.4306"

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.PelletizerDisc">
  <Data property="Diameter">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
      </Data>
      <Data property="Value">
        <Double>20.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="MaterialOfConstructionCode">
    <String>1.4306</String>
  </Data>
</Object>
```

PlateHeatExchanger

Plant::ProcessEquipment::PlateHeatExchanger

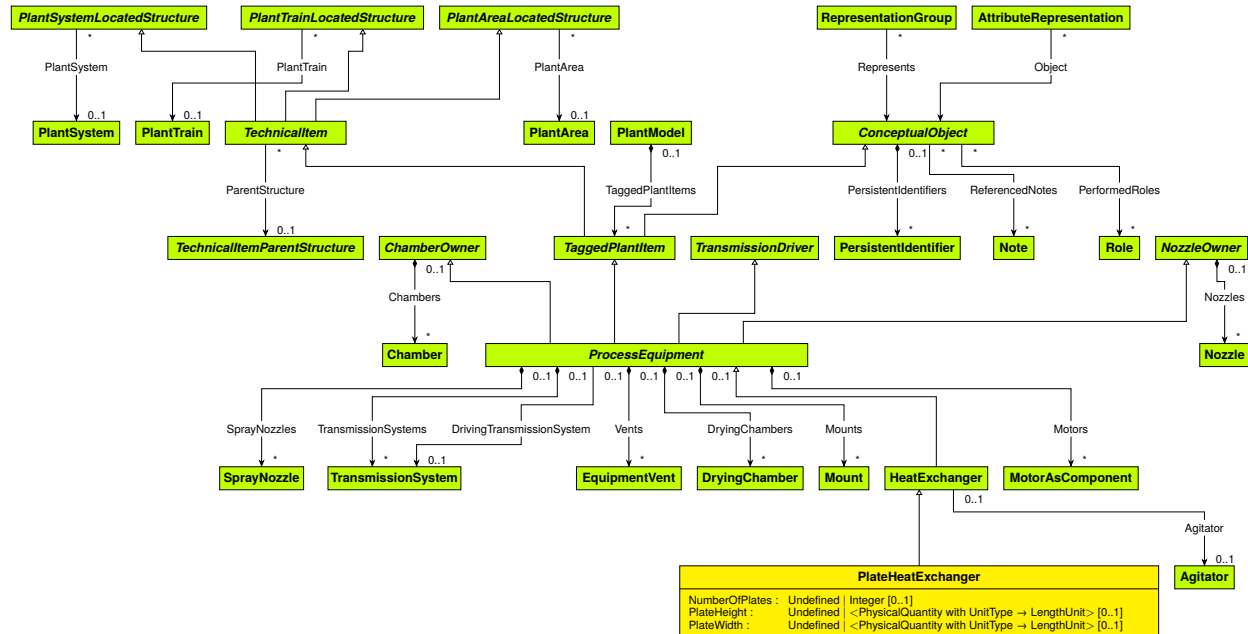
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/PlateHeatExchanger>

PLATE HEAT EXCHANGER

A heat exchanger that uses metal plates to transfer heat between two fluids.



Supertypes

- *HeatExchanger*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>NumberOfPlates</i>	0..1	<i>Undefined Integer</i>	no	no
<i>PlateHeight</i>	0..1	<i>Undefined <PhysicalQuantity with UnitType → LengthUnit></i>	no	no
<i>PlateWidth</i>	0..1	<i>Undefined <PhysicalQuantity with UnitType → LengthUnit></i>	no	no

NumberOfPlates**Data Property**

The number of plates in the *PlateHeatExchanger*.

Type: *Undefined* | *Integer*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS364229>

NUMBER OF PLATES

Example

20

PlateHeight**Data Property**

The height of the plates in the *PlateHeatExchanger*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → LengthUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PlateHeight>

PLATE HEIGHT

Example

```
PhysicalQuantity(
  Unit = Millimetre,
  Value = 850.0)
```

PlateWidth**Data Property**

The width of the plates in the *PlateHeatExchanger*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → LengthUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/PlateWidth>

PLATE WIDTH

Example

```
PhysicalQuantity(  
    Unit = Millimetre,  
    Value = 1100.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.PlateHeatExchanger">  
  <Data property="NumberOfPlates">  
    <Integer>20</Integer>  
  </Data>  
  <Data property="PlateHeight">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Millimetre"/>  
      </Data>  
      <Data property="Value">  
        <Double>850.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
  <Data property="PlateWidth">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Millimetre"/>  
      </Data>  
      <Data property="Value">  
        <Double>1100.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
</Object>
```

PressureVessel

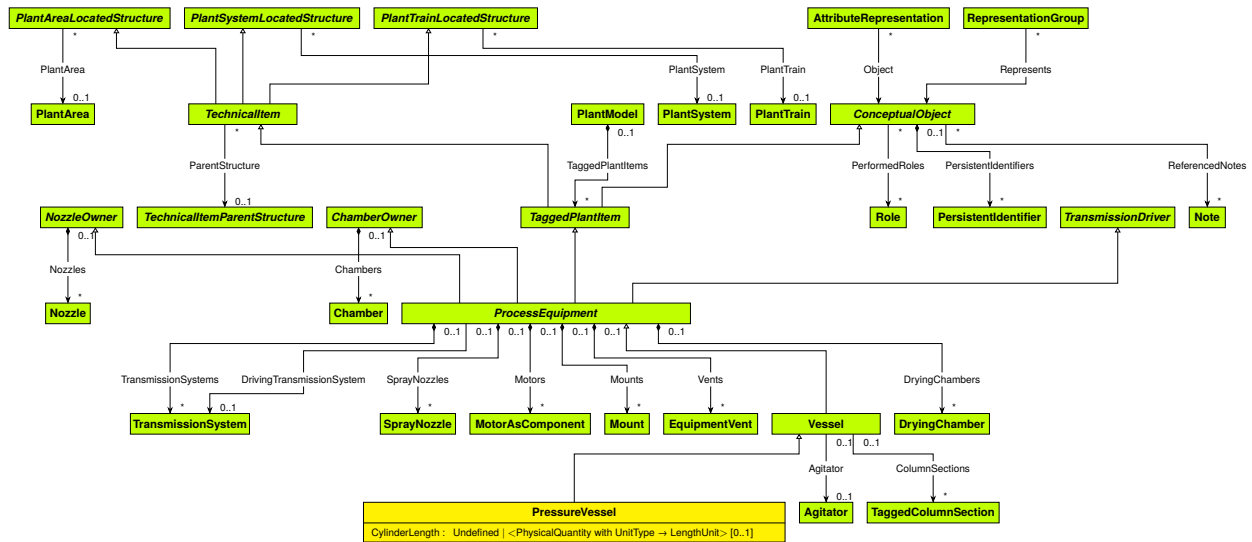
Plant::ProcessEquipment::PressureVessel

Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS427229>

PRESSURE VESSEL



Supertypes

- *Vessel*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>CylinderLength</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no

CylinderLength

Data Property

The cylinder length of the *PressureVessel*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → LengthUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/CylinderLength>

CYLINDER LENGTH

Example

PhysicalQuantity(

Unit = *Metre*,

Value = 2.0)

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.PressureVessel">
  <Data property="CylinderLength">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Metre"/>
      </Data>
      <Data property="Value">
        <Double>2.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

ProcessColumn

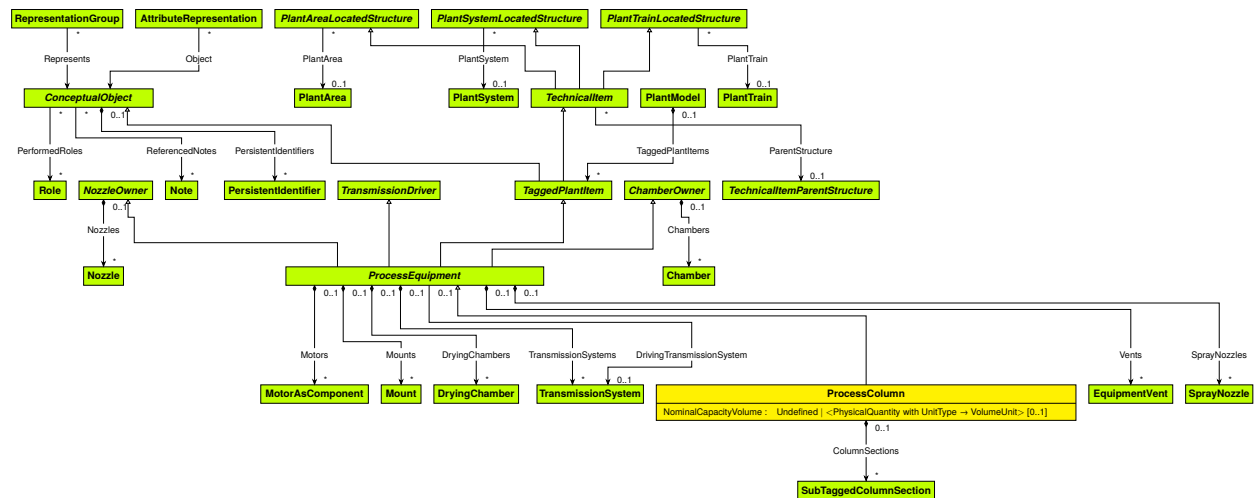
Plant::ProcessEquipment::ProcessColumn

Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS4316825224>

PROCESS COLUMN



Supertypes

- *ProcessEquipment*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>ColumnSections</i>	*	<i>SubTaggedColumnSection</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>NominalCapacityVolume</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VolumeUnit></i>	no	no

ColumnSections

Composition Property

The column sections of the *ProcessColumn*.

Type: *SubTaggedColumnSection*

Multiplicity: *

Ordered: no

NominalCapacityVolume

Data Property

The nominal volumetric capacity of the *ProcessColumn*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → VolumeUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalCapacityVolume>

NOMINAL CAPACITY VOLUME

Example

```
PhysicalQuantity(
  Unit = MetreCubed,
  Value = 7.2)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.ProcessColumn">
  <Data property="NominalCapacityVolume">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.VolumeUnit.MetreCubed"/>
      </Data>
      <Data property="Value">
        <Double>7.2</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

(continues on next page)

(continued from previous page)

```
</Data>
</Object>
```

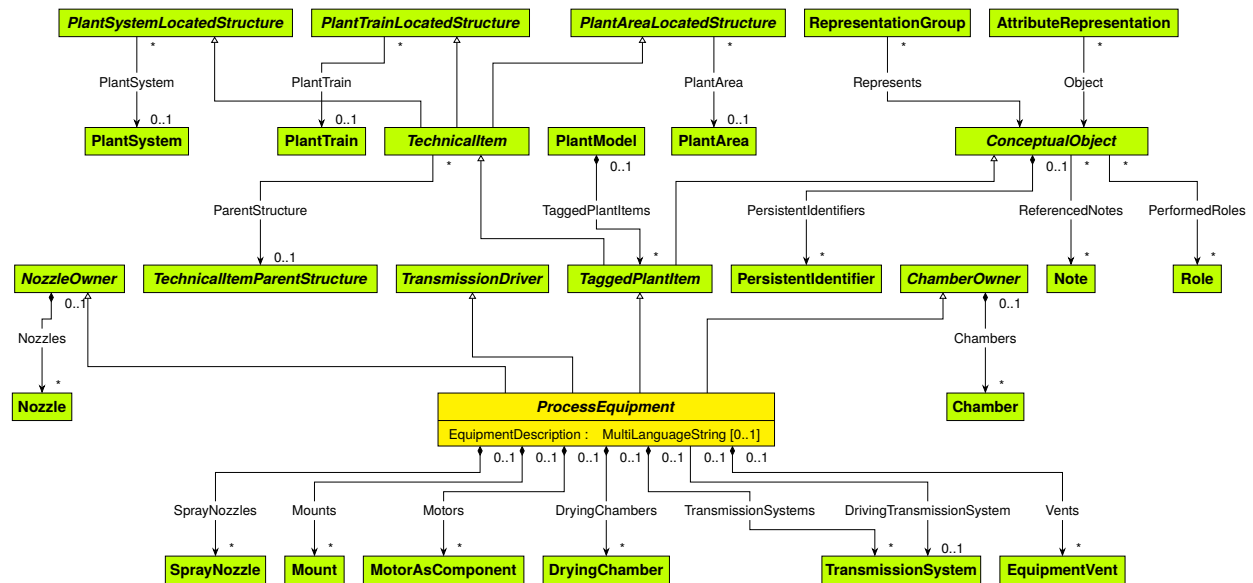
ProcessEquipment

Plant::ProcessEquipment::ProcessEquipment

Overview

Abstract Class

An apparatus or machine.



Supertypes

- *ChamberOwner*
- *NozzleOwner*
- *TaggedPlantItem*
- *TransmissionDriver*

Subtypes

- *Agglomerator*
- *Agitator*
- *Blower*
- *Burner*
- *Centrifuge*
- *Compressor*
- *CoolingTower*
- *Dryer*

- *ElectricGenerator*
- *Extruder*
- *Fan*
- *Feeder*
- *Filter*
- *HeatExchanger*
- *Heater*
- *Mill*
- *Mixer*
- *MobileTransportSystem*
- *Motor*
- *PackagingSystem*
- *ProcessColumn*
- *Pump*
- *Separator*
- *Sieve*
- *StationaryTransportSystem*
- *Turbine*
- *Vessel*
- *WasteGasEmitter*
- *Weigher*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>DryingChambers</i>	*	<i>DryingChamber</i>	no
<i>Motors</i>	*	<i>MotorAsComponent</i>	no
<i>Mounts</i>	*	<i>Mount</i>	no
<i>SprayNozzles</i>	*	<i>SprayNozzle</i>	no
<i>TransmissionSystems</i>	*	<i>TransmissionSystem</i>	no
<i>Vents</i>	*	<i>EquipmentVent</i>	no

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>DrivingTransmissionSystem</i>	0..1	<i>TransmissionSystem</i>	no	no	0..1

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>EquipmentDescription</i>	0..1	<i>MultiLanguageString</i>	no	no

DrivingTransmissionSystem

Reference Property

The *TransmissionSystem* that drives the ProcessEquipment. If given, it must be among the Plant.ProcessEquipment.Equipment.TransmissionSystems of the same ProcessEquipment.

Type: *TransmissionSystem*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

DryingChambers

Composition Property

The DryingChambers of the ProcessEquipment.

Type: *DryingChamber*

Multiplicity: *

Ordered: no

EquipmentDescription

Data Property

A short description of the *ProcessEquipment* in natural language.

Type: *MultiLanguageString*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS2181987301>

EQUIPMENT DESCRIPTION ASSIGNMENT CLASS

Example

```
MultiLanguageString(  
  SingleLanguageStrings = [  
    SingleLanguageString(  
      Language = "de",  
      Value = "Prozessgaskühler"),
```

```
SingleLanguageString(  
  Language = "en",  
  Value = "process gas cooler"))
```

Motors

Composition Property

The motors that are components of the ProcessEquipment.

Type: *MotorAsComponent*

Multiplicity: *

Ordered: no

Mounts

Composition Property

The *Mounts* that are attached to the ProcessEquipment.

Type: *Mount*

Multiplicity: *

Ordered: no

SprayNozzles

Composition Property

The SprayNozzles of the ProcessEquipment.

Type: *SprayNozzle*

Multiplicity: *

Ordered: no

TransmissionSystems

Composition Property

The TransmissionSystems that are components of the ProcessEquipment.

Type: *TransmissionSystem*

Multiplicity: *

Ordered: no

Vents

Composition Property

The EquipmentVents that are components of the ProcessEquipment.

Type: *EquipmentVent*

Multiplicity: *

Ordered: no

ProcessNozzle

Plant::ProcessEquipment::ProcessNozzle

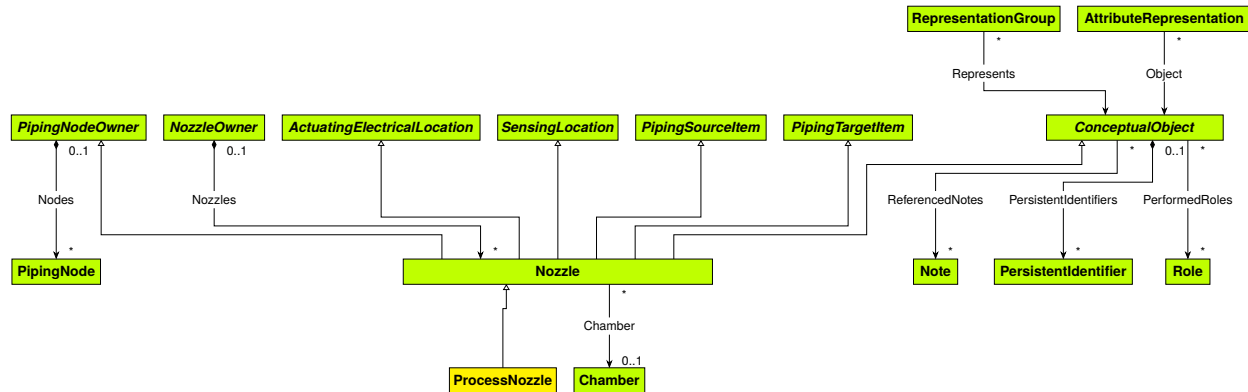
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS4316825252>

PROCESS NOZZLE

A *Nozzle* intended to enable transfer of process material (from <http://data.posccaesar.org/rdl/RDS4316825252>).



Supertypes

- *Nozzle*

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.ProcessNozzle"/>
```

Pump

Plant::ProcessEquipment::Pump

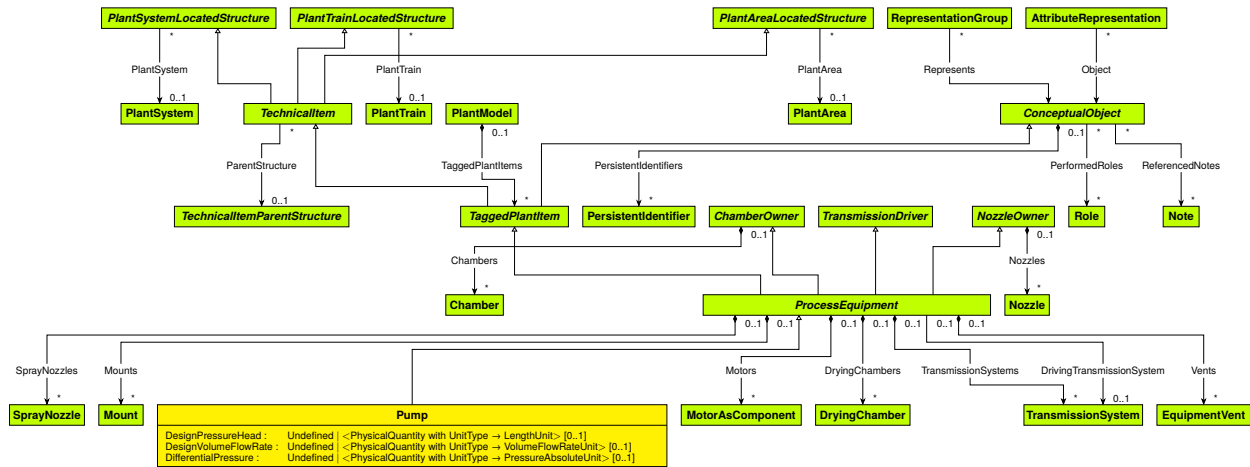
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS327239>

PUMP

A machine that is capable of pumping but may require parts and subsystems for that capability.



Supertypes

- *ProcessEquipment*

Subtypes

- *CentrifugalPump*
- *EjectorPump*
- *ReciprocatingPump*
- *RotaryPump*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignPressureHead</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no
<i>DesignVolumeFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VolumeFlowRateUnit></i>	no	no
<i>DifferentialPressure</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PressureAbsoluteUnit></i>	no	no

DesignPressureHead

Data Property

The pressure head for which the *Pump* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → LengthUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignPressureHead>

DESIGN PRESSURE HEAD

Example

```
PhysicalQuantity(  
  Unit = Metre,  
  Value = 40.0)
```

DesignVolumeFlowRate

Data Property

The volume flow rate for which the *Pump* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VolumeFlowRateUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS14286227>

DESIGN VOLUME FLOW RATE

Example

```
PhysicalQuantity(  
  Unit = MetreCubedPerHour,  
  Value = 420.0)
```

DifferentialPressure

Data Property

The differential pressure of the *Pump*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PressureAbsoluteUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS361574>

DIFFERENTIAL PRESSURE

Example

```
PhysicalQuantity(  
  Unit = Bar,  
  Value = 4.8)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Pump">
  <Data property="DesignPressureHead">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Metre"/>
      </Data>
      <Data property="Value">
        <Double>40.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignVolumeFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour"
→"/>
      </Data>
      <Data property="Value">
        <Double>420.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DifferentialPressure">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PressureAbsoluteUnit.Bar"/>
      </Data>
      <Data property="Value">
        <Double>4.8</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

RadialFan

Plant::ProcessEquipment::RadialFan

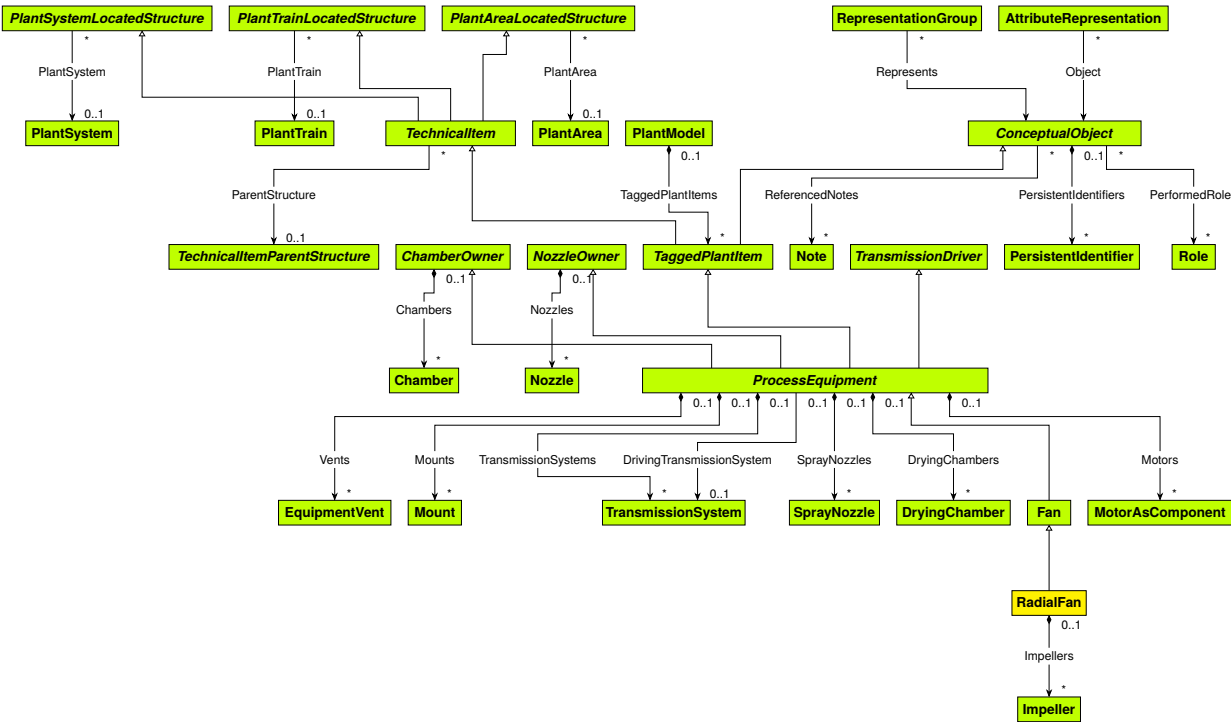
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS414089>

RADIAL FAN

A ‘fan’ with axial inlet and radial outlet (from <http://data.posccaesar.org/rdl/RDS414089>).



Supertypes

- *Fan*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Impellers</i>	*	<i>Impeller</i>	no

Impellers

Composition Property

The impellers of the RadialFan.

Type: *Impeller*

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.RadialFan"/>
```

RailWaggon

Plant::ProcessEquipment::RailWaggon

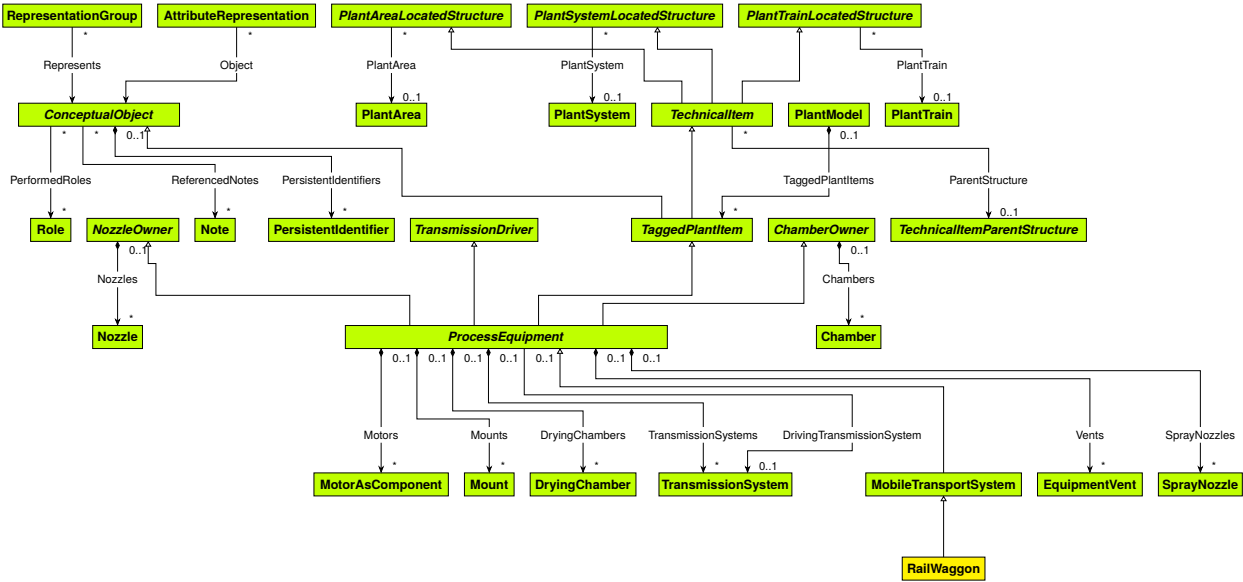
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS11524697>

RAIL WAGGON

A non self driving vehicle and mobile transport system intended to ride on rails.



Supertypes

- *MobileTransportSystem*

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.RailWaggon"/>
```

ReciprocatingCompressor

Plant::ProcessEquipment::ReciprocatingCompressor

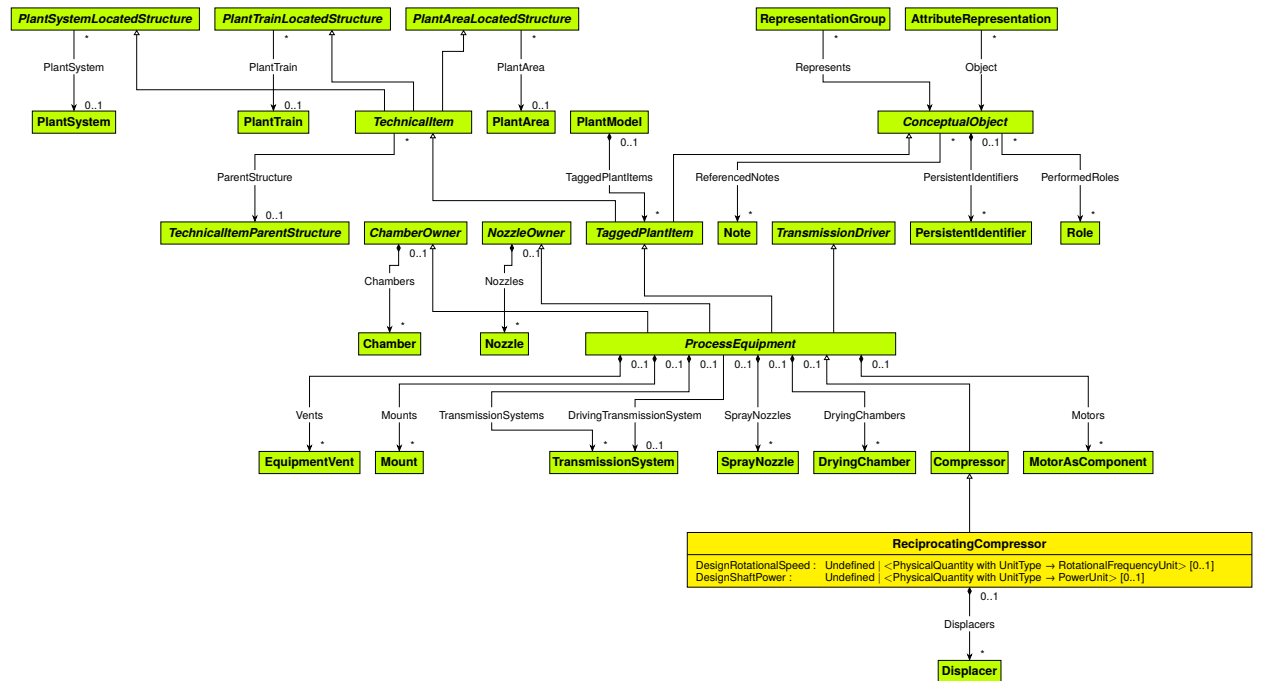
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS417284>

RECIPROCATING COMPRESSOR

A positive displacement compressor in which forced reduction of gas volume takes place by the movement of a displacing element in a cylinder or enclosure (from <http://data.posccaesar.org/rdl/RDS417284>).



Supertypes

- *Compressor*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Displacers</i>	*	<i>Displacer</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignRotationalSpeed</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → Rotational-FrequencyUnit></i>	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no

DesignRotationalSpeed

Data Property

The rotational speed for which the *ReciprocatingCompressor* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(
  Unit = ReciprocalMinute,
  Value = 180.0)
```

DesignShaftPower

Data Property

The shaft power for which the *ReciprocatingCompressor* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PowerUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(
  Unit = Kilowatt,
  Value = 400.0)
```

Displacers

Composition Property

The displacers of the *ReciprocatingCompressor*.

Type: *Displacer*

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.ReciprocatingCompressor">
  <Data property="DesignRotationalSpeed">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ ReciprocalMinute"/>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

(continues on next page)

(continued from previous page)

```

    <Data property="Value">
      <Double>180.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="DesignShaftPower">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
    </Data>
    <Data property="Value">
      <Double>400.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>

```

ReciprocatingExtruder

Plant::ProcessEquipment::ReciprocatingExtruder

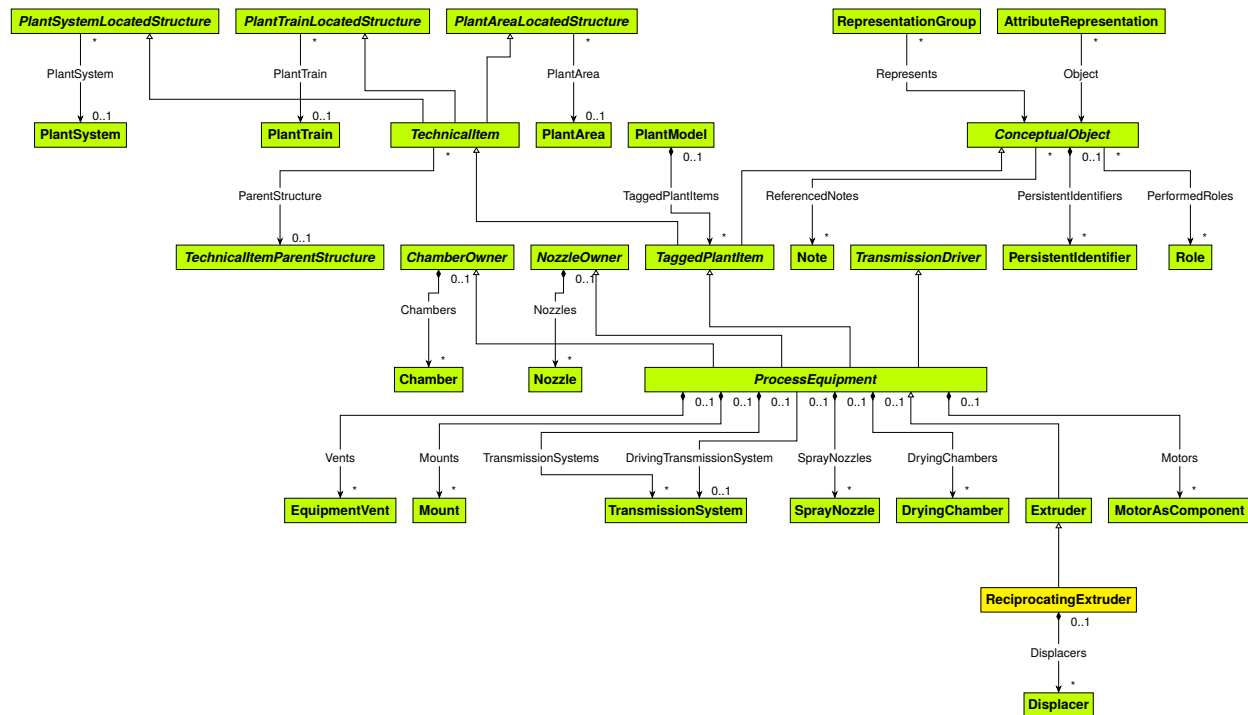
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS412409911>

PISTON EXTRUDER

An extruder that uses a piston in a batch process (from <http://data.posccaesar.org/rdl/RDS412409911>).



Supertypes

- *Extruder*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Displacers</i>	*	<i>Displacer</i>	no

Displacers

Composition Property

The displacers of the ReciprocatingExtruder.

Type: *Displacer*

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.ReciprocatingExtruder"/>
```

ReciprocatingPressureAgglomerator

Plant::ProcessEquipment::ReciprocatingPressureAgglomerator

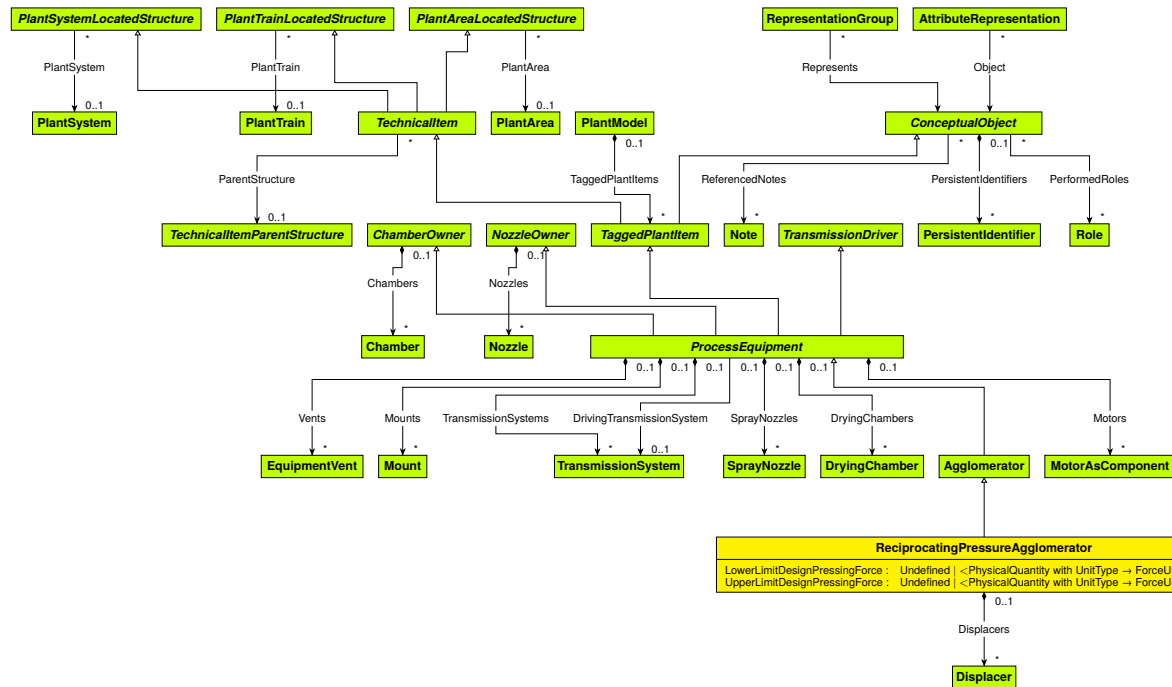
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/ReciprocatingPressureAgglomerator>

RECIPROCATING PRESSURE AGGLOMERATOR

An *Agglomerator* which uses pistons to produce pressure and to form material (from <http://data.15926.org/rdl/RDS2228720>).



Supertypes

- *Agglomerator*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Displacers</i>	*	<i>Displacer</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>LowerLimitDesignPressingForce</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → ForceUnit></i>	no	no
<i>UpperLimitDesignPressingForce</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → ForceUnit></i>	no	no

Displacers

Composition Property

The displacers of the ReciprocatingPressureAgglomerator.

Type: *Displacer*

Multiplicity: *

Ordered: no

LowerLimitDesignPressingForce

Data Property

The lower limit for the pressing force for which the *ReciprocatingPressureAgglomerator* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → ForceUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LowerLimitDesignPressingForce>

LOWER LIMIT DESIGN PRESSING FORCE

Example

```
PhysicalQuantity(
    Unit = Newton,
    Value = 20.0)
```

UpperLimitDesignPressingForce

Data Property

The upper limit for the pressing force for which the *ReciprocatingPressureAgglomerator* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → ForceUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitDesignPressingForce>

UPPER LIMIT DESIGN PRESSING FORCE

Example

```
PhysicalQuantity(
    Unit = Newton,
    Value = 70.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.ReciprocatingPressureAgglomerator">
  <Data property="LowerLimitDesignPressingForce">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.ForceUnit.Newton"/>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

(continues on next page)

(continued from previous page)

```

    <Data property="Value">
      <Double>20.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="UpperLimitDesignPressingForce">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.ForceUnit.Newton"/>
    </Data>
    <Data property="Value">
      <Double>70.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>

```

ReciprocatingPump

Plant::ProcessEquipment::ReciprocatingPump

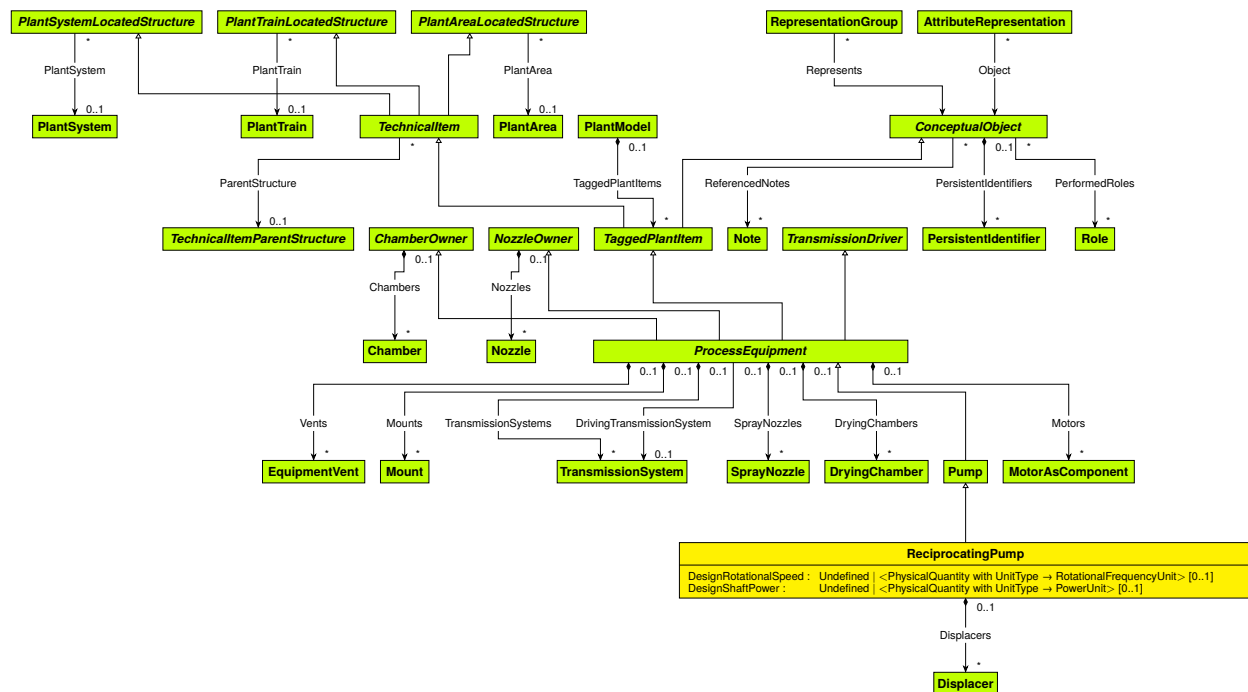
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS416969>

RECIPROCATING PUMP

a positive displacement pump which contains a displacing element intended to be moved in a reciprocating movement to exert pressure on a fluid, typically moving within a cylindrical space (from <http://data.posccaesar.org/rdl/RDS416969>).



Supertypes

- *Pump*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Displacers</i>	*	<i>Displacer</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignRotationalSpeed</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → RotationalFrequencyUnit></i>	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no

DesignRotationalSpeed

Data Property

The rotational speed for which the *ReciprocatingPump* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(
  Unit = ReciprocalMinute,
  Value = 180.0)
```

DesignShaftPower

Data Property

The shaft power for which the *ReciprocatingPump* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PowerUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(  
    Unit = Kilowatt,  
    Value = 400.0)
```

Displacers

Composition Property

The displacers of the ReciprocatingPump.

Type: *Displacer*

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.ReciprocatingPump">  
  <Data property="DesignRotationalSpeed">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.  
↪ReciprocalMinute"/>  
      </Data>  
      <Data property="Value">  
        <Double>180.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
  <Data property="DesignShaftPower">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>  
      </Data>  
      <Data property="Value">  
        <Double>400.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
</Object>
```

RevolvingSieve

Plant::ProcessEquipment::RevolvingSieve

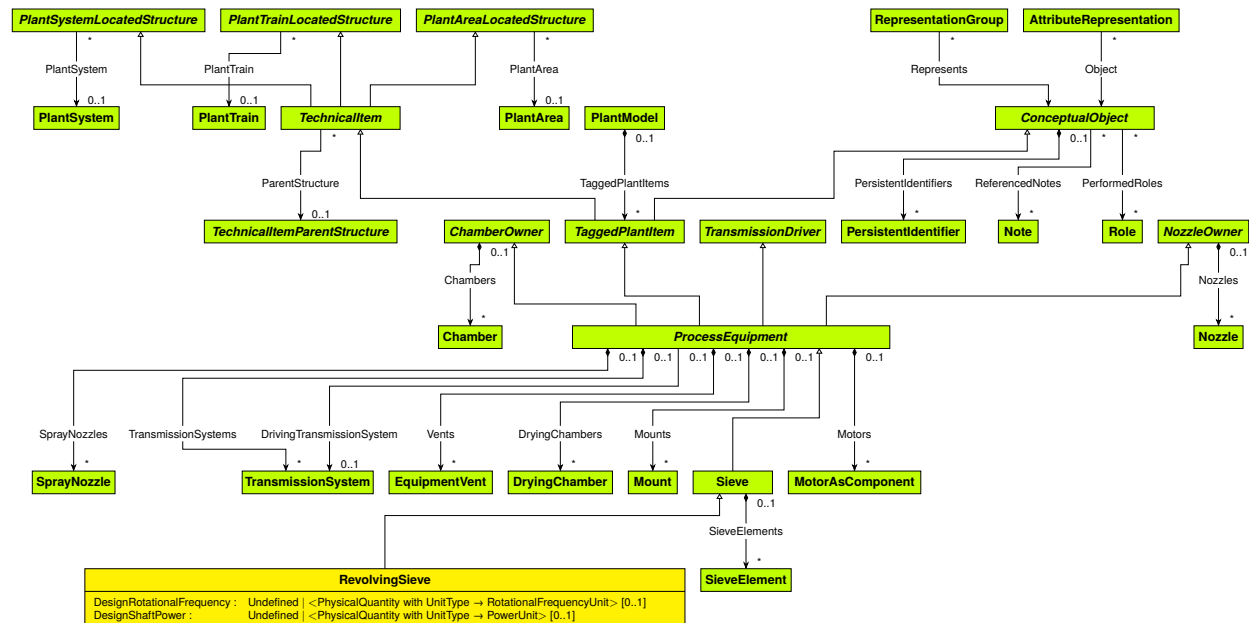
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/RevolvingSieve>

REVOLVING SIEVE

A revolving sieve that intends to sift out finer from coarser parts.



Supertypes

- *Sieve*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignRotationalFrequency</i>	0..1	<i>Undefined <PhysicalQuantity with UnitType → RotationalFrequencyUnit></i>	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined <PhysicalQuantity with UnitType → PowerUnit></i>	no	no

DesignRotationalFrequency

Data Property

The rotational frequency for which the *RevolvingSieve* is designed.

Type: *Undefined | <PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalFrequency>

DESIGN ROTATIONAL FREQUENCY

Example

```
PhysicalQuantity(  
  Unit = ReciprocalMinute,  
  Value = 180.0)
```

DesignShaftPower

Data Property

The shaft power for which the *RevolvingSieve* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(  
  Unit = Kilowatt,  
  Value = 400.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.RevolvingSieve">  
  <Data property="DesignRotationalFrequency">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.  
↪ReciprocalMinute"/>  
      </Data>  
      <Data property="Value">  
        <Double>180.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
  <Data property="DesignShaftPower">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>  
      </Data>
```

(continues on next page)

Plant::ProcessEquipment::RotaryCompressor

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Displacers</i>	*	<i>Displacer</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignRotationalSpeed</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → Rotational-FrequencyUnit></i>	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no

DesignRotationalSpeed

Data Property

The rotational speed for which the *RotaryCompressor* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(  
    Unit = ReciprocalMinute,  
    Value = 180.0)
```

DesignShaftPower

Data Property

The shaft power for which the *RotaryCompressor* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PowerUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(
  Unit = Kilowatt,
  Value = 400.0)
```

Displacers**Composition Property**

The displacers of the RotaryCompressor.

Type: *Displacer*

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.RotaryCompressor">
  <Data property="DesignRotationalSpeed">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ReciprocalMinute"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignShaftPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>400.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

RotaryMixer

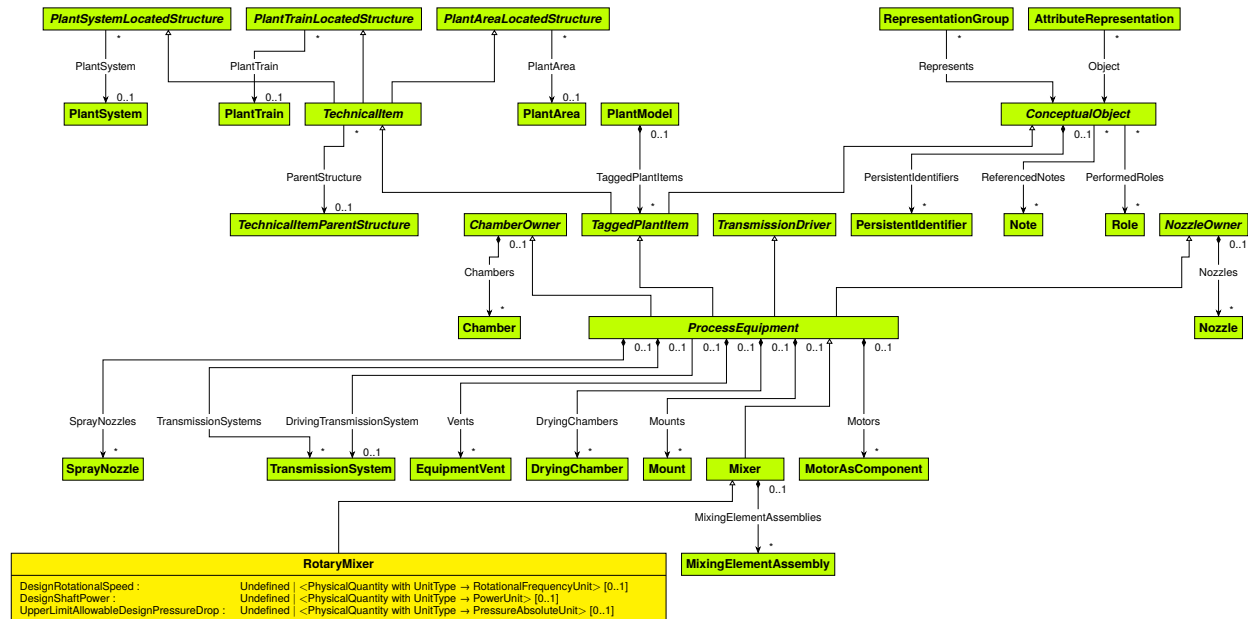
Plant::ProcessEquipment::RotaryMixer

Overview**Concrete Class**

RDL: <http://sandbox.dexpi.org/rdl/RotaryMixer>

ROTARY MIXER

A *Mixer* machine that mixes by means of rotating components.



Supertypes

- *Mixer*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignRotationalSpeed</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → RotationalFrequencyUnit></i>	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no
<i>UpperLimitAllowableDesignPressureDrop</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PressureAbsoluteUnit></i>	no	no

DesignRotationalSpeed

Data Property

The rotational speed for which the *RotaryMixer* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(
  Unit = ReciprocalMinute,
  Value = 180.0)
```

DesignShaftPower**Data Property**

The shaft power for which the *RotaryMixer* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(
  Unit = Kilowatt,
  Value = 400.0)
```

UpperLimitAllowableDesignPressureDrop**Data Property**

The upper limit for the pressure drop for which the *RotaryMixer* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PressureAbsoluteUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitAllowableDesignPressureDrop>

UPPER LIMIT ALLOWABLE DESIGN PRESSURE DROP

Example

```
PhysicalQuantity(
  Unit = Bar,
  Value = 2.0)
```

Example in DEXPI XML

```

<Object type="Plant/ProcessEquipment.RotaryMixer">
  <Data property="DesignRotationalSpeed">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ReciprocalMinute"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignShaftPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>400.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="UpperLimitAllowableDesignPressureDrop">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PressureAbsoluteUnit.Bar"/>
      </Data>
      <Data property="Value">
        <Double>2.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>

```

RotaryPump

Plant::ProcessEquipment::RotaryPump

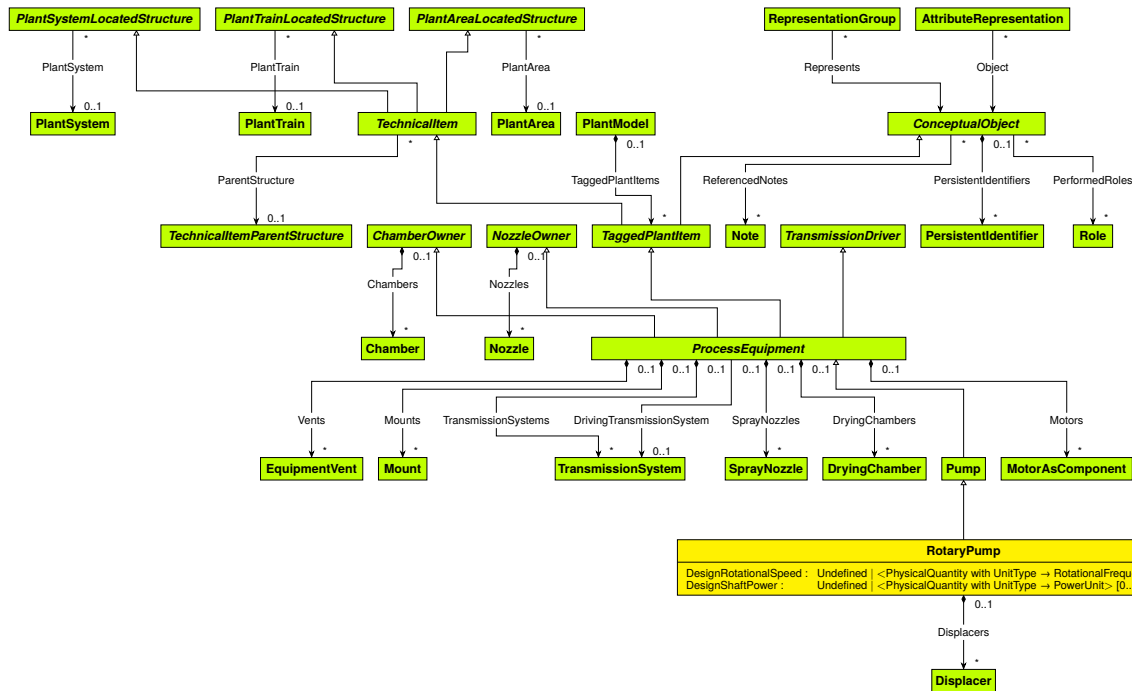
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS420749>

ROTARY PUMP

A positive displacement pump that consists of a chamber containing gears, cams, screws, vanes, plungers or similar elements actuated by relative rotation of the drive shaft or casing and which has no separate inlet and outlet valves (from <http://data.posccaesar.org/rdl/RDS420749>).



Supertypes

- *Pump*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Displacers</i>	*	<i>Displacer</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignRotationalSpeed</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → RotationalFrequencyUnit></i>	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no

DesignRotationalSpeed

Data Property

The rotational speed for which the *RotaryPump* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(  
    Unit = ReciprocalMinute,  
    Value = 180.0)
```

DesignShaftPower

Data Property

The shaft power for which the *RotaryPump* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PowerUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(  
    Unit = Kilowatt,  
    Value = 400.0)
```

Displacers

Composition Property

The displacers of the *RotaryPump*.

Type: *Displacer*

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.RotaryPump">  
  <Data property="DesignRotationalSpeed">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.  
↪ ReciprocalMinute"/>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
</Object>
```

(continues on next page)

(continued from previous page)

```

<Data property="Value">
  <Double>180.0</Double>
</Data>
</AggregatedDataValue>
</Data>
<Data property="DesignShaftPower">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
    </Data>
    <Data property="Value">
      <Double>400.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>

```

RotatingExtruder

Plant::ProcessEquipment::RotatingExtruder

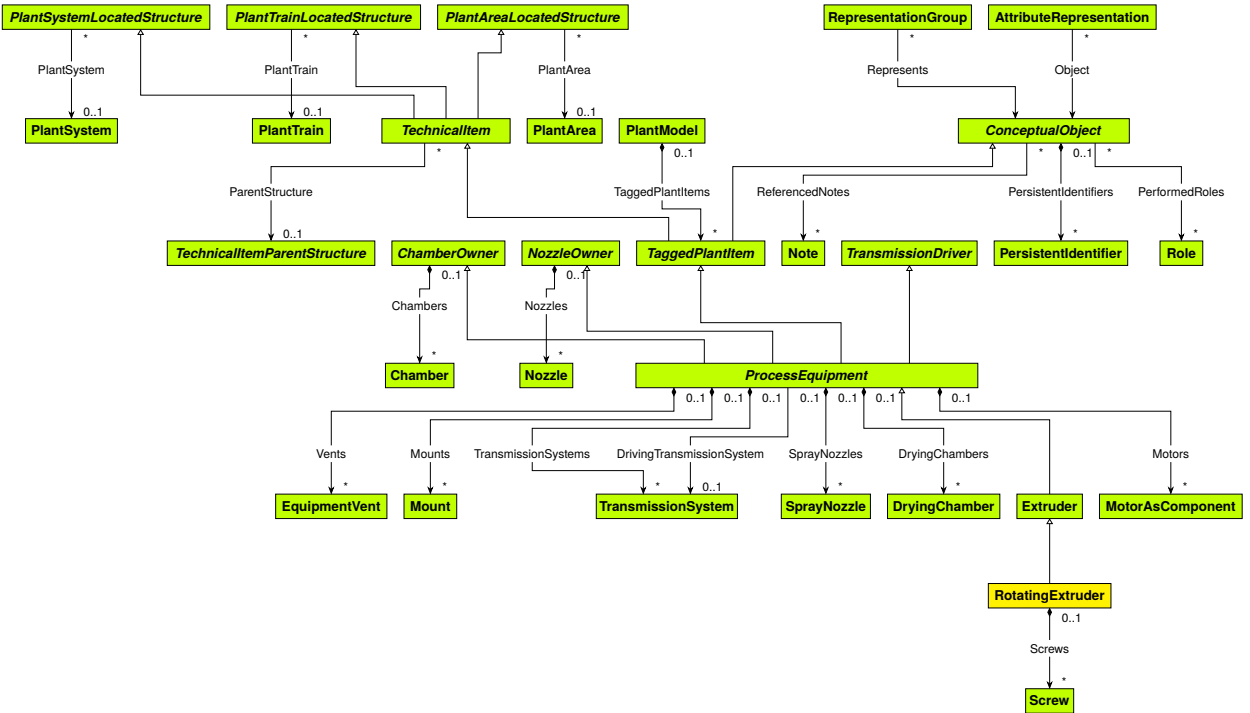
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS394045941>

AUGER EXTRUDER

An extruder that operates in a continuous process. Typically using a screw to build up pressure in the melt. It can incorporate a mixing stage with a forming stage (from <http://data.posccaesar.org/rdl/RDS394045941>).



Supertypes

- *Extruder*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Screws</i>	*	<i>Screw</i>	no

Screws

Composition Property

The screws of the RotatingExtruder.

Type: *Screw*

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.RotatingExtruder"/>
```

RotatingGrowthAgglomerator

Plant::ProcessEquipment::RotatingGrowthAgglomerator

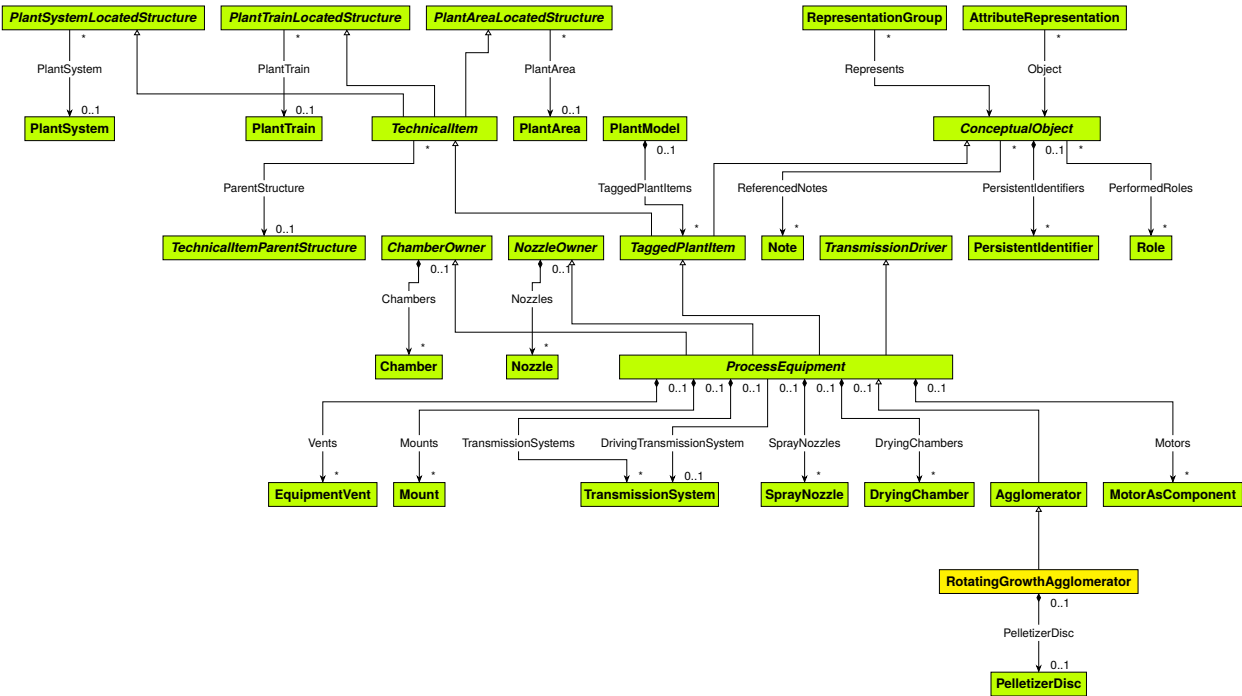
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/RotatingGrowthAgglomerator>

ROTATING GROWTH AGGLOMERATOR

An *Agglomerator* which uses a pelletizer disc to produce pellets.



Supertypes

- *Agglomerator*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>PelletizerDisc</i>	0..1	<i>PelletizerDisc</i>	no

PelletizerDisc

Composition Property

The pelletizing disc of the RotatingGrowthAgglomerator.

Type: *PelletizerDisc*

Multiplicity: 0..1

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.RotatingGrowthAgglomerator"/>
```


Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>LowerLimitDesignPressingForce</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → ForceUnit></i>	no	no
<i>UpperLimitDesignPressingForce</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → ForceUnit></i>	no	no

BriquettingRollers

Composition Property

The briquetting rollers of the *RotatingPressureAgglomerator*.

Type: *BriquettingRoller*

Multiplicity: *

Ordered: no

LowerLimitDesignPressingForce

Data Property

The lower limit for the pressing force for which the *RotatingPressureAgglomerator* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → ForceUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/LowerLimitDesignPressingForce>

LOWER LIMIT DESIGN PRESSING FORCE

Example

```
PhysicalQuantity(
  Unit = Newton,
  Value = 20.0)
```

UpperLimitDesignPressingForce

Data Property

The upper limit for the pressing force for which the *RotatingPressureAgglomerator* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → ForceUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitDesignPressingForce>

UPPER LIMIT DESIGN PRESSING FORCE

Example

```
PhysicalQuantity(  
    Unit = Newton,  
    Value = 70.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.RotatingPressureAgglomerator">  
  <Data property="LowerLimitDesignPressingForce">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.ForceUnit.Newton"/>  
      </Data>  
      <Data property="Value">  
        <Double>20.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
  <Data property="UpperLimitDesignPressingForce">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.ForceUnit.Newton"/>  
      </Data>  
      <Data property="Value">  
        <Double>70.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
</Object>
```

Screw

Plant::ProcessEquipment::Screw

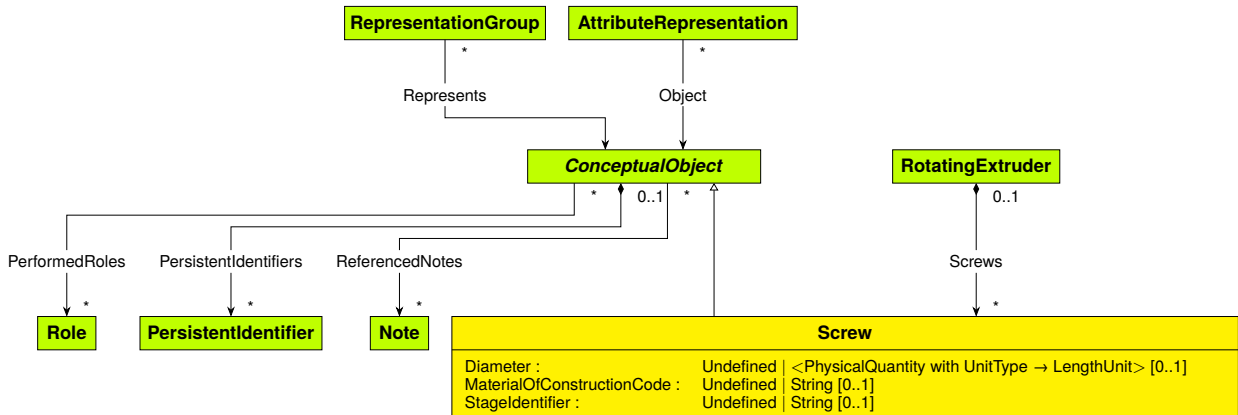
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS7219994>

SCREW

A shaft with a helical shaped shaft design (from <http://data.posccaesar.org/rdl/RDS7219994>).



Supertypes

- *ConceptualObject*

Members

Data Properties

Name	Multi- plicity	Type	Or- dered	Unique
<i>Diameter</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no
<i>MaterialOfConstruction- Code</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>StageIdentifier</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

Diameter

Data Property

The diameter of the *Screw*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → LengthUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS350954>

DIAMETER

Example

```

PhysicalQuantity(
    Unit = Centimetre,
    Value = 20.0)

```

MaterialOfConstructionCode

Data Property

A code that gives the material of construction of the *Screw*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS1460719741>

MATERIAL OF CONSTRUCTION CODE ASSIGNMENT CLASS

Example

"1.4306"

StageIdentifier

Data Property

The stage identifier of the *Screw*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/StageIdentifierAssignmentClass>

STAGE IDENTIFIER ASSIGNMENT CLASS

Example

"s1"

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Screw">
  <Data property="Diameter">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
      </Data>
      <Data property="Value">
        <Double>20.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="MaterialOfConstructionCode">
    <String>1.4306</String>
  </Data>
```

(continues on next page)

Efficiency

Data Property

The efficiency of the *SedimentalCentrifuge*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PercentageUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS362654>

EFFICIENCY

Example

PhysicalQuantity(

Unit = *Percent*,

Value = 90.0)

SedimentalCentrifugeDrum

Composition Property

The sedimental centrifuge drum of the *SedimentalCentrifuge*.

Type: *SedimentalCentrifugeDrum*

Multiplicity: 0..1

Ordered: no

Example in DEXPI XML

```

<Object type="Plant/ProcessEquipment.SedimentalCentrifuge">
  <Data property="Efficiency">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PercentageUnit.Percent"/>
      </Data>
      <Data property="Value">
        <Double>90.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>

```

SedimentalCentrifugeDrum

Plant::ProcessEquipment::SedimentalCentrifugeDrum

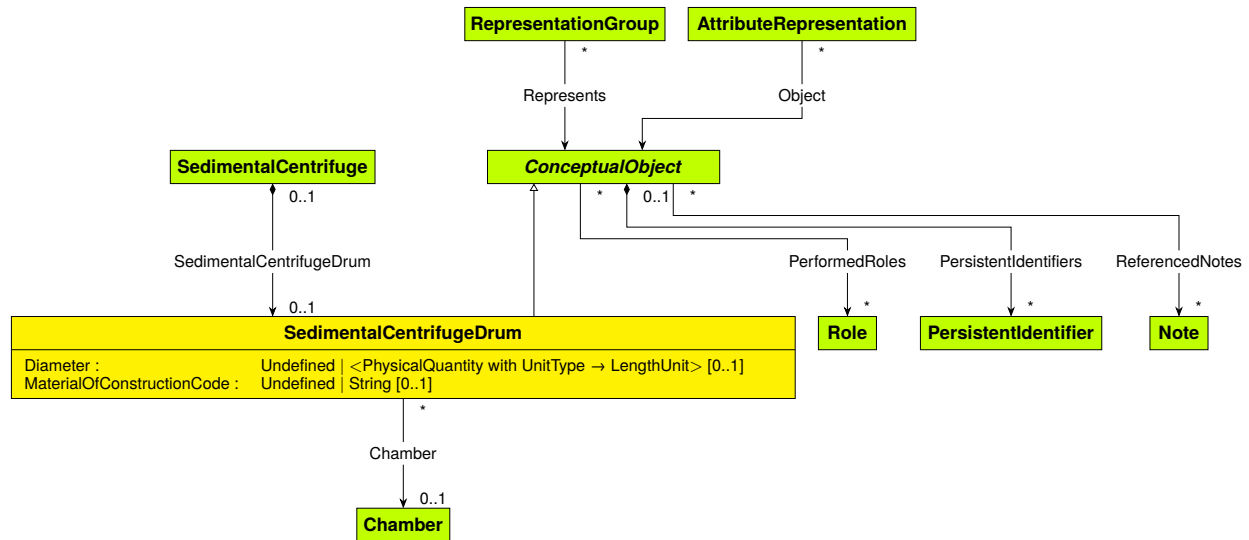
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/SedimentalCentrifugeDrum>

SEDIMENTAL CENTRIFUGE DRUM

A `SedimentalCentrifugeDrum` is a drum and a component of a *SedimentalCentrifuge*.



Supertypes

- *ConceptualObject*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Chamber</i>	0..1	<i>Chamber</i>	no	no	*

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Diameter</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType -> LengthUnit></i>	no	no
<i>MaterialOfConstruction-Code</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

Chamber

Reference Property

The *Chamber* in which the *SedimentalCentrifugeDrum* is located, if applicable. The Chamber must be a component of the same object as the *SedimentalCentrifugeDrum*.

Type: *Chamber*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

Diameter

Data Property

The diameter of the *SedimentalCentrifugeDrum*.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS350954>

DIAMETER

Example

```
PhysicalQuantity(  
    Unit = Centimetre,  
    Value = 20.0)
```

MaterialOfConstructionCode

Data Property

A code that gives the material of construction of the *SedimentalCentrifugeDrum*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS1460719741>

MATERIAL OF CONSTRUCTION CODE ASSIGNMENT CLASS

Example

"1.4306"

Example in DEXPI XML

```

<Object type="Plant/ProcessEquipment.SedimentalCentrifugeDrum">
  <Data property="Diameter">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
      </Data>
      <Data property="Value">
        <Double>20.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="MaterialOfConstructionCode">
    <String>1.4306</String>
  </Data>
</Object>

```

Separator

Plant::ProcessEquipment::Separator

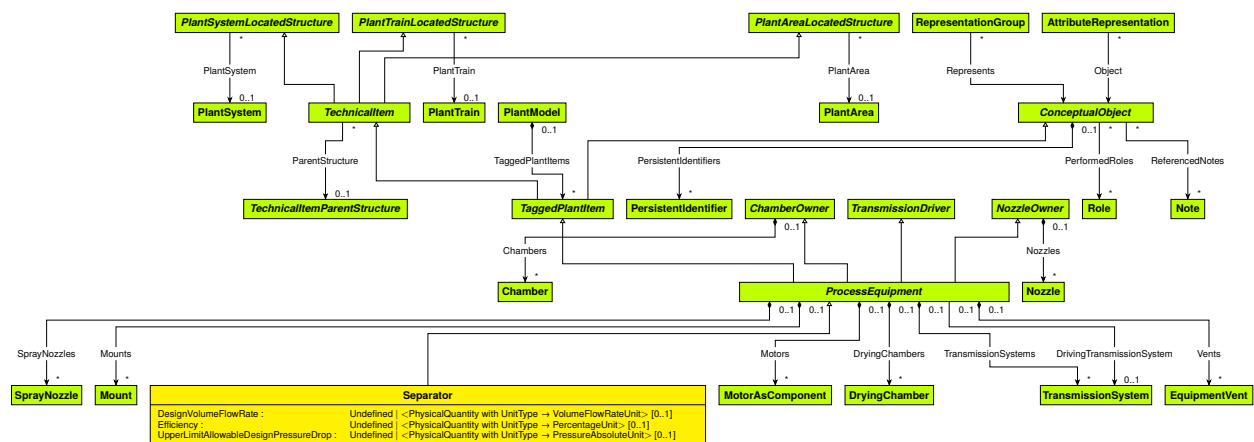
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS2194378711>

SEPARATOR

A 'device' intended to separate different types of substances



Supertypes

- *ProcessEquipment*

Subtypes

- *ElectricalSeparator*
- *GravitationalSeparator*
- *MechanicalSeparator*
- *ScrubbingSeparator*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignVolumeFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VolumeFlowRateUnit></i>	no	no
<i>Efficiency</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PercentageUnit></i>	no	no
<i>UpperLimitAllowableDesign-PressureDrop</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PressureAbsoluteUnit></i>	no	no

DesignVolumeFlowRate

Data Property

The volume flow rate for which the *Separator* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → VolumeFlowRateUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS14286227>

DESIGN VOLUME FLOW RATE

Example

```
PhysicalQuantity(
  Unit = MetreCubedPerHour,
  Value = 420.0)
```

Efficiency

Data Property

The efficiency of the *Separator*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PercentageUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS362654>

EFFICIENCY

Example

PhysicalQuantity(

Unit = *Percent*,

Value = 90.0)

UpperLimitAllowableDesignPressureDrop

Data Property

The upper limit for the pressure drop for which the *Separator* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PressureAbsoluteUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitAllowableDesignPressureDrop>

UPPER LIMIT ALLOWABLE DESIGN PRESSURE DROP

Example

PhysicalQuantity(

Unit = *Bar*,

Value = 2.0)

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Separator">
  <Data property="DesignVolumeFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour
↪"/>
      </Data>
      <Data property="Value">
        <Double>420.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="Efficiency">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
```

(continues on next page)

(continued from previous page)

```

<Data property="Unit">
  <DataReference data="Core/PhysicalQuantities.PercentageUnit.Percent"/>
</Data>
<Data property="Value">
  <Double>90.0</Double>
</Data>
</AggregatedDataValue>
</Data>
<Data property="UpperLimitAllowableDesignPressureDrop">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.PressureAbsoluteUnit.Bar"/>
    </Data>
    <Data property="Value">
      <Double>2.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>

```

Ship

Plant::ProcessEquipment::Ship

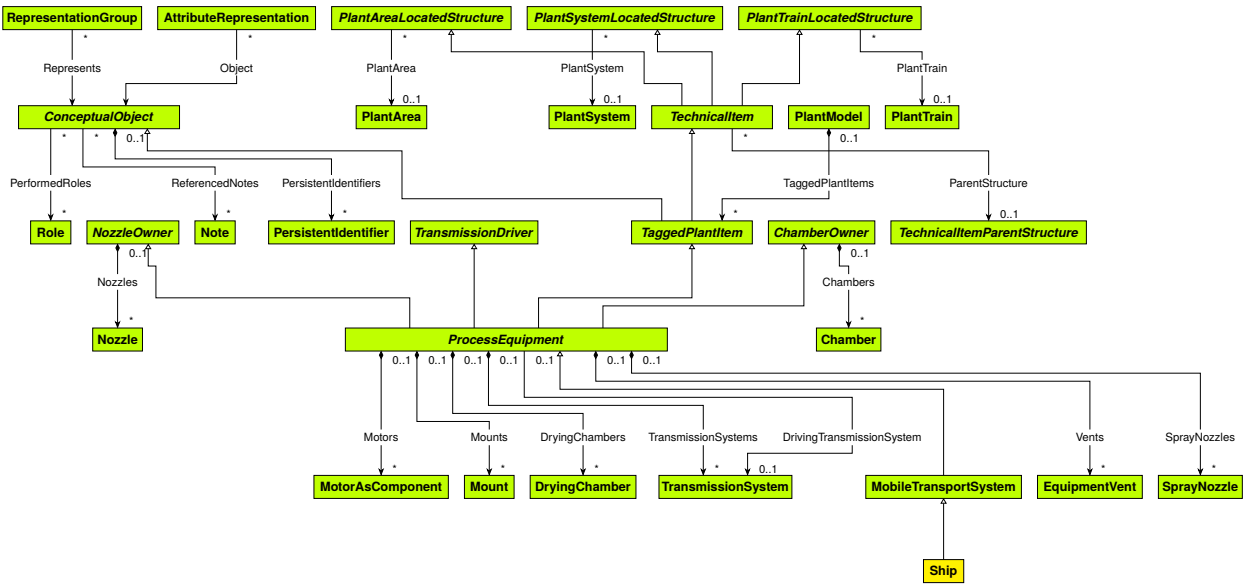
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS11523932>

SHIP

A watercraft and *MobileTransportSystem* that is a sea-going vessel of considerable size.



Supertypes

- *MobileTransportSystem*

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Ship"/>
```

Sieve

Plant::ProcessEquipment::Sieve

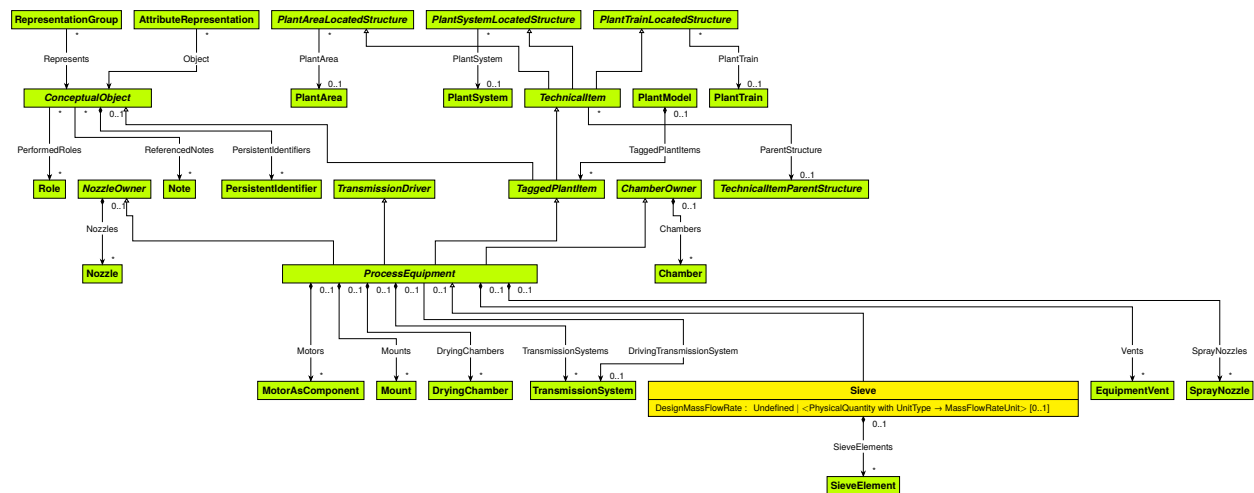
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/Sieve>

SIEVE

A device that removes particles from a fluid when the fluid passes through or separates particles or molecules according to their size.



Supertypes

- *ProcessEquipment*

Subtypes

- *RevolvingSieve*
- *StationarySieve*
- *VibratingSieve*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>SieveElements</i>	*	<i>SieveElement</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignMassFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → MassFlowRateUnit></i>	no	no

DesignMassFlowRate

Data Property

The mass flow rate for which the *Sieve* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → MassFlowRateUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS14286182>

DESIGN MASS FLOW RATE

Example

```
PhysicalQuantity(
  Unit = KilogramPerSecond,
  Value = 420.0)
```

SieveElements

Composition Property

The sieve elements of the Sieve.

Type: *SieveElement*

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Sieve">
  <Data property="DesignMassFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
```

(continues on next page)

(continued from previous page)

```

<DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerSecond"/
</Data>
<Data property="Value">
  <Double>420.0</Double>
</Data>
</AggregatedDataValue>
</Data>
</Object>

```

SieveElement

Plant::ProcessEquipment::SieveElement

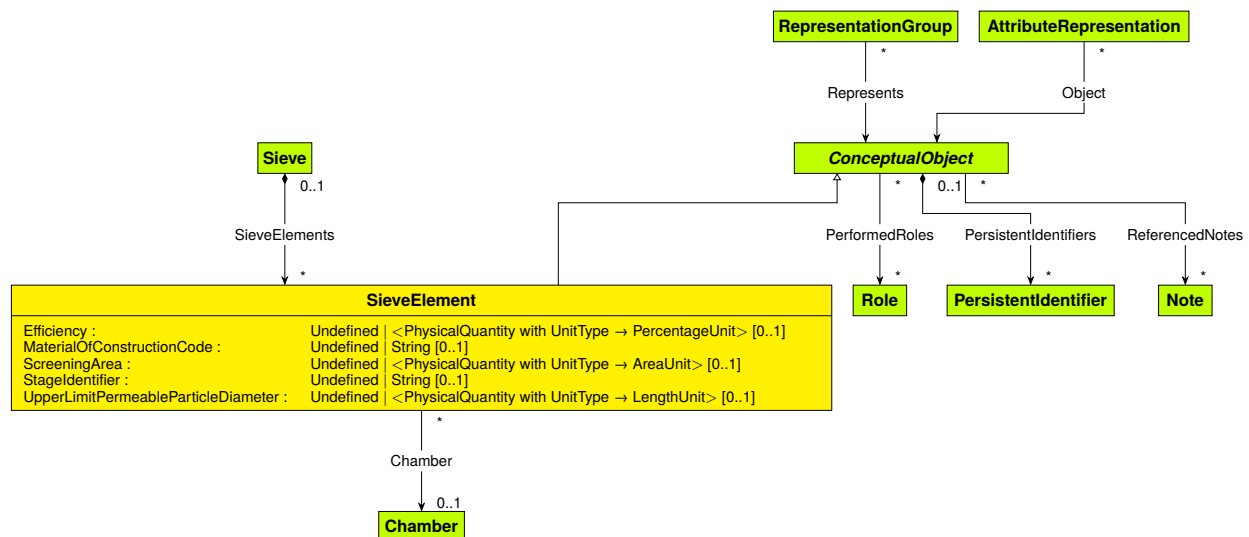
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/SieveElement>

SIEVE ELEMENT

A screening unit that is a component of a sieve.



Supertypes

- *ConceptualObject*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Chamber</i>	0..1	<i>Chamber</i>	no	no	*

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Efficiency</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PercentageUnit></i>	no	no
<i>MaterialOfConstructionCode</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>ScreeningArea</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → AreaUnit></i>	no	no
<i>StageIdentifier</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>UpperLimitPermeableParticleDiameter</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no

Chamber

Reference Property

The *Chamber* in which the SieveElement is located, if applicable. The Chamber must be a component of the same object as the SieveElement.

Type: *Chamber*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

Efficiency

Data Property

The efficiency of the *SieveElement*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PercentageUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS362654>

EFFICIENCY

Example

```
PhysicalQuantity(
  Unit = Percent,
  Value = 90.0)
```

MaterialOfConstructionCode

Data Property

A code that gives the material of construction of the *SieveElement*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS1460719741>

MATERIAL OF CONSTRUCTION CODE ASSIGNMENT CLASS

Example

"1.4306"

ScreeningArea

Data Property

The filter area of the *SieveElement*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → AreaUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/ScreeningArea>

SCREENING AREA

Example

```
PhysicalQuantity(  
    Unit = MetreSquared,  
    Value = 6.0)
```

StageIdentifier

Data Property

The stage identifier of the *SieveElement*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/StageIdentifierAssignmentClass>

STAGE IDENTIFIER ASSIGNMENT CLASS

Example

"s1"

UpperLimitPermeableParticleDiameter**Data Property**

The maximum of the particle size passing through the *SieveElement*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → LengthUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitPermeableParticleDiameter>

UPPER LIMIT PERMEABLE PARTICLE DIAMETER

Example

PhysicalQuantity(
 Unit = *Micrometre*,
 Value = 400.0)

Example in DEXPI XML

```

<Object type="Plant/ProcessEquipment.SieveElement">
  <Data property="Efficiency">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PercentageUnit.Percent"/>
      </Data>
      <Data property="Value">
        <Double>90.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="MaterialOfConstructionCode">
    <String>1.4306</String>
  </Data>
  <Data property="ScreeningArea">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.AreaUnit.MetreSquared"/>
      </Data>
      <Data property="Value">
        <Double>6.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="StageIdentifier">
    <String>s1</String>
  </Data>
</Object>

```

(continues on next page)

(continued from previous page)

```

</Data>
<Data property="UpperLimitPermeableParticleDiameter">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.LengthUnit.Micrometre"/>
    </Data>
    <Data property="Value">
      <Double>400.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>

```

Silo

Plant::ProcessEquipment::Silo

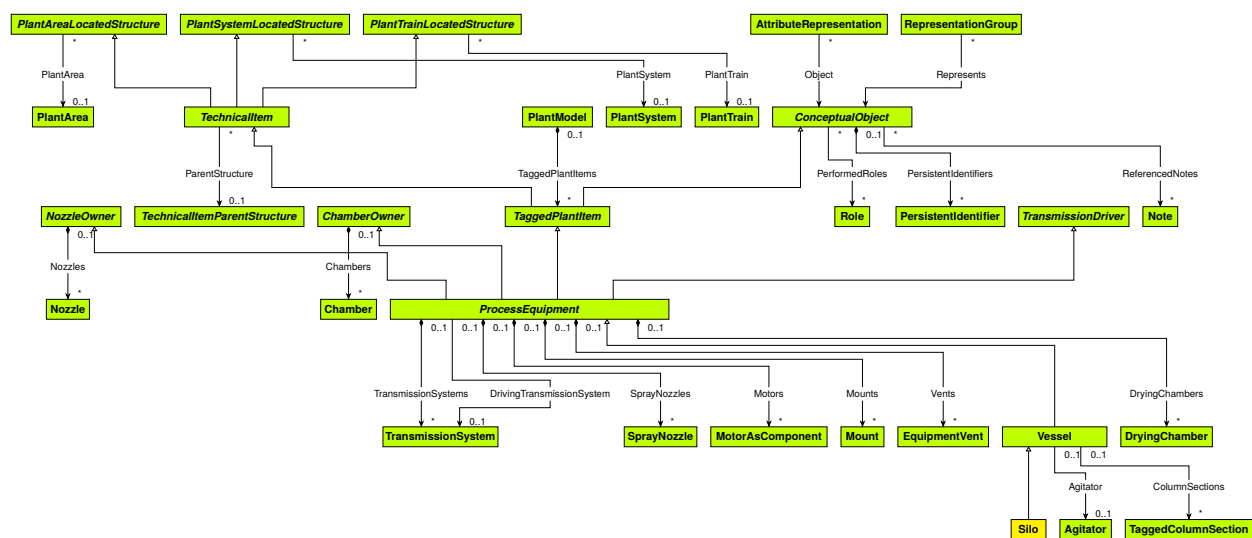
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS1022399>

SILO

A *Vessel* with a conical shape that is intended to store solids in bulk (from <http://data.15926.org/rdl/RDS1022399>).



Supertypes

- *Vessel*

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Silo"/>
```


SpiralHeatExchanger

Plant::ProcessEquipment::SpiralHeatExchanger

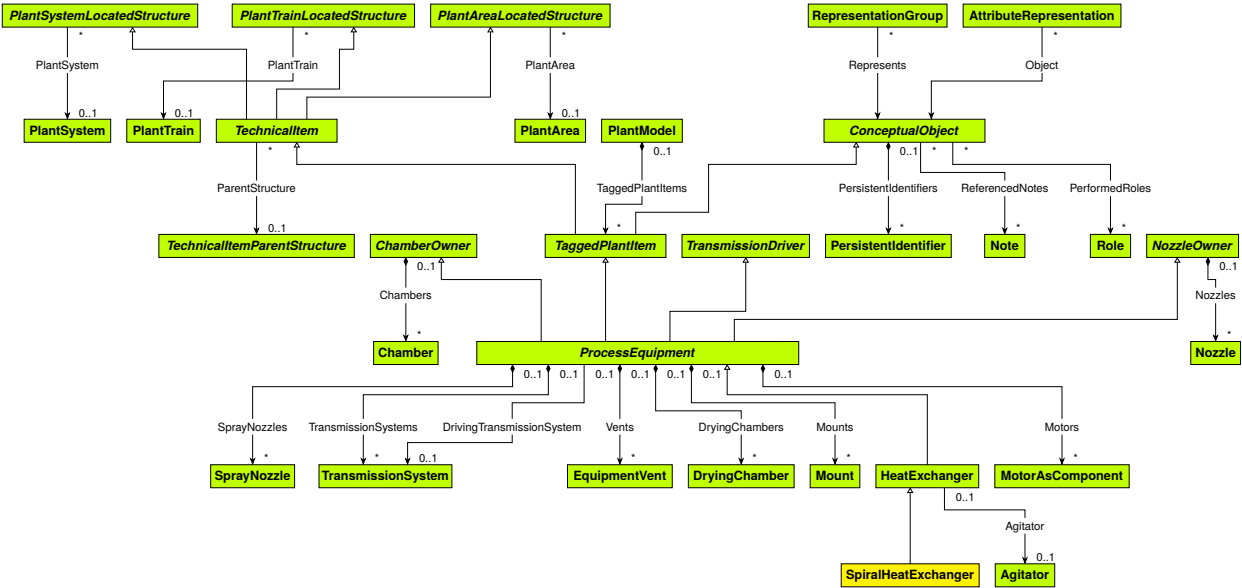
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/SpiralHeatExchanger>

SPIRAL HEAT EXCHANGER

A *HeatExchanger* in which a pair of plates is formed into a spiral.



Supertypes

- *HeatExchanger*

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.SpiralHeatExchanger"/>
```

SprayCooler

Plant::ProcessEquipment::SprayCooler

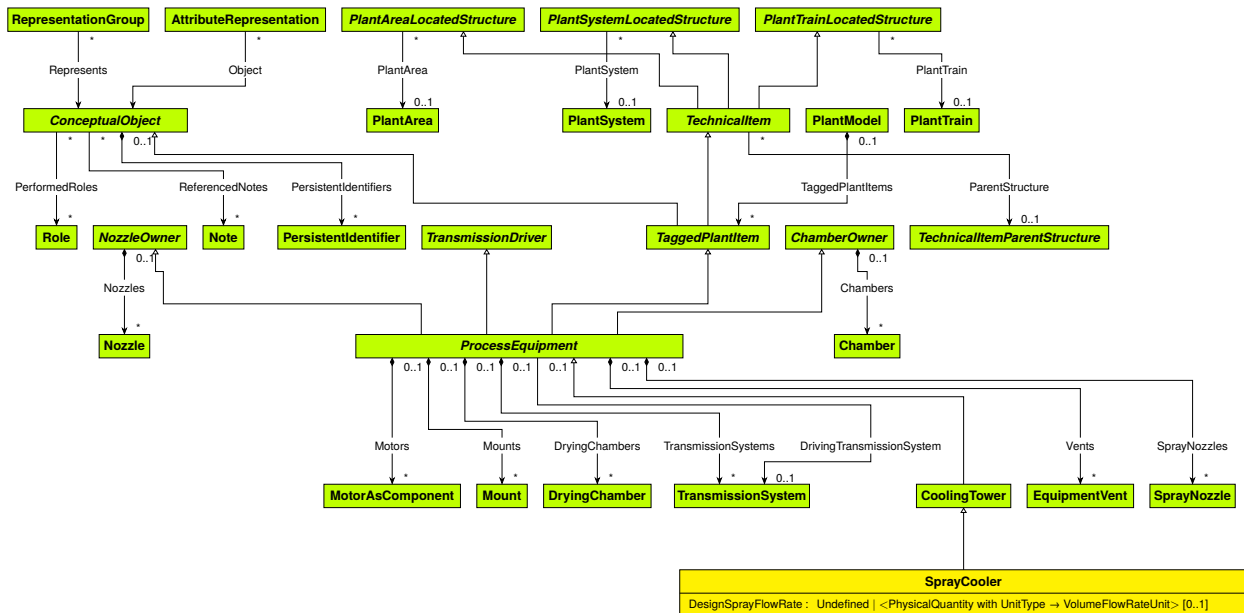
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/SprayCooler>

SPRAY COOLER

A *CoolingTower* that is based on spraying a coolant on a heated surface to be cooled.



Supertypes

- *CoolingTower*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Design-SprayFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VolumeFlowRateUnit></i>	no	no

DesignSprayFlowRate

Data Property

The spray volume flow rate for the motive fluid for which the *SprayCooler* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → VolumeFlowRateUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignSprayFlowRate>

DESIGN SPRAY FLOW RATE

Example

PhysicalQuantity(
Unit = MetreCubedPerHour,
Value = 40.0)

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.SprayCooler">  
  <Data property="DesignSprayFlowRate">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour  
      </Data>  
    </Data>  
    <Data property="Value">  
      <Double>40.0</Double>  
    </Data>  
  </AggregatedDataValue>  
</Data>  
</Object>
```

SprayNozzle

Plant::ProcessEquipment::SprayNozzle

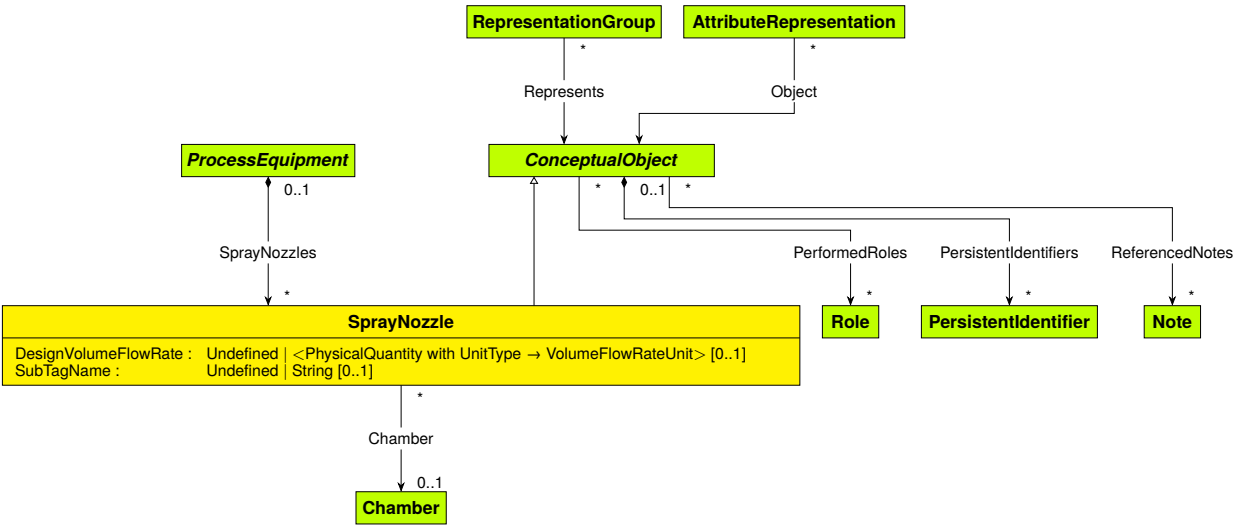
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS5855670>

SPRAY NOZZLE

A nozzle where liquid is introduced under pressure



Supertypes

- *ConceptualObject*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Chamber</i>	0..1	<i>Chamber</i>	no	no	*

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignVolumeFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VolumeFlowRateUnit></i>	no	no
<i>SubTagName</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

Chamber

Reference Property

The *Chamber* in which the *SprayNozzle* is located, if applicable. The Chamber must be a component of the same object as the *SprayNozzle*.

Type: *Chamber*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

DesignVolumeFlowRate

Data Property

The volume flow rate for which the *SprayNozzle* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → VolumeFlowRateUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS14286227>

DESIGN VOLUME FLOW RATE

Example

```
PhysicalQuantity(
  Unit = MetreCubedPerHour,
  Value = 420.0)
```

SubTagName

Data Property

The sub tag name of the *SprayNozzle*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SubTagNameAssignmentClass>

SUB TAG NAME ASSIGNMENT CLASS

Example

"ST1"

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.SprayNozzle">
  <Data property="DesignVolumeFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour
↪"/>
      </Data>
      <Data property="Value">
        <Double>420.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="SubTagName">
    <String>ST1</String>
  </Data>
</Object>
```

StaticMixer

Plant::ProcessEquipment::StaticMixer

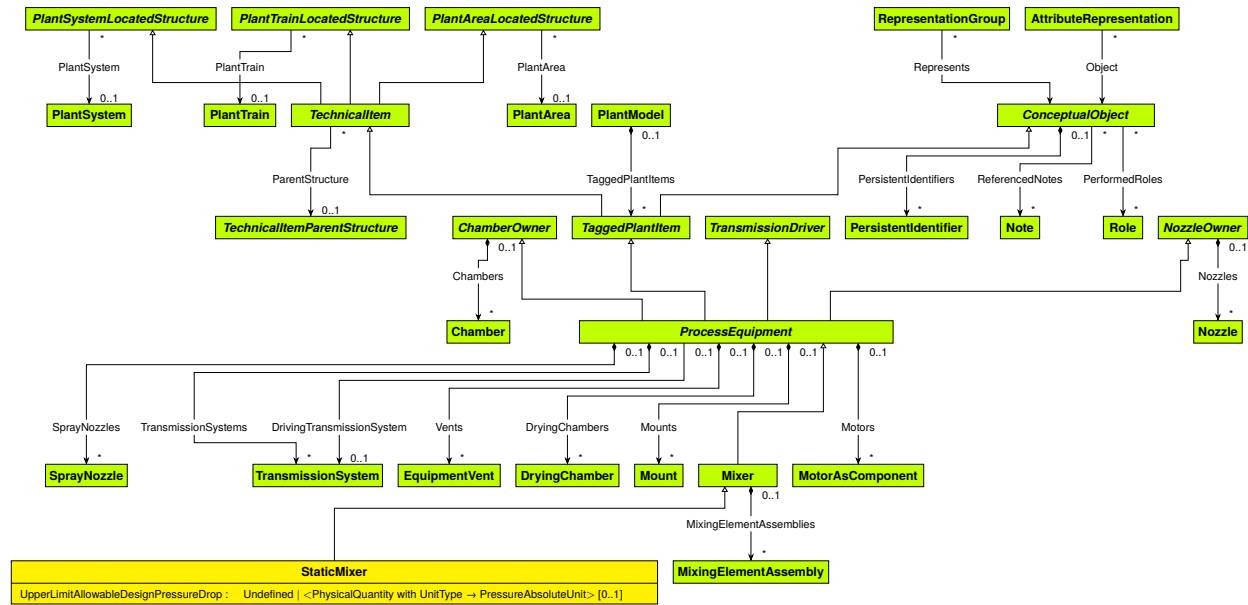
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS1016684>

STATIC MIXER

A physical object that is intended to mix fluid by means of diverging the flow with static obstacles or by increasing locally the velocity.



Supertypes

- *Mixer*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>UpperLimitAllowableDesignPressureDrop</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PressureAbsoluteUnit></i>	no	no

UpperLimitAllowableDesignPressureDrop

Data Property

The upper limit for the pressure drop for which the *StaticMixer* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PressureAbsoluteUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/UpperLimitAllowableDesignPressureDrop>

UPPER LIMIT ALLOWABLE DESIGN PRESSURE DROP

Example

 $PhysicalQuantity($

Unit = Bar,

Value = 2.0)

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.StaticMixer">
  <Data property="UpperLimitAllowableDesignPressureDrop">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PressureAbsoluteUnit.Bar"/>
      </Data>
      <Data property="Value">
        <Double>2.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

Stationary Sieve

Plant::ProcessEquipment::StationarySieve

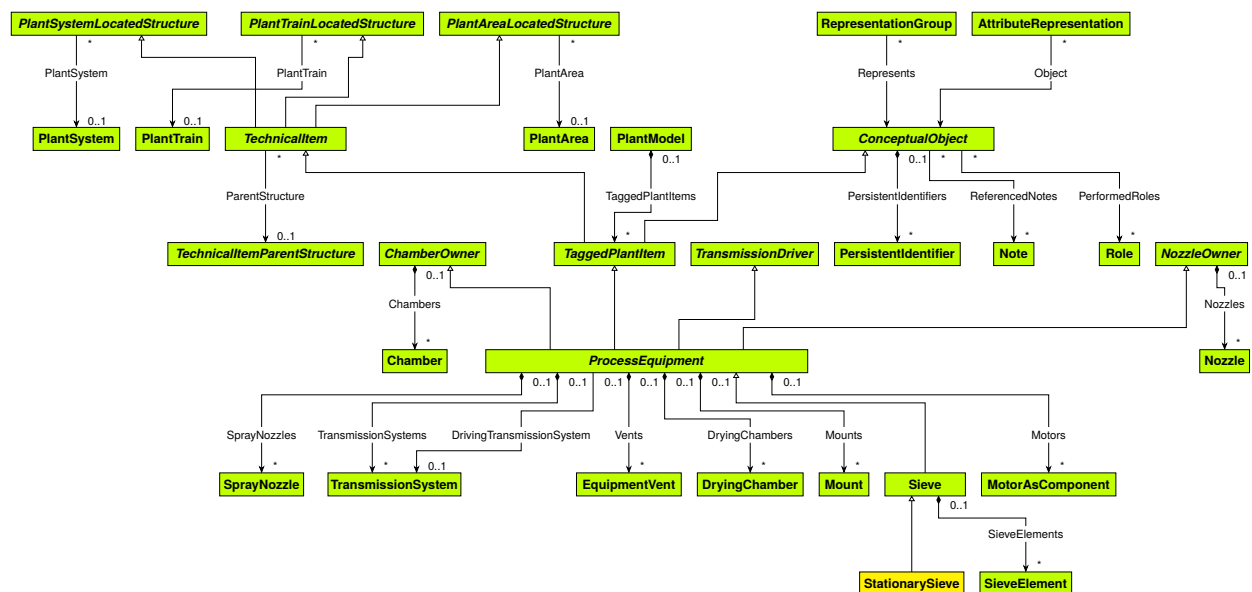
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/StationaryScreen>

STATIONARY SCREEN

A *Sieve* consisting of rakes or sieves, that, during operation, remains in a fixed position (from <http://data.15926.org/rdl/RDS2226669>).



Supertypes

- *Sieve*

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.StationarySieve"/>
```

StationaryTransportSystem

Plant::ProcessEquipment::StationaryTransportSystem

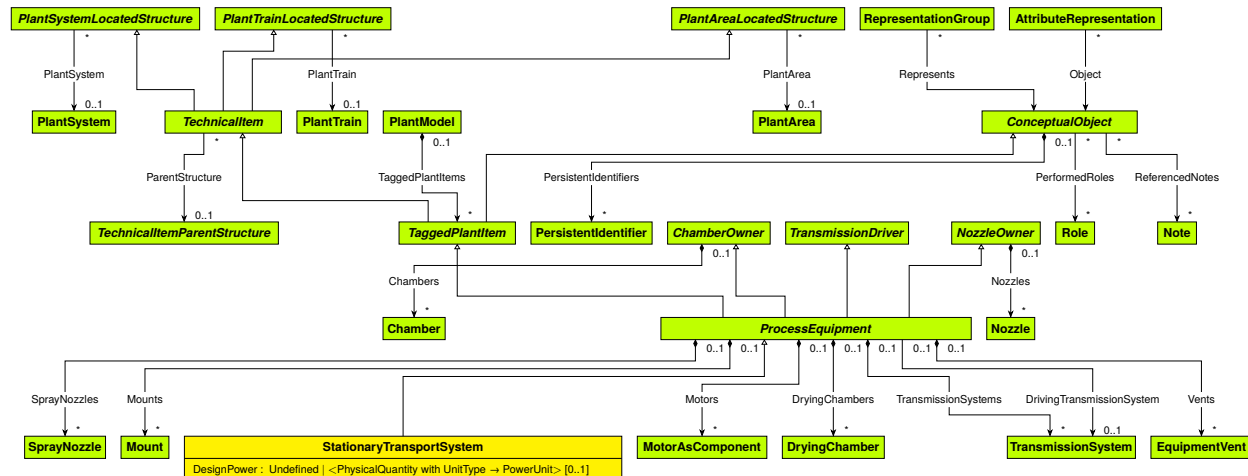
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/StationaryTransportSystem>

STATIONARY TRANSPORT SYSTEM

A transport system that is intended to transport, store or load/unload material and that, as a whole, remains in one place.



Supertypes

- *ProcessEquipment*

Subtypes

- *Conveyor*
- *Lift*
- *LoadingUnloadingSystem*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no

DesignPower

Data Property

The power for which the *StationaryTransportSystem* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PowerUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignPower>

DESIGN POWER

Example

```
PhysicalQuantity(
    Unit = Kilowatt,
    Value = 500.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.StationaryTransportSystem">
  <Data property="DesignPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>500.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

SteamGenerator

Plant::ProcessEquipment::SteamGenerator

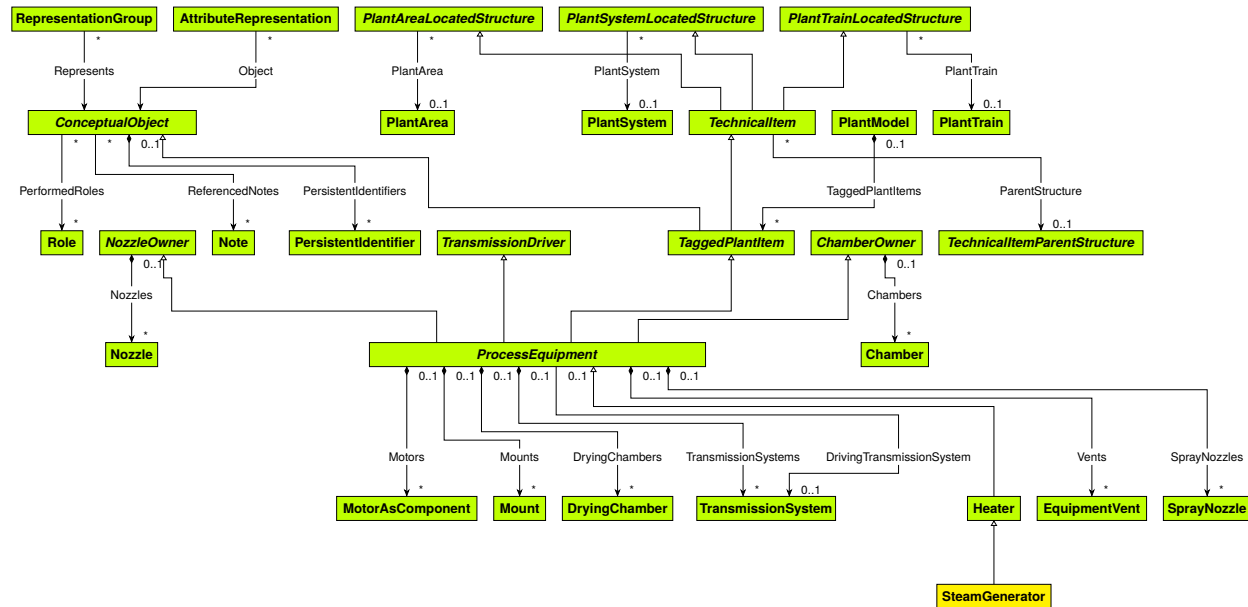
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS13306207>

STEAM GENERATOR

A boiler that is intended to generate steam (from <http://data.posccaesar.org/rdl/RDS13306207>).



Supertypes

- *Heater*

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.SteamGenerator"/>
```

SteamTurbine

Plant::ProcessEquipment::SteamTurbine

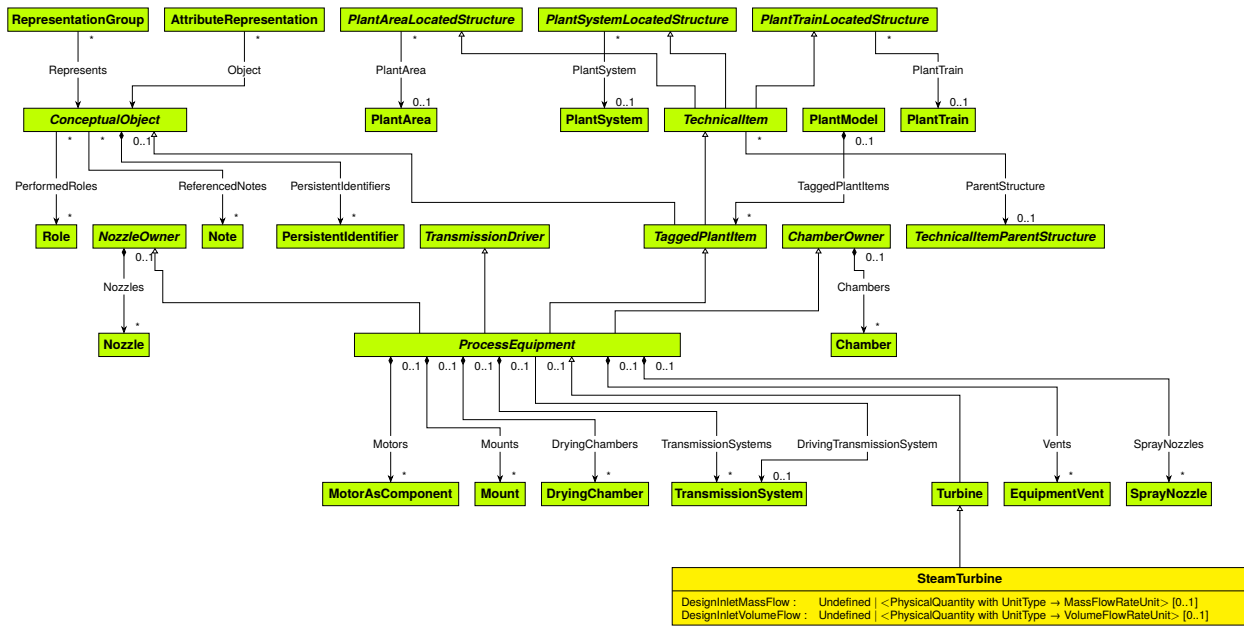
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS416744>

STEAM TURBINE

A turbine that is a heat engine in which energy of steam is transformed into kinetic energy by means of expansion through nozzles and the kinetic energy of the resulting jet is in turn converted into force doing work on rings of blading mounted on a rotating shaft (from <http://data.posccaesar.org/rdl/RDS416744>).



Supertypes

- *Turbine*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignInletMassFlow</i>	0..1	<i>Undefined</i> <PhysicalQuantity with UnitType → MassFlowRateUnit>	no	no
<i>DesignInletVolumeFlow</i>	0..1	<i>Undefined</i> <PhysicalQuantity with UnitType → VolumeFlowRateUnit>	no	no

DesignInletMassFlow

Data Property

The inlet mass flow for which the *SteamTurbine* is designed.

Type: *Undefined* | <PhysicalQuantity with UnitType → MassFlowRateUnit>

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignInletMassFlow>

DESIGN INLET MASS FLOW

Example

```
PhysicalQuantity(  
  Unit = KilogramPerHour,  
  Value = 800.0)
```

DesignInletVolumeFlow

Data Property

The inlet volume flow for which the *SteamTurbine* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VolumeFlowRateUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignInletVolumeFlow>

DESIGN INLET VOLUME FLOW

Example

```
PhysicalQuantity(  
  Unit = LitrePerSecond,  
  Value = 2.8)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.SteamTurbine">  
  <Data property="DesignInletMassFlow">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour"/>  
      </Data>  
      <Data property="Value">  
        <Double>800.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
  <Data property="DesignInletVolumeFlow">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.LitrePerSecond"/>  
      </Data>  
      <Data property="Value">  
        <Double>2.8</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
</Object>
```

SubTaggedColumnSection

Plant::ProcessEquipment::SubTaggedColumnSection

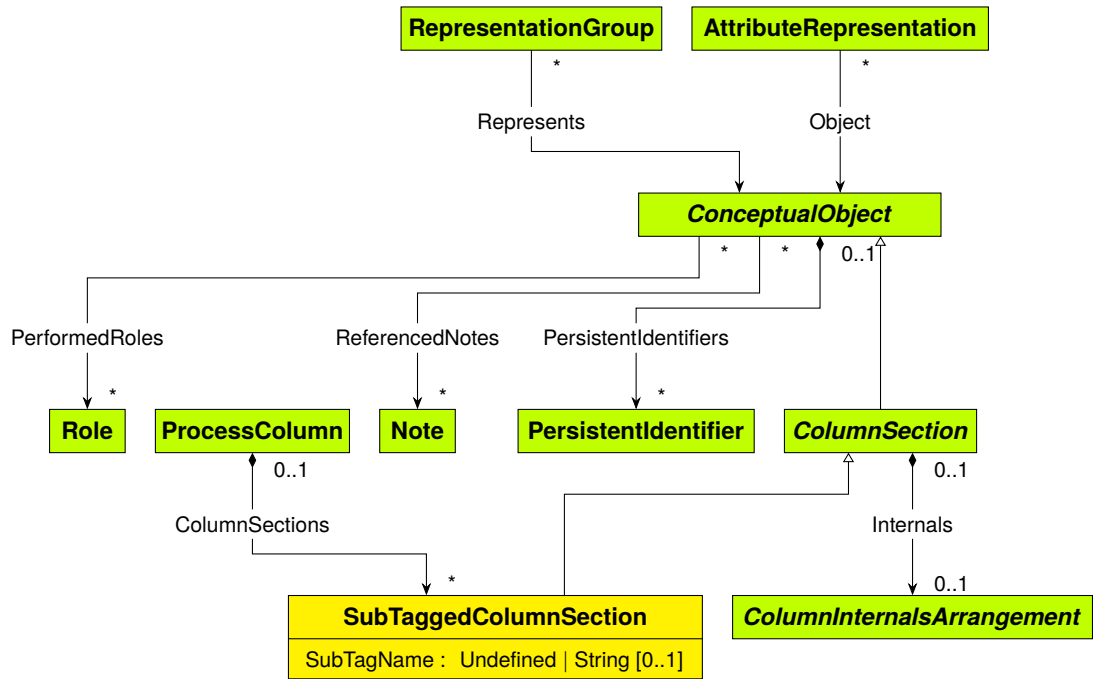
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/ColumnSection>

COLUMN SECTION

A sub tagged column section.



Supertypes

- *ColumnSection*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>SubTagName</i>	0..1	<i>Undefined String</i>	no	no

SubTagName

Data Property

The sub tag name of the *SubTaggedColumnSection*.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SubTagNameAssignmentClass>

SUB TAG NAME ASSIGNMENT CLASS

Example

"ST1"

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.SubTaggedColumnSection">
  <Data property="SubTagName">
    <String>ST1</String>
  </Data>
</Object>
```

TaggedColumnSection

Plant::ProcessEquipment::TaggedColumnSection

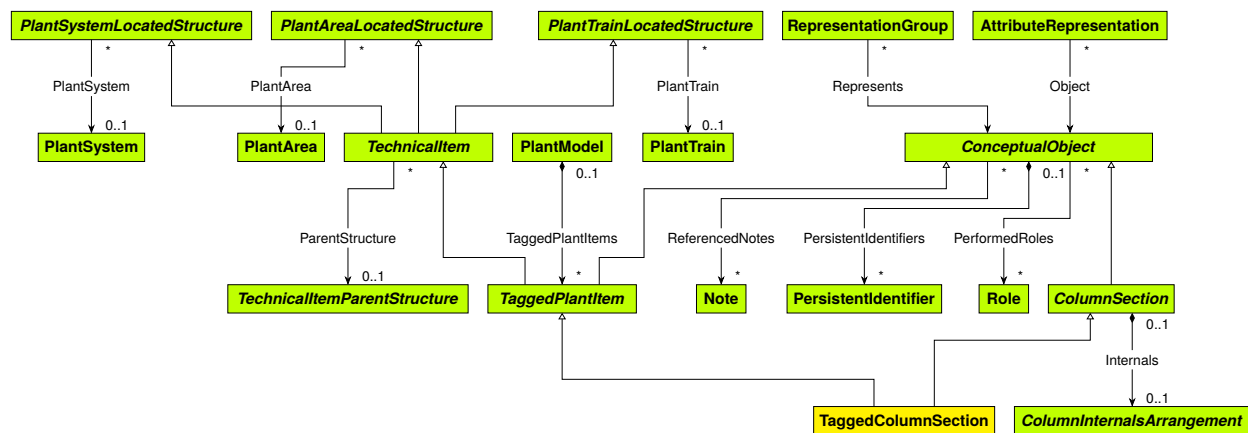
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/ColumnSection>

COLUMN SECTION

A fully tagged column section.



Supertypes

- *ColumnSection*
- *TaggedPlantItem*

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.TaggedColumnSection"/>
```

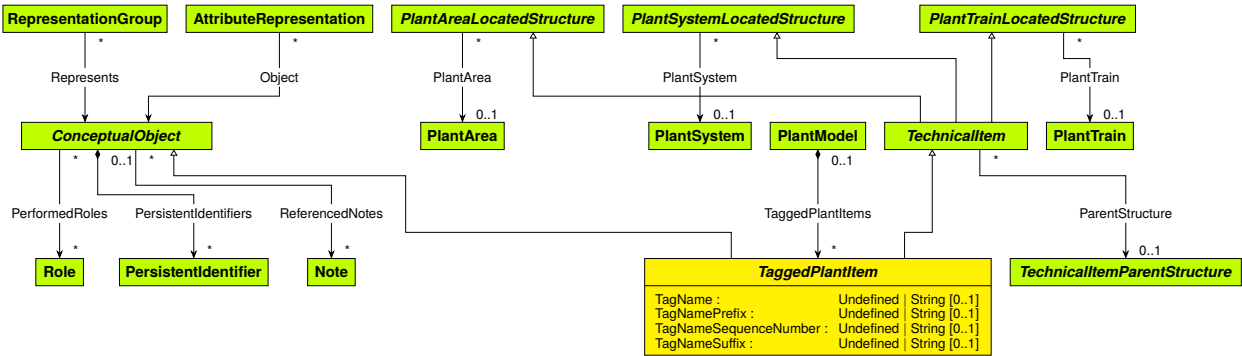
TaggedPlantItem

Plant::ProcessEquipment::TaggedPlantItem

Overview

Abstract Class

A fully tagged item in a plant.



Supertypes

- *ConceptualObject*
- *TechnicalItem*

Subtypes

- *ProcessEquipment*
- *TaggedColumnSection*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>TagName</i>	0..1	Undefined String	no	no
<i>TagNamePrefix</i>	0..1	Undefined String	no	no
<i>TagNameSequenceNumber</i>	0..1	Undefined String	no	no
<i>TagNameSuffix</i>	0..1	Undefined String	no	no

TagName

Data Property

The tag number of the *TaggedPlantItem*. See also *TagNamePrefix*, *TagNameSequenceNumber*, and *TagNameSuffix*.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/TagNameAssignmentClass>

TAG NAME ASSIGNMENT CLASS

Example

"P4714-A"

TagNamePrefix

Data Property

The prefix part of the tag number of the *TaggedPlantItem*. For example, the prefix of the tag number “P4714-A” is “P”. The prefix often indicates the type of the equipment item, e.g., “P” can indicate a pump. See also *TagName*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/TagNamePrefixAssignmentClass>

TAG NAME PREFIX ASSIGNMENT CLASS

Example

"P"

TagNameSequenceNumber

Data Property

The sequence number part of the tag number of the *TaggedPlantItem*. For example, the sequence number of the tag number “P4714-A” is “4714”.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/TagNameSequenceNumberAssignmentClass>

TAG NAME SEQUENCE NUMBER ASSIGNMENT CLASS

Example

"4714"

TagNameSuffix

Data Property

The suffix part of the tag number of an *TaggedPlantItem* item. For example, the suffix of the tag number “P4714-A” is “A”.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/TagNameSuffixAssignmentClass>

TAG NAME SUFFIX ASSIGNMENT CLASS

Example

"A"

Tank

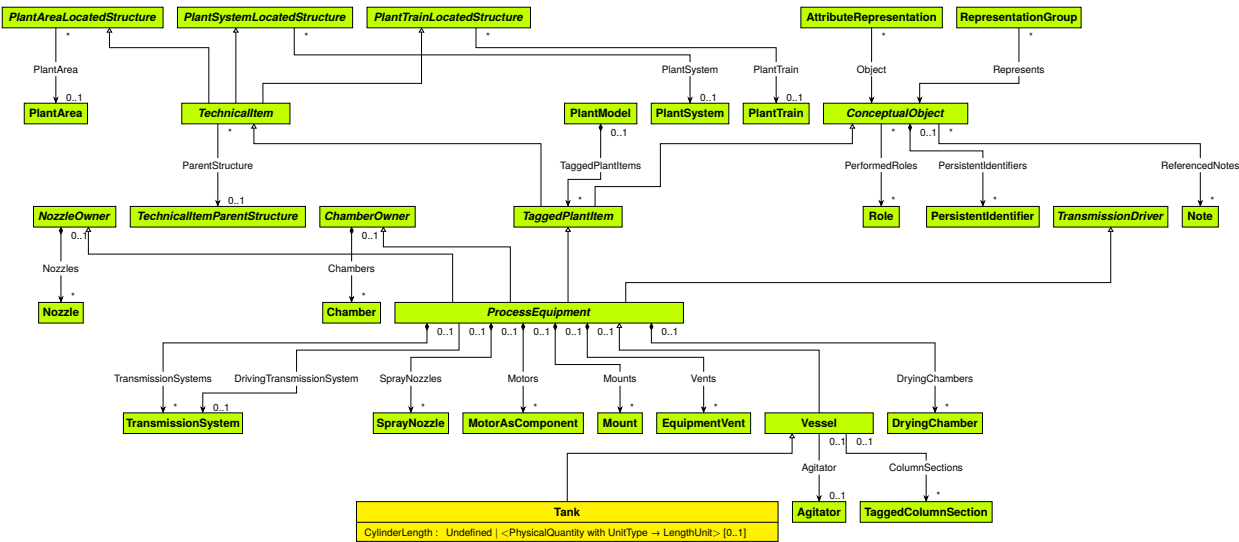
Plant::ProcessEquipment::Tank

Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS445139>

TANK



Supertypes

- *Vessel*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>CylinderLength</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → LengthUnit></i>	no	no

CylinderLength

Data Property

The cylinder length of the *Tank*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → LengthUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/CylinderLength>

CYLINDER LENGTH

Example

```
PhysicalQuantity(
  Unit = Metre,
  Value = 2.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Tank">
  <Data property="CylinderLength">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Metre"/>
      </Data>
      <Data property="Value">
        <Double>2.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

ThinFilmEvaporator

Plant::ProcessEquipment::ThinFilmEvaporator

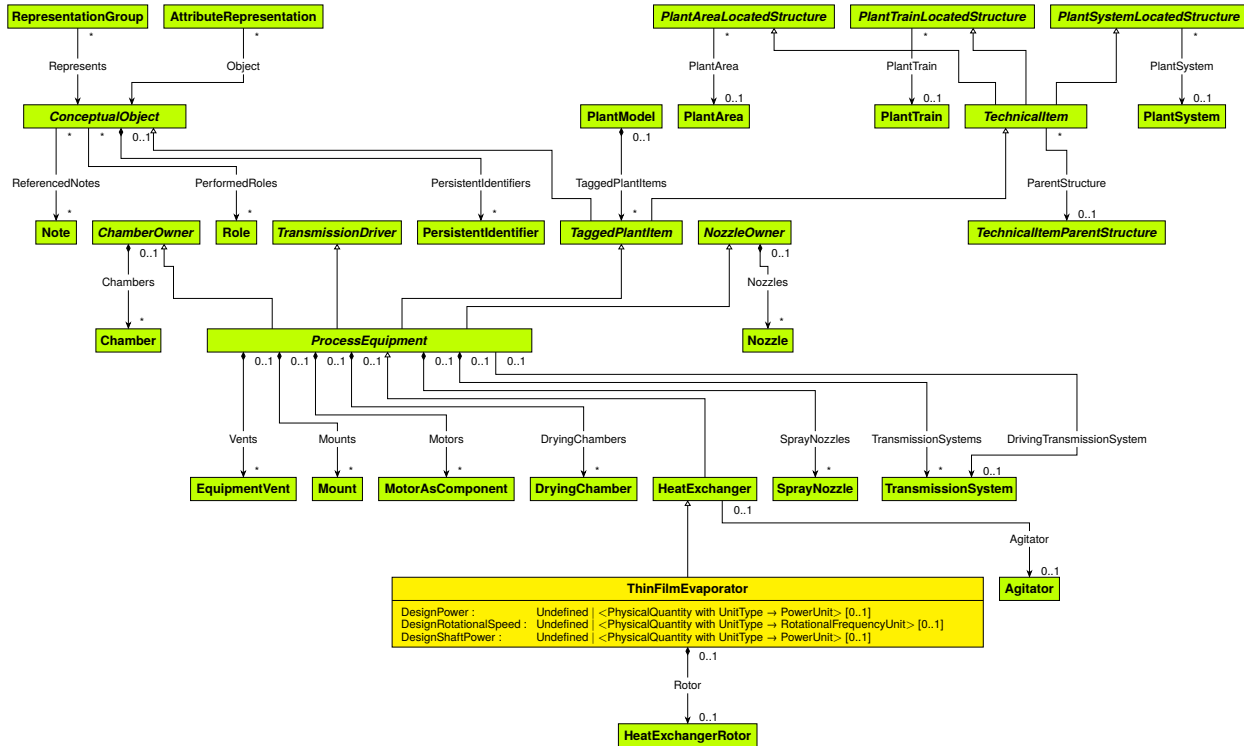
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/ThinFilmEvaporator>

THIN FILM EVAPORATOR

A *HeatExchanger* and evaporator for the purification of temperature-sensitive products by evaporation, where a thin film of the liquid product on the inner side of a vertical evaporation pipe is generated by a rotating wiper system.



Supertypes

- *HeatExchanger*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Rotor</i>	0..1	<i>HeatExchangerRotor</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no
<i>DesignRotational-Speed</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → Rotational-FrequencyUnit></i>	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no

DesignPower

Data Property

The power for which the *ThinFilmEvaporator* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PowerUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignPower>

DESIGN POWER

Example

```
PhysicalQuantity(  
  Unit = Kilowatt,  
  Value = 500.0)
```

DesignRotationalSpeed

Data Property

The rotational speed for which the *ThinFilmEvaporator* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(  
  Unit = ReciprocalMinute,  
  Value = 180.0)
```

DesignShaftPower

Data Property

The shaft power for which the *ThinFilmEvaporator* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(  
    Unit = Kilowatt,  
    Value = 400.0)
```

Rotor

Composition Property

The rotor of the *ThinFilmEvaporator*.

Type: *HeatExchangerRotor*

Multiplicity: 0..1

Ordered: no

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.ThinFilmEvaporator">  
  <Data property="DesignPower">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>  
      </Data>  
      <Data property="Value">  
        <Double>500.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
  <Data property="DesignRotationalSpeed">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.  
↪ ReciprocalMinute"/>  
      </Data>  
      <Data property="Value">  
        <Double>180.0</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
</Object>
```

(continues on next page)

(continued from previous page)

```
</Data>
</AggregatedDataValue>
</Data>
<Data property="DesignShaftPower">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
    </Data>
    <Data property="Value">
      <Double>400.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>
```

TransmissionDriver

Plant::ProcessEquipment::TransmissionDriver

Overview

Abstract Class

A Driver applied to drive *TransmissionSystems*.

TransmissionDriver

Subtypes

- *MotorAsComponent*
- *ProcessEquipment*
- *TransmissionSystem*

TransmissionSystem

Plant::ProcessEquipment::TransmissionSystem

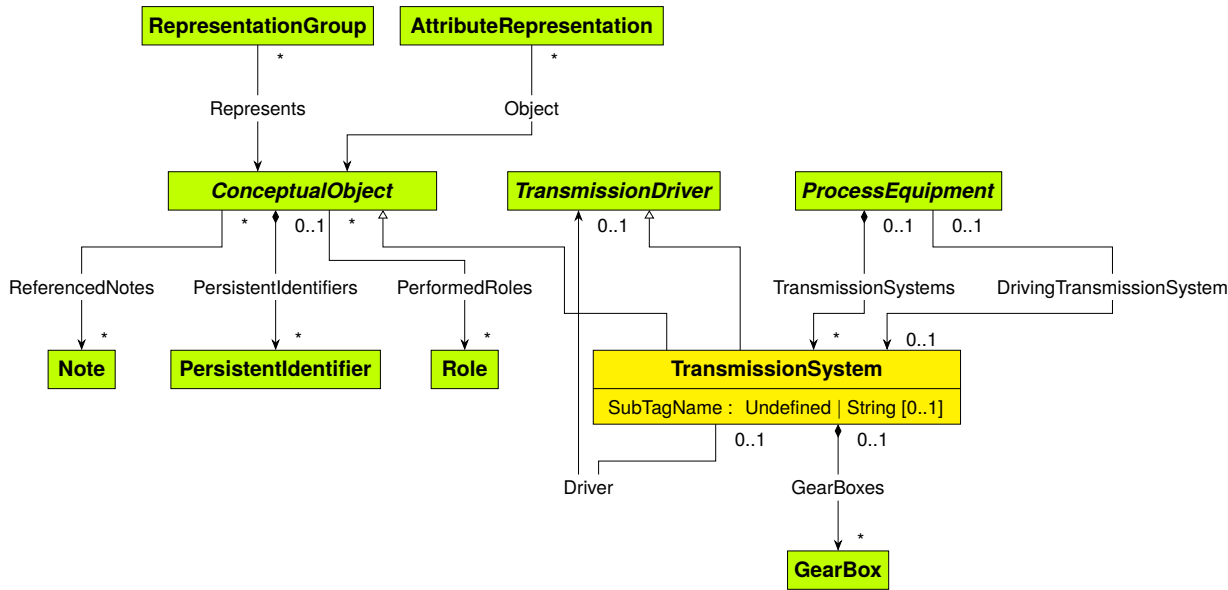
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS13105034>

TRANSMISSION SYSTEM

A rotating equipment system that can transmit mechanical power from its *Driver* to driven *Plant.ProcessEquipment.Equipment* by means of an assembly of transmission components and auxiliaries.



Supertypes

- *ConceptualObject*
- *TransmissionDriver*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>GearBoxes</i>	*	<i>GearBox</i>	no

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Driver</i>	0..1	<i>TransmissionDriver</i>	no	no	0..1

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>SubTagName</i>	0..1	<i>Undefined String</i>	no	no

Driver

Reference Property

The *TransmissionDriver* that drives the TransmissionSystem.

Type: *TransmissionDriver*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

GearBoxes

Composition Property

The gear boxes that are components of the TransmissionSystem.

Type: *GearBox*

Multiplicity: *

Ordered: no

SubTagName

Data Property

The sub tag name of the *TransmissionSystem*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/SubTagNameAssignmentClass>

SUB TAG NAME ASSIGNMENT CLASS

Example

"ST1"

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.TransmissionSystem">
  <Data property="SubTagName">
    <String>ST1</String>
  </Data>
</Object>
```

TransportableContainer

*Plant::ProcessEquipment::*TransportableContainer

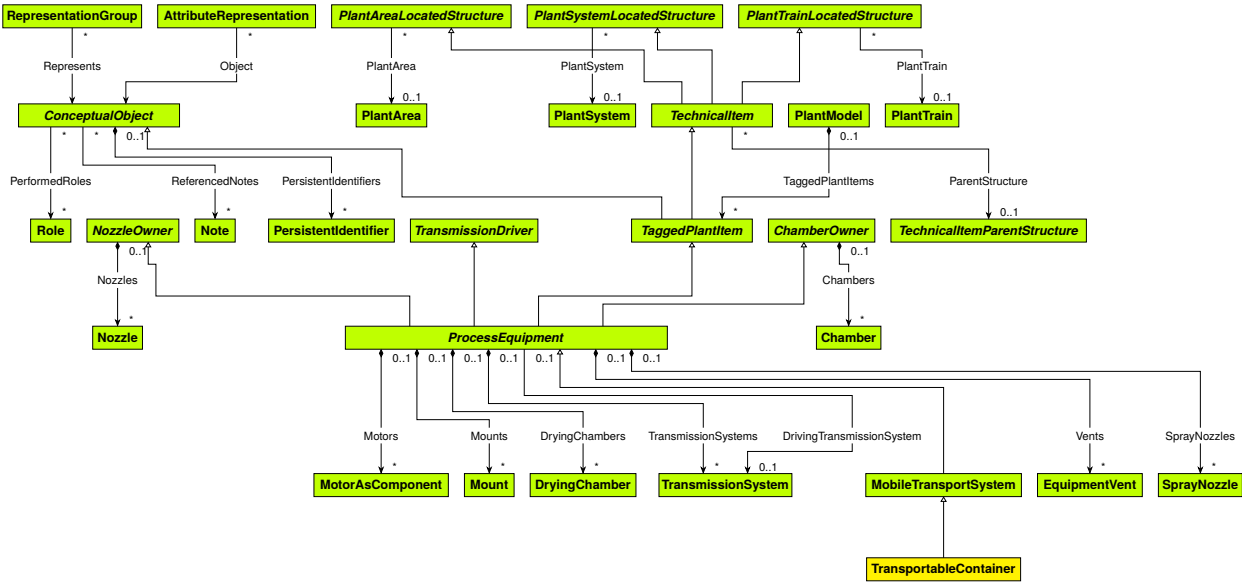
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS22164402859>

TRANSPORTABLE CONTAINER

A ‘container’ that is a transportable, with strength suitable to withstand shipment, storage, and handling (from <http://data.posccaesar.org/rdl/RDS22164402859>).



Supertypes

- *MobileTransportSystem*

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.TransportableContainer"/>
```

Truck

Plant::ProcessEquipment::Truck

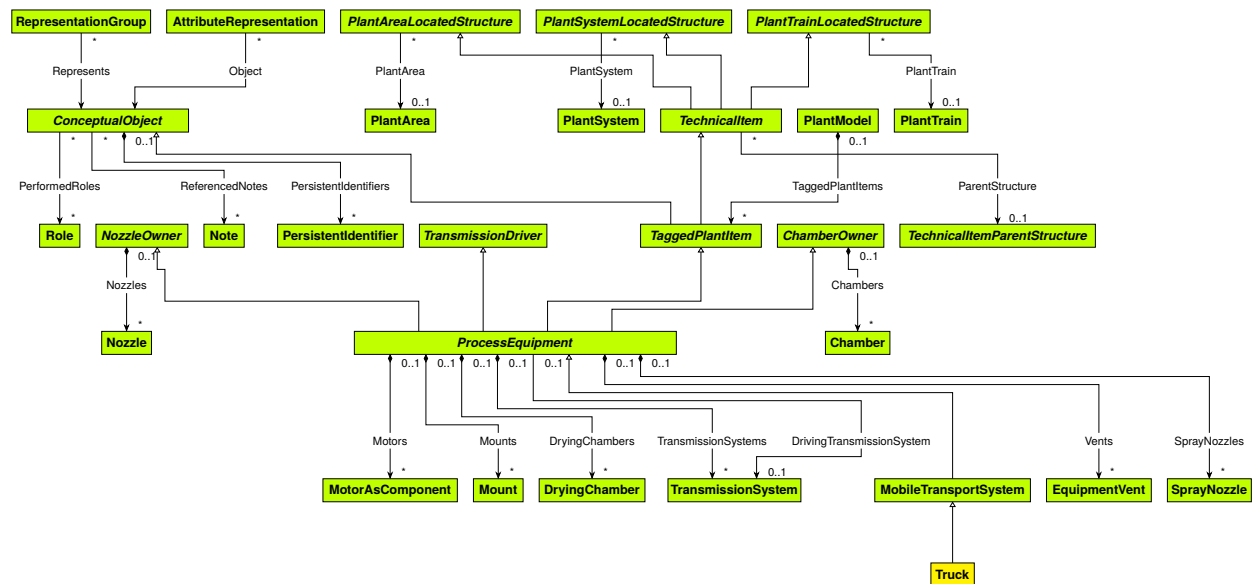
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS11524112>

TRUCK

An automotive vehicle that is long, low and open intended for carrying goods by road (from <http://data.posccaesar.org/rdl/RDS11524112>).



Supertypes

- *MobileTransportSystem*

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Truck"/>
```

TubeBundle

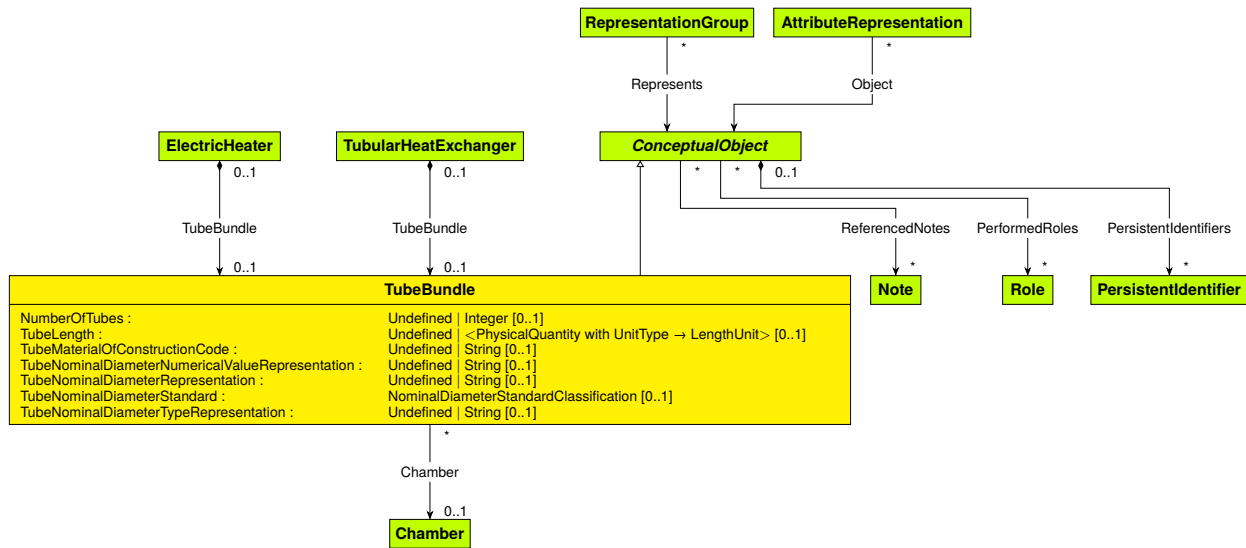
Plant::ProcessEquipment::TubeBundle

Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS415259>

TUBE BUNDLE



Supertypes

- *ConceptualObject*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Chamber</i>	0..1	<i>Chamber</i>	no	no	*

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>NumberOfTubes</i>	0..1	<i>Undefined Integer</i>	no	no
<i>TubeLength</i>	0..1	<i>Undefined <PhysicalQuantity with Unit-Type → LengthUnit></i>	no	no
<i>TubeMaterialOfConstructionCode</i>	0..1	<i>Undefined String</i>	no	no
<i>TubeNominalDiameterNumericalValueRepresentation</i>	0..1	<i>Undefined String</i>	no	no
<i>TubeNominalDiameterRepresentation</i>	0..1	<i>Undefined String</i>	no	no
<i>TubeNominalDiameterStandard</i>	0..1	<i>NominalDiameterStandardClassification</i>	no	no
<i>TubeNominalDiameterTypeRepresentation</i>	0..1	<i>Undefined String</i>	no	no

Chamber

Reference Property

The *Chamber* in which the TubeBundle is located, if applicable. The Chamber must be a component of the same object as the TubeBundle.

Type: *Chamber*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

NumberOfTubes

Data Property

The number of tubes of the *TubeBundle*.

Type: *Undefined* | *Integer*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS363959>

NUMBER OF TUBES

Example

36

TubeLength

Data Property

The length of the tubes of the *TubeBundle*.

Type: *Undefined* | $\langle \textit{PhysicalQuantity}$ with $\textit{UnitType} \rightarrow \textit{LengthUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS363869>

TUBE LENGTH

Example

```
PhysicalQuantity(  
  Unit = Millimetre,  
  Value = 2200.0)
```

TubeMaterialOfConstructionCode

Data Property

A code that gives the material of construction of the tubes of the *TubeBundle*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/TubeMaterialOfConstructionCodeAssignmentClass>

TUBE MATERIAL OF CONSTRUCTION CODE ASSIGNMENT CLASS

Example

"1.4306"

TubeNominalDiameterNumericalValueRepresentation

Data Property

A readable representation of the numerical value of the nominal diameter of the tubes.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/TubeNominalDiameterNumericalValueRepresentationAssignmentClass>

TUBE NOMINAL DIAMETER NUMERICAL VALUE REPRESENTATION ASSIGNMENT CLASS

Example

"25"

TubeNominalDiameterRepresentation

Data Property

A readable representation of the nominal diameter of the tubes.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/TubeNominalDiameterRepresentationAssignmentClass>

TUBE NOMINAL DIAMETER REPRESENTATION ASSIGNMENT CLASS

Example

"DN 25"

TubeNominalDiameterStandard

Data Property

The nominal diameter of the tubes, given as a reference to a nominal diameter standard and value.

Type: *NominalDiameterStandardClassification*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/TubeNominalDiameterStandardSpecialization>

TUBE NOMINAL DIAMETER STANDARD SPECIALIZATION

Example

Din2448ObjectDn25

TubeNominalDiameterTypeRepresentation

Data Property

A readable representation of the type of the nominal diameter of the tubes.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/TubeNominalDiameterTypeRepresentationAssignmentClass>

TUBE NOMINAL DIAMETER TYPE REPRESENTATION ASSIGNMENT CLASS

Example

"DN"

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.TubeBundle">
  <Data property="NumberOfTubes">
    <Integer>36</Integer>
  </Data>
  <Data property="TubeLength">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.LengthUnit.Millimetre"/>
      </Data>
      <Data property="Value">
        <Double>2200.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
```

(continues on next page)

Supertypes

- *HeatExchanger*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>TubeBundle</i>	0..1	<i>TubeBundle</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>TemaStandardType</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

TemaStandardType

Data Property

The type of the *TubularHeatExchanger* according to the Tubular Exchanger Manufacturers Association, Inc. (TEMA, url{<http://www.tema.org>}). This is a three-letter code.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/TemaStandardTypeAssignmentClass>

TEMA STANDARD TYPE ASSIGNMENT CLASS

Example

"AEL"

TubeBundle

Composition Property

The tube bundle of the *TubularHeatExchanger*.

Type: *TubeBundle*

Multiplicity: 0..1

Ordered: no

Example in DEXPI XML


```
<Object type="Plant/ProcessEquipment.TubularHeatExchanger">
  <Data property="TemaStandardType">
    <String>AEL</String>
  </Data>
</Object>
```

Turbine

Plant::ProcessEquipment::Turbine

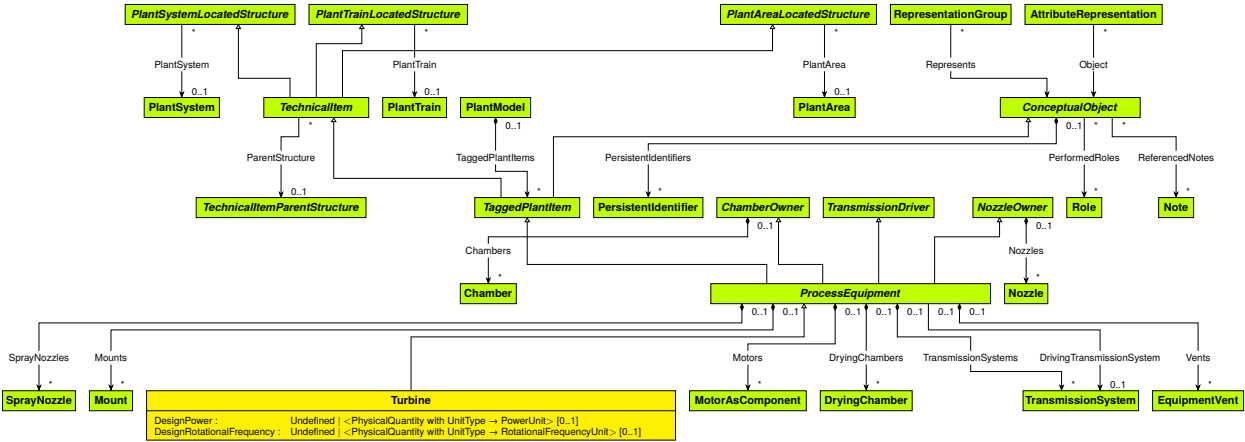
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS313289>

TURBINE

An object that is a rotary mechanical device that extracts energy from a fluid flow and converts it into useful work (from <http://data.15926.org/rdl/RDS313289>).



Supertypes

- *ProcessEquipment*

Subtypes

- *GasTurbine*
- *SteamTurbine*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no
<i>DesignRotational-Frequency</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → RotationalFrequencyUnit></i>	no	no

DesignPower

Data Property

The power for which the *Turbine* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → PowerUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignPower>

DESIGN POWER

Example

```
PhysicalQuantity(  
  Unit = Kilowatt,  
  Value = 500.0)
```

DesignRotationalFrequency

Data Property

The rotational frequency for which the *Turbine* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalFrequency>

DESIGN ROTATIONAL FREQUENCY

Example

```
PhysicalQuantity(  
  Unit = ReciprocalMinute,  
  Value = 180.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Turbine">
  <Data property="DesignPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>500.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
  <Data property="DesignRotationalFrequency">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ ReciprocalMinute"/>
      </Data>
      <Data property="Value">
        <Double>180.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

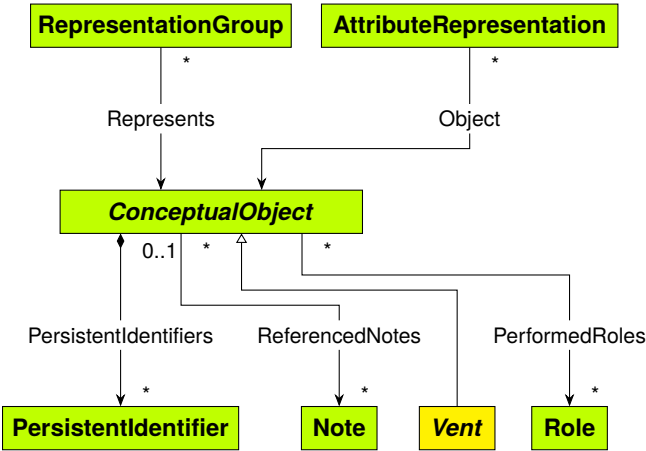
Vent

Plant::ProcessEquipment::Vent

Overview

Abstract Class

An object that has the capability of venting by being or having an opening that allows air, smoke, or gas to enter or leave a closed space to relief pressure and prevent vacuum.



Supertypes

- *ConceptualObject*

Subtypes

- *EquipmentVent*
- *VentLine*

Vessel

Plant::ProcessEquipment::Vessel

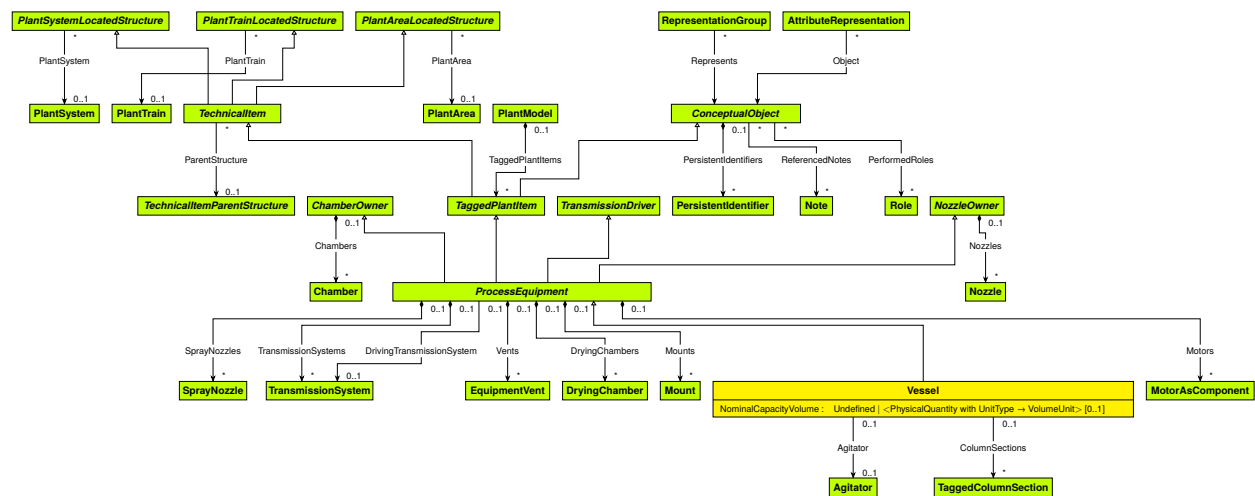
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS414674>

VESSEL

A container intended for storage and/or processing of fluids or solids.



Supertypes

- *ProcessEquipment*

Subtypes

- *PressureVessel*
- *Silo*
- *Tank*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Agitator</i>	0..1	<i>Agitator</i>	no	no	0..1
<i>ColumnSections</i>	*	<i>TaggedColumnSection</i>	no	no	0..1

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>NominalCapacityVolume</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VolumeUnit></i>	no	no

Agitator

Reference Property

The *Agitator* of the Vessel, if applicable.

Type: *Agitator*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

ColumnSections

Reference Property

The column sections of the Vessel, if applicable.

Type: *TaggedColumnSection*

Multiplicity: *

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

NominalCapacityVolume

Data Property

The nominal volumetric capacity of the *Vessel*.

Type: *Undefined* | *<PhysicalQuantity with UnitType → VolumeUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/NominalCapacityVolume>

NOMINAL CAPACITY VOLUME

Example

```
PhysicalQuantity(  
    Unit = MetreCubed,  
    Value = 7.2)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Vessel">  
  <Data property="NominalCapacityVolume">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.VolumeUnit.MetreCubed"/>  
      </Data>  
      <Data property="Value">  
        <Double>7.2</Double>  
      </Data>  
    </AggregatedDataValue>  
  </Data>  
</Object>
```

VibratingSieve

Plant::ProcessEquipment::VibratingSieve

Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/VibratingScreen>

VIBRATING SCREEN

A *Sieve* where the product to be sieved is transported over the mesh by vibration of the latter (from <http://data.15926.org/rdl/RDS2226670>).

Name	Multiplicity	Type	Ordered	Unique
<i>DesignPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity</i> with <i>UnitType</i> $\rightarrow$ <i>PowerUnit</i> <i>></i>	no	no

Example

```
PhysicalQuantity(
  Unit = Kilowatt,
  Value = 500.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.VibratingSieve">
  <Data property="DesignPower">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>500.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
```

WasteGasEmitter

Plant::ProcessEquipment::WasteGasEmitter

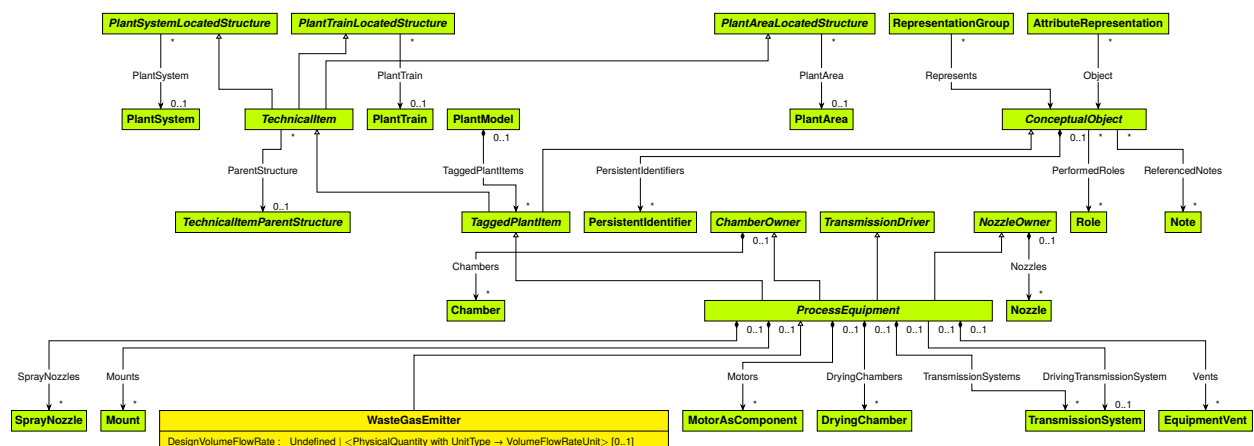
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/WasteGasEmitter>

WASTE GAS EMITTER

A physical object that is intended to release/emit waste gas from the process.



Supertypes

- *ProcessEquipment*

Subtypes

- *Chimney*
- *Flare*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignVolumeFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → VolumeFlowRateUnit></i>	no	no

DesignVolumeFlowRate

Data Property

The volume flow rate for which the *WasteGasEmitter* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → VolumeFlowRateUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS14286227>

DESIGN VOLUME FLOW RATE

Example

```
PhysicalQuantity(
  Unit = MetreCubedPerHour,
  Value = 420.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.WasteGasEmitter">
  <Data property="DesignVolumeFlowRate">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour"
↪      />
      </Data>
      <Data property="Value">
        <Double>420.0</Double>
      </Data>
```

(continues on next page)

(continued from previous page)

```

</AggregatedDataValue>
</Data>
</Object>

```

Weigher

Plant::ProcessEquipment::Weigher

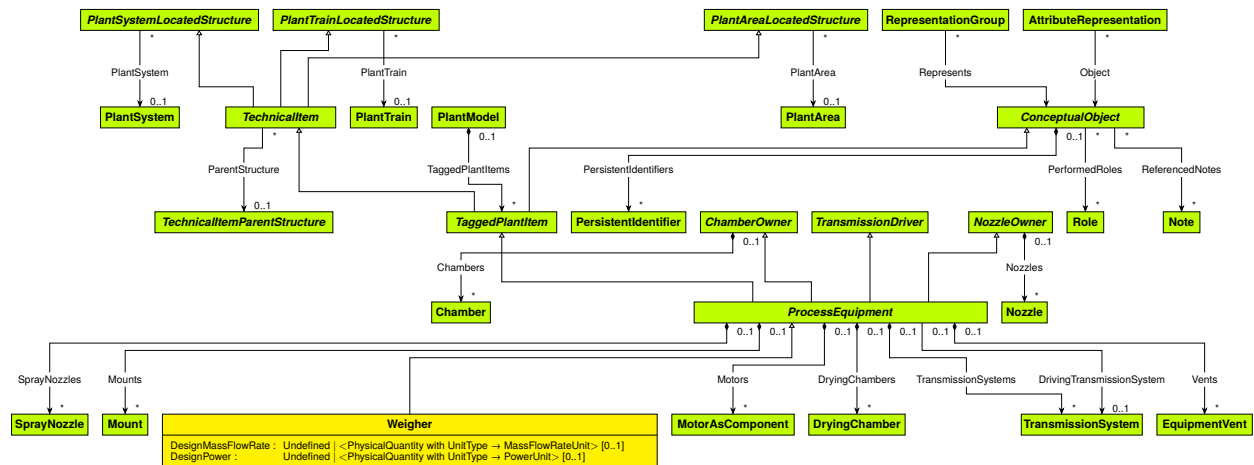
Overview

Concrete Class

RDL: <http://sandbox.dexpi.org/rdl/Weigher>

WEIGHER

A functional object that is capable of weighing.



Supertypes

- *ProcessEquipment*

Subtypes

- *BatchWeigher*
- *ContinuousWeigher*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignMassFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → MassFlowRateUnit></i>	no	no
<i>DesignPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no

DesignMassFlowRate

Data Property

The mass flow rate for which the *Weigher* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://data.posccaesar.org/rdl/RDS14286182>

DESIGN MASS FLOW RATE

Example

```
PhysicalQuantity(  
    Unit = KilogramPerSecond,  
    Value = 420.0)
```

DesignPower

Data Property

The power for which the *Weigher* is designed.

Type: *Undefined* | $\langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignPower>

DESIGN POWER

Example

```
PhysicalQuantity(  
    Unit = Kilowatt,  
    Value = 500.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.Weigher">  
  <Data property="DesignMassFlowRate">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerSecond"/>  
      </Data>  
      <Data property="Value">
```

(continues on next page)

(continued from previous page)

```
<Double>420.0</Double>
</Data>
</AggregatedDataValue>
</Data>
<Data property="DesignPower">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
    </Data>
    <Data property="Value">
      <Double>500.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>
```

WetCoolingTower

Plant::ProcessEquipment::WetCoolingTower

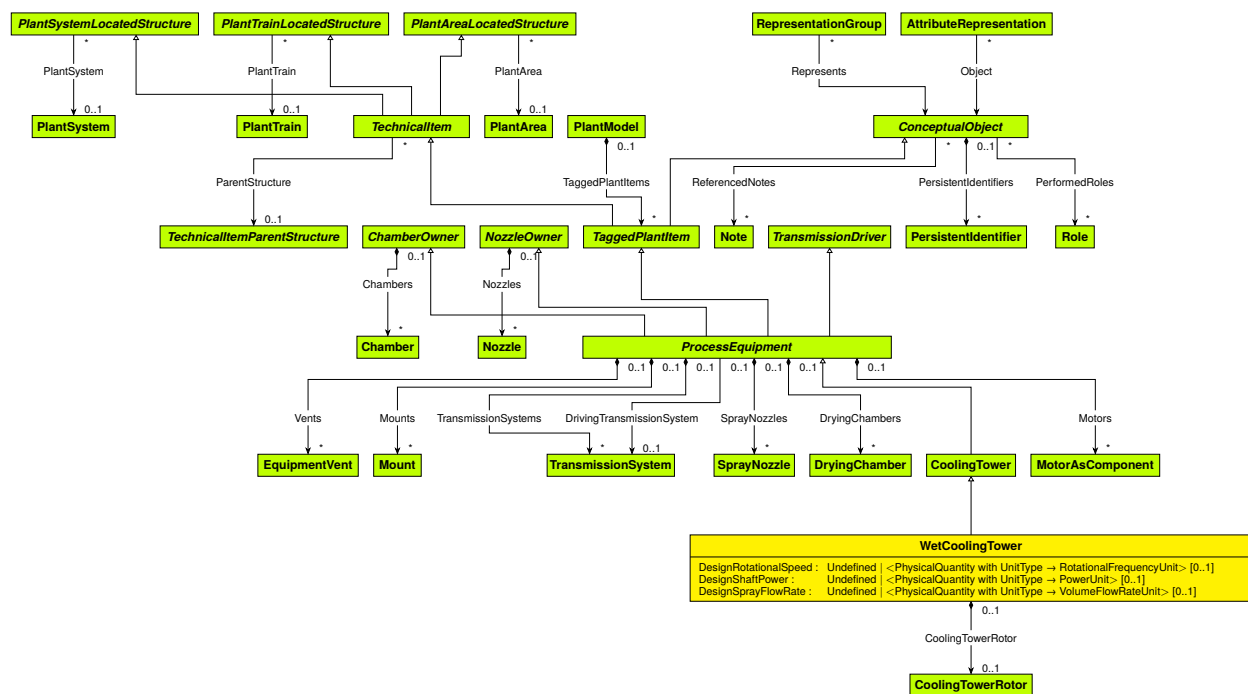
Overview

Concrete Class

RDL: <http://data.posccaesar.org/rdl/RDS14071846>

WET COOLING TOWER

A *Cooling Tower* that derives its primary cooling effect from the evaporation that takes place when air and water are brought into direct contact.



Supertypes

- *CoolingTower*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>CoolingTowerRotor</i>	0..1	<i>CoolingTowerRotor</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>DesignRotationalSpeed</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → Rotational-FrequencyUnit></i>	no	no
<i>DesignShaftPower</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → PowerUnit></i>	no	no
<i>Design-SprayFlowRate</i>	0..1	<i>Undefined</i> <i><PhysicalQuantity with UnitType → Volume-FlowRateUnit></i>	no	no

CoolingTowerRotor

Composition Property

The cooling tower rotor of the *WetCoolingTower*.

Type: *CoolingTowerRotor*

Multiplicity: 0..1

Ordered: no

DesignRotationalSpeed

Data Property

The rotational speed for which the *WetCoolingTower* is designed.

Type: *Undefined* | *<PhysicalQuantity with UnitType → RotationalFrequencyUnit>*

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignRotationalSpeed>

DESIGN ROTATIONAL SPEED

Example

```
PhysicalQuantity(
    Unit = ReciprocalMinute,
```

Value = 180.0)

DesignShaftPower

Data Property

The shaft power for which the *WetCoolingTower* is designed.

Type: *Undefined* | <*PhysicalQuantity* with *UnitType* → *PowerUnit*>

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignShaftPower>

DESIGN SHAFT POWER

Example

```
PhysicalQuantity(  
  Unit = Kilowatt,  
  Value = 400.0)
```

DesignSprayFlowRate

Data Property

The spray volume flow rate for the motive fluid for which the *WetCoolingTower* is designed.

Type: *Undefined* | <*PhysicalQuantity* with *UnitType* → *VolumeFlowRateUnit*>

Multiplicity: 0..1

Ordered: no

Unique: no

RDL: <http://sandbox.dexpi.org/rdl/DesignSprayFlowRate>

DESIGN SPRAY FLOW RATE

Example

```
PhysicalQuantity(  
  Unit = MetreCubedPerHour,  
  Value = 40.0)
```

Example in DEXPI XML

```
<Object type="Plant/ProcessEquipment.WetCoolingTower">  
  <Data property="DesignRotationalSpeed">  
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
      <Data property="Unit">  
        <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.  
↪ReciprocalMinute"/>  
      </DataReference>  
    </AggregatedDataValue>  
  </Data>  
</Object>
```

(continues on next page)

(continued from previous page)

```

    </Data>
    <Data property="Value">
      <Double>180.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="DesignShaftPower">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
    </Data>
    <Data property="Value">
      <Double>400.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
<Data property="DesignSprayFlowRate">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.MetreCubedPerHour
↪"/>
    </Data>
    <Data property="Value">
      <Double>40.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>

```

Plant::ProcessEquipment

Overview

Package

Members

Classes

Name	Type
<i>AccessNozzle</i>	Concrete Class
<i>Agglomerator</i>	Concrete Class
<i>Agitator</i>	Concrete Class
<i>AgitatorRotor</i>	Concrete Class
<i>AirCoolingSystem</i>	Concrete Class
<i>AirEjector</i>	Concrete Class
<i>AlternatingCurrentGenerator</i>	Concrete Class
<i>AlternatingCurrentMotor</i>	Concrete Class
<i>AlternatingCurrentMotorAsComponent</i>	Concrete Class
<i>AxialBlower</i>	Concrete Class
<i>AxialCompressor</i>	Concrete Class
<i>AxialFan</i>	Concrete Class

continues on next page

Table 9 – continued from previous page

Name	Type
<i>BatchWeigher</i>	Concrete Class
<i>Blower</i>	Concrete Class
<i>Boiler</i>	Concrete Class
<i>BriquettingRoller</i>	Concrete Class
<i>Burner</i>	Concrete Class
<i>CentrifugalBlower</i>	Concrete Class
<i>CentrifugalCompressor</i>	Concrete Class
<i>CentrifugalPump</i>	Concrete Class
<i>Centrifuge</i>	Concrete Class
<i>Chamber</i>	Concrete Class
<i>ChamberOwner</i>	Abstract Class
<i>Chimney</i>	Concrete Class
<i>ColumnInternalsArrangement</i>	Abstract Class
<i>ColumnPackingsArrangement</i>	Concrete Class
<i>ColumnSection</i>	Abstract Class
<i>ColumnTraysArrangement</i>	Concrete Class
<i>CombustionEngine</i>	Concrete Class
<i>CombustionEngineAsComponent</i>	Concrete Class
<i>Compressor</i>	Concrete Class
<i>ContinuousWeigher</i>	Concrete Class
<i>ConvectionDryer</i>	Concrete Class
<i>Conveyor</i>	Concrete Class
<i>CoolingTower</i>	Concrete Class
<i>CoolingTowerRotor</i>	Concrete Class
<i>Crusher</i>	Concrete Class
<i>CrusherElement</i>	Concrete Class
<i>DirectCurrentGenerator</i>	Concrete Class
<i>DirectCurrentMotor</i>	Concrete Class
<i>DirectCurrentMotorAsComponent</i>	Concrete Class
<i>Displacer</i>	Concrete Class
<i>DryCoolingTower</i>	Concrete Class
<i>Dryer</i>	Concrete Class
<i>DryingChamber</i>	Concrete Class
<i>EjectorPump</i>	Concrete Class
<i>ElectricGenerator</i>	Concrete Class
<i>ElectricHeater</i>	Concrete Class
<i>ElectricalSeparator</i>	Concrete Class
<i>EquipmentVent</i>	Concrete Class
<i>Extruder</i>	Concrete Class
<i>Fan</i>	Concrete Class
<i>Feeder</i>	Concrete Class
<i>Filter</i>	Concrete Class
<i>FilterUnit</i>	Concrete Class
<i>FilteringCentrifuge</i>	Concrete Class
<i>FilteringCentrifugeDrum</i>	Concrete Class
<i>Flare</i>	Concrete Class
<i>ForkliftTruck</i>	Concrete Class
<i>Furnace</i>	Concrete Class
<i>GasFilter</i>	Concrete Class
<i>GasTurbine</i>	Concrete Class

continues on next page

Table 9 – continued from previous page

Name	Type
<i>GearBox</i>	Concrete Class
<i>GravitationalSeparator</i>	Concrete Class
<i>Grinder</i>	Concrete Class
<i>GrindingElement</i>	Concrete Class
<i>HeatExchanger</i>	Concrete Class
<i>HeatExchangerRotor</i>	Concrete Class
<i>HeatedSurfaceDryer</i>	Concrete Class
<i>Heater</i>	Concrete Class
<i>Impeller</i>	Concrete Class
<i>InstrumentNozzle</i>	Concrete Class
<i>Kneader</i>	Concrete Class
<i>Lift</i>	Concrete Class
<i>LiquidFilter</i>	Concrete Class
<i>LoadingUnloadingSystem</i>	Concrete Class
<i>MechanicalSeparator</i>	Concrete Class
<i>Mill</i>	Concrete Class
<i>Mixer</i>	Concrete Class
<i>MixingElementAssembly</i>	Concrete Class
<i>MobileTransportSystem</i>	Concrete Class
<i>Motor</i>	Concrete Class
<i>MotorAsComponent</i>	Concrete Class
<i>Mount</i>	Concrete Class
<i>Nozzle</i>	Concrete Class
<i>NozzleOwner</i>	Abstract Class
<i>PackagingSystem</i>	Concrete Class
<i>PelletizerDisc</i>	Concrete Class
<i>PlateHeatExchanger</i>	Concrete Class
<i>PressureVessel</i>	Concrete Class
<i>ProcessColumn</i>	Concrete Class
<i>ProcessEquipment</i>	Abstract Class
<i>ProcessNozzle</i>	Concrete Class
<i>Pump</i>	Concrete Class
<i>RadialFan</i>	Concrete Class
<i>RailWaggon</i>	Concrete Class
<i>ReciprocatingCompressor</i>	Concrete Class
<i>ReciprocatingExtruder</i>	Concrete Class
<i>ReciprocatingPressureAgglomerator</i>	Concrete Class
<i>ReciprocatingPump</i>	Concrete Class
<i>RevolvingSieve</i>	Concrete Class
<i>RotaryCompressor</i>	Concrete Class
<i>RotaryMixer</i>	Concrete Class
<i>RotaryPump</i>	Concrete Class
<i>RotatingExtruder</i>	Concrete Class
<i>RotatingGrowthAgglomerator</i>	Concrete Class
<i>RotatingPressureAgglomerator</i>	Concrete Class
<i>Screw</i>	Concrete Class
<i>ScrubbingSeparator</i>	Concrete Class
<i>SedimentalCentrifuge</i>	Concrete Class
<i>SedimentalCentrifugeDrum</i>	Concrete Class
<i>Separator</i>	Concrete Class

continues on next page

Table 9 – continued from previous page

Name	Type
<i>Ship</i>	Concrete Class
<i>Sieve</i>	Concrete Class
<i>SieveElement</i>	Concrete Class
<i>Silo</i>	Concrete Class
<i>SpiralHeatExchanger</i>	Concrete Class
<i>SprayCooler</i>	Concrete Class
<i>SprayNozzle</i>	Concrete Class
<i>StaticMixer</i>	Concrete Class
<i>StationarySieve</i>	Concrete Class
<i>StationaryTransportSystem</i>	Concrete Class
<i>SteamGenerator</i>	Concrete Class
<i>SteamTurbine</i>	Concrete Class
<i>SubTaggedColumnSection</i>	Concrete Class
<i>TaggedColumnSection</i>	Concrete Class
<i>TaggedPlantItem</i>	Abstract Class
<i>Tank</i>	Concrete Class
<i>ThinFilmEvaporator</i>	Concrete Class
<i>TransmissionDriver</i>	Abstract Class
<i>TransmissionSystem</i>	Concrete Class
<i>TransportableContainer</i>	Concrete Class
<i>Truck</i>	Concrete Class
<i>TubeBundle</i>	Concrete Class
<i>TubularHeatExchanger</i>	Concrete Class
<i>Turbine</i>	Concrete Class
<i>Vent</i>	Abstract Class
<i>Vessel</i>	Concrete Class
<i>VibratingSieve</i>	Concrete Class
<i>WasteGasEmitter</i>	Concrete Class
<i>Weigher</i>	Concrete Class
<i>WetCoolingTower</i>	Concrete Class

Plant

3.2.8 Overview

Model

URI: <https://data.dexpi.org/models/2.0.0/Plant.xml>

[Plant.xml](#)

Types for the representation of engineering information and associated graphical representations.

3.2.9 Members

Packages

Name
<i>Diagram</i>
<i>Enumerations</i>
<i>Instrumentation</i>
<i>Piping</i>
<i>PlantStructure</i>
<i>ProcessEquipment</i>

Classes

Name	Type
<i>PlantModel</i>	Concrete Class

3.3 Process

3.3.1 Enumerations

CompositionBasis

Process::Enumerations::CompositionBasis

Overview

Enumeration

<<enumeration>> CompositionBasis
Mass Mole

Members

Literals

Name
Mass
Mole

CompositionDisplay

Process::Enumerations::CompositionDisplay

Overview

Enumeration

<<enumeration>> CompositionDisplay
AbsoluteValue Fraction Percent

Members

Literals

Name
AbsoluteValue
Fraction
Percent

CompressionMethod

*Process::Enumerations::*CompressionMethod

Overview

Enumeration

<<enumeration>> CompressionMethod
AxialMotion Blower CentrifugalMotion CustomMethod Ejector Fan ReciprocatingMotion RotaryMotion Unspecified

Members

Literals

Name
AxialMotion
Blower
CentrifugalMotion
CustomMethod
Ejector
Fan
ReciprocatingMotion
RotaryMotion
Unspecified

EngineDriveMethod

Process::Enumerations::EngineDriveMethod

Overview

Enumeration

<<enumeration>> EngineDriveMethod
Diesel GasTurbine OttoCycle Unspecified

Members

Literals

Name
Diesel
GasTurbine
OttoCycle
Unspecified

HeatExchangeMethod

Process::Enumerations::HeatExchangeMethod

Overview

Enumeration

<<enumeration>> HeatExchangeMethod
Generic Plate Spiral Tubular

Members

Literals

Name
Generic
Plate
Spiral
Tubular

InformationVariantType

*Process::Enumerations::*InformationVariantType

Overview

Enumeration

<<enumeration>> InformationVariantType
Boolean Double Integer

Members

Literals

Name
Boolean
Double
Integer

MeasuredQuantity

Process::Enumerations::MeasuredQuantity

Overview

Enumeration

<<enumeration>> MeasuredQuantity
AudioVisual
Density
ElectricCurrent
ElectricPotential
ElectromagneticField
Energy
Flow
Humidity
Level
MultipleQuantities
NumberOfEvents
Power
Pressure
PressureDifference
Quality
Radiation
SpatialDimension
Time
Velocity
VibrationOrTorque
WeightMassForce

Members

Literals

Name
AudioVisual
Density
ElectricCurrent
ElectricPotential
ElectromagneticField
Energy
Flow
Humidity
Level
MultipleQuantities
NumberOfEvents
Power
Pressure
PressureDifference
Quality
Radiation
SpatialDimension
Time
Velocity
VibrationOrTorque
WeightMassForce

MotorDriveMethod

*Process::Enumerations::*MotorDriveMethod

Overview

Enumeration

<<enumeration>> MotorDriveMethod
AlternatingCurrent DirectCurrent StepperMotor Unspecified

Members

Literals

Name
AlternatingCurrent
DirectCurrent
StepperMotor
Unspecified

PortDirection

Process::Enumerations::PortDirection

Overview

Enumeration

<<enumeration>> PortDirection
Inlet Outlet

Members

Literals

Name
Inlet
Outlet

ProcessStepHierarchyLevel

Process::Enumerations::ProcessStepHierarchyLevel

Overview

Enumeration

<<enumeration>> ProcessStepHierarchyLevel
ControlFunction ElementaryFunction Process ProcessSection ProcessTrain SafetyFunction SupportFunction UnitOperation

Members

Literals

Name
ControlFunction
ElementaryFunction
Process
ProcessSection
ProcessTrain
SafetyFunction
SupportFunction
UnitOperation

PumpingMethod

Process::Enumerations::PumpingMethod

Overview

Enumeration

<<enumeration>> PumpingMethod
CentrifugalMotion CustomMethod Eductor PositiveDisplacement RotaryMotion Unspecified

Members

Literals

Name
CentrifugalMotion
CustomMethod
Eductor
PositiveDisplacement
RotaryMotion
Unspecified

ReactionProcessType

*Process::Enumerations::*ReactionProcessType

Overview

Enumeration

<<enumeration>> ReactionProcessType
FluidizedBed PackedBed Tank Tubular Unspecified

Members

Literals

Name
FluidizedBed
PackedBed
Tank
Tubular
Unspecified

TrayRole

*Process::Enumerations::*TrayRole

Overview

Enumeration

<<enumeration>> TrayRole
Bottom Feed Monitored Top

Members

Literals

Name
Bottom
Feed
Monitored
Top

TurbineDriveMethod

*Process::Enumerations::*TurbineDriveMethod

Overview

Enumeration

<<enumeration>> TurbineDriveMethod
Expander Unspecified WaterTurbine WindTurbine

Members

Literals

Name
Expander
Unspecified
WaterTurbine
WindTurbine

Process::Enumerations

Overview

Package

The Enumerations package contains enumerations for various aspects of process information.

Members

Data Types

Name	Type
<i>CompositionBasis</i>	Enumeration
<i>CompositionDisplay</i>	Enumeration
<i>CompressionMethod</i>	Enumeration
<i>EngineDriveMethod</i>	Enumeration
<i>HeatExchangeMethod</i>	Enumeration
<i>InformationVariantType</i>	Enumeration
<i>MeasuredQuantity</i>	Enumeration
<i>MotorDriveMethod</i>	Enumeration
<i>PortDirection</i>	Enumeration
<i>ProcessStepHierarchyLevel</i>	Enumeration
<i>PumpingMethod</i>	Enumeration
<i>ReactionProcessType</i>	Enumeration
<i>TrayRole</i>	Enumeration
<i>TurbineDriveMethod</i>	Enumeration

3.3.2 Process

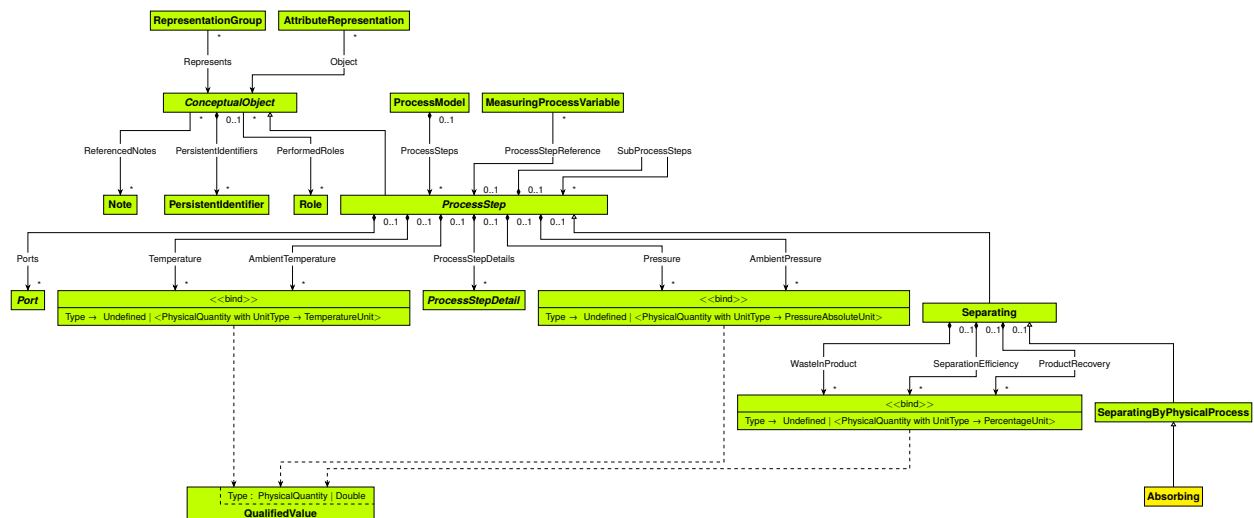
Absorbing

Process::Process::Absorbing

Overview

Concrete Class

A subtype of *SeparatingByPhysicalProcess* where separation is based on mass transfer of material from a gas to a liquid. This type is also used to represent desorbing, where the mass transfer is from the liquid to the gas.



Supertypes

- *SeparatingByPhysicalProcess*

Example in DEXPI XML

```
<Object type="Process/Process.Adsorbing"/>
```

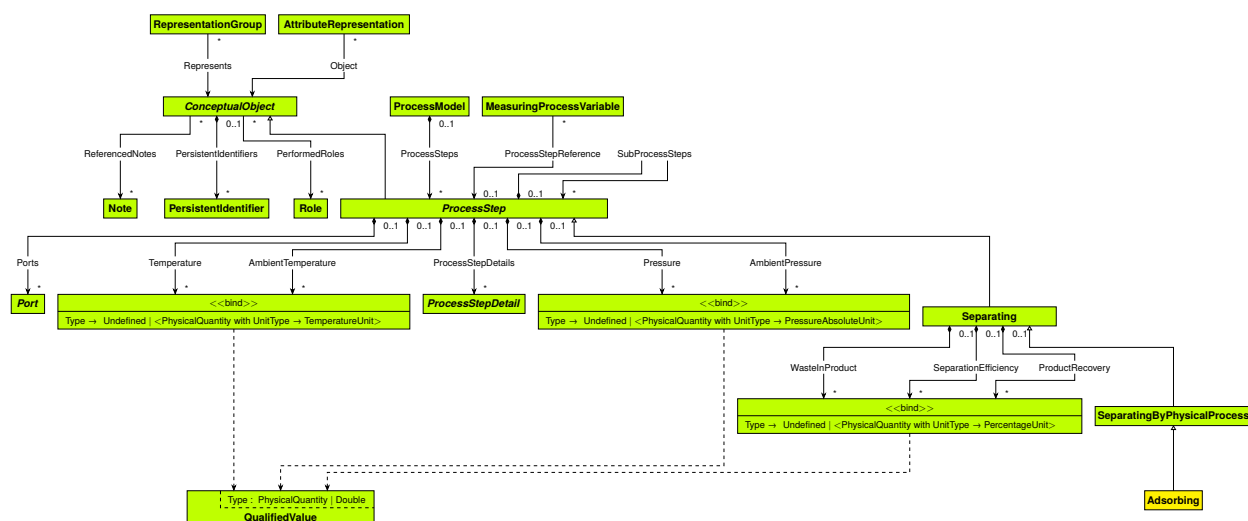
Adsorbing

Process::Process::Adsorbing

Overview

Concrete Class

A subtype of *SeparatingByPhysicalProcess* where separation is based on mass transfer of material from a fluid to the surface of a solid. This type is also used to represent the reverse process, where the mass transfer is from the surface to the fluid.



Supertypes

- *SeparatingByPhysicalProcess*

Example in DEXPI XML

```
<Object type="Process/Process.Adsorbing"/>
```

Agglomerating

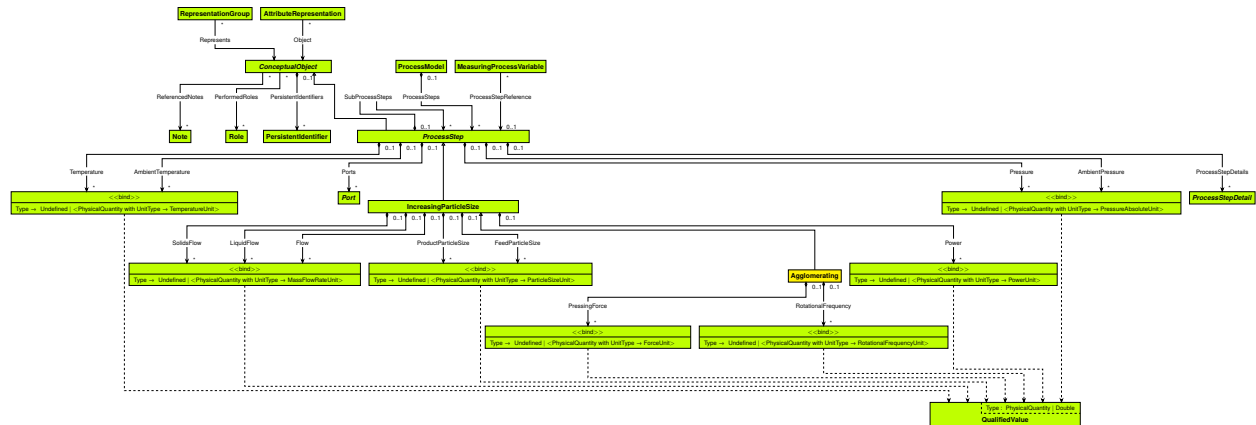
Process::Process::Agglomerating

Overview

Concrete Class

A solids-processing process for *IncreasingParticleSize* in which particles bind together and coalesce into larger particles, held together by relatively weak forces.

See also *Flocculating* and *Pelletizing*. Note that Pelletizing is a subtype of *FormingSolidMaterial*.



Supertypes

- *IncreasingParticleSize*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>PressingForce</i>	*	< <i>QualifiedValue</i> with <i>Type</i> $\rightarrow$ <i>Undefined</i> < <i>PhysicalQuantity</i> with <i>UnitType</i> $\rightarrow$ <i>ForceUnit</i> >>	no
<i>RotationalFrequency</i>	*	< <i>QualifiedValue</i> with <i>Type</i> $\rightarrow$ <i>Undefined</i> < <i>PhysicalQuantity</i> with <i>UnitType</i> $\rightarrow$ <i>RotationalFrequencyUnit</i> >>	no

PressingForce

Composition Property

The pressing force required in the process.

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{ForceUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
    QualifiedValue(
        Value = PhysicalQuantity(
            Unit = Newton,
            Value = 10000.0))
]
```

RotationalFrequency

Composition Property

The rotational frequency of the process.

Type: $\langle \text{QualifiedValue with } Type \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{RotationalFrequencyUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = ReciprocalMinute,
      Value = 120.0))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.Agglomerating">
  <Components property="PressingForce">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.ForceUnit.Newton"/>
          </Data>
          <Data property="Value">
            <Double>10000.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="RotationalFrequency">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ ReciprocalMinute"/>
          </Data>
          <Data property="Value">
            <Double>120.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
```

(continues on next page)

(continued from previous page)

```

</Components>
</Object>

```

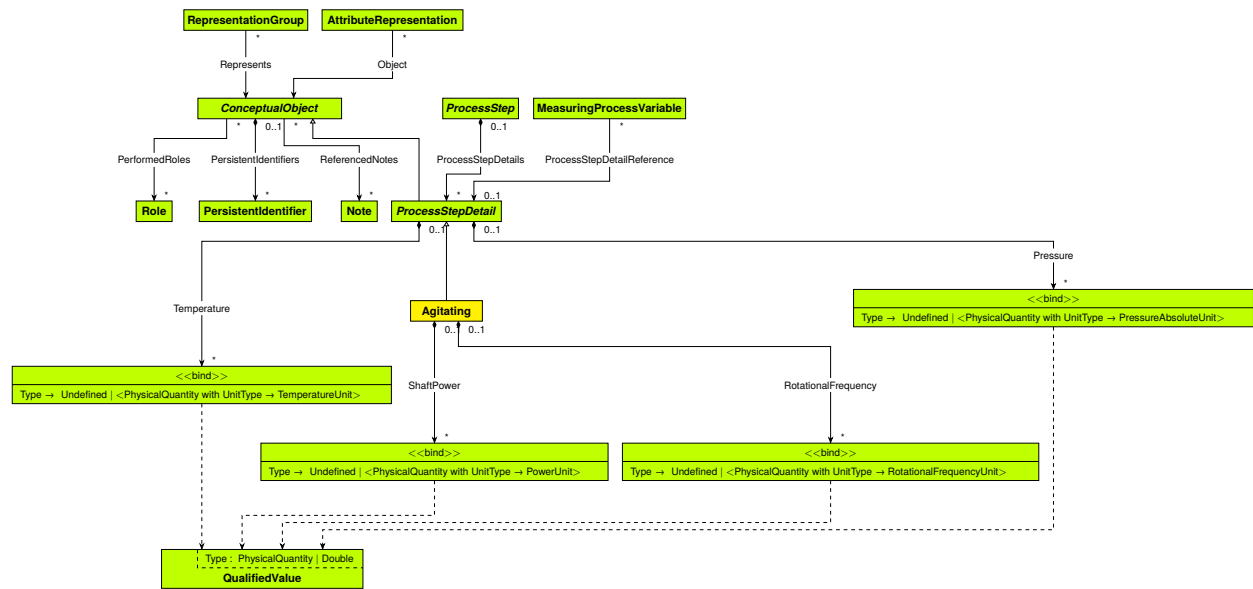
Agitating

Process::Process::Agitating

Overview

Concrete Class

This *ProcessStepDetail* provides shear forces to liquids in another process, with the purpose of mixing, increasing heat transfer rates or increasing mass transfer rates.



Supertypes

- *ProcessStepDetail*

Members

Composition Properties

Name	Multi- plicity	Type	Or- dered
<i>RotationalFrequency</i>	*	<i><QualifiedValue with Type -> Undefined <PhysicalQuantity with UnitType -> RotationalFrequencyUnit>></i>	no
<i>ShaftPower</i>	*	<i><QualifiedValue with Type -> Undefined <PhysicalQuantity with UnitType -> PowerUnit>></i>	no

RotationalFrequency

Composition Property

The rotational frequency of the *Agitating*.

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{RotationalFrequencyUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = ReciprocalMinute,
      Value = 120.0))
]
```

ShaftPower

Composition Property

The shaft power demand of the *Agitating*.

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{PowerUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Kilowatt,
      Value = 80.0))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.Agitating">
  <Components property="RotationalFrequency">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ReciprocalMinute"/>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```

(continues on next page)

(continued from previous page)

```

    </Data>
    <Data property="Value">
      <Double>120.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>
</Components>
<Components property="ShaftPower">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
        </Data>
        <Data property="Value">
          <Double>80.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
</Object>

```

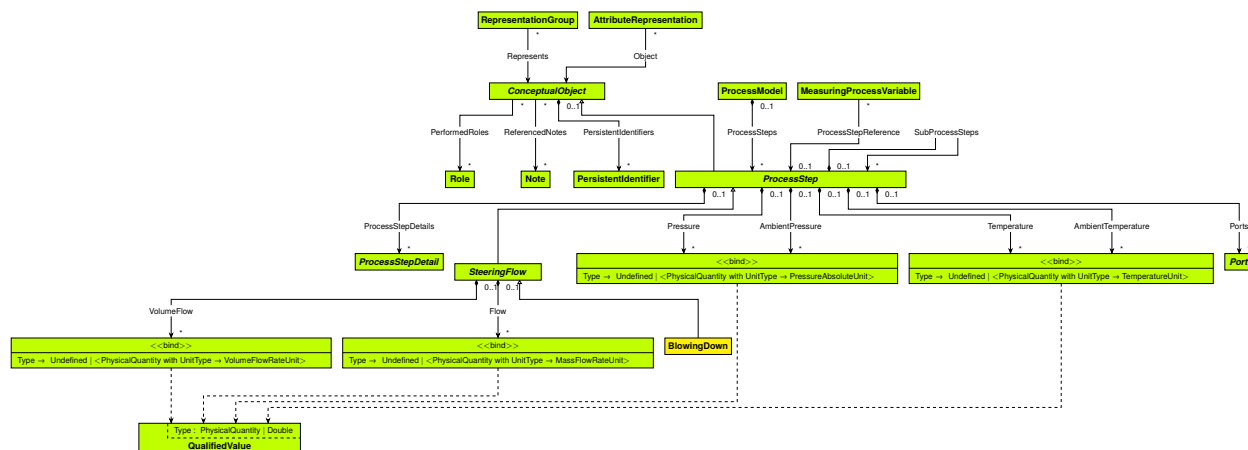
BlowingDown

Process::Process::BlowingDown

Overview

Concrete Class

This *SteeringFlow* process is used to depressurise or blow down a process, removing the contents of a process rapidly and safely to prevent unsafe conditions or shut down the process. This process is usually realized as a set of blow-down or depressurisation valves.



Supertypes

- *SteeringFlow*

Example in DEXPI XML

```
<Object type="Process/Process.BlowingDown"/>
```

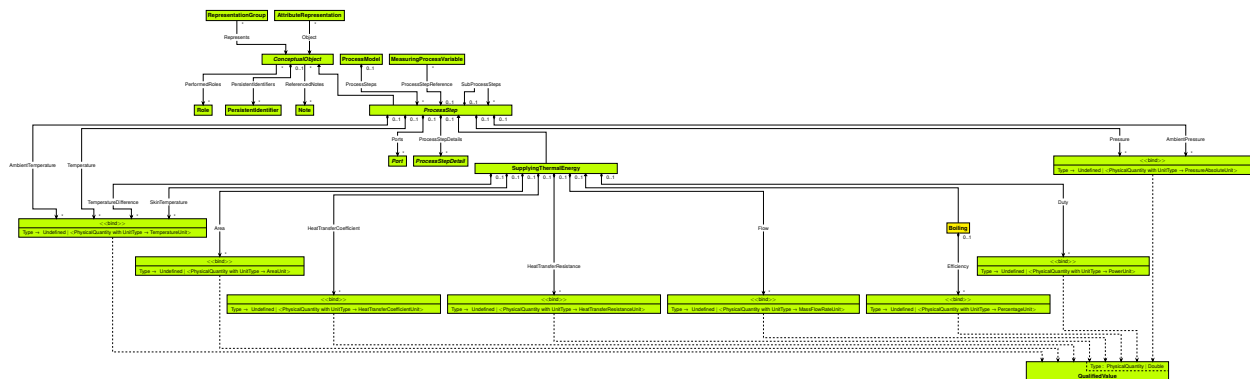
Boiling

Process::Process::Boiling

Overview

Concrete Class

This *SupplyingThermalEnergy* process vaporises a fluid. This process can be realized as a boiler, heat exchanger or fired heater. The process can also be used to represent the re-boiling subprocess in a distillation process.



Supertypes

- *SupplyingThermalEnergy*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Efficiency</i>	*	$\langle \textit{QualifiedValue} \text{ with } \textit{Type} \rightarrow \textit{Undefined} \mid \langle \textit{PhysicalQuantity} \text{ with } \textit{UnitType} \rightarrow \textit{PercentageUnit} \rangle \rangle$	no

Efficiency

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PercentageUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
    QualifiedValue(
        Value = PhysicalQuantity(
            Unit = Percent,
            Value = 78.0))
]
```

Example in DEXPI XML

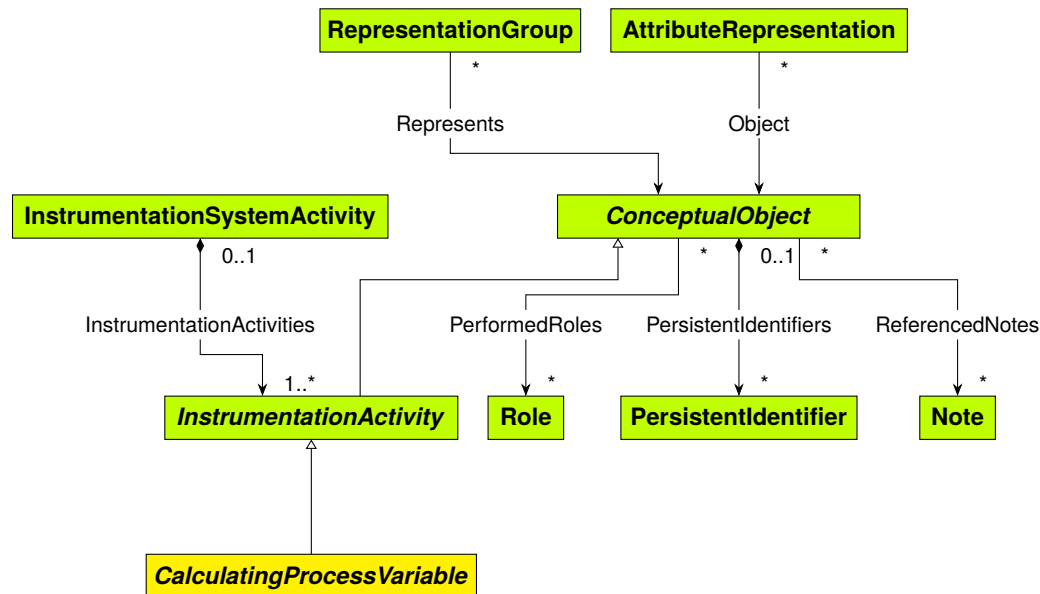
```
<Object type="Process/Process.Boiling">
  <Components property="Efficiency">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.PercentageUnit.Percent"/>
          </Data>
          <Data property="Value">
            <Double>78.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```

CalculatingProcessVariable

Process::Process::CalculatingProcessVariable

Overview**Abstract Class**

This *InstrumentationActivity* calculates an output that is some mathematical function of the inputs. This is a parent type for specialized calculations with specific purposes.



Supertypes

- *InstrumentationActivity*

Subtypes

- *CalculatingRatio*
- *CalculatingSplitRange*
- *TransformingProcessVariable*

CalculatingRatio

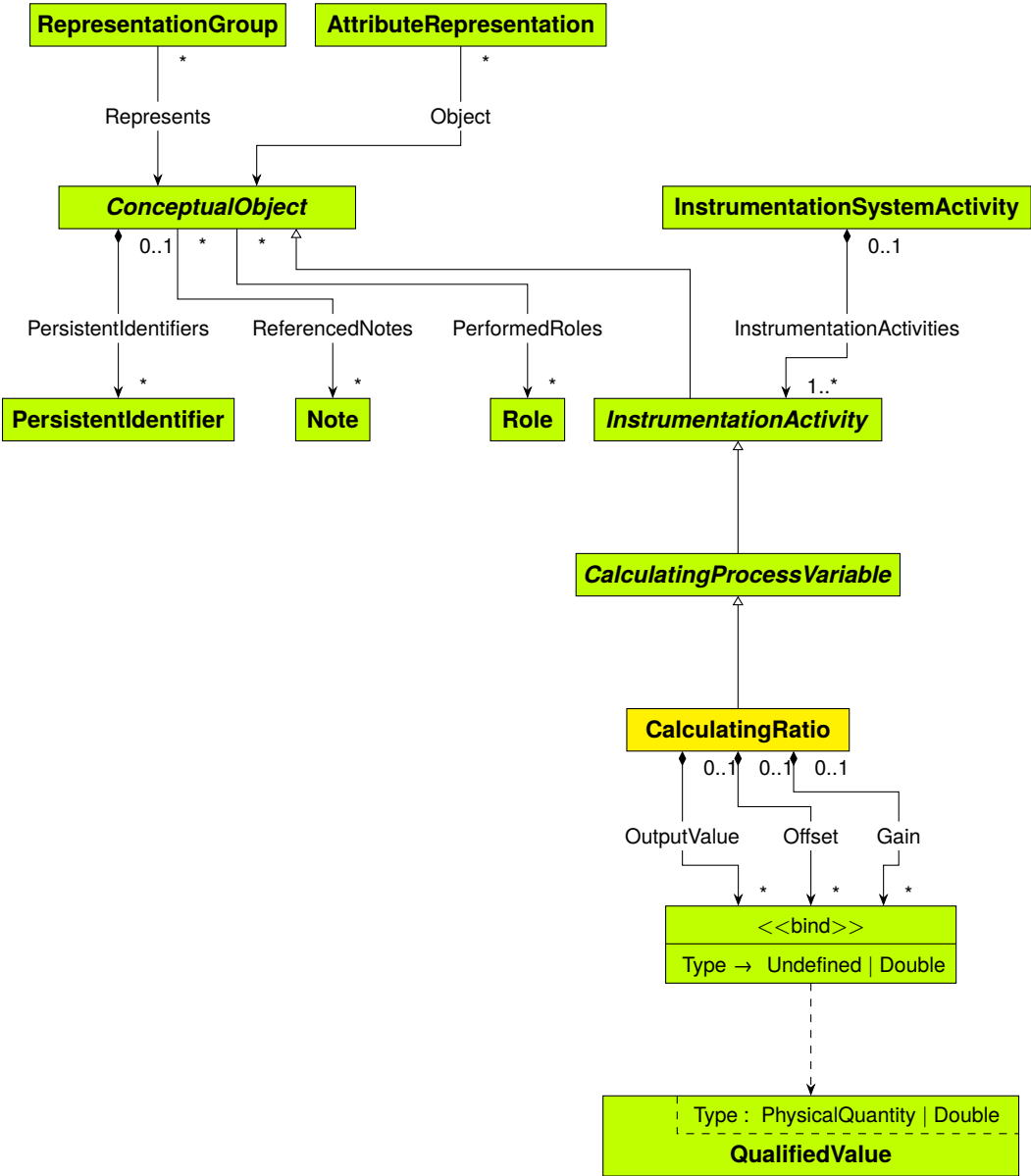
Process::Process::CalculatingRatio

Overview

Concrete Class

This *CalculatingProcessVariable* process calculates an output signal that is the ratio of the two input signals. It can be used to specify and design ratio control functions.

If the two input signals are Dividend and Divisor, then $\text{OutputValue} = \text{Gain} * \text{Dividend} / \text{Divisor} + \text{Offset}$.



Supertypes

- *CalculatingProcessVariable*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Gain</i>	*	<QualifiedValue with Type → Undefined Double>	no
<i>Offset</i>	*	<QualifiedValue with Type → Undefined Double>	no
<i>OutputValue</i>	*	<QualifiedValue with Type → Undefined Double>	no

Gain**Composition Property**

Multiplier in the ratio calculation.

Type: $\langle \textit{QualifiedValue}$ with $\textit{Type} \rightarrow \textit{Undefined} \mid \textit{Double} \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
    QualifiedValue(  
        Value = 2.5)  
]
```

Offset**Composition Property**

Offset added to the ratio calculation.

Type: $\langle \textit{QualifiedValue}$ with $\textit{Type} \rightarrow \textit{Undefined} \mid \textit{Double} \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
    QualifiedValue(  
        Value = 3.0)  
]
```

OutputValue**Composition Property**

The calculated ratio.

Type: $\langle \textit{QualifiedValue}$ with $\textit{Type} \rightarrow \textit{Undefined} \mid \textit{Double} \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
    QualifiedValue(  
        Value = 4.3)  
]
```


Example in DEXPI XML

```
<Object type="Process/Process.CalculatingRatio">
  <Components property="Gain">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <Double>2.5</Double>
      </Data>
    </Object>
  </Components>
  <Components property="Offset">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <Double>3.0</Double>
      </Data>
    </Object>
  </Components>
  <Components property="OutputValue">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <Double>4.3</Double>
      </Data>
    </Object>
  </Components>
</Object>
```

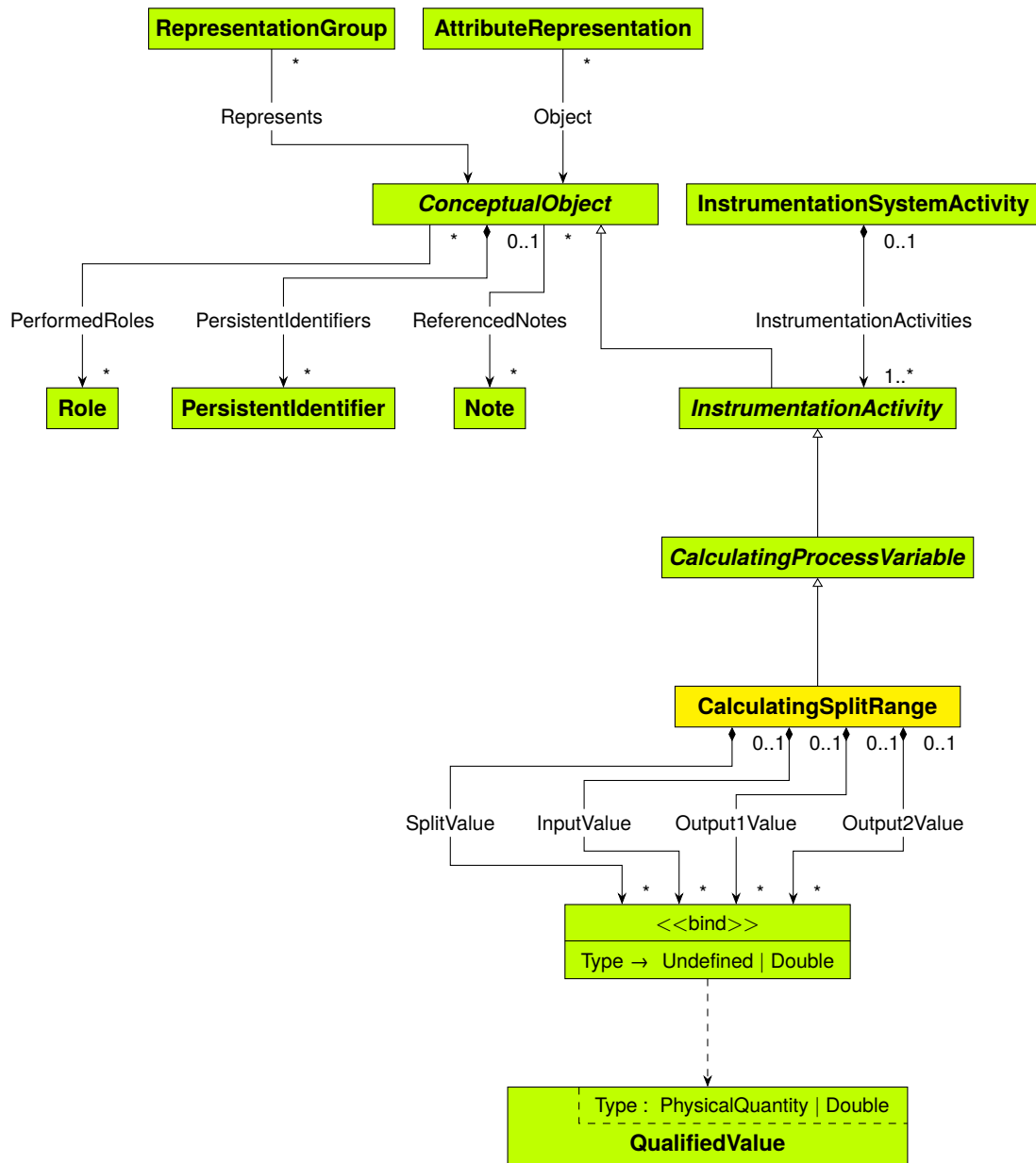
CalculatingSplitRange

Process::Process::CalculatingSplitRange

Overview

Concrete Class

This *CalculatingProcessVariable* process calculates two output signals so that a single input signal can be used to manipulate two control elements using a split range. It can be used to specify and design split range control functions.



Supertypes

- *CalculatingProcessVariable*

Members

Composition Properties

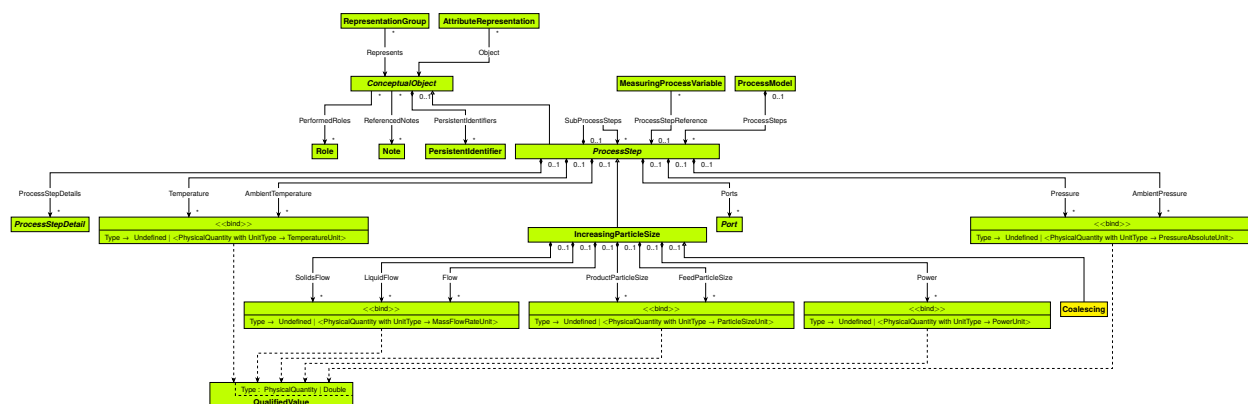
Name	Multiplicity	Type	Ordered
<i>InputValue</i>	*	<QualifiedValue with Type → Undefined Double>	no
<i>Output1Value</i>	*	<QualifiedValue with Type → Undefined Double>	no
<i>Output2Value</i>	*	<QualifiedValue with Type → Undefined Double>	no
<i>SplitValue</i>	*	<QualifiedValue with Type → Undefined Double>	no

InputValue**Composition Property****Type:** $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid Double \rangle$ **Multiplicity:** ***Ordered:** no**Output1Value****Composition Property****Type:** $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid Double \rangle$ **Multiplicity:** ***Ordered:** no**Output2Value****Composition Property****Type:** $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid Double \rangle$ **Multiplicity:** ***Ordered:** no**SplitValue****Composition Property****Type:** $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid Double \rangle$ **Multiplicity:** ***Ordered:** no**Example in DEXPI XML**

```
<Object type="Process/Process.CalculatingSplitRange"/>
```

Coalescing*Process::Process::Coalescing***Overview****Concrete Class**

This is an *IncreasingParticleSize* process in which small droplets of a liquid phase are made to coalesce into larger droplets, which can then be separated from the mixture.



Supertypes

- *IncreasingParticleSize*

Example in DEXPI XML

```
<Object type="Process/Process.Coalescing"/>
```

Composition

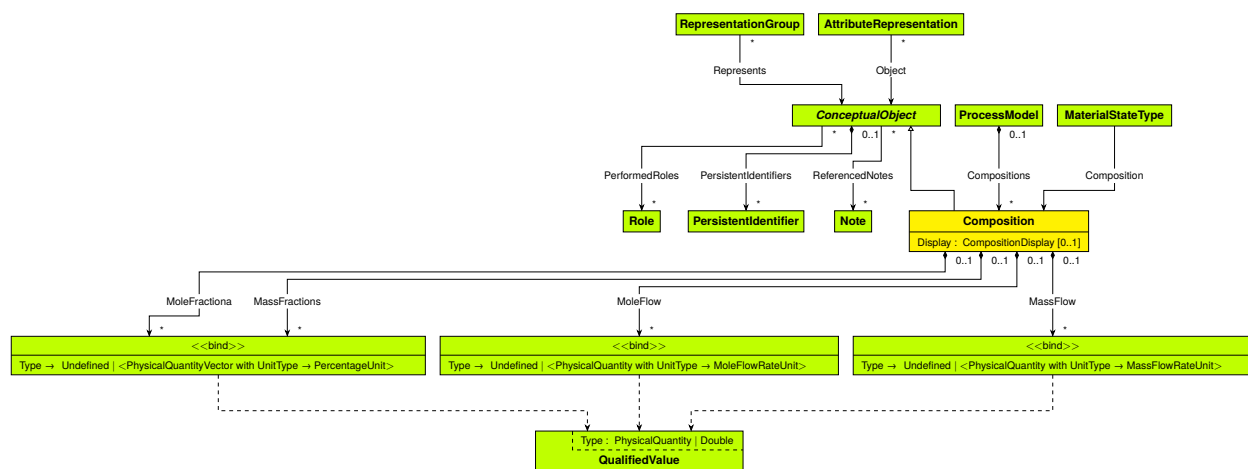
Process::Process::Composition

Overview

Concrete Class

This data item is an array of numbers that describe the chemical composition of a material. The order of items in the array is determined by the list of chemical components (*ListOfMaterialComponents*) in the corresponding *Material-Template* object (see *MaterialTemplate.ListOfComponents*).

Note: ISO 15926 uses the concept of CONCENTRATION and does not model the concept of material composition as an attribute.



Supertypes

- *ConceptualObject*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>MassFlow</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MassFlowRateUnit>></i>	no
<i>MassFractions</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantityVector with UnitType → PercentageUnit>></i>	no
<i>MoleFlow</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MoleFlowRateUnit>></i>	no
<i>MoleFractions</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantityVector with UnitType → PercentageUnit>></i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Display</i>	0..1	<i>CompositionDisplay</i>	no	no

Display

Data Property

The display form for the composition: Fraction, Percent, or AbsoluteValue (mass or molar flow).

Type: *CompositionDisplay*

Multiplicity: 0..1

Ordered: no

Unique: no

MassFlow

Composition Property

Vector of mass flow rates for each component.

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → MassFlowRateUnit>>*

Multiplicity: *

Ordered: no

Example

[

QualifiedValue(

```
Value = PhysicalQuantity(  
  Unit = KilogramPerHour,  
  Value = 270.0))  
]
```

MassFractions

Composition Property

Vector of mass fraction values for each component.

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantityVector with UnitType} \rightarrow \text{PercentageUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Percent,  
      Value = mimi [19.0, 81.0]))  
]
```

MoleFlow

Composition Property

Vector of mole flow rates for each component.

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MoleFlowRateUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = KilomolePerSecond,  
      Value = 8.5))  
]
```

MoleFractiona

Composition Property

Vector of mole fraction values for each component.

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantityVector with UnitType → PercentageUnit>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantityVector(
      Unit = Percent,
      Values = [
        17.0,
        83.0]))
]
```

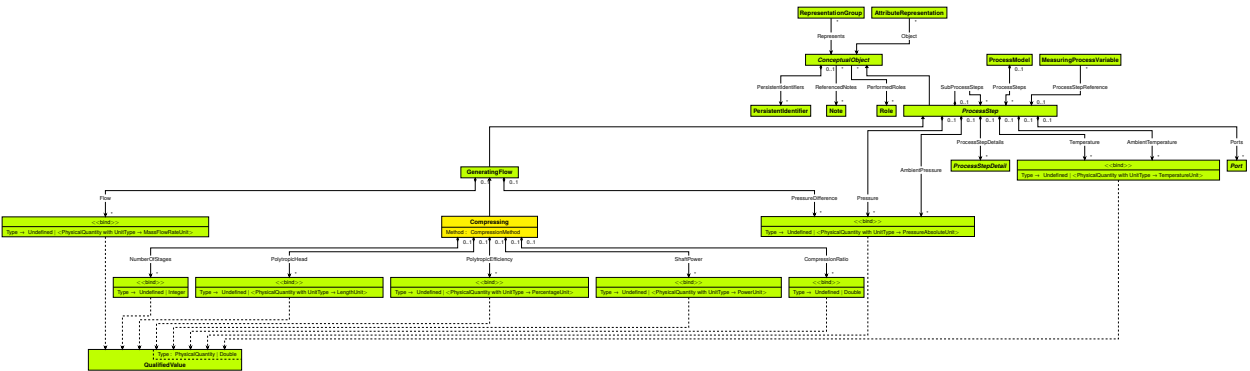
Compressing

Process::Process::Compressing

Overview

Concrete Class

This *GeneratingFlow* process represents compression of gases to increase pressure and generate a flow of gas. The physical principle used to compress the gas can be specified by selecting a value from the CompressionMethod enumeration.



Supertypes

- *GeneratingFlow*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>CompressionRatio</i>	*	<QualifiedValue with Type → Undefined Double>	no
<i>NumberOfStages</i>	*	<QualifiedValue with Type → Undefined Integer>	no
<i>PolytropicEfficiency</i>	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PercentageUnit>>	no
<i>PolytropicHead</i>	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → LengthUnit>>	no
<i>ShaftPower</i>	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PowerUnit>>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Method</i>	1	<i>CompressionMethod</i>	no	no

CompressionRatio

Composition Property

Ratio between outlet and inlet absolute pressure.

Type: <QualifiedValue with Type → Undefined | Double>

Multiplicity: *

Ordered: no

Method

Data Property

Compression method.

Type: *CompressionMethod*

Multiplicity: 1

Ordered: no

Unique: no

NumberOfStages

Composition Property

Number of compressor stages of the *Compressing*.

Type: <QualifiedValue with Type → Undefined | Integer>

Multiplicity: ***Ordered:** no**PolytropicEfficiency****Composition Property**

The polytropic efficiency of the *Compressing*.

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → PercentageUnit>>

Multiplicity: ***Ordered:** no**Example**

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Percent,
      Value = 82.0))
]
```

PolytropicHead**Composition Property**

The polytropic head generated by the *Compressing*.

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → LengthUnit>>

Multiplicity: ***Ordered:** no**Example**

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Metre,
      Value = 1800.0))
]
```

ShaftPower**Composition Property**

The mechanical power consumption of the *Compressing*.

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → PowerUnit>>

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Kilowatt,  
      Value = 80.0))  
]
```

Example in DEXPI XML

```
<Object type="Process/Process.Compressing">  
  <Components property="PolytropicEfficiency">  
    <Object type="Core/QualifiedValue">  
      <Data property="Value">  
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
          <Data property="Unit">  
            <DataReference data="Core/PhysicalQuantities.PercentageUnit.Percent"/>  
          </Data>  
          <Data property="Value">  
            <Double>82.0</Double>  
          </Data>  
        </AggregatedDataValue>  
      </Data>  
    </Object>  
  </Components>  
  <Components property="PolytropicHead">  
    <Object type="Core/QualifiedValue">  
      <Data property="Value">  
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
          <Data property="Unit">  
            <DataReference data="Core/PhysicalQuantities.LengthUnit.Metre"/>  
          </Data>  
          <Data property="Value">  
            <Double>1800.0</Double>  
          </Data>  
        </AggregatedDataValue>  
      </Data>  
    </Object>  
  </Components>  
  <Components property="ShaftPower">  
    <Object type="Core/QualifiedValue">  
      <Data property="Value">  
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
          <Data property="Unit">  
            <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>  
          </Data>  
          <Data property="Value">
```

(continues on next page)

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Height</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle \rangle$	no
<i>NumberOfTheoreticalStages</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \text{Integer} \rangle$	no

Height

Composition Property

The height of the packing.

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Centimetre,
      Value = 12.4))
]
```

NumberOfTheoreticalStages

Composition Property

The number of theoretical stages realised in the packing.

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \text{Integer} \rangle$

Multiplicity: *

Ordered: no

Example in DEXPI XML

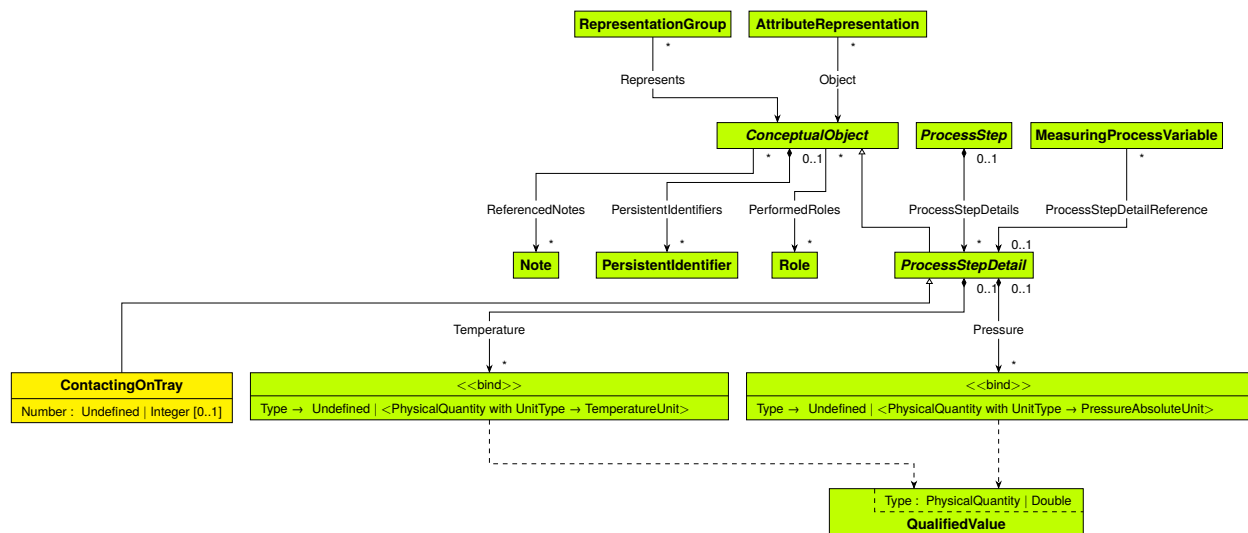
```
<Object type="Process/Process.ContactingInPacking">
  <Components property="Height">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```

(continues on next page)

```
</Data>
<Data property="Value">
  <Double>12.4</Double>
</Data>
</AggregatedDataValue>
</Data>
</Object>
</Components>
</Object>
```

This *ProcessStepDetail* is used to represent a process carried out in an assembly of trays. Its role is to represent trays shown in process flow diagrams and to support the functional breakdown of complex processing systems. See also *ContactingInPacking*.

- Bottom
- Feed
- Monitored
- Top



- *ProcessStepDetail*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Number</i>	0..1	<i>Undefined</i> <i>Integer</i>	no	no

Number

Data Property

Type: *Undefined* | *Integer*

Multiplicity: 0..1

Ordered: no

Unique: no

Example in DEXPI XML

```
<Object type="Process/Process.ContactingOnTray"/>
```

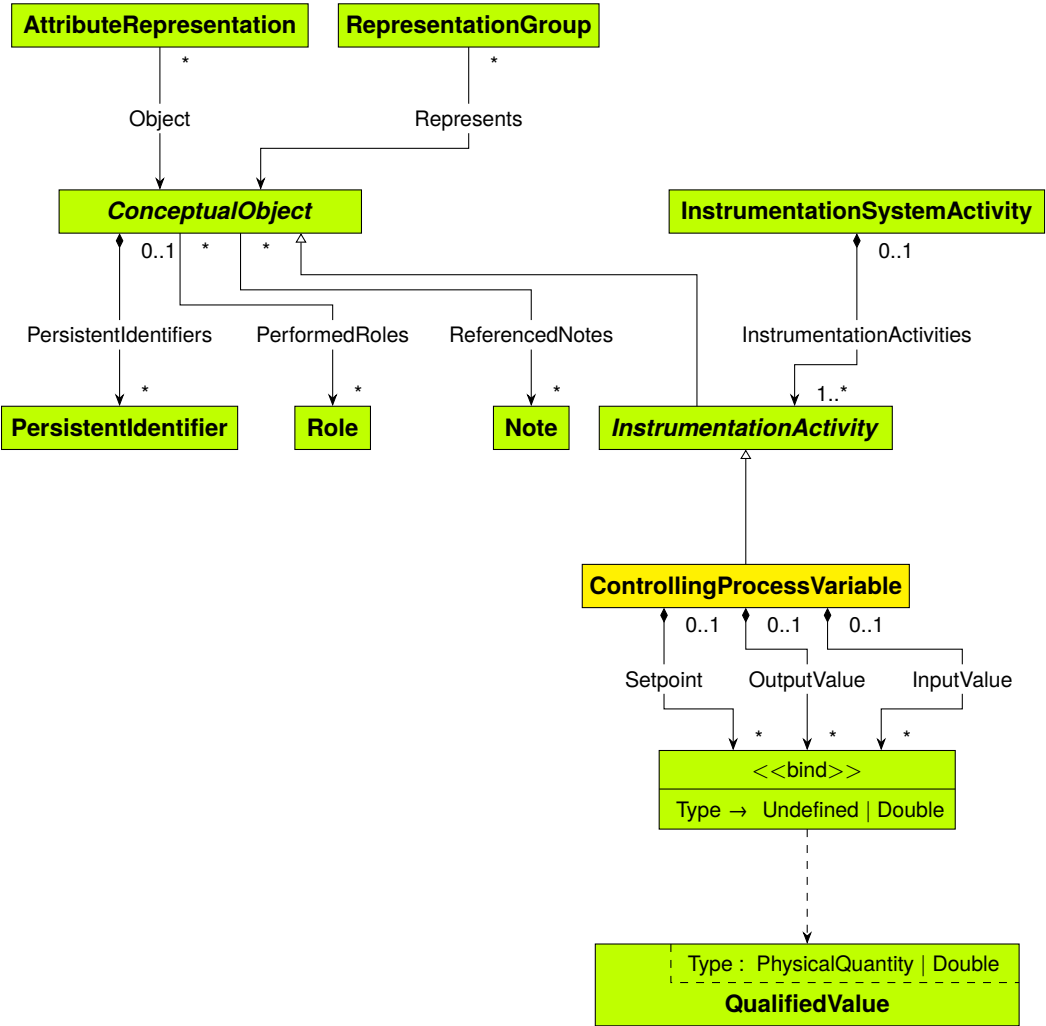
ControllingProcessVariable

Process::Process::ControllingProcessVariable

Overview

Concrete Class

This *InstrumentationActivity* represents a process control activity. A block of this type calculates the value of a manipulated variable so that a measured variable is held at a specified set point. The set point can be supplied as an external *InformationFlow*.



Supertypes

- *InstrumentationActivity*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>InputValue</i>	*	<QualifiedValue with Type → Undefined Double>	no
<i>OutputValue</i>	*	<QualifiedValue with Type → Undefined Double>	no
<i>Setpoint</i>	*	<QualifiedValue with Type → Undefined Double>	no

InputValue

Composition Property

Type: <QualifiedValue with Type → Undefined | Double>

Multiplicity: *

Ordered: no

OutputValue

Composition Property

Type: $\langle \text{QualifiedValue with } Type \rightarrow \text{Undefined} \mid \text{Double} \rangle$

Multiplicity: *

Ordered: no

Setpoint

Composition Property

Type: $\langle \text{QualifiedValue with } Type \rightarrow \text{Undefined} \mid \text{Double} \rangle$

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Process/Process.ControllingProcessVariable"/>
```

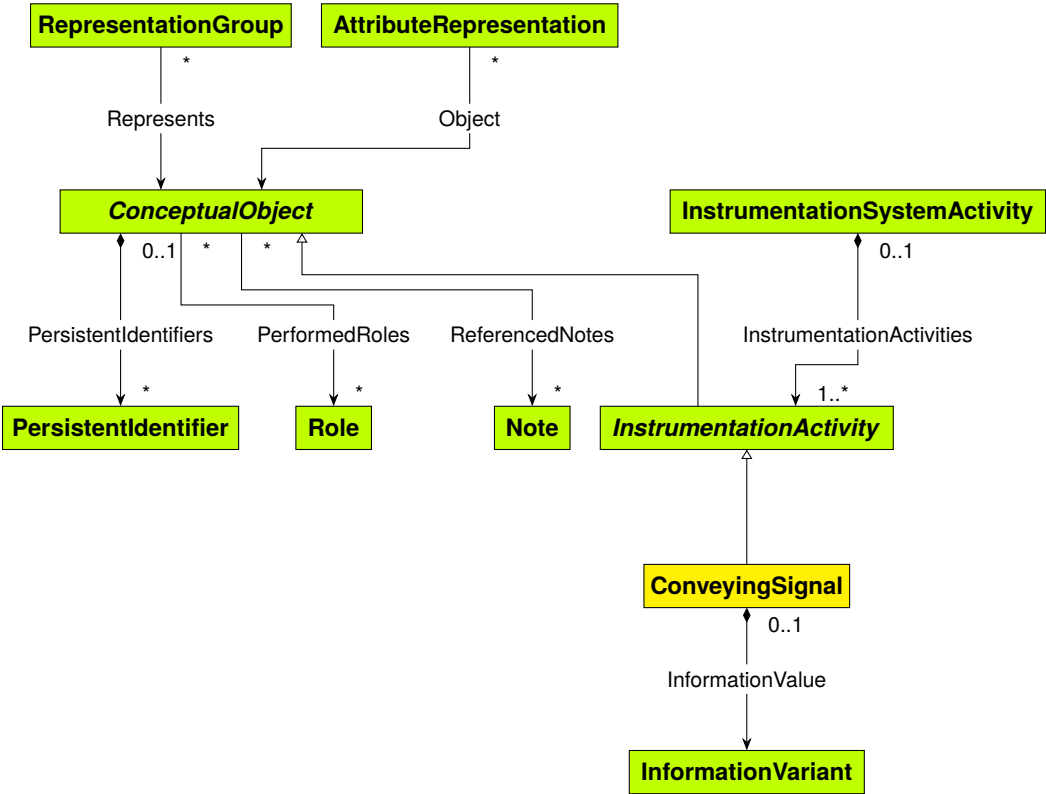
ConveyingSignal

Process::Process::ConveyingSignal

Overview

Concrete Class

This process transmits a signal between two processes. In practices, this process is only used if there will be a need to invest in specialized equipment and facilities, or if the properties of the signal are different at the input and output of the process. Most signal connections are represented by *InformationFlow* objects between *InformationPort* instances.



Supertypes

- *InstrumentationActivity*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>InformationValue</i>	1	<i>InformationVariant</i>	no

InformationValue

Composition Property

Type: *InformationVariant*

Multiplicity: 1

Ordered: no

Example in DEXPI XML

```
<Object type="Process/Process.ConveyingSignal"/>
```

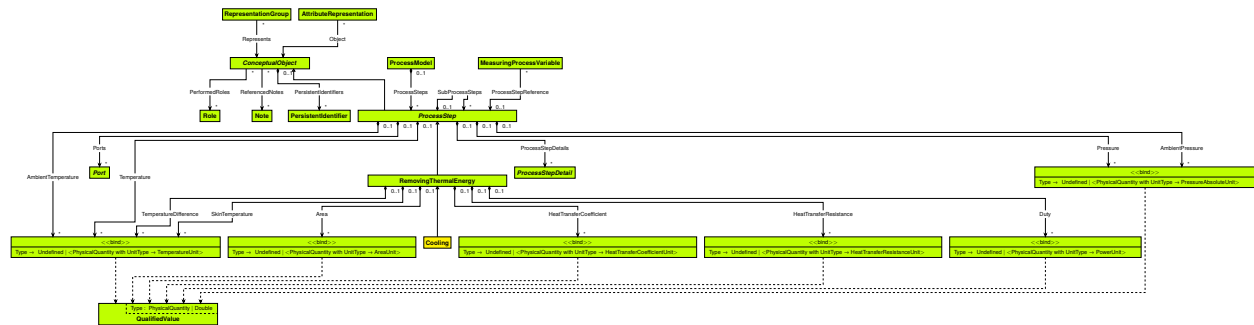
Cooling

Process::Process::Cooling

Overview

Concrete Class

This *RemovingThermalEnergy* process has the purpose of reducing the enthalpy content of a material through heat exchange with the environment or a cooling medium. See also *ExchangingThermalEnergy*, which can be used to model the same process in cases where the cooling medium system is defined.



Supertypes

- *RemovingThermalEnergy*

Example in DEXPI XML

```
<Object type="Process/Process.Cooling"/>
```

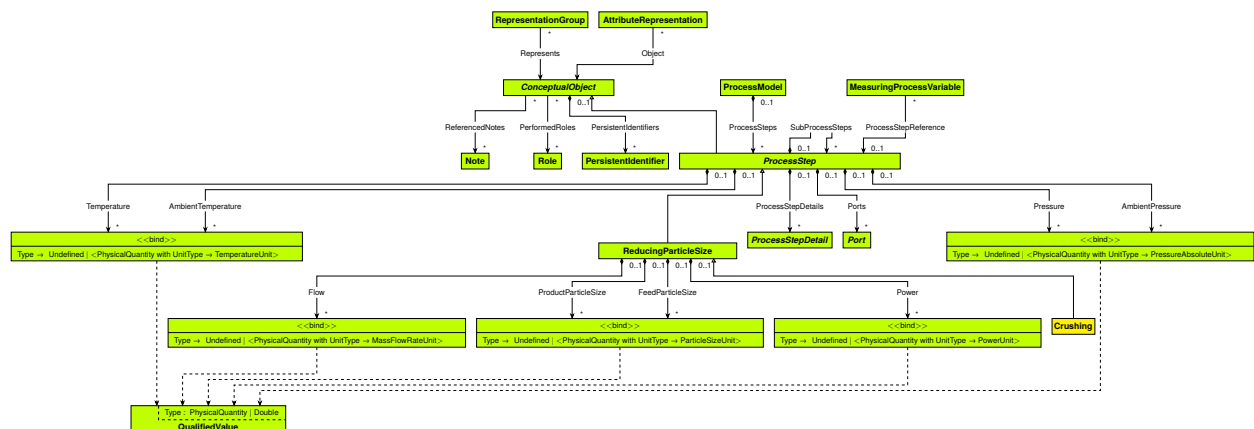
Crushing

Process::Process::Crushing

Overview

Concrete Class

This *ReducingParticleSize* process takes solid material and breaks it into smaller pieces by an externally applied mechanical force. See also *Grinding* and *Milling*.



Duty

Composition Property

Type: $\langle \text{QualifiedValue with } Type \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{PowerUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Kilowatt,  
      Value = 4.6))  
]
```

Example in DEXPI XML

```
<Object type="Process/Process.Crystallizing">  
  <Components property="Duty">  
    <Object type="Core/QualifiedValue">  
      <Data property="Value">  
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
          <Data property="Unit">  
            <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>  
          </Data>  
          <Data property="Value">  
            <Double>4.6</Double>  
          </Data>  
        </AggregatedDataValue>  
      </Data>  
    </Object>  
  </Components>  
</Object>
```

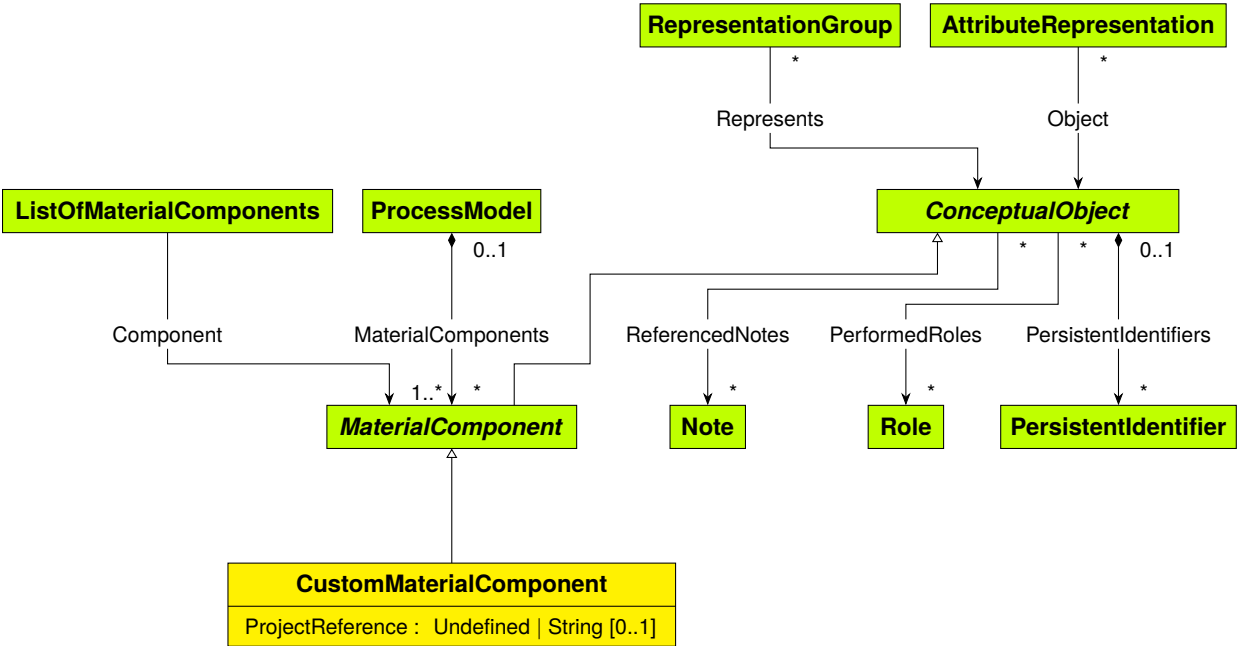
CustomMaterialComponent

Process::Process::CustomMaterialComponent

Overview

Concrete Class

A CustomMaterialComponent defines a custom chemical compound used to specify the chemical composition of a material. These compounds are usually pseudocomponents that are used to simulate the thermodynamic properties of complex mixtures of organic compounds with high molecular weight.



Supertypes

- *MaterialComponent*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>ProjectReference</i>	0..1	<i>Undefined String</i>	no	no

ProjectReference

Data Property

Reference to project identifier or documentation.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

Example in DEXPI XML

```
<Object type="Process/Process.CustomMaterialComponent"/>
```

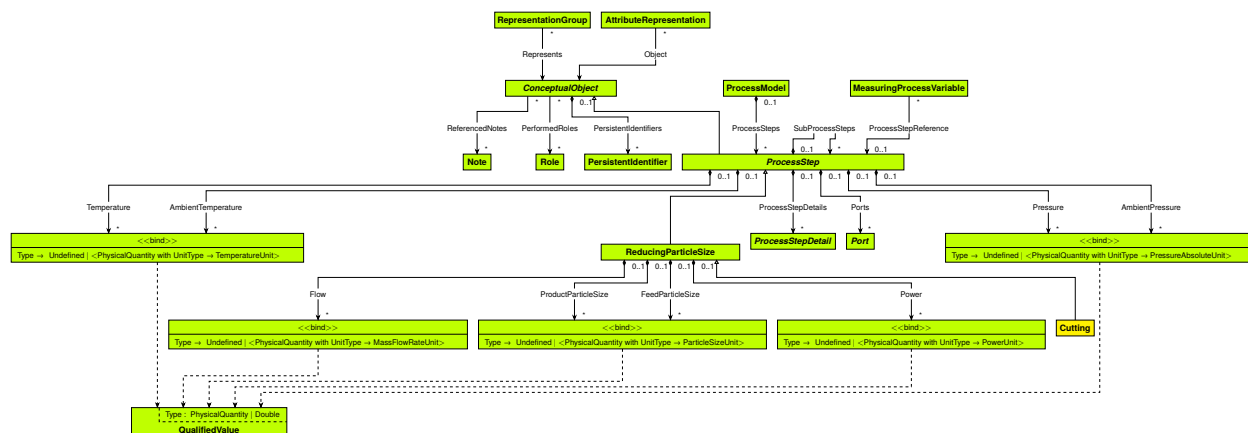
Cutting

Process::Process::Cutting

Overview

Concrete Class

This *ReducingParticleSize* process separates solid material into smaller parts using an edged implement or knife.



Supertypes

- *ReducingParticleSize*

Example in DEXPI XML

```
<Object type="Process/Process.Cutting"/>
```

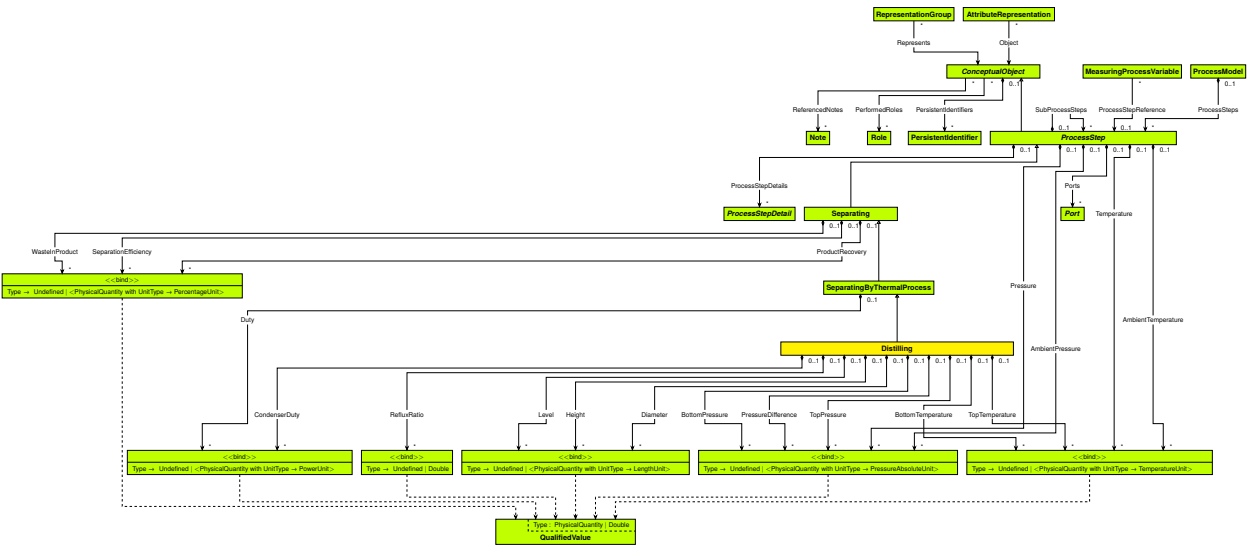
Distilling

Process::Process::Distilling

Overview

Concrete Class

This *SeparatingByThermalProcess* process uses boiling, condensing and countercurrent contacting of liquid and vapour to separate chemical components with a difference in relative volatility.



Supertypes

- *SeparatingByThermalProcess*

Subtypes

- *StabilizingDistilling*
- *StrippingDistilling*
- *VacuumDistilling*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>BottomPressure</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PressureAbsoluteUnit} \rangle \rangle$	no
<i>BottomTemperature</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{TemperatureUnit} \rangle \rangle$	no
<i>CondenserDuty</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle \rangle$	no
<i>Diameter</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle \rangle$	no
<i>Height</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle \rangle$	no
<i>Level</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle \rangle$	no
<i>PressureDifference</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PressureAbsoluteUnit} \rangle \rangle$	no
<i>RefluxRatio</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \text{Double} \rangle$	no
<i>TopPressure</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PressureAbsoluteUnit} \rangle \rangle$	no
<i>TopTemperature</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{TemperatureUnit} \rangle \rangle$	no

BottomPressure

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PressureAbsoluteUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Bar,
      Value = 2.2))
]
```

BottomTemperature

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{TemperatureUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = DegreeCelsius,
      Value = 98.0))
]
```

CondenserDuty**Composition Property**

Duty of the condenser.

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → PowerUnit>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Kilowatt,
      Value = 72.9))
]
```

Diameter**Composition Property**

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → LengthUnit>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Centimetre,
      Value = 42.0))
]
```

Height

Composition Property

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow LengthUnit \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Centimetre,  
      Value = 90.0))  
]
```

Level

Composition Property

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow LengthUnit \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Centimetre,  
      Value = 8.2))  
]
```

PressureDifference

Composition Property

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow PressureAbsoluteUnit \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Centimetre,  
      Value = 8.2))  
]
```

Unit = Bar,
Value = 0.9))

]

RefluxRatio

Composition Property

Reflux ratio.

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \text{Double} \rangle$

Multiplicity: *

Ordered: no

Example

[

QualifiedValue(
Value = 0.24)

]

TopPressure

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PressureAbsoluteUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

[

QualifiedValue(
Value = PhysicalQuantity(
Unit = Bar,
Value = 1.3))

]

TopTemperature

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{TemperatureUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
    QualifiedValue(
        Value = PhysicalQuantity(
            Unit = DegreeCelsius,
            Value = 71.0))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.Distilling">
  <Components property="BottomPressure">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.PressureAbsoluteUnit.Bar"/>
          </Data>
          <Data property="Value">
            <Double>2.2</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="BottomTemperature">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.TemperatureUnit.DegreeCelsius"/>
          </Data>
          <Data property="Value">
            <Double>98.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="CondenserDuty">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
          </Data>
          <Data property="Value">
            <Double>72.9</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>

```

(continues on next page)

(continued from previous page)

```

    </Data>
  </Object>
</Components>
<Components property="Diameter">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
        </Data>
        <Data property="Value">
          <Double>42.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
<Components property="Height">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
        </Data>
        <Data property="Value">
          <Double>90.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
<Components property="Level">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
        </Data>
        <Data property="Value">
          <Double>8.2</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
<Components property="PressureDifference">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.PressureAbsoluteUnit.Bar"/>
        </Data>

```

(continues on next page)

(continued from previous page)

```

        <Data property="Value">
          <Double>0.9</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
<Components property="RefluxRatio">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <Double>0.24</Double>
    </Data>
  </Object>
</Components>
<Components property="TopPressure">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.PressureAbsoluteUnit.Bar"/>
        </Data>
        <Data property="Value">
          <Double>1.3</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
<Components property="TopTemperature">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.TemperatureUnit.DegreeCelsius"/>
        </Data>
        <Data property="Value">
          <Double>71.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
</Object>

```

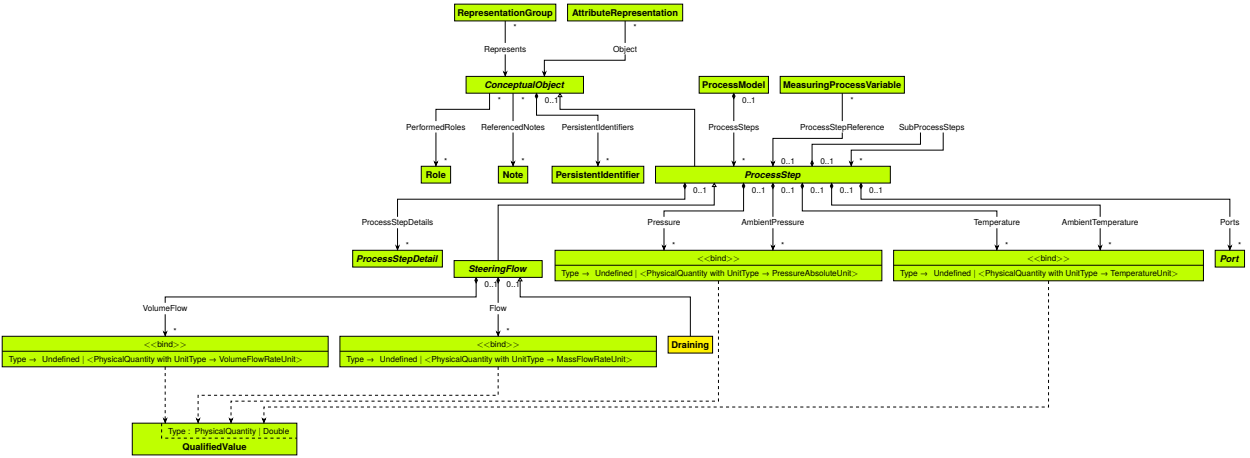
Draining

Process::Process::Draining

Overview

Concrete Class

This supporting *SteeringFlow* process is used to manage leakages and handle the emptying of a plant for maintenance.



Supertypes

- *SteeringFlow*

Example in DEXPI XML

```
<Object type="Process/Process.Draining"/>
```

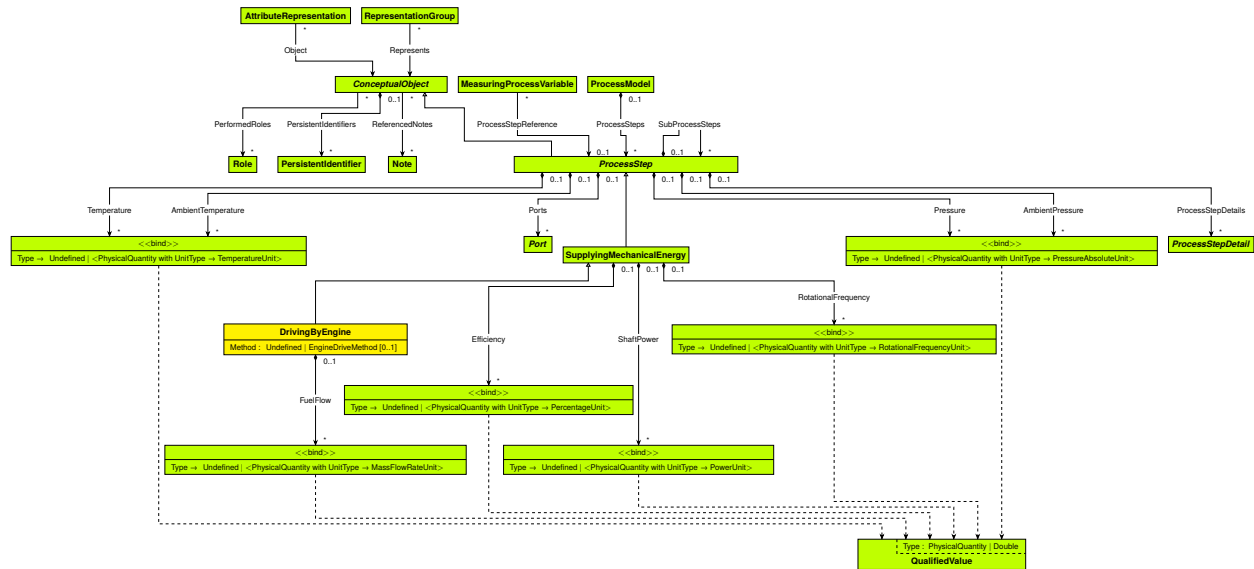
DrivingByEngine

Process::Process::DrivingByEngine

Overview

Concrete Class

This *SupplyingMechanicalEnergy* process uses an internal combustion engine as its source of energy.



Supertypes

- *SupplyingMechanicalEnergy*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>FuelFlow</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MassFlowRateUnit>></i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Method</i>	0..1	<i>Undefined EngineDriveMethod</i>	no	no

FuelFlow

Composition Property

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → MassFlowRateUnit>>*

Multiplicity: *

Ordered: no

Example

[
QualifiedValue(


```

    Value = PhysicalQuantity(
      Unit = KilogramPerHour,
      Value = 18.5))
]

```

Method

Data Property

Type: *Undefined* | *EngineDriveMethod*

Multiplicity: 0..1

Ordered: no

Unique: no

Example in DEXPI XML

```

<Object type="Process/Process.DrivingByEngine">
  <Components property="FuelFlow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
          </Data>
          <Data property="Value">
            <Double>18.5</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>

```

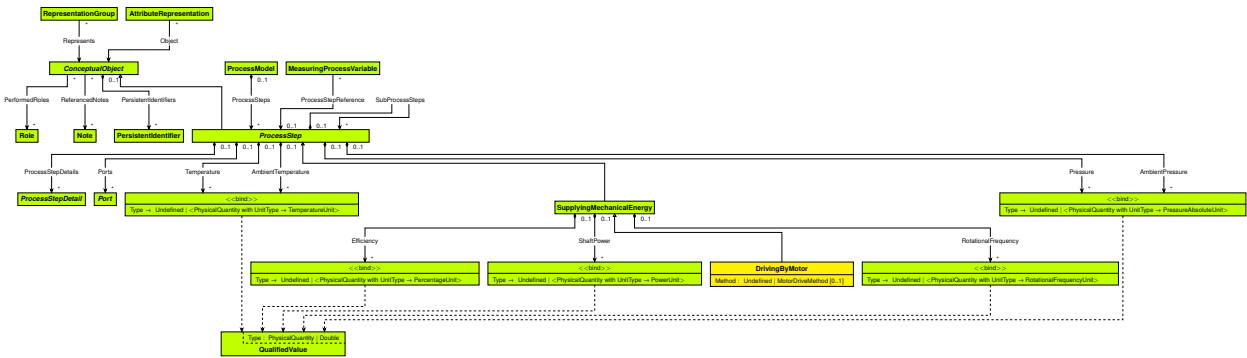
DrivingByMotor

Process::Process::DrivingByMotor

Overview

Concrete Class

This *SupplyingMechanicalEnergy* process uses an electric motor as its source of energy.



Supertypes

- *SupplyingMechanicalEnergy*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Method</i>	0..1	<i>Undefined MotorDriveMethod</i>	no	no

Method

Data Property

Type: *Undefined | MotorDriveMethod*

Multiplicity: 0..1

Ordered: no

Unique: no

Example in DEXPI XML

```
<Object type="Process/Process.DrivingByMotor"/>
```

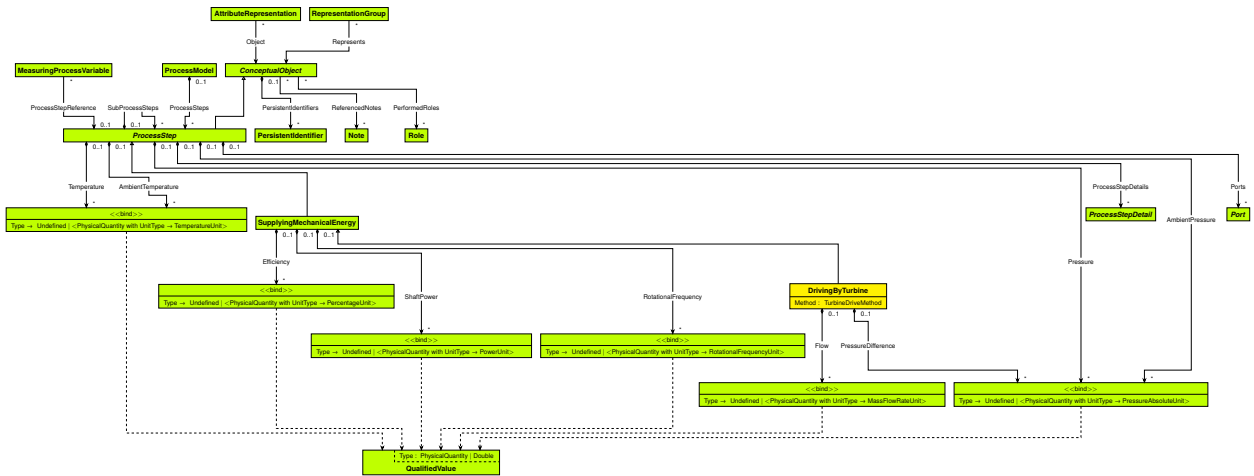
DrivingByTurbine

Process::Process::DrivingByTurbine

Overview

Concrete Class

This *SupplyingMechanicalEnergy* process uses an turbine as its source of energy.



Supertypes

- *SupplyingMechanicalEnergy*

Members

Composition Properties

Name	Multi- plicity	Type	Ordered
<i>Flow</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MassFlowRateUnit>></i>	no
<i>PressureDifference</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PressureAbsoluteUnit>></i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Method</i>	1	<i>TurbineDriveMethod</i>	no	no

Flow

Composition Property

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → MassFlowRateUnit>>*

Multiplicity: *

Ordered: no

Example

[
QualifiedValue(
Value = PhysicalQuantity(

```
        Unit = KilogramPerHour,  
        Value = 102.0))  
]
```

Method

Data Property

Type: *TurbineDriveMethod*

Multiplicity: 1

Ordered: no

Unique: no

PressureDifference

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PressureAbsoluteUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
    QualifiedValue(  
        Value = PhysicalQuantity(  
            Unit = Bar,  
            Value = 3.7))  
]
```

Example in DEXPI XML

```
<Object type="Process/Process.DrivingByTurbine">  
  <Components property="Flow">  
    <Object type="Core/QualifiedValue">  
      <Data property="Value">  
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
          <Data property="Unit">  
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour  
→"/>  
          </Data>  
          <Data property="Value">  
            <Double>102.0</Double>  
          </Data>  
        </AggregatedDataValue>  
      </Data>  
    </Object>  
  </Components>
```

(continues on next page)

(continued from previous page)

```

<Components property="PressureDifference">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.PressureAbsoluteUnit.Bar"/>
        </Data>
        <Data property="Value">
          <Double>3.7</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
</Object>

```

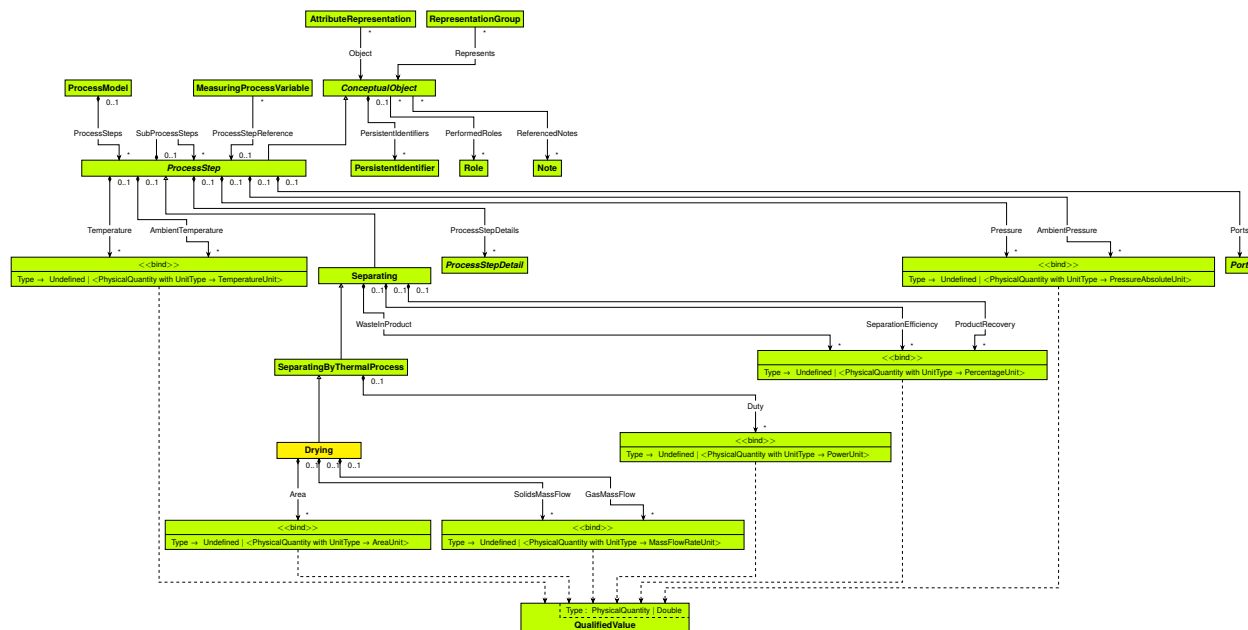
Drying

Process::Process::Drying

Overview

Concrete Class

This *SeparatingByThermalProcess* process removes moisture or liquids from a solid material, producing a solid product with a lower content of the liquid components.



Supertypes

- *SeparatingByThermalProcess*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Area</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{AreaUnit} \rangle \rangle$	no
<i>GasMass-Flow</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$	no
<i>SolidsMass-Flow</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$	no

Area

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{AreaUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = CentimetreSquared,
      Value = 3420.0))
]
```

GasMassFlow

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KilogramPerHour,
      Value = 25.8))
]
```

SolidsMassFlow**Composition Property**

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KilogramPerHour,
      Value = 11.0))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.Drying">
  <Components property="Area">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.AreaUnit.CentimetreSquared"/>
          </Data>
          <Data property="Value">
            <Double>3420.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="GasMassFlow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
          </Data>
          <Data property="Value">
            <Double>25.8</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="SolidsMassFlow">
```

(continues on next page)

(continued from previous page)

```
<Object type="Core/QualifiedValue">
  <Data property="Value">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour"
        />
      </Data>
      <Data property="Value">
        <Double>11.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
</Components>
</Object>
```

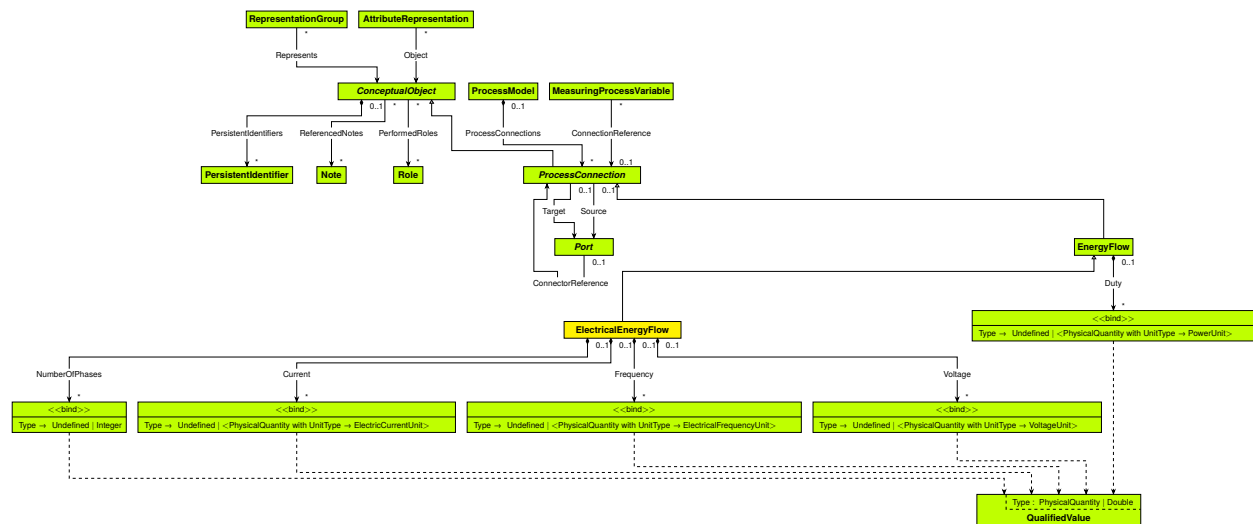
ElectricalEnergyFlow

Process::Process::ElectricalEnergyFlow

Overview

Concrete Class

A *ProcessConnection* that defines the flow of energy between two instances of *ElectricalEnergyPort*. This is used to model flows of electricity into out of the port. The form of electrical energy is defined by the properties of the object.



Supertypes

- *EnergyFlow*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Current</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{ElectricCurrentUnit} \rangle \rangle$	no
<i>Frequency</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{ElectricalFrequencyUnit} \rangle \rangle$	no
<i>NumberOfPhases</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \text{Integer} \rangle$	no
<i>Voltage</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VoltageUnit} \rangle \rangle$	no

Current

Composition Property

Current.

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{ElectricCurrentUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
    QualifiedValue(
        Value = PhysicalQuantity(
            Unit = Ampere,
            Value = 16.0))
]
```

Frequency

Composition Property

Frequency of transmission (zero for DC).

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{ElectricalFrequencyUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

NumberOfPhases

Composition Property

Number of phases (zero for DC).

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \text{Integer} \rangle$

Multiplicity: *

Ordered: no

Voltage

Composition Property

Voltage.

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VoltageUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Volt,  
      Value = 230.0))  
]
```

Example in DEXPI XML

```
<Object type="Process/Process.ElectricalEnergyFlow">  
  <Components property="Current">  
    <Object type="Core/QualifiedValue">  
      <Data property="Value">  
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
          <Data property="Unit">  
            <DataReference data="Core/PhysicalQuantities.ElectricCurrentUnit.Ampere"/>  
          </Data>  
          <Data property="Value">  
            <Double>16.0</Double>  
          </Data>  
        </AggregatedDataValue>  
      </Data>  
    </Object>  
  </Components>  
  <Components property="Voltage">  
    <Object type="Core/QualifiedValue">  
      <Data property="Value">  
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
          <Data property="Unit">  
            <DataReference data="Core/PhysicalQuantities.VoltageUnit.Volt"/>  
          </Data>  
          <Data property="Value">  
            <Double>230.0</Double>  
          </Data>  
        </AggregatedDataValue>  
      </Data>  
    </Object>  
  </Components>  
</Object>
```

(continues on next page)

(continued from previous page)

```
</Object>
</Components>
</Object>
```

ElectricalEnergyPort

Process::Process::ElectricalEnergyPort

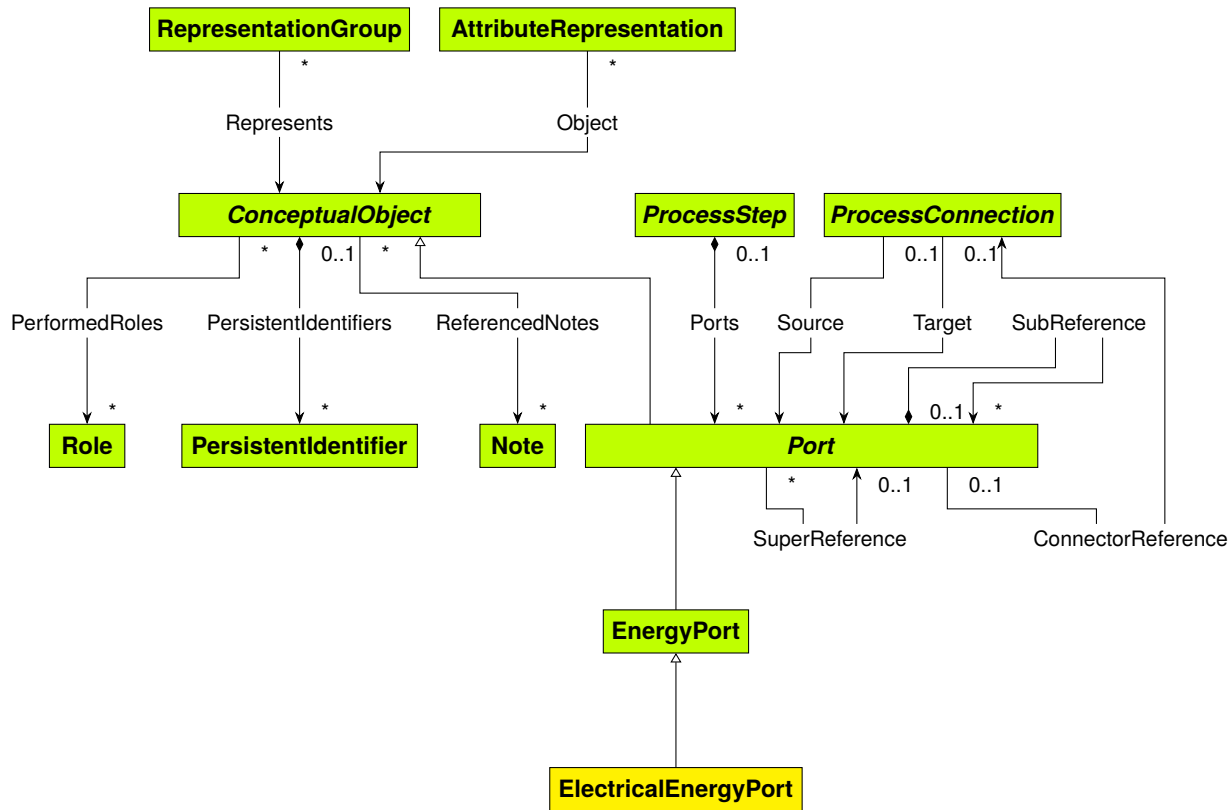
Overview

Concrete Class

An *EnergyPort* that models the exchange of electrical energy between *ProcessStep* objects. The form of electrical energy exchanged is defined by the values of the properties of the *ElectricalEnergyFlow* into or out of the port.

Roles referenced via *PerformedRoles* should have one of the following *Names*:

- Instrumentation
- Power
- Utility



Supertypes

- *EnergyPort*

Example in DEXPI XML

```
<Object type="Process/Process.ElectricalEnergyPort"/>
```

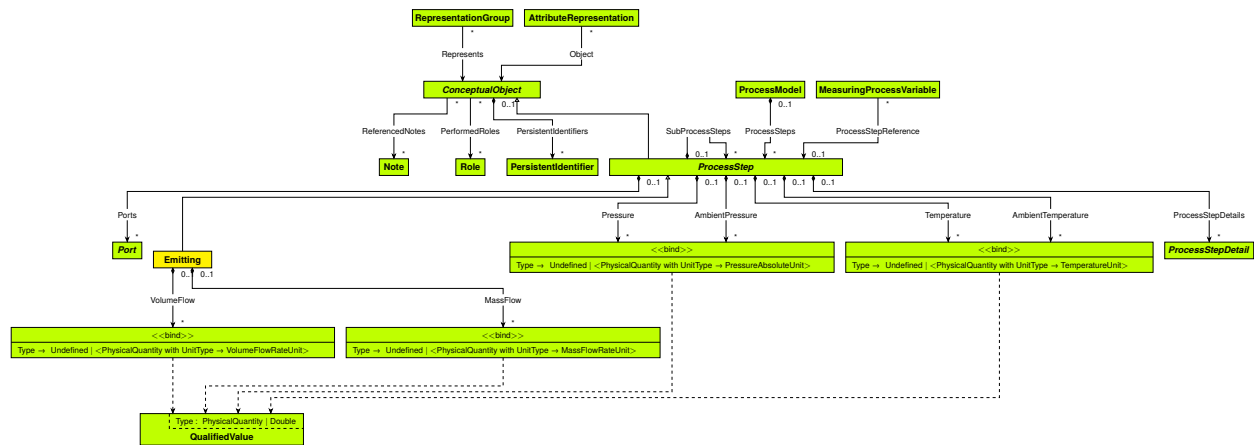
Emitting

Process::Process::Emitting

Overview

Concrete Class

This type represents the destination for a flow out of a process that can be regarded as an emission to the environment. It can be used instead of a *Sink*. It is provided as a convenience for calculating emissions and sustainability calculations.



Supertypes

- *ProcessStep*

Members

Composition Properties

Name	Multi- plicity	Type	Or- dered
<i>Mass-Flow</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MassFlowRateUnit>></i>	no
<i>Volume-Flow</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → VolumeFlowRateUnit>></i>	no

MassFlow

Composition Property

Mass flow rate.

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → MassFlowRateUnit>>*

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KilogramPerHour,
      Value = 7.4))
]
```

VolumeFlow

Composition Property

Volume flow rate.

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → VolumeFlowRateUnit>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = LitrePerSecond,
      Value = 0.9))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.Emitting">
  <Components property="MassFlow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
          </Data>
          <Data property="Value">
            <Double>7.4</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
```

(continues on next page)

(continued from previous page)

```
<Components property="VolumeFlow">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.
↪LitrePerSecond"/>
        </Data>
        <Data property="Value">
          <Double>0.9</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
</Object>
```

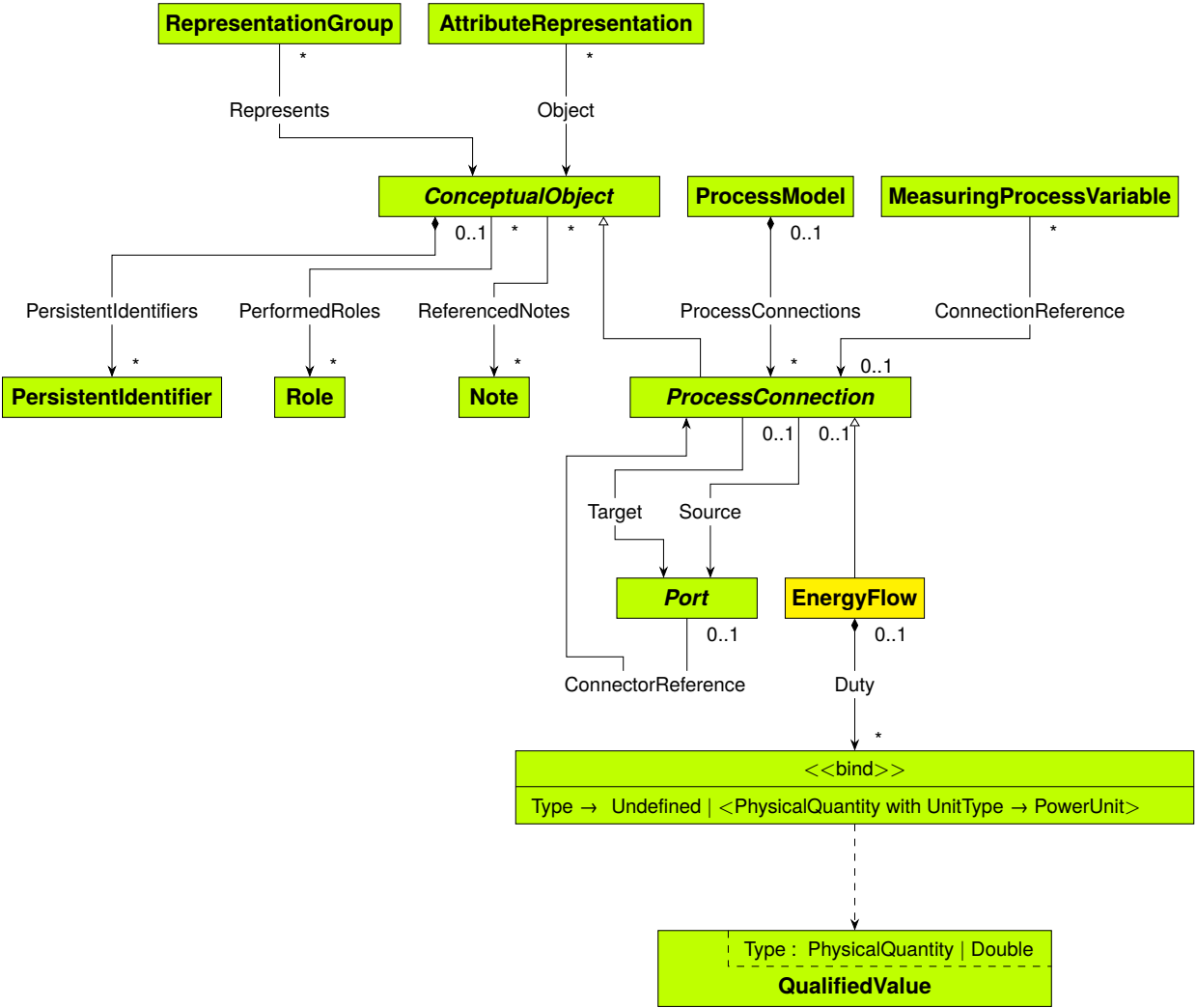
EnergyFlow

Process::Process::EnergyFlow

Overview

Concrete Class

A *ProcessConnection* that defines the flow of energy between two instances of *EnergyPort*. This is used to model flows of energy where the form of energy is unspecified. This type usually abstract and child types of this type are used to model energy flows with a specified form of energy.



Supertypes

- *ProcessConnection*

Subtypes

- *ElectricalEnergyFlow*
- *MechanicalEnergyFlow*
- *ThermalEnergyFlow*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Duty</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PowerUnit>></i>	no

Duty

Composition Property

Rate of energy flow for this connection.

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{PowerUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Kilowatt,  
      Value = 200.0))  
]
```

Example in DEXPI XML

```
<Object type="Process/Process.EnergyFlow">  
  <Components property="Duty">  
    <Object type="Core/QualifiedValue">  
      <Data property="Value">  
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
          <Data property="Unit">  
            <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>  
          </Data>  
          <Data property="Value">  
            <Double>200.0</Double>  
          </Data>  
        </AggregatedDataValue>  
      </Data>  
    </Object>  
  </Components>  
</Object>
```

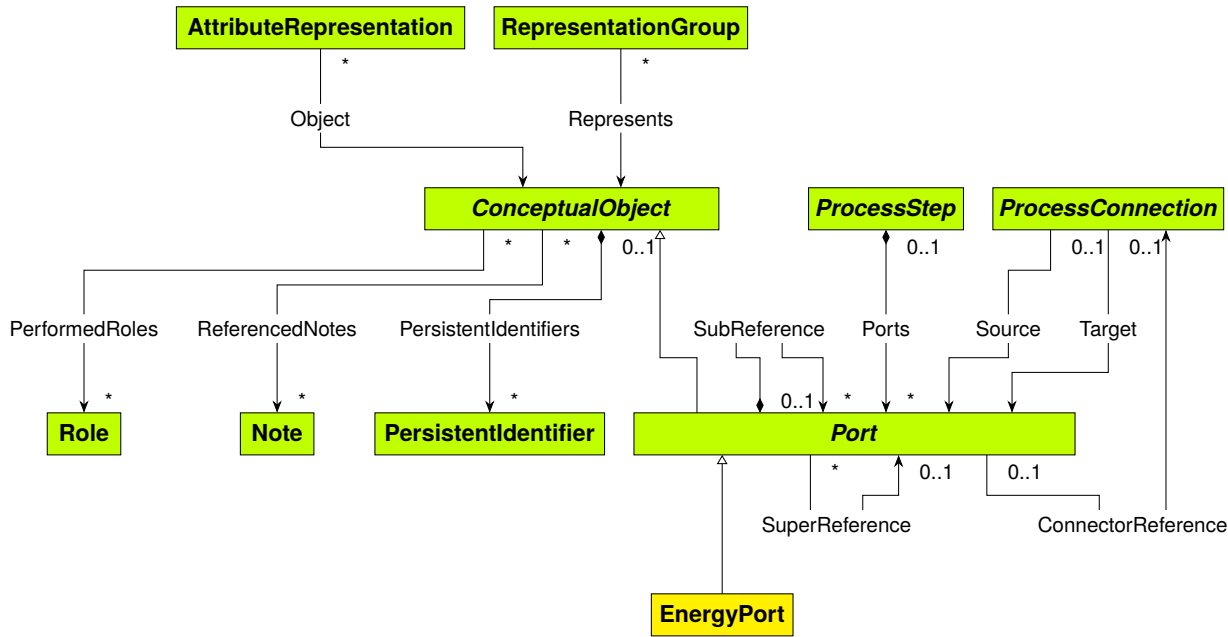
EnergyPort

Process::Process::EnergyPort

Overview

Concrete Class

A *Port* that models the transfer of energy, of whatever form, into our out of a *ProcessStep* object. This type is parent to specific port types that specify the form of energy exchanged. This type can be used to represent energy requirements where the type of energy is unspecified. In practice, this type is abstract, and the child types are used in modelling.



Supertypes

- *Port*

Subtypes

- *ElectricalEnergyPort*
- *MechanicalEnergyPort*
- *ThermalEnergyPort*

Example in DEXPI XML

```
<Object type="Process/Process.EnergyPort"/>
```

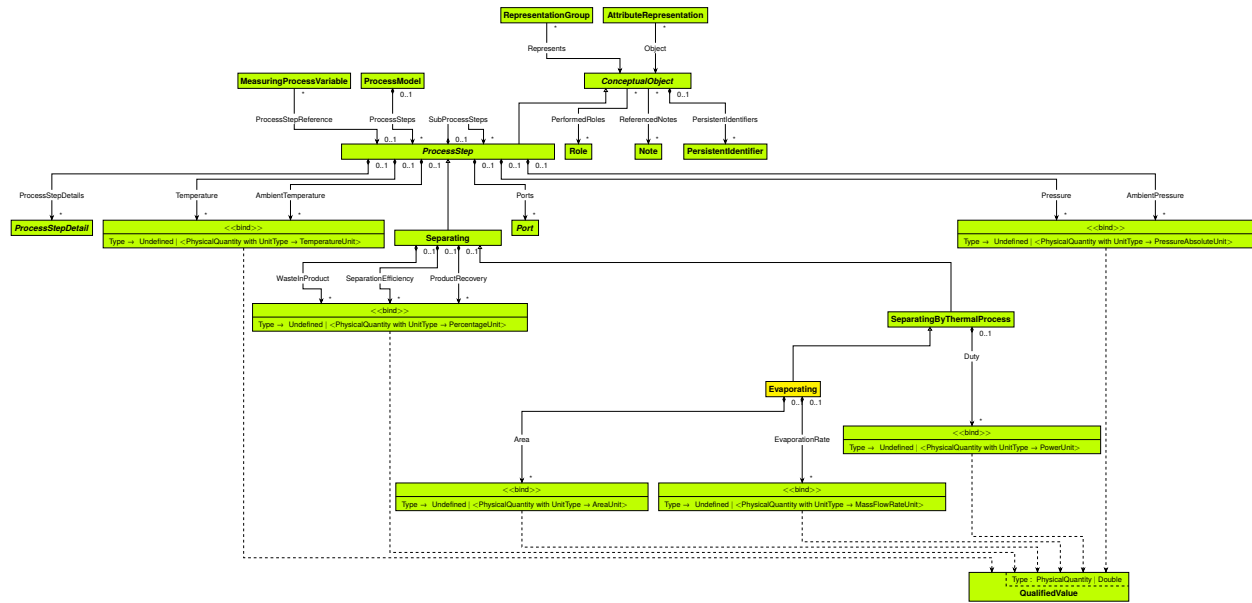
Evaporating

Process::Process::Evaporating

Overview

Concrete Class

This *SeparatingByThermalProcess* process applies thermal energy to drive off moisture or other volatile liquid as gas. This results in a vapour product and a more concentrated liquid or solid product.



Supertypes

- *SeparatingByThermalProcess*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Area</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → AreaUnit>></i>	no
<i>EvaporationRate</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MassFlowRateUnit>></i>	no

Area

Composition Property

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → AreaUnit>>*

Multiplicity: *

Ordered: no

Example

[

```

QualifiedValue(
    Value = PhysicalQuantity(
        Unit = CentimetreSquared,
        Value = 1200.0))

```

]

EvaporationRate**Composition Property****Type:** $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$ **Multiplicity:** ***Ordered:** no**Example**

[

```

QualifiedValue(
  Value = PhysicalQuantity(
    Unit = KilogramPerHour,
    Value = 11.2))

```

]

Example in DEXPI XML

```

<Object type="Process/Process.Evaporating">
  <Components property="Area">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.AreaUnit.CentimetreSquared"/>
          </Data>
          <Data property="Value">
            <Double>1200.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="EvaporationRate">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
          </Data>
          <Data property="Value">
            <Double>11.2</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>

```

(continues on next page)

(continued from previous page)

```
</Components>
</Object>
```

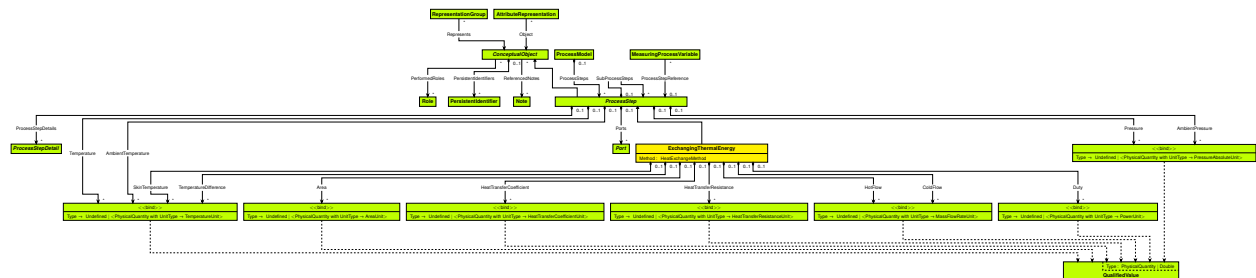
Exchanging Thermal Energy

Process::Process::ExchangingThermalEnergy

Overview

Concrete Class

This process transfers thermal energy between two or more streams of material. This process is realized by a heat exchanger.



Supertypes

- *ProcessStep*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
Area	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → AreaUnit>>	no
ColdFlow	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MassFlowRateUnit>>	no
Duty	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PowerUnit>>	no
HeatTransfer-Coefficient	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → HeatTransferCoefficientUnit>>	no
HeatTransfer-Resistance	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → HeatTransferResistanceUnit>>	no
HotFlow	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MassFlowRateUnit>>	no
SkinTemperature	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → TemperatureUnit>>	no
TemperatureDifference	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → TemperatureUnit>>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Method</i>	1	<i>HeatExchangeMethod</i>	no	no

Area

Composition Property

Type: <*QualifiedValue* with *Type* → *Undefined* | <*PhysicalQuantity* with *UnitType* → *AreaUnit*>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = CentimetreSquared,
      Value = 140000.0))
]
```

ColdFlow

Composition Property

Type: <*QualifiedValue* with *Type* → *Undefined* | <*PhysicalQuantity* with *UnitType* → *MassFlowRateUnit*>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KilogramPerHour,
      Value = 17.2))
]
```

Duty

Composition Property

Type: <*QualifiedValue* with *Type* → *Undefined* | <*PhysicalQuantity* with *UnitType* → *PowerUnit*>>

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Kilowatt,  
      Value = 80.0))  
]
```

HeatTransferCoefficient**Composition Property**

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{HeatTransferCoefficientUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = KilowattPerMetreSquaredKelvin,  
      Value = 18.0))  
]
```

HeatTransferResistance**Composition Property**

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{HeatTransferResistanceUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = MetreSquaredKelvinPerWatt,  
      Value = 5.6e-05))  
]
```

HotFlow**Composition Property**

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{MassFlowRateUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KilogramPerHour,
      Value = 22.0))
]
```

Method**Data Property**

Type: *HeatExchangeMethod*

Multiplicity: 1

Ordered: no

Unique: no

SkinTemperature**Composition Property**

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{TemperatureUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = DegreeCelsius,
      Value = 293.15))
]
```

TemperatureDifference**Composition Property**

Type: $\langle \text{QualifiedValue with } Type \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{TemperatureUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Kelvin,
      Value = 42.0))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.ExchangingThermalEnergy">
  <Components property="Area">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.AreaUnit.CentimetreSquared"/>
          </Data>
          <Data property="Value">
            <Double>14000.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="ColdFlow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
          </Data>
          <Data property="Value">
            <Double>17.2</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="Duty">
```

(continues on next page)

(continued from previous page)

```

<Object type="Core/QualifiedValue">
  <Data property="Value">
    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
      </Data>
      <Data property="Value">
        <Double>80.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
</Components>
<Components property="HeatTransferCoefficient">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.HeatTransferCoefficientUnit.
↪KilowattPerMetreSquaredKelvin"/>
        </Data>
        <Data property="Value">
          <Double>18.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
<Components property="HeatTransferResistance">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.HeatTransferResistanceUnit.
↪MetreSquaredKelvinPerWatt"/>
        </Data>
        <Data property="Value">
          <Double>5.6e-05</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
<Components property="HotFlow">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
        </Data>
        <Data property="Value">

```

(continues on next page)

(continued from previous page)

```

        <Double>22.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
</Components>
<Components property="SkinTemperature">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.TemperatureUnit.DegreeCelsius"/>
        </Data>
        <Data property="Value">
          <Double>293.15</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
<Components property="TemperatureDifference">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.TemperatureUnit.Kelvin"/>
        </Data>
        <Data property="Value">
          <Double>42.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
</Object>

```

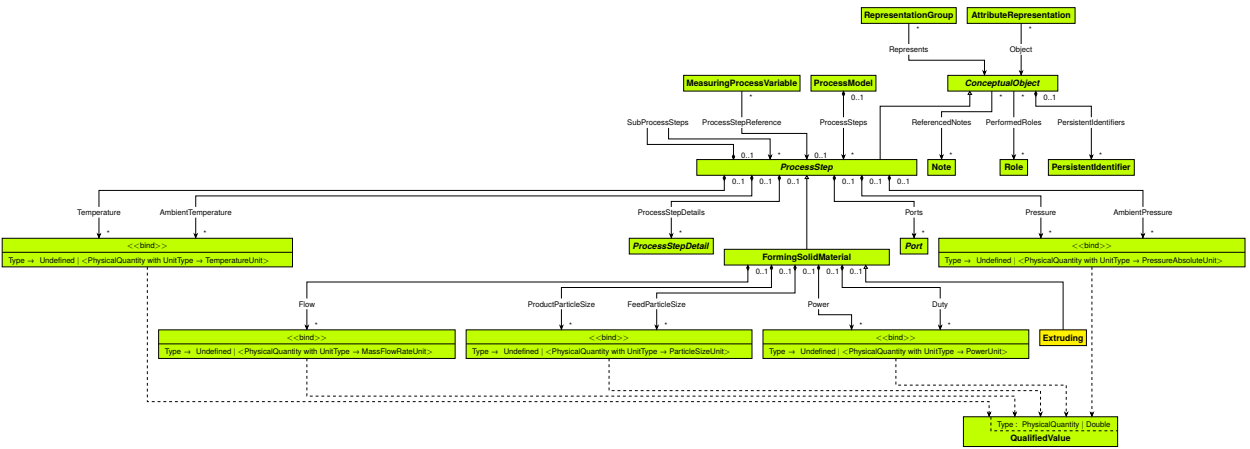
Extruding

Process::Process::Extruding

Overview

Concrete Class

This *FormingSolidMaterial* process takes solid material and forms it by forcing it through a specially designed opening, so that the material is given a specific shape.



Supertypes

- *FormingSolidMaterial*

Example in DEXPI XML

```
<Object type="Process/Process.Extruding"/>
```

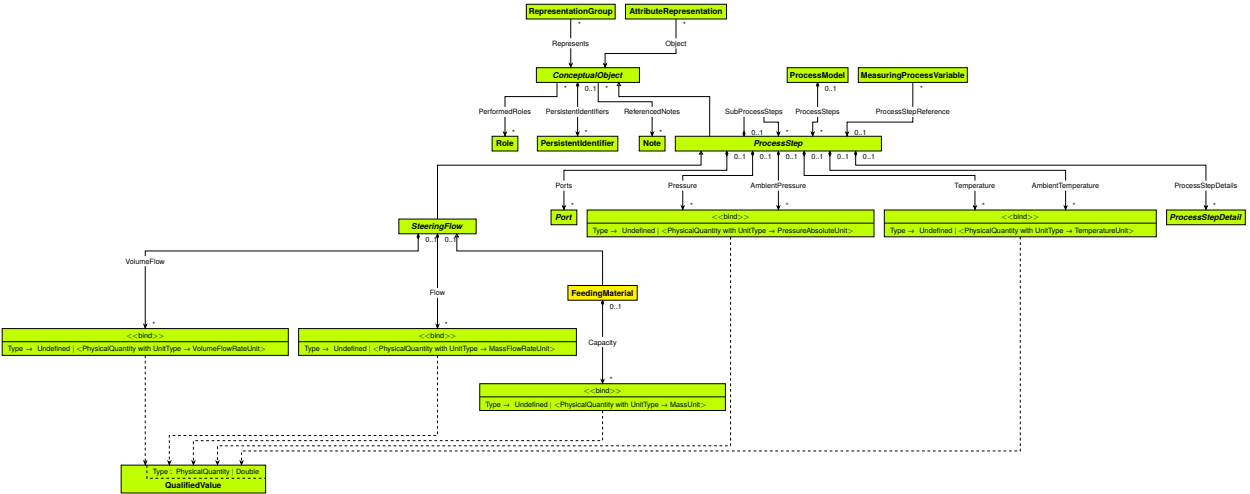
FeedingMaterial

Process::Process::FeedingMaterial

Overview

Concrete Class

This *SteeringFlow* process supplies material to a process at a controllable rate. This process can be used to represent solid feeding processes or dosing systems.



Supertypes

- *SteeringFlow*

Members

Composition Properties

Name	Multiplic- ity	Type	Or- dered
<i>Capacity</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassUnit} \rangle \rangle$	no

Capacity

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Kilogram,
      Value = 24.0))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.FeedingMaterial">
  <Components property="Capacity">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassUnit.Kilogram"/>
          </Data>
          <Data property="Value">
            <Double>24.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```

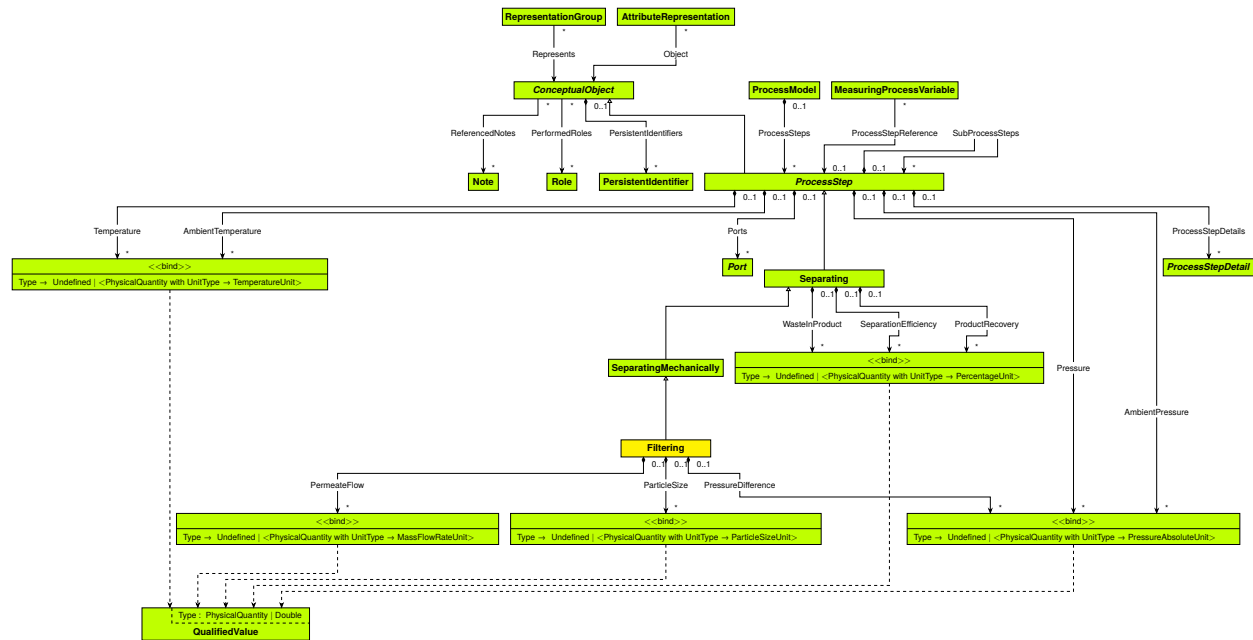
Filtering

Process::Process::Filtering

Overview

Concrete Class

This *SeparatingMechanically* process separates solids from fluids by passing the fluid through a porous barrier.



Supertypes

- *SeparatingMechanically*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>ParticleSize</i>	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → ParticleSizeUnit>>	no
<i>Permeate-Flow</i>	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MassFlowRateUnit>>	no
<i>PressureDifference</i>	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PressureAbsoluteUnit>>	no

ParticleSize

Composition Property

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → ParticleSizeUnit>>

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Millimetre,  
      Value = 0.8))  
]
```

PermeateFlow

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = KilogramPerHour,  
      Value = 18.0))  
]
```

PressureDifference

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PressureAbsoluteUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Bar,  
      Value = 2.8))  
]
```

Example in DEXPI XML

```

<Object type="Process/Process.Filtering">
  <Components property="ParticleSize">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.ParticleSizeUnit.Millimetre"/>
          </Data>
          <Data property="Value">
            <Double>0.8</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="PermeateFlow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
          </Data>
          <Data property="Value">
            <Double>18.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="PressureDifference">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.PressureAbsoluteUnit.Bar"/>
          </Data>
          <Data property="Value">
            <Double>2.8</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>

```

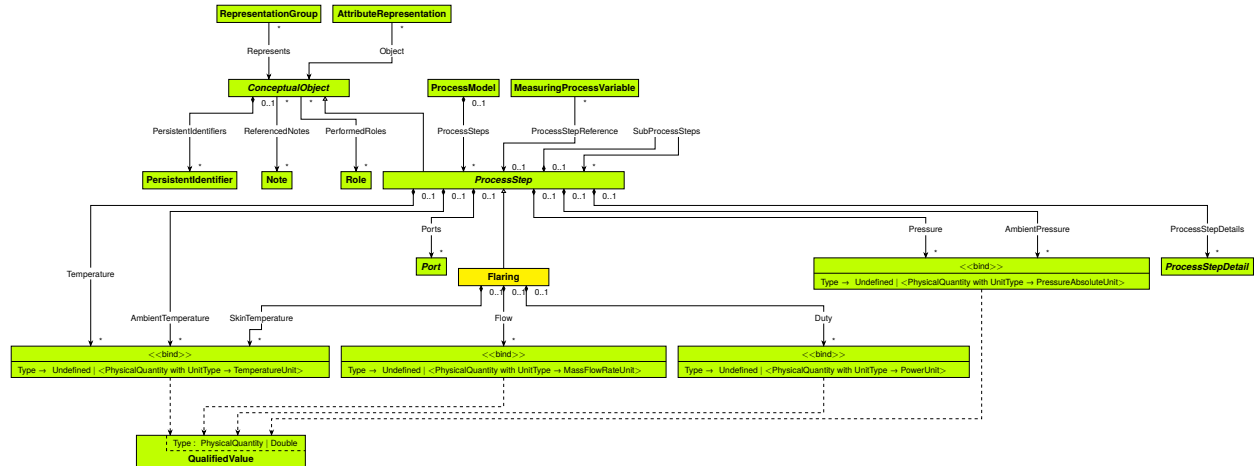
Flaring

Process::Process::Flaring

Overview

Concrete Class

This *SupplyingThermalEnergy* process represents the flare system in a process plant.



Supertypes

- *ProcessStep*

Members

Composition Properties

Name	Multi- plicity	Type	Or- dered
<i>Duty</i>	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PowerUnit>>	no
<i>Flow</i>	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MassFlowRateUnit>>	no
<i>SkinTemperature</i>	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → TemperatureUnit>>	no

Duty

Composition Property

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → PowerUnit>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Kilowatt,
      Value = 92.0))
]
```

Flow**Composition Property**

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → MassFlowRateUnit>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KilogramPerHour,
      Value = 83.0))
]
```

SkinTemperature**Composition Property**

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → TemperatureUnit>>

Multiplicity: *

Ordered: no

Example

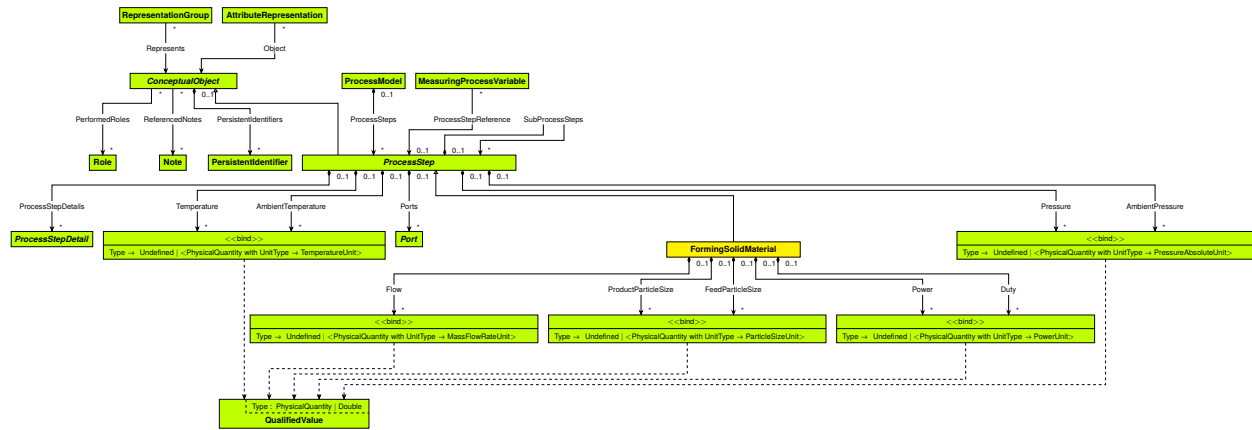
```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = DegreeCelsius,
      Value = 293.15))
]
```

Example in DEXPI XML

```

<Object type="Process/Process.Flaring">
  <Components property="Duty">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
          </Data>
          <Data property="Value">
            <Double>92.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="Flow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
          </Data>
          <Data property="Value">
            <Double>83.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="SkinTemperature">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.TemperatureUnit.DegreeCelsius"/>
          </Data>
          <Data property="Value">
            <Double>293.15</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>

```

Supertypes

- *ProcessStep*

Subtypes

- *Extruding*
- *Pelletizing*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Duty</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle \rangle$	no
<i>FeedParticle-Size</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{ParticleSizeUnit} \rangle \rangle$	no
<i>Flow</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$	no
<i>Power</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle \rangle$	no
<i>ProductParticle-Size</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{ParticleSizeUnit} \rangle \rangle$	no

Duty

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Kilowatt,
      Value = 92.0))
]
```

FeedParticleSize**Composition Property**

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → ParticleSizeUnit>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Millimetre,
      Value = 0.5))
]
```

Flow**Composition Property**

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → MassFlowRateUnit>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KilogramPerHour,
      Value = 76.0))
]
```

Power

Composition Property

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{PowerUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Kilowatt,  
      Value = 92.0))  
]
```

ProductParticleSize

Composition Property

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{ParticleSizeUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Millimetre,  
      Value = 2.6))  
]
```

Example in DEXPI XML

```
<Object type="Process/Process.FormingSolidMaterial">  
  <Components property="Duty">  
    <Object type="Core/QualifiedValue">  
      <Data property="Value">  
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
          <Data property="Unit">  
            <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>  
          </Data>  
          <Data property="Value">  
            <Double>92.0</Double>  
          </Data>  
        </AggregatedDataValue>  
      </Data>  
    </Object>  
  </Components>  
</Object>
```

(continues on next page)

(continued from previous page)

```

        </AggregatedDataValue>
    </Data>
</Object>
</Components>
<Components property="FeedParticleSize">
    <Object type="Core/QualifiedValue">
        <Data property="Value">
            <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
                <Data property="Unit">
                    <DataReference data="Core/PhysicalQuantities.ParticleSizeUnit.Millimetre"/>
                </Data>
                <Data property="Value">
                    <Double>0.5</Double>
                </Data>
            </AggregatedDataValue>
        </Data>
    </Object>
</Components>
<Components property="Flow">
    <Object type="Core/QualifiedValue">
        <Data property="Value">
            <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
                <Data property="Unit">
                    <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
→"/>
                </Data>
                <Data property="Value">
                    <Double>76.0</Double>
                </Data>
            </AggregatedDataValue>
        </Data>
    </Object>
</Components>
<Components property="Power">
    <Object type="Core/QualifiedValue">
        <Data property="Value">
            <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
                <Data property="Unit">
                    <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
                </Data>
                <Data property="Value">
                    <Double>92.0</Double>
                </Data>
            </AggregatedDataValue>
        </Data>
    </Object>
</Components>
<Components property="ProductParticleSize">
    <Object type="Core/QualifiedValue">
        <Data property="Value">
            <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
                <Data property="Unit">

```

(continues on next page)

(continued from previous page)

```

    <DataReference data="Core/PhysicalQuantities.ParticleSizeUnit.Millimetre"/>
  </Data>
  <Data property="Value">
    <Double>2.6</Double>
  </Data>
</AggregatedDataValue>
</Data>
</Object>
</Components>
</Object>

```

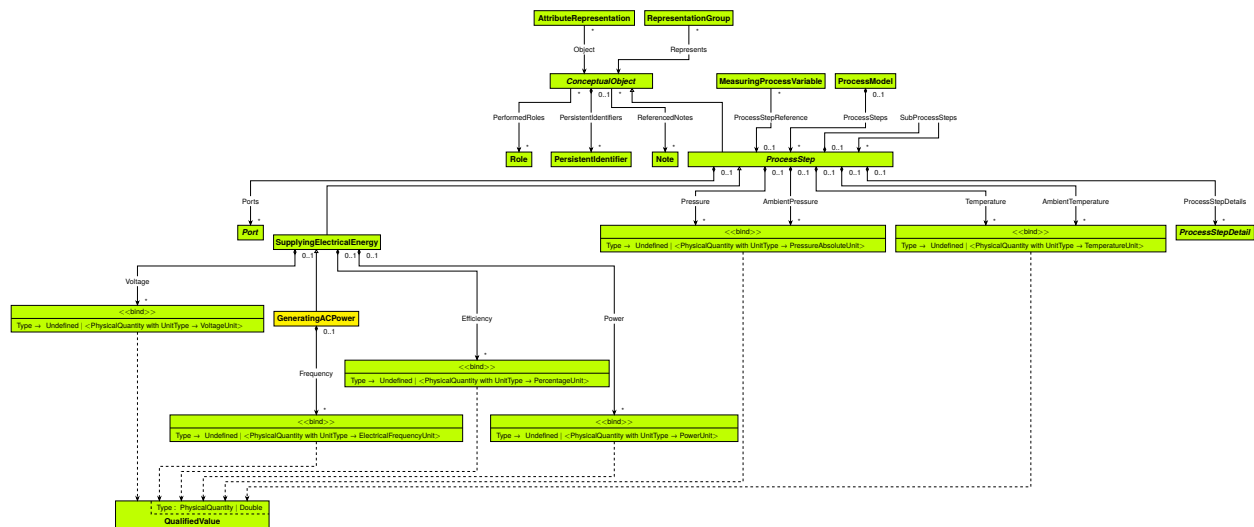
GeneratingACPower

Process::Process::GeneratingACPower

Overview

Concrete Class

This *SupplyingElectricalEnergy* process produces alternating current electrical energy from mechanical energy.



Supertypes

- *SupplyingElectricalEnergy*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
Frequency	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → ElectricalFrequencyUnit>></i>	no

Frequency

Composition Property

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → ElectricalFrequencyUnit>>

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Process/Process.GeneratingACPower"/>
```

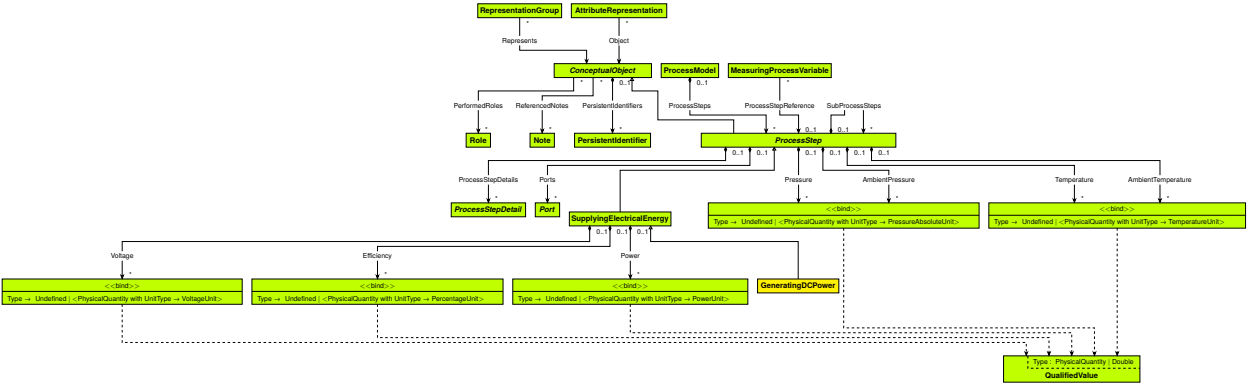
GeneratingDCPower

Process::Process::GeneratingDCPower

Overview

Concrete Class

This *SupplyingElectricalEnergy* process produces alternating current electrical energy from mechanical energy.



Supertypes

- *SupplyingElectricalEnergy*

Example in DEXPI XML

```
<Object type="Process/Process.GeneratingDCPower"/>
```

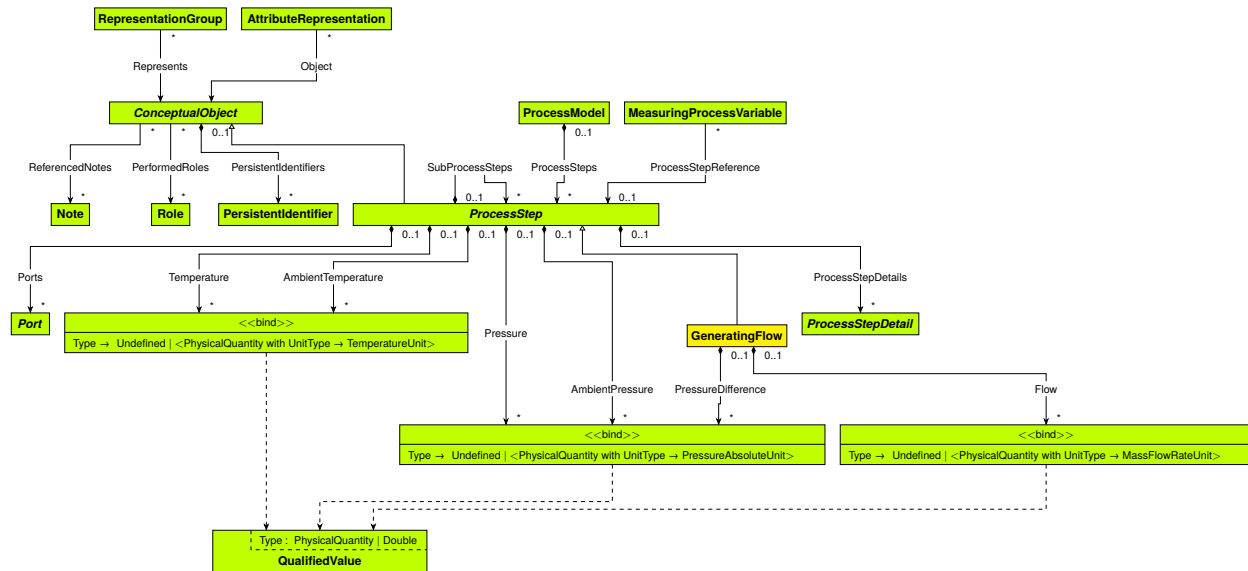
GeneratingFlow

Process::Process::GeneratingFlow

Overview

Concrete Class

This is a base type for processes with the purpose of generating flow and/or increasing pressure of a fluid stream. The child types are *Pumping* and *Compressing*.



Supertypes

- *ProcessStep*

Subtypes

- *Compressing*
- *Pumping*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Flow</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MassFlowRateUnit>></i>	no
<i>PressureDifference</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PressureAbsoluteUnit>></i>	no

Flow

Composition Property

The mass flow rate of the *GeneratingFlow*.

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → MassFlowRateUnit>>*

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KilogramPerHour,
      Value = 92.0))
]
```

PressureDifference**Composition Property**

Pressure difference across process.

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → PressureAbsoluteUnit>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Bar,
      Value = 2.2))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.GeneratingFlow">
  <Components property="Flow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
          </Data>
          <Data property="Value">
            <Double>92.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="PressureDifference">
    <Object type="Core/QualifiedValue">
```

(continues on next page)

(continued from previous page)

```

<Data property="Value">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.PressureAbsoluteUnit.Bar"/>
    </Data>
    <Data property="Value">
      <Double>2.2</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>
</Components>
</Object>

```

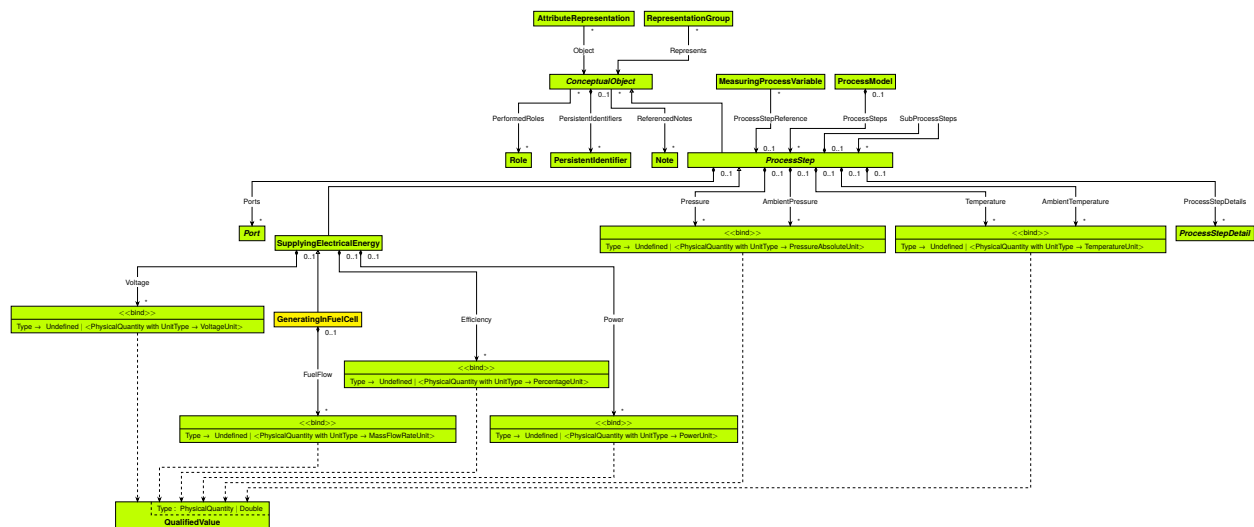
GeneratingInFuelCell

Process::Process::GeneratingInFuelCell

Overview

Concrete Class

This process generates electricity using a fuel cell.



Supertypes

- *SupplyingElectricalEnergy*

Members

Composition Properties

Name	Multi- plicity	Type	Or- dered
<i>FuelFlow</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MassFlowRateUnit>></i>	no

FuelFlow

Composition Property

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → MassFlowRateUnit>>*

Multiplicity: *

Ordered: no

Example

```
[
    QualifiedValue(
        Value = PhysicalQuantity(
            Unit = KilogramPerHour,
            Value = 12.2))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.GeneratingInFuelCell">
  <Components property="FuelFlow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
          </Data>
          <Data property="Value">
            <Double>12.2</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```

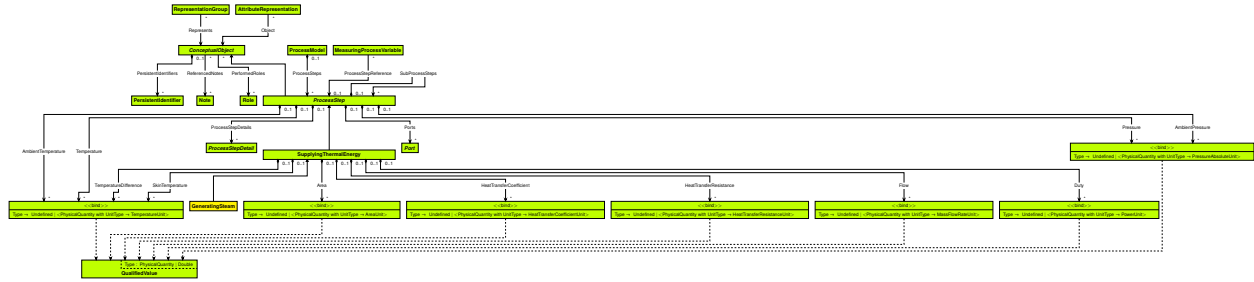
GeneratingSteam

Process::Process::GeneratingSteam

Overview

Concrete Class

This process produces steam for use in other processes.



Supertypes

- *SupplyingThermalEnergy*

Example in DEXPI XML

```
<Object type="Process/Process.GeneratingSteam"/>
```

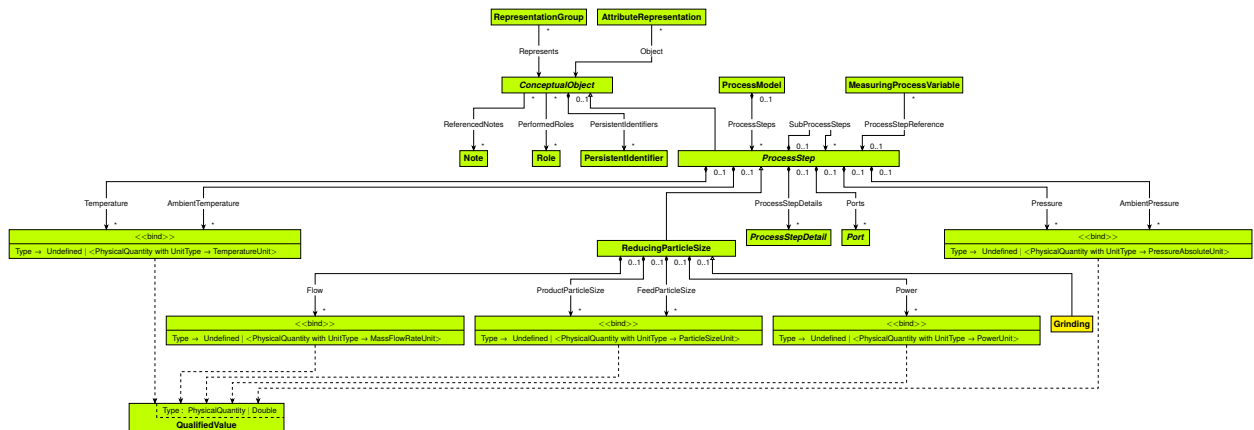
Grinding

Process::Process::Grinding

Overview

Concrete Class

This *ReducingParticleSize* process produces fine particles by abrasion of solid material.



Supertypes

- *ReducingParticleSize*

Example in DEXPI XML

```
<Object type="Process/Process.Grinding"/>
```

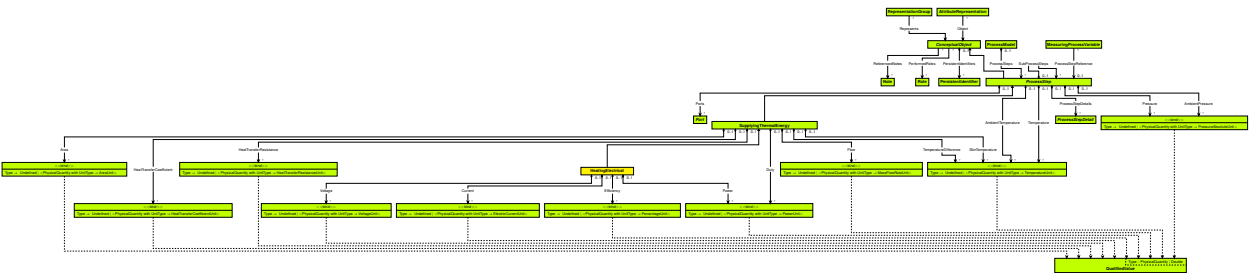
HeatingElectrical

Process::Process::HeatingElectrical

Overview

Concrete Class

This *SupplyingThermalEnergy* process increases the enthalpy content of one or more material streams through the application of electrical energy.



Supertypes

- *SupplyingThermalEnergy*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Current</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → ElectricCurrentUnit>></i>	no
<i>Efficiency</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PercentageUnit>></i>	no
<i>Power</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PowerUnit>></i>	no
<i>Voltage</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → VoltageUnit>></i>	no

Current

Composition Property

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → ElectricCurrentUnit>>*

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Ampere,  
      Value = 800.0))  
]
```

Efficiency

Composition Property

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → PercentageUnit>>

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Percent,  
      Value = 92.0))  
]
```

Power

Composition Property

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → PowerUnit>>

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Kilowatt,  
      Value = 184.0))  
]
```


Voltage

Composition Property

Type: $\langle \text{QualifiedValue with } Type \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{VoltageUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Volt,
      Value = 230.0))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.HeatingElectrical">
  <Components property="Current">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.ElectricCurrentUnit.Ampere"/>
          </Data>
          <Data property="Value">
            <Double>800.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="Efficiency">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.PercentageUnit.Percent"/>
          </Data>
          <Data property="Value">
            <Double>92.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="Power">
    <Object type="Core/QualifiedValue">
```

(continues on next page)

(continued from previous page)

```
<Data property="Value">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
    </Data>
    <Data property="Value">
      <Double>184.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>
</Components>
<Components property="Voltage">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.VoltageUnit.Volt"/>
        </Data>
        <Data property="Value">
          <Double>230.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
</Object>
```

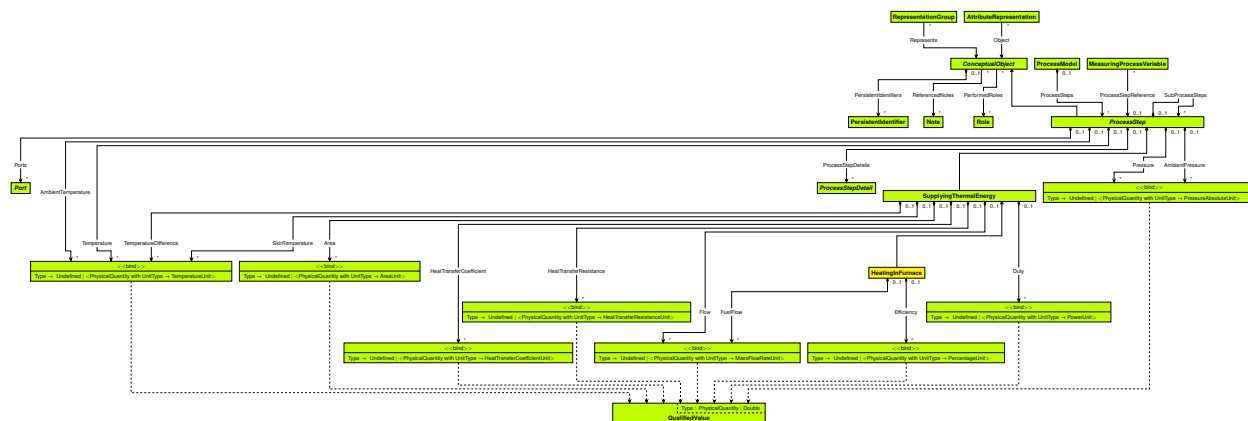
HeatingInFurnace

Process::Process::HeatingInFurnace

Overview

Concrete Class

This *SupplyingThermalEnergy* process increases the enthalpy content of one or more material streams by passing them through a fired heater or furnace.



Supertypes

- *SupplyingThermalEnergy*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Efficiency</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PercentageUnit} \rangle \rangle$	no
<i>FuelFlow</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$	no

Efficiency

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PercentageUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
    QualifiedValue(
        Value = PhysicalQuantity(
            Unit = Percent,
            Value = 80.0))
]
```

FuelFlow

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
    QualifiedValue(
        Value = PhysicalQuantity(
            Unit = KilogramPerHour,
            Value = 12.0))
]
```

]

Example in DEXPI XML

```

<Object type="Process/Process.HeatingInFurnace">
  <Components property="Efficiency">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.PercentageUnit.Percent"/>
          </Data>
          <Data property="Value">
            <Double>80.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="FuelFlow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
→"/>
          </Data>
          <Data property="Value">
            <Double>12.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>

```

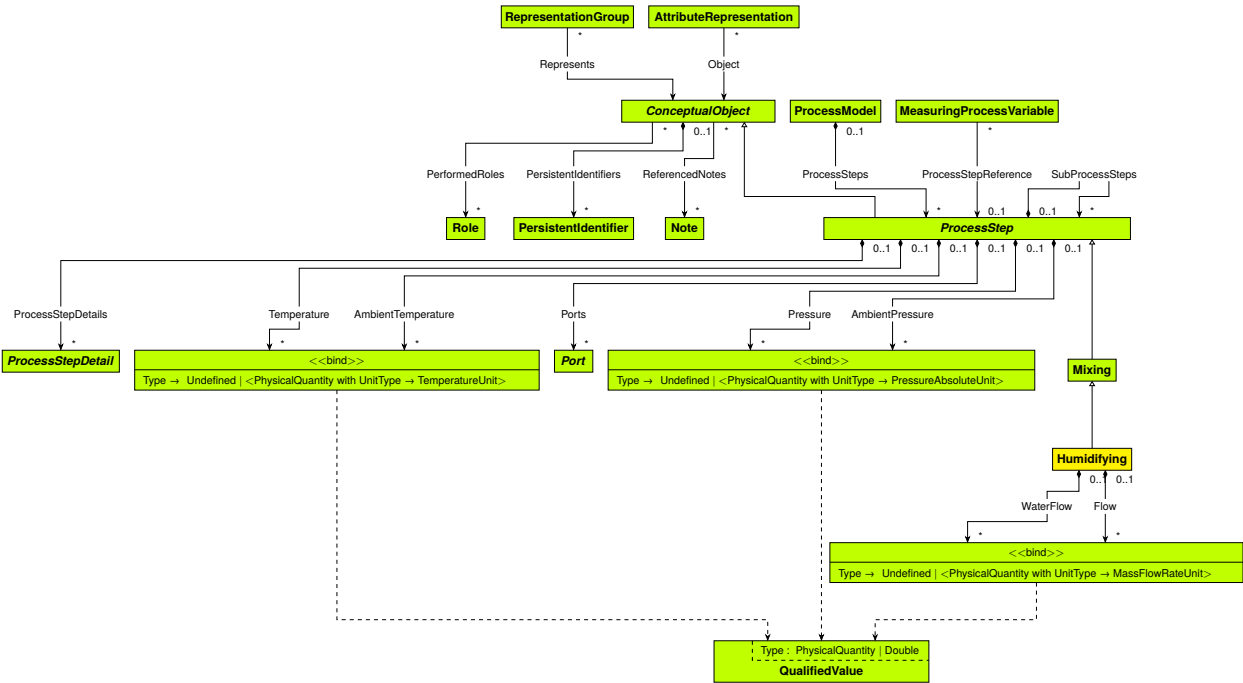
Humidifying

Process::Process::Humidifying

Overview

Concrete Class

This *Mixing* process increases the water or liquid content of a material stream by mixing a feed with the water or liquid.



Supertypes

- *Mixing*

Members

Composition Properties

Name	Multi- plicity	Type	Or- dered
Flow	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MassFlowRateUnit>>	no
Water- Flow	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MassFlowRateUnit>>	no

Flow

Composition Property

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → MassFlowRateUnit>>

Multiplicity: *

Ordered: no

Example

[
 QualifiedValue(
 Value = PhysicalQuantity(

```

    Unit = KilogramPerHour,
    Value = 45.0))

```

WaterFlow

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```

[
    QualifiedValue(
        Value = PhysicalQuantity(
            Unit = KilogramPerHour,
            Value = 8.0))
]

```

Example in DEXPI XML

```

<Object type="Process/Process.Humidifying">
  <Components property="Flow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
          </Data>
          <Data property="Value">
            <Double>45.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="WaterFlow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
          </Data>
          <Data property="Value">
            <Double>8.0</Double>

```

(continues on next page)

Process::Process::IncreasingParticleSize

Composition Properties

Name	Multiplicity	Type	Ordered
<i>FeedParticleSize</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{ParticleSizeUnit} \rangle \rangle$	no
<i>Flow</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$	no
<i>LiquidFlow</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$	no
<i>Power</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle \rangle$	no
<i>ProductParticleSize</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{ParticleSizeUnit} \rangle \rangle$	no
<i>SolidsFlow</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$	no

FeedParticleSize

Composition Property

Characteristic particle size for the feed.

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{ParticleSizeUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Millimetre,
      Value = 0.2))
]
```

Flow

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
```



```

    Value = PhysicalQuantity(
      Unit = KilogramPerHour,
      Value = 82.0))

```

```

]

```

LiquidFlow

Composition Property

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → MassFlowRateUnit>>

Multiplicity: *

Ordered: no

Example

```

[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KilogramPerHour,
      Value = 34.0))
]

```

Power

Composition Property

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → PowerUnit>>

Multiplicity: *

Ordered: no

Example

```

[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Kilowatt,
      Value = 820.0))
]

```

ProductParticleSize

Composition Property

Characteristic particle size for the product.

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → ParticleSizeUnit>>

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Millimetre,  
      Value = 8.0))  
]
```

SolidsFlow

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = KilogramPerHour,  
      Value = 48.0))  
]
```

Example in DEXPI XML

```
<Object type="Process/Process.IncreasingParticleSize">  
  <Components property="FeedParticleSize">  
    <Object type="Core/QualifiedValue">  
      <Data property="Value">  
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
          <Data property="Unit">  
            <DataReference data="Core/PhysicalQuantities.ParticleSizeUnit.Millimetre"/>  
          </Data>  
          <Data property="Value">  
            <Double>0.2</Double>  
          </Data>  
        </AggregatedDataValue>  
      </Data>  
    </Object>  
  </Components>  
  <Components property="Flow">  
    <Object type="Core/QualifiedValue">
```

(continues on next page)

(continued from previous page)

```

    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
        </Data>
        <Data property="Value">
          <Double>82.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
<Components property="LiquidFlow">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
        </Data>
        <Data property="Value">
          <Double>34.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
<Components property="Power">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
        </Data>
        <Data property="Value">
          <Double>820.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
<Components property="ProductParticleSize">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.ParticleSizeUnit.Millimetre"/>
        </Data>
        <Data property="Value">
          <Double>8.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>

```

(continues on next page)

(continued from previous page)

```

        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="SolidsFlow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
          </Data>
          <Data property="Value">
            <Double>48.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>

```

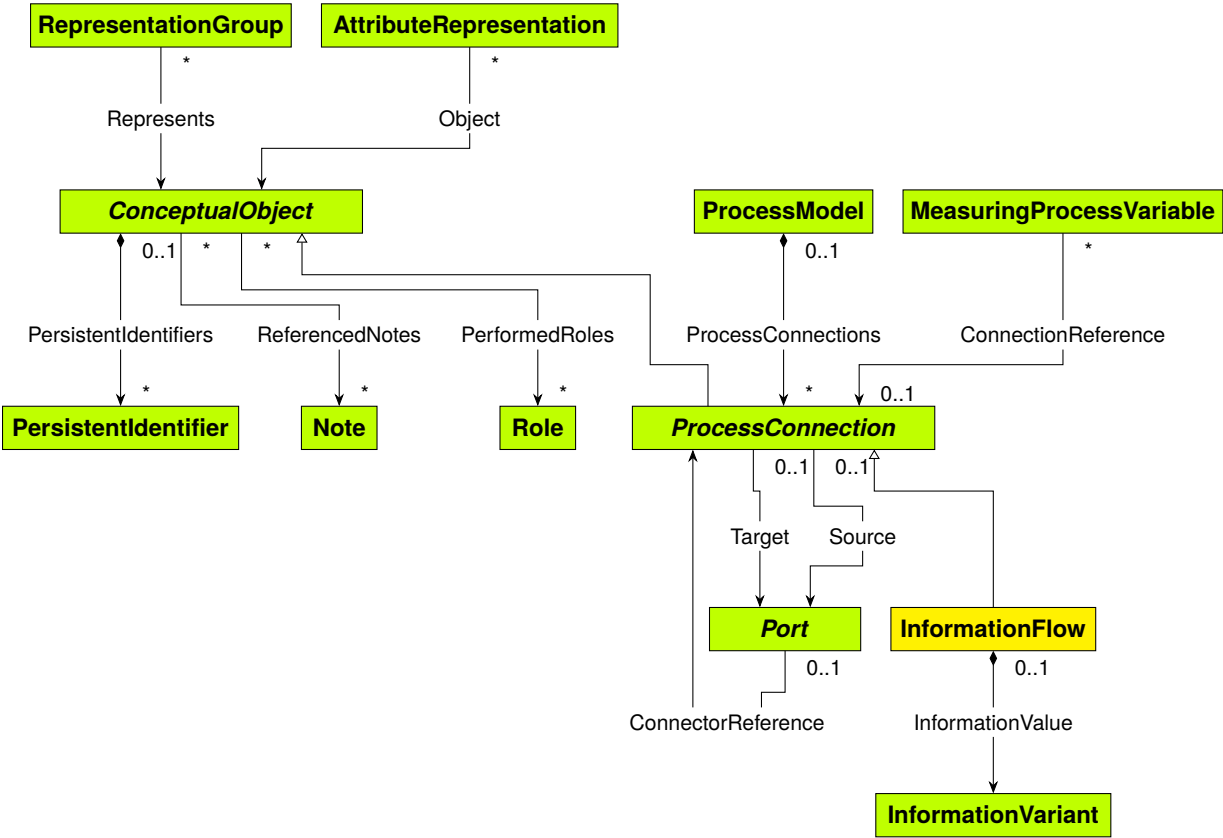
InformationFlow

Process::Process::InformationFlow

Overview

Concrete Class

A *ProcessConnection* that defines the flow of information between two instances of *InformationPort*. This is used to model signals and other information flows into out of the port.



Supertypes

- *ProcessConnection*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>InformationValue</i>	1	<i>InformationVariant</i>	no

InformationValue

Composition Property

Type: *InformationVariant*

Multiplicity: 1

Ordered: no

Example in DEXPI XML

```
<Object type="Process/Process.InformationFlow"/>
```

InformationPort

Process::Process::InformationPort

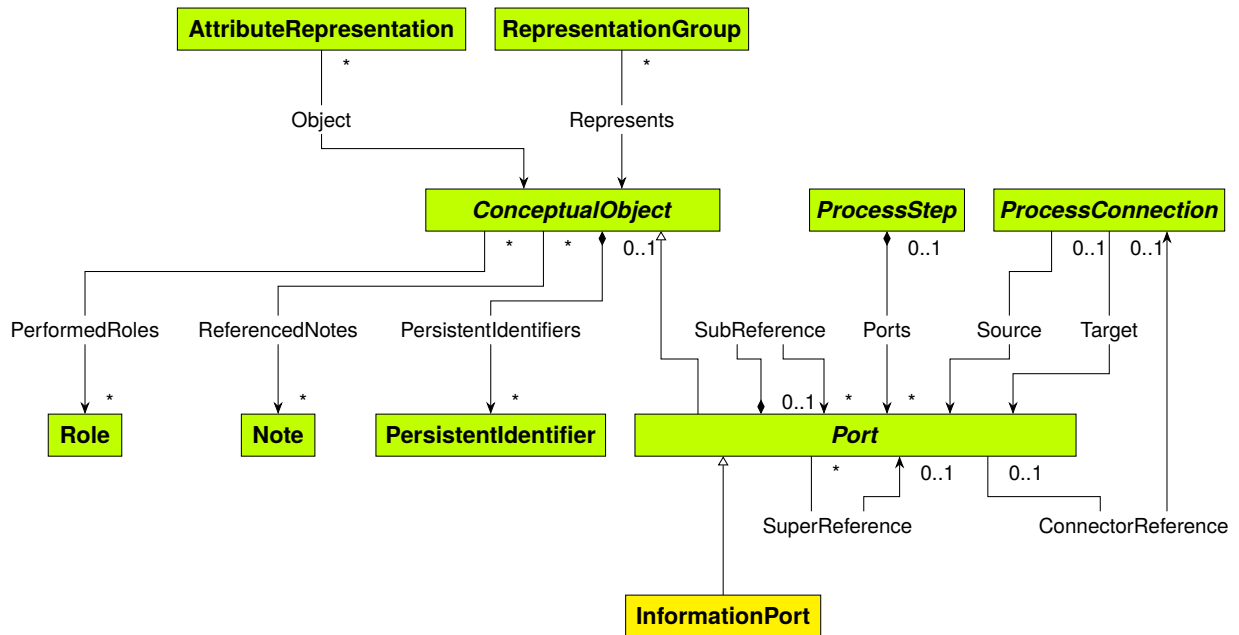
Overview

Concrete Class

A *Port* that models the transfer of information, of whatever form, into our out of a *ProcessStep* object. This type is used to represent control signals.

Roles referenced via *PerformedRoles* should have one of the following *Names*:

- Feedback
- Input
- Output
- Override
- Setpoint



Supertypes

- *Port*

Example in DEXPI XML

```
<Object type="Process/Process.InformationPort"/>
```

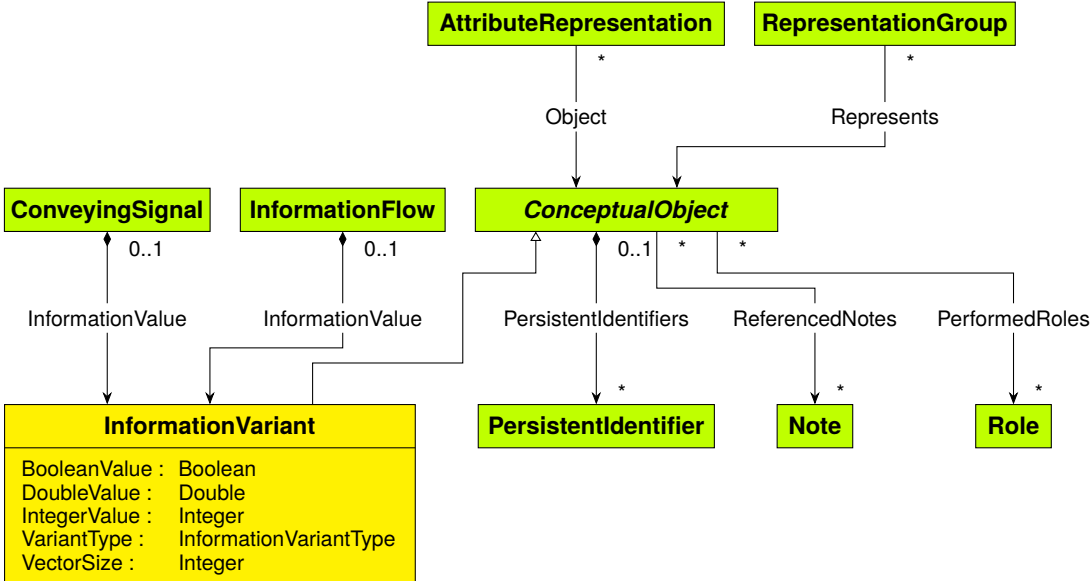
InformationVariant

Process::Process::InformationVariant

Overview

Concrete Class

This data structure is embedded in the *InformationFlow* object and can be configured to represent scalars and arrays of boolean, integer and real variables.



Supertypes

- *ConceptualObject*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>BooleanValue</i>	1	<i>Boolean</i>	no	no
<i>DoubleValue</i>	1	<i>Double</i>	no	no
<i>IntegerValue</i>	1	<i>Integer</i>	no	no
<i>VariantType</i>	1	<i>InformationVariantType</i>	no	no
<i>VectorSize</i>	1	<i>Integer</i>	no	no

BooleanValue

Data Property

Type: *Boolean*

Multiplicity: 1

Ordered: no

Unique: no

DoubleValue**Data Property**

Type: *Double*

Multiplicity: 1

Ordered: no

Unique: no

IntegerValue**Data Property**

Type: *Integer*

Multiplicity: 1

Ordered: no

Unique: no

VariantType**Data Property**

Type: *InformationVariantType*

Multiplicity: 1

Ordered: no

Unique: no

VectorSize**Data Property**

Type: *Integer*

Multiplicity: 1

Ordered: no

Unique: no

Example in DEXPI XML

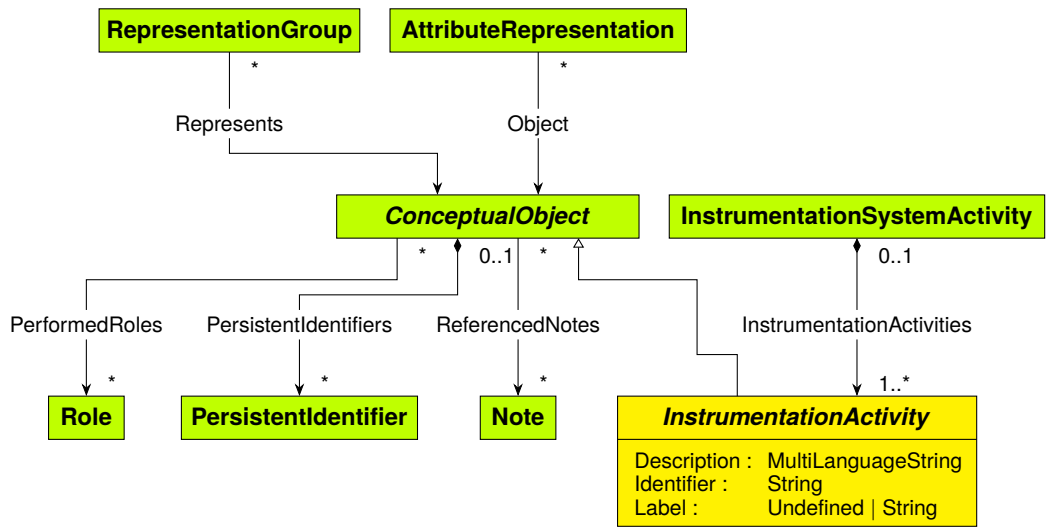
```
<Object type="Process/Process.InformationVariant"/>
```

InstrumentationActivity

Process::Process::InstrumentationActivity

Overview**Abstract Class**

This is the base type for control and instrumentation functions that are specified as part of the process design.



Supertypes

- *ConceptualObject*

Subtypes

- *CalculatingProcessVariable*
- *ControllingProcessVariable*
- *ConveyingSignal*
- *MeasuringProcessVariable*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Description</i>	1	<i>MultiLanguageString</i>	no	no
<i>Identifier</i>	1	<i>String</i>	no	no
<i>Label</i>	1	<i>Undefined String</i>	no	no

Description

Data Property

Type: *MultiLanguageString*

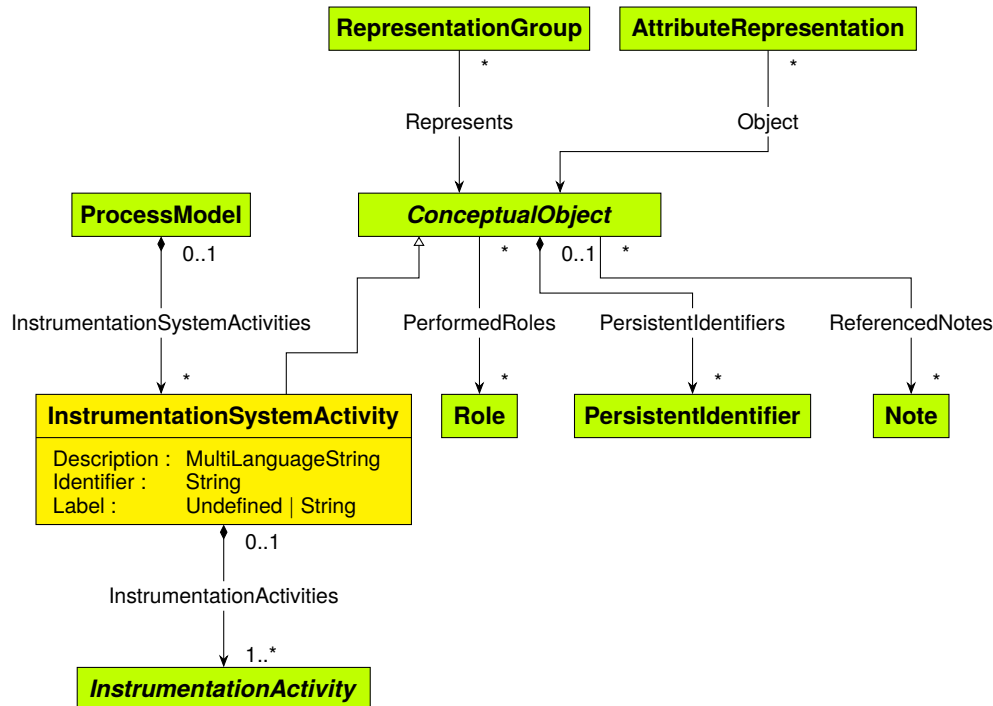
Multiplicity: 1

Ordered: no

Unique: no

Identifier**Data Property****Type:** *String***Multiplicity:** 1**Ordered:** no**Unique:** no**Label****Data Property****Type:** *Undefined* | *String***Multiplicity:** 1**Ordered:** no**Unique:** no**InstrumentationSystemActivity***Process::Process::InstrumentationSystemActivity***Overview****Concrete Class**

An *InstrumentationSystemActivity* is a collection of related *InstrumentationActivity* instances that together perform a specified function. Thus a *MeasuringProcessVariable* and a *ControllingProcessVariable* activity can be grouped into a single *InstrumentationSystemActivity* that represents the control loop.



Supertypes

- *ConceptualObject*

Members**Composition Properties**

Name	Multiplicity	Type	Ordered
<i>InstrumentationActivities</i>	1..*	<i>InstrumentationActivity</i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Description</i>	1	<i>MultiLanguageString</i>	no	no
<i>Identifier</i>	1	<i>String</i>	no	no
<i>Label</i>	1	<i>Undefined String</i>	no	no

Description**Data Property**

Type: *MultiLanguageString*

Multiplicity: 1

Ordered: no

Unique: no

Identifier**Data Property**

Type: *String*

Multiplicity: 1

Ordered: no

Unique: no

InstrumentationActivities**Composition Property**

Type: *InstrumentationActivity*

Multiplicity: 1..*

Ordered: no

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Flow</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$	no
<i>Power</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle \rangle$	no
<i>RotationalFrequency</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{RotationalFrequencyUnit} \rangle \rangle$	no

Flow

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KilogramPerHour,
      Value = 120.0))
]
```

Power

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Kilowatt,
      Value = 920.0))
]
```

RotationalFrequency

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{RotationalFrequencyUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = ReciprocalMinute,
      Value = 150.0))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.Kneading">
  <Components property="Flow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
          </Data>
          <Data property="Value">
            <Double>120.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="Power">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
          </Data>
          <Data property="Value">
            <Double>920.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="RotationalFrequency">
```

(continues on next page)

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>PressureDifference</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PressureAbsoluteUnit>></i>	no

PressureDifference

Composition Property

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → PressureAbsoluteUnit>>*

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Bar,
      Value = 1.9))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.LimitingFlow">
  <Components property="PressureDifference">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.PressureAbsoluteUnit.Bar"/>
          </Data>
          <Data property="Value">
            <Double>1.9</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```

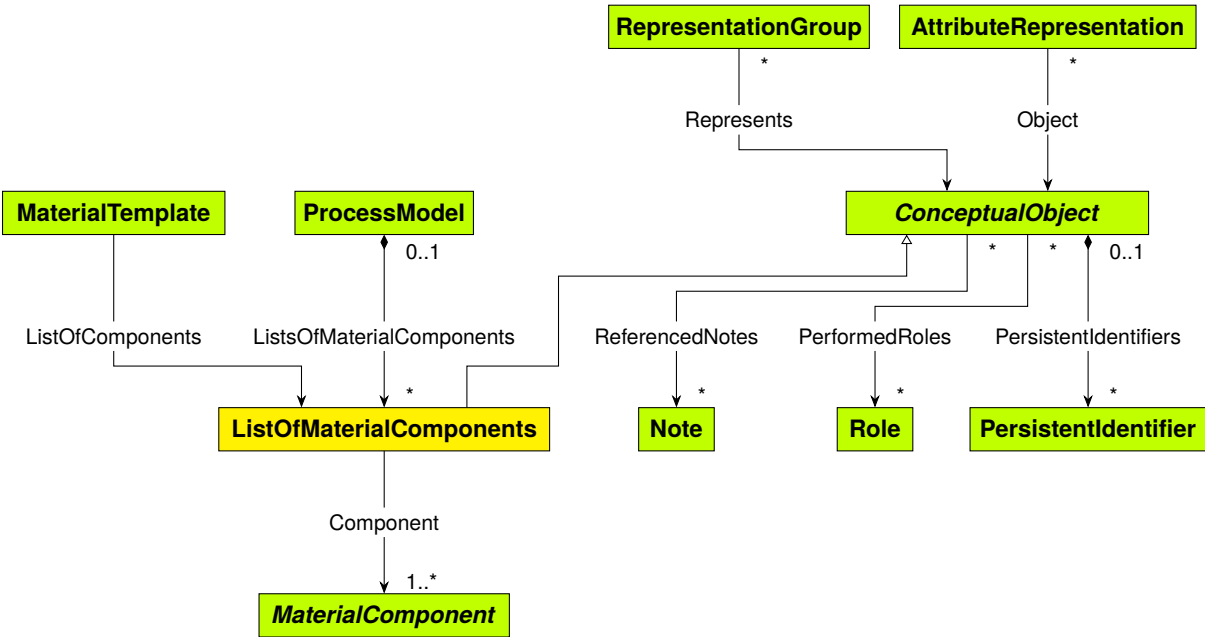

ListOfMaterialComponents

Process::Process::ListOfMaterialComponents

Overview

Concrete Class

This is an ordered list of *PureMaterialComponent* and *CustomMaterialComponent* objects. This is used in the *MaterialTemplate* to determine the structure of composition vectors in the *Stream* and *MaterialStateType* objects.



Supertypes

- *ConceptualObject*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Component</i>	1..*	<i>MaterialComponent</i>	no	no	1

Component

Reference Property

Type: *MaterialComponent*

Multiplicity: 1..*

Ordered: no

Unique: no

Opposite Multiplicity: 1

Example in DEXPI XML

```
<Object type="Process/Process.ListOfMaterialComponents"/>
```

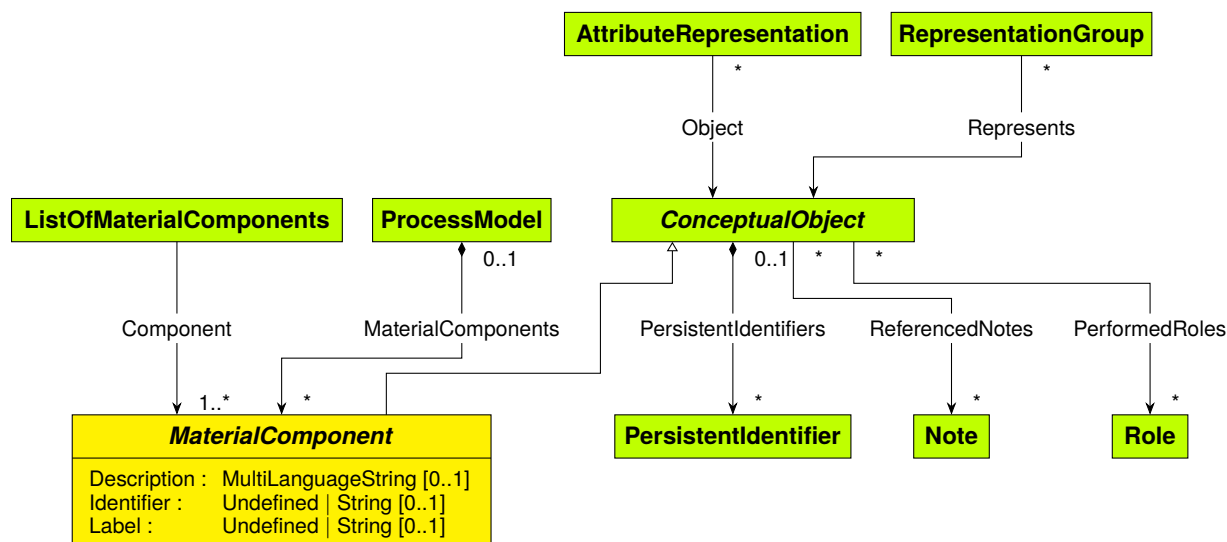
MaterialComponent

Process::Process::MaterialComponent

Overview

Abstract Class

This is the base type to chemical components in the model. It is parent class for *PureMaterialComponent* and *CustomMaterialComponent*.



Supertypes

- *ConceptualObject*

Subtypes

- *CustomMaterialComponent*
- *PureMaterialComponent*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Description</i>	0..1	<i>MultiLanguageString</i>	no	no
<i>Identifier</i>	0..1	<i>Undefined String</i>	no	no
<i>Label</i>	0..1	<i>Undefined String</i>	no	no

Description

Data Property

Descriptive text for the *MaterialComponent*.

Type: *MultiLanguageString*

Multiplicity: 0..1

Ordered: no

Unique: no

Identifier

Data Property

Unique identifier for the *MaterialComponent*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

Label

Data Property

Display label for the *MaterialComponent*.

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

MaterialPort

Process::Process::MaterialPort

Overview

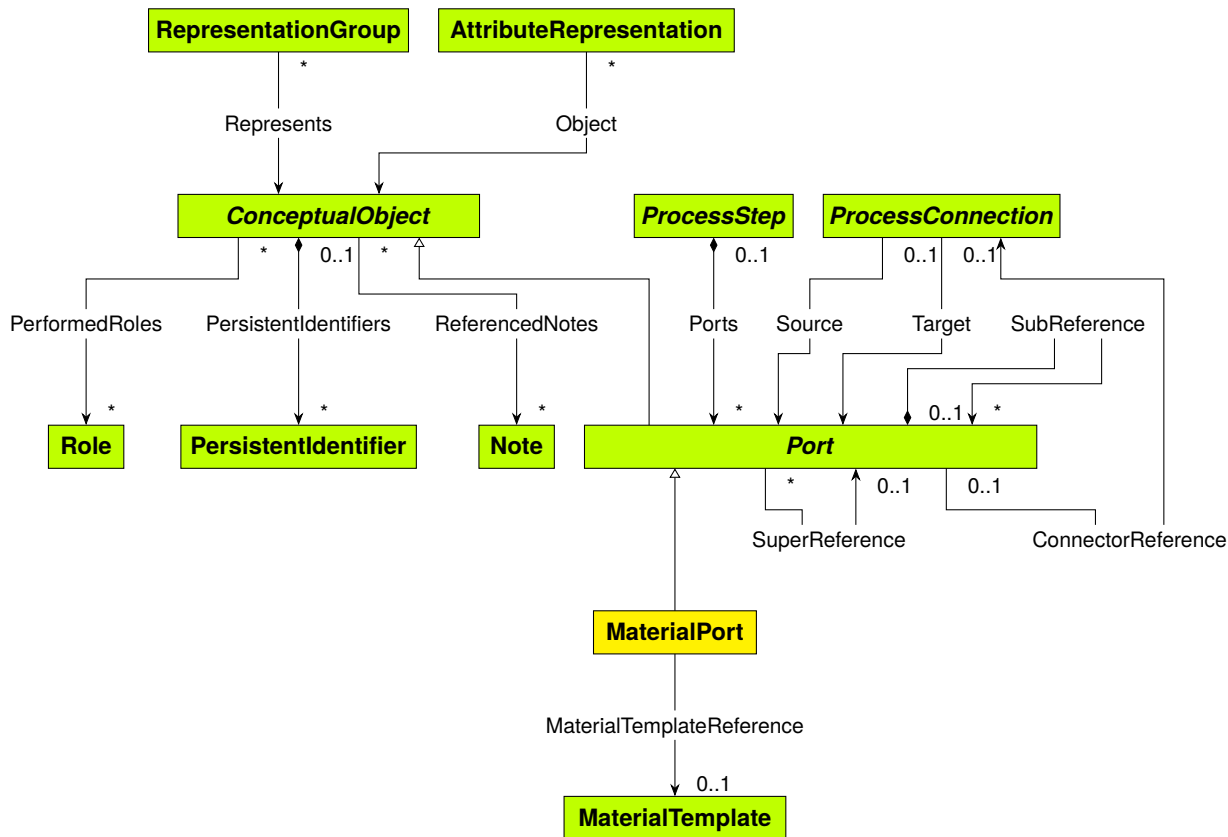
Concrete Class

A *Port* that models the transfer of material, of whatever form, into or out of a *ProcessStep* object. This type is used for flow of material: solids, liquids and gases.

Roles referenced via *PerformedRoles* should have one of the following *Names*:

- Air
- BottomProduct
- ChemicalFeed
- Discharge
- Drain
- Exhaust

- Feed
- Fuel
- HeavyLiquid
- Inlet
- LeanSolvent
- LightLiquid
- Liquid
- LiquidInlet
- LiquidOutlet
- Outlet
- Product
- Reboil
- Reflux
- Relief
- RichSolvent
- Sidedraw
- Suction
- TopProduct
- Vapour
- VapourInlet
- VapourOutlet
- Vent
- Waste



Supertypes

- *Port*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>MaterialTemplateReference</i>	0..1	<i>MaterialTemplate</i>	no	no	1

MaterialTemplateReference

Reference Property

Type: *MaterialTemplate*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 1

Example in DEXPI XML

```
<Object type="Process/Process.MaterialPort"/>
```

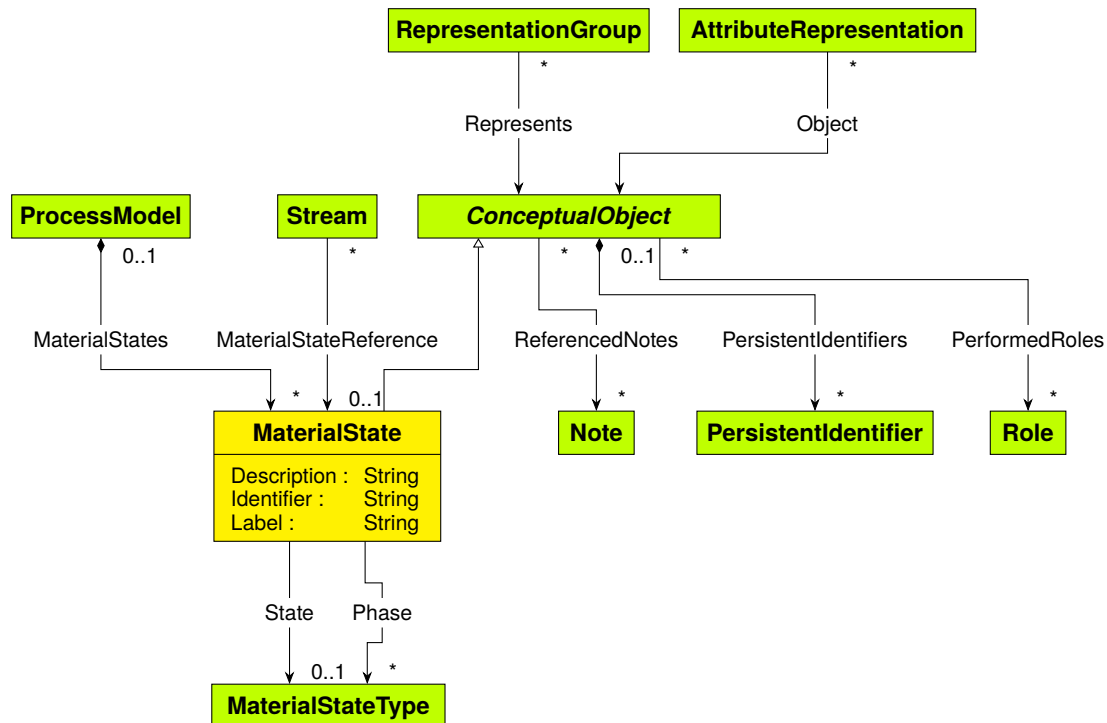
MaterialState

Process::Process::MaterialState

Overview

Concrete Class

This data structure contains the properties for a *Stream*. It contains one or more *MaterialState* objects: one for the total properties and then one for each of the phases present.



Supertypes

- *ConceptualObject*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Phase</i>	*	<i>MaterialStateType</i>	no	no	1
<i>State</i>	0..1	<i>MaterialStateType</i>	no	no	1

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Description</i>	1	<i>String</i>	no	no
<i>Identifier</i>	1	<i>String</i>	no	no
<i>Label</i>	1	<i>String</i>	no	no

Description

Data Property

Type: *String*

Multiplicity: 1

Ordered: no

Unique: no

Identifier

Data Property

Type: *String*

Multiplicity: 1

Ordered: no

Unique: no

Label

Data Property

Type: *String*

Multiplicity: 1

Ordered: no

Unique: no

Phase

Reference Property

Type: *MaterialStateType*

Multiplicity: *

Ordered: no

Unique: no

Opposite Multiplicity: 1

State

Reference Property

Type: *MaterialStateType*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: 1

Example in DEXPI XML

```
<Object type="Process/Process.MaterialState"/>
```

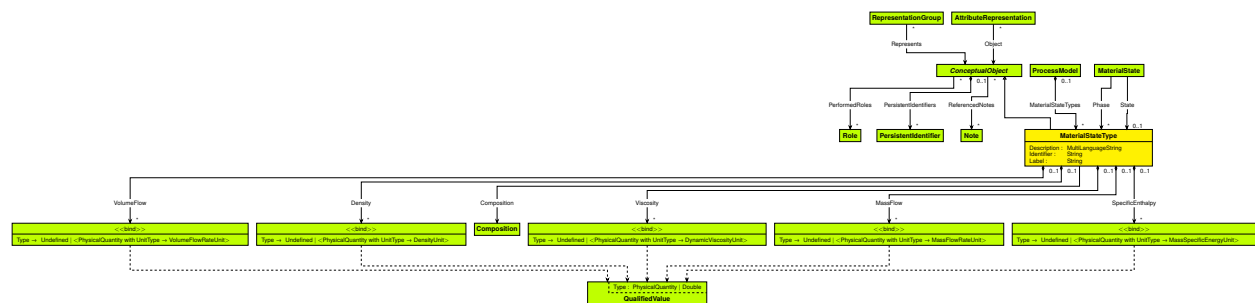
MaterialStateType

Process::Process::MaterialStateType

Overview

Concrete Class

This data structure represents the properties of a material or a single material phase. The *MaterialState* data structure for a *Stream* is built up of one or more of these data structures: one for the total properties and then one for each of the phases present.



Supertypes

- *ConceptualObject*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Density</i>	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → DensityUnit>>	no
<i>MassFlow</i>	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MassFlowRateUnit>>	no
<i>SpecificEnthalpy</i>	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MassSpecificEnergyUnit>>	no
<i>Viscosity</i>	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → DynamicViscosityUnit>>	no
<i>Volume-Flow</i>	*	<QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → VolumeFlowRateUnit>>	no

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Composition</i>	1	<i>Composition</i>	no	no	1

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Description</i>	1	<i>MultiLanguageString</i>	no	no
<i>Identifier</i>	1	<i>String</i>	no	no
<i>Label</i>	1	<i>String</i>	no	no

Composition

Reference Property

Type: *Composition*

Multiplicity: 1

Ordered: no

Unique: no

Opposite Multiplicity: 1

Density

Composition Property

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → DensityUnit>>

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = KilogramPerMetreCubed,  
      Value = 5400.0))  
]
```

Description**Data Property**

Type: *MultiLanguageString*

Multiplicity: 1

Ordered: no

Unique: no

Identifier**Data Property**

Type: *String*

Multiplicity: 1

Ordered: no

Unique: no

Label**Data Property**

Type: *String*

Multiplicity: 1

Ordered: no

Unique: no

MassFlow**Composition Property**

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KilogramPerHour,
      Value = 1820.0))
]
```

SpecificEnthalpy**Composition Property**

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → MassSpecificEnergyUnit>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KilojoulePerKilogram,
      Value = 419.0))
]
```

Viscosity**Composition Property**

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → DynamicViscosityUnit>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = PascalSecond,
      Value = 0.000281))
]
```

VolumeFlow

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VolumeFlowRateUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = LitrePerSecond,
      Value = 2.8))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.MaterialStateType">
  <Components property="Density">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.DensityUnit.
↪KilogramPerMetreCubed"/>
          </Data>
          <Data property="Value">
            <Double>5400.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="MassFlow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
          </Data>
          <Data property="Value">
            <Double>1820.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
```

(continues on next page)

(continued from previous page)

```

<Components property="SpecificEnthalpy">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.MassSpecificEnergyUnit.
↪KilojoulePerKilogram"/>
        </Data>
        <Data property="Value">
          <Double>419.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
<Components property="Viscosity">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.DynamicViscosityUnit.
↪PascalSecond"/>
        </Data>
        <Data property="Value">
          <Double>0.000281</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
<Components property="VolumeFlow">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.
↪LitrePerSecond"/>
        </Data>
        <Data property="Value">
          <Double>2.8</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
</Object>

```

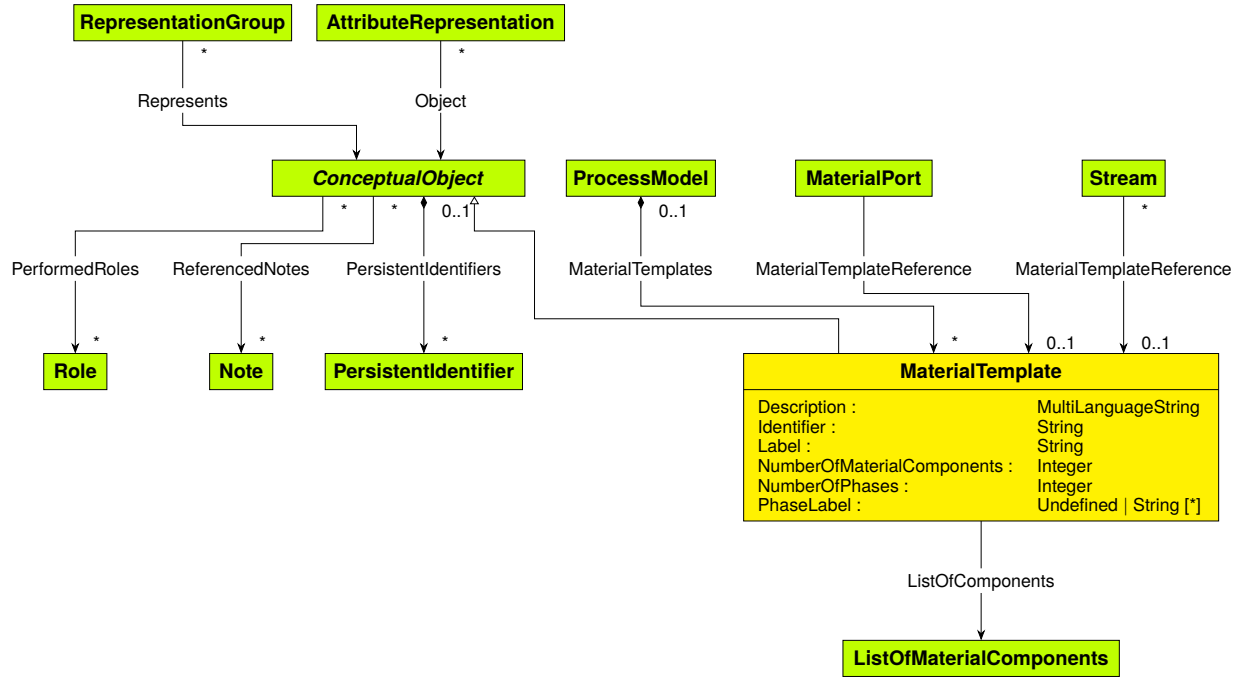
MaterialTemplate

Process::Process::MaterialTemplate

Overview

Concrete Class

This data structure defines the structure of *Stream* objects that correspond to a specific type of material. This structure contains information about the thermodynamic model used to represent this type of material. It contains the *ListOfMaterialComponents*: pure and pseudo components used to represent the system. It also determines the legend for a stream table when the model is used to represent a Process Flow Diagram.



Supertypes

- *ConceptualObject*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>ListOfComponents</i>	1	<i>ListOfMaterialComponents</i>	no	no	1

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Description</i>	1	<i>MultiLanguageString</i>	no	no
<i>Identifier</i>	1	<i>String</i>	no	no
<i>Label</i>	1	<i>String</i>	no	no
<i>NumberOfMaterialComponents</i>	1	<i>Integer</i>	no	no
<i>NumberOfPhases</i>	1	<i>Integer</i>	no	no
<i>PhaseLabel</i>	*	<i>Undefined String</i>	no	no

Description

Data Property

Type: *MultiLanguageString*

Multiplicity: 1

Ordered: no

Unique: no

Identifier

Data Property

Type: *String*

Multiplicity: 1

Ordered: no

Unique: no

Label

Data Property

Type: *String*

Multiplicity: 1

Ordered: no

Unique: no

ListOfComponents

Reference Property

Type: *ListOfMaterialComponents*

Multiplicity: 1

Ordered: no

Unique: no

Opposite Multiplicity: 1

NumberOfMaterialComponents

Data Property

Type: *Integer*

Multiplicity: 1

Ordered: no

Unique: no

NumberOfPhases

Data Property

Type: *Integer*

Multiplicity: 1

Ordered: no

Unique: no

PhaseLabel

Data Property

Type: *Undefined* | *String*

Multiplicity: *

Ordered: no

Unique: no

Example in DEXPI XML

```
<Object type="Process/Process.MaterialTemplate"/>
```

MeasuringProcessVariable

Process::Process::MeasuringProcessVariable

Overview

Concrete Class

This *InstrumentationActivity* measures a process variable and supplies its value at an outlet *InformationPort*. The measured variable is identified by reference to a parameter in any process step or port.



- ## Members

Name	Multiplicity	Type	Ordered
<i>InputValue</i>	*	$\langle \textit{QualifiedValue} \text{ with } \textit{Type} \rightarrow \textit{Undefined} \mid \textit{Double} \rangle$	no
<i>MeasuredVariable</i>	*	$\langle \textit{QualifiedValue} \text{ with } \textit{Type} \rightarrow \textit{Undefined} \mid \textit{Double} \rangle$	no
<i>OutputValue</i>	*	$\langle \textit{QualifiedValue} \text{ with } \textit{Type} \rightarrow \textit{Undefined} \mid \textit{Double} \rangle$	no

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>ConnectionReference</i>	0..1	<i>ProcessConnection</i>	no	no	*
<i>MeasuredVariableReference</i>	0..1	<i><QualifiedValue with Type → Undefined PhysicalQuantity></i>	no	no	*
<i>ProcessStepDetailReference</i>	0..1	<i>ProcessStepDetail</i>	no	no	*
<i>ProcessStepReference</i>	0..1	<i>ProcessStep</i>	no	no	*

ConnectionReference

Reference Property

Type: *ProcessConnection*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

InputValue

Composition Property

Type: $\langle \textit{QualifiedValue} \text{ with } Type \rightarrow Undefined \mid Double \rangle$

Multiplicity: *

Ordered: no

MeasuredVariable

Composition Property

Type: $\langle \textit{QualifiedValue} \text{ with } Type \rightarrow Undefined \mid Double \rangle$

Multiplicity: *

Ordered: no

MeasuredVariableReference

Reference Property

Type: $\langle \textit{QualifiedValue} \text{ with } Type \rightarrow Undefined \mid \textit{PhysicalQuantity} \rangle$

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

OutputValue

Composition Property

Type: $\langle \textit{QualifiedValue} \text{ with } Type \rightarrow Undefined \mid Double \rangle$

Multiplicity: *

Ordered: no

ProcessStepDetailReference

Reference Property

Type: *ProcessStepDetail*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

ProcessStepReference

Reference Property

Type: *ProcessStep*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

Example in DEXPI XML

```
<Object type="Process/Process.MeasuringProcessVariable"/>
```

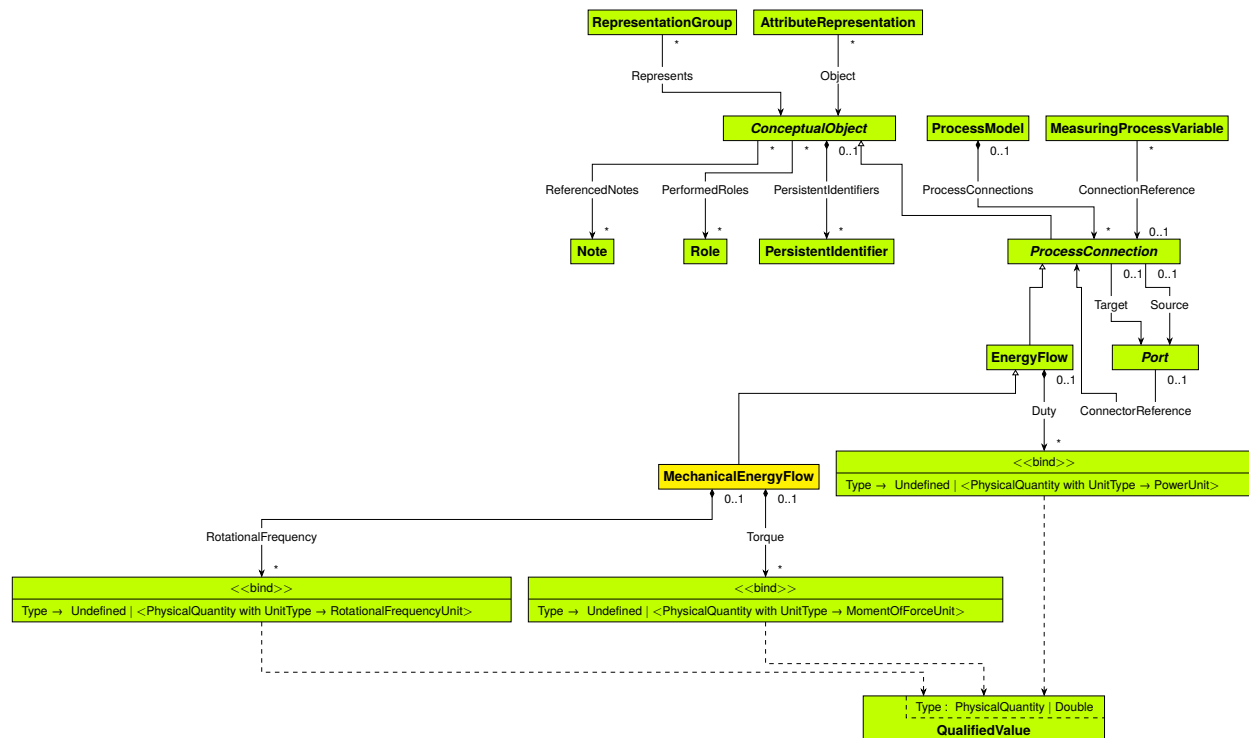
MechanicalEnergyFlow

Process::Process::MechanicalEnergyFlow

Overview

Concrete Class

A *ProcessConnection* that defines the flow of mechanical energy two instances of *MechanicalEnergyPort*.



Supertypes

- *EnergyFlow*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>RotationalFrequency</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{RotationalFrequencyUnit} \rangle \rangle$	no
<i>Torque</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MomentOfForceUnit} \rangle \rangle$	no

Rotational Frequency

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{RotationalFrequencyUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = ReciprocalMinute,
      Value = 900.0))
]
```

Torque**Composition Property**

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MomentOfForceUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = NewtonMetre,
      Value = 120.0))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.MechanicalEnergyFlow">
  <Components property="RotationalFrequency">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ReciprocalMinute"/>
          </Data>
          <Data property="Value">
            <Double>900.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="Torque">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
```

(continues on next page)

(continued from previous page)

```

<Data property="Unit">
  <DataReference data="Core/PhysicalQuantities.MomentOfForceUnit.NewtonMetre"/>
</Data>
<Data property="Value">
  <Double>120.0</Double>
</Data>
</AggregatedDataValue>
</Data>
</Object>
</Components>
</Object>

```

MechanicalEnergyPort

Process::Process::MechanicalEnergyPort

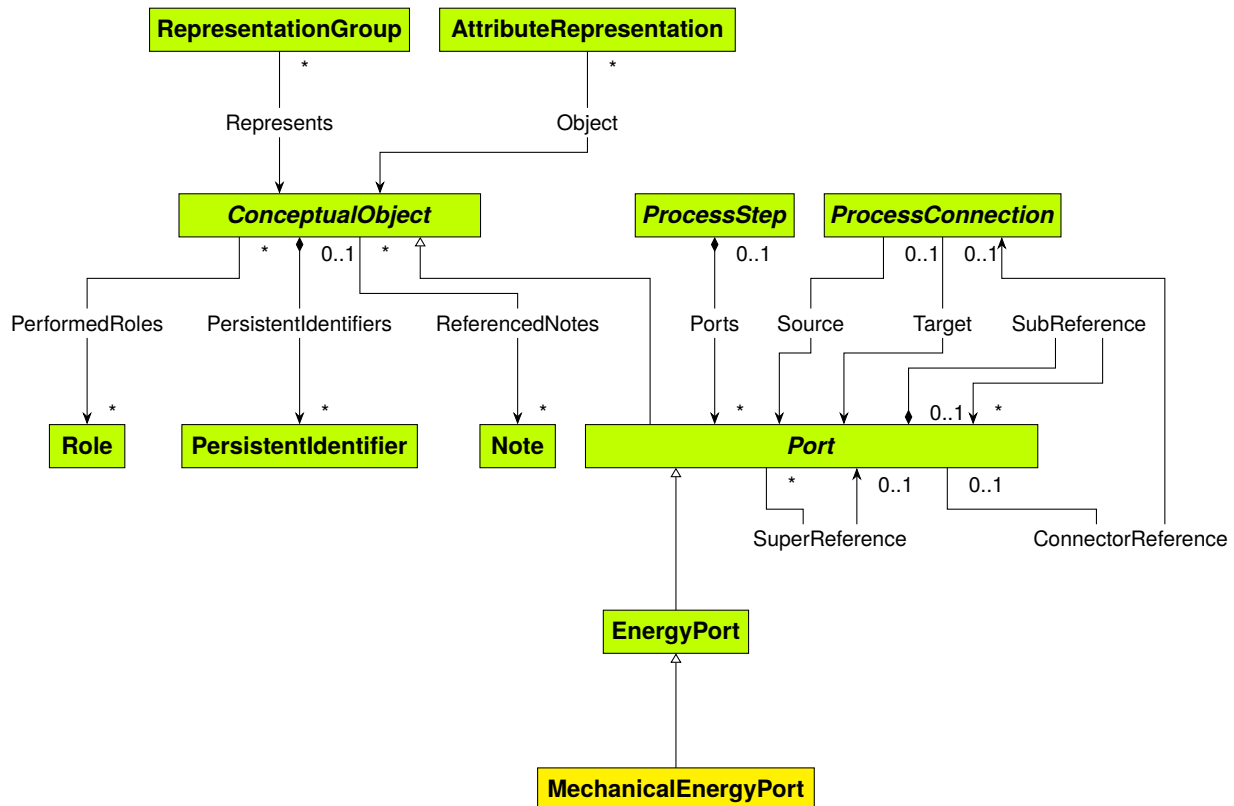
Overview

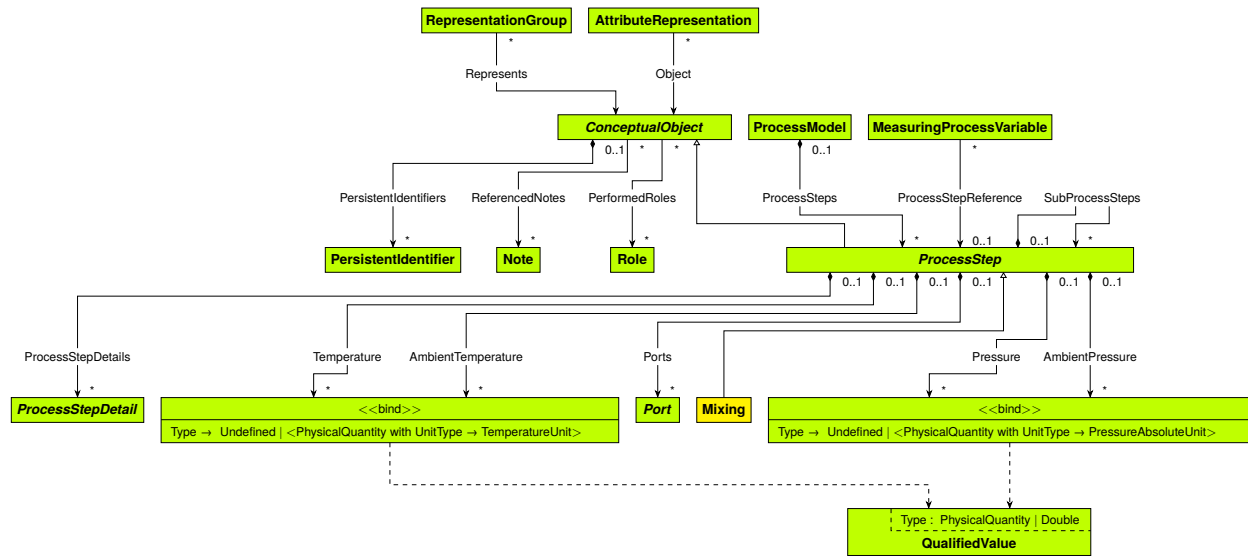
Concrete Class

A *Port* that models the transfer of mechanical energy into or out of a *ProcessStep* object.

Roles referenced via *PerformedRoles* should have one of the following *Names*:

- Input
- Output





Supertypes

- *ProcessStep*

Subtypes

- *Humidifying*
- *Kneading*
- *MixingSimple*
- *RotaryMixing*
- *StaticMixing*

Example in DEXPI XML

```
<Object type="Process/Process.Mixing"/>
```

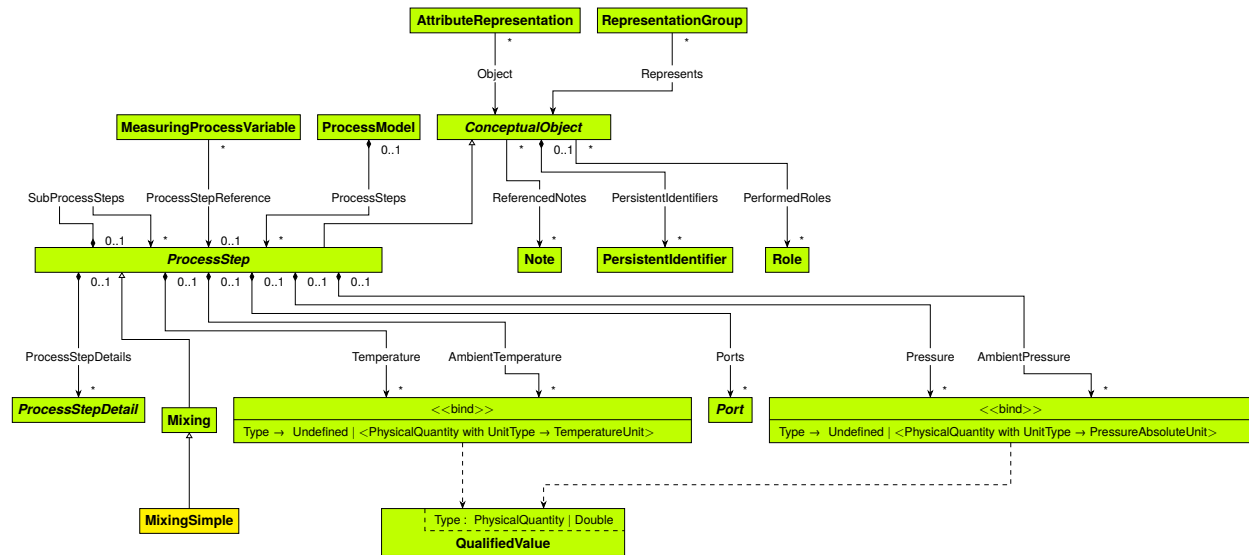
MixingSimple

Process::Process::MixingSimple

Overview

Concrete Class

This is a *Mixing* process that simply combines one or more inlet flows into an outlet flow. This process step is used to represent converging branches of flows in a PFD or BFD. It is the counterpart of *Splitting*.



Supertypes

- *Mixing*

Example in DEXPI XML

```
<Object type="Process/Process.MixingSimple"/>
```

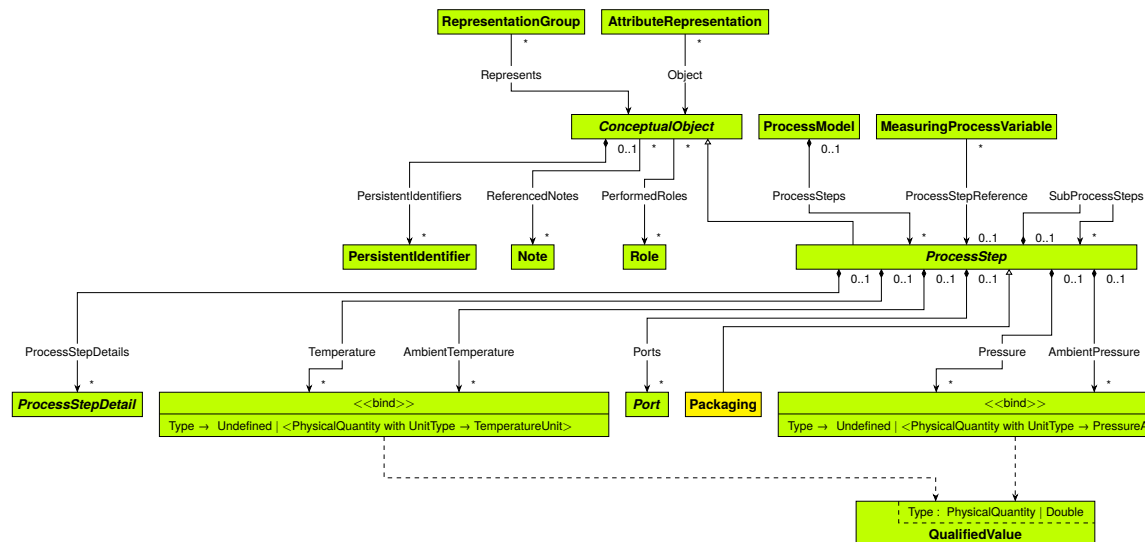
Packaging

Process::Process::Packaging

Overview

Concrete Class

This process takes a flow of material and processes it into packages that allow transport and distribution.



Supertypes

- *ProcessStep*

Example in DEXPI XML

```
<Object type="Process/Process.Packaging"/>
```

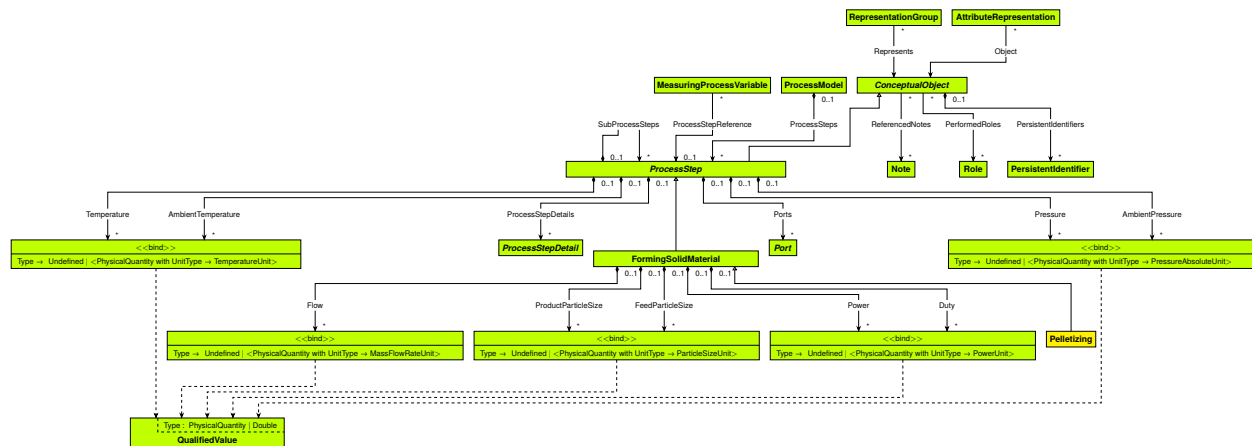
Pelletizing

Process::Process::Pelletizing

Overview

Concrete Class

This *FormingSolidMaterial* process takes solid material and forms it into, pellets, small particles of uniform size and shape.



Supertypes

- *FormingSolidMaterial*

Example in DEXPI XML

```
<Object type="Process/Process.Pelletizing"/>
```

Port

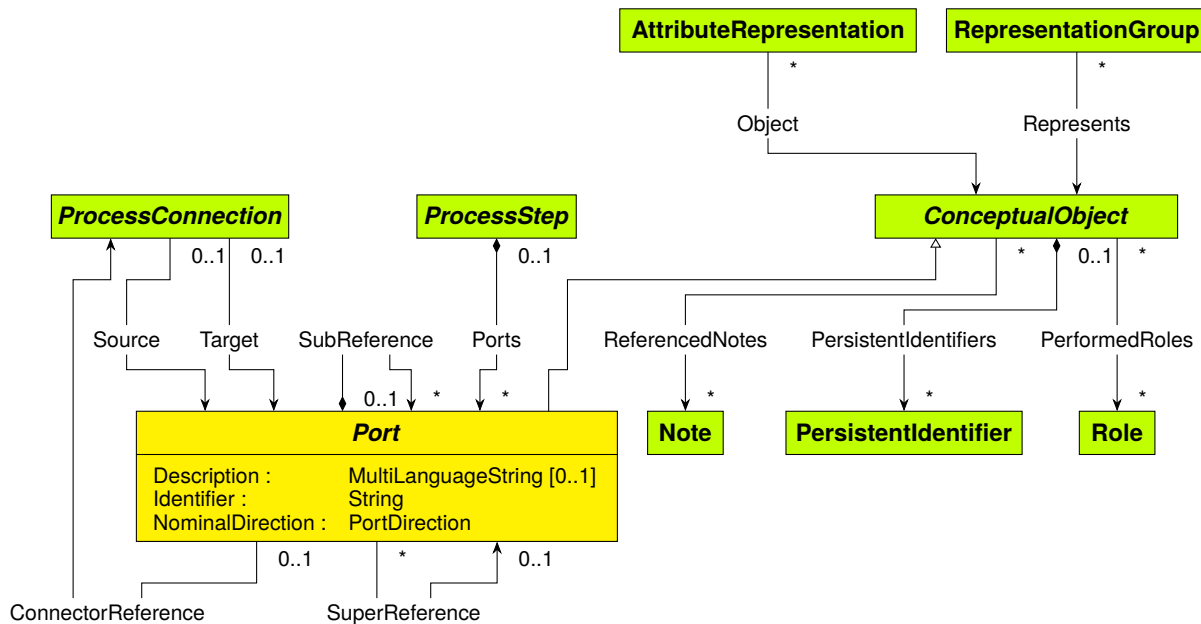
Process::Process::Port

Overview

Abstract Class

A port represents a transfer of material, energy or information into or out of a process step. It provides the anchor point for connecting two process steps to each other. The term is derived from SysML and IDEF0 representations of processes.

Note that a port in DEXPI Process is a conceptual object. It represents a contract regarding the material, energy or information coming into or out of a process step.



Supertypes

- *ConceptualObject*

Subtypes

- *EnergyPort*
- *InformationPort*
- *MaterialPort*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>SubReference</i>	*	<i>Port</i>	no

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>ConnectorReference</i>	1	<i>ProcessConnection</i>	no	no	0..1
<i>SuperReference</i>	0..1	<i>Port</i>	no	no	*

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Description</i>	0..1	<i>MultiLanguageString</i>	no	no
<i>Identifier</i>	1	<i>String</i>	no	no
<i>NominalDirection</i>	1	<i>PortDirection</i>	no	no

ConnectorReference

Reference Property

Type: *ProcessConnection*

Multiplicity: 1

Ordered: no

Unique: no

Opposite Multiplicity: 0..1

Description

Data Property

Type: *MultiLanguageString*

Multiplicity: 0..1

Ordered: no

Unique: no

Identifier

Data Property

Type: *String*

Multiplicity: 1

Ordered: no

Unique: no

NominalDirection

Data Property

Type: *PortDirection*

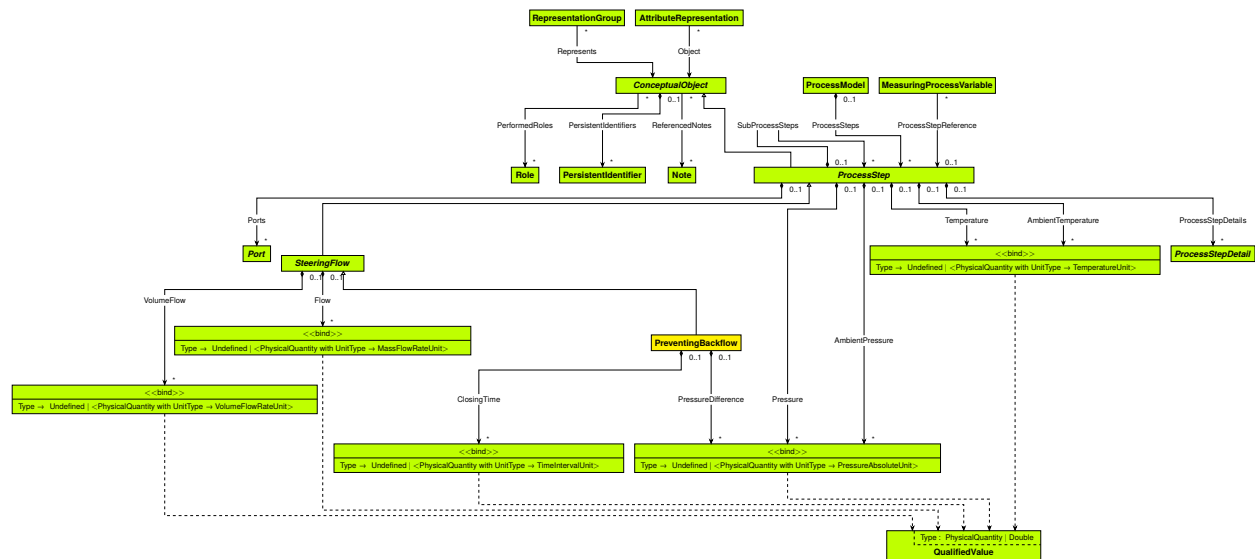
Multiplicity: 1

Ordered: no

Unique: no

SubReference**Composition Property****Type:** *Port***Multiplicity:** ***Ordered:** no**SuperReference****Reference Property****Type:** *Port***Multiplicity:** 0..1**Ordered:** no**Unique:** no**Opposite Multiplicity:** ***PreventingBackflow***Process::Process::PreventingBackflow***Overview****Concrete Class**

This *SteeringFlow* process is used to ensure that a flow of material cannot be reversed. This process is used in safety design and is realized by a check valve or non-return valve.

**Supertypes**

- *SteeringFlow*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>ClosingTime</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{TimeIntervalUnit} \rangle \rangle$	no
<i>PressureDifference</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PressureAbsoluteUnit} \rangle \rangle$	no

ClosingTime

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{TimeIntervalUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Millisecond,
      Value = 800.0))
]
```

PressureDifference

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PressureAbsoluteUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Process/Process.PreventingBackflow">
  <Components property="ClosingTime">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.TimeIntervalUnit.Millisecond"/>
          </Data>
          <Data property="Value">
            <Double>800.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```

(continues on next page)

(continued from previous page)

```
</Data>
</AggregatedDataValue>
</Data>
</Object>
</Components>
</Object>
```

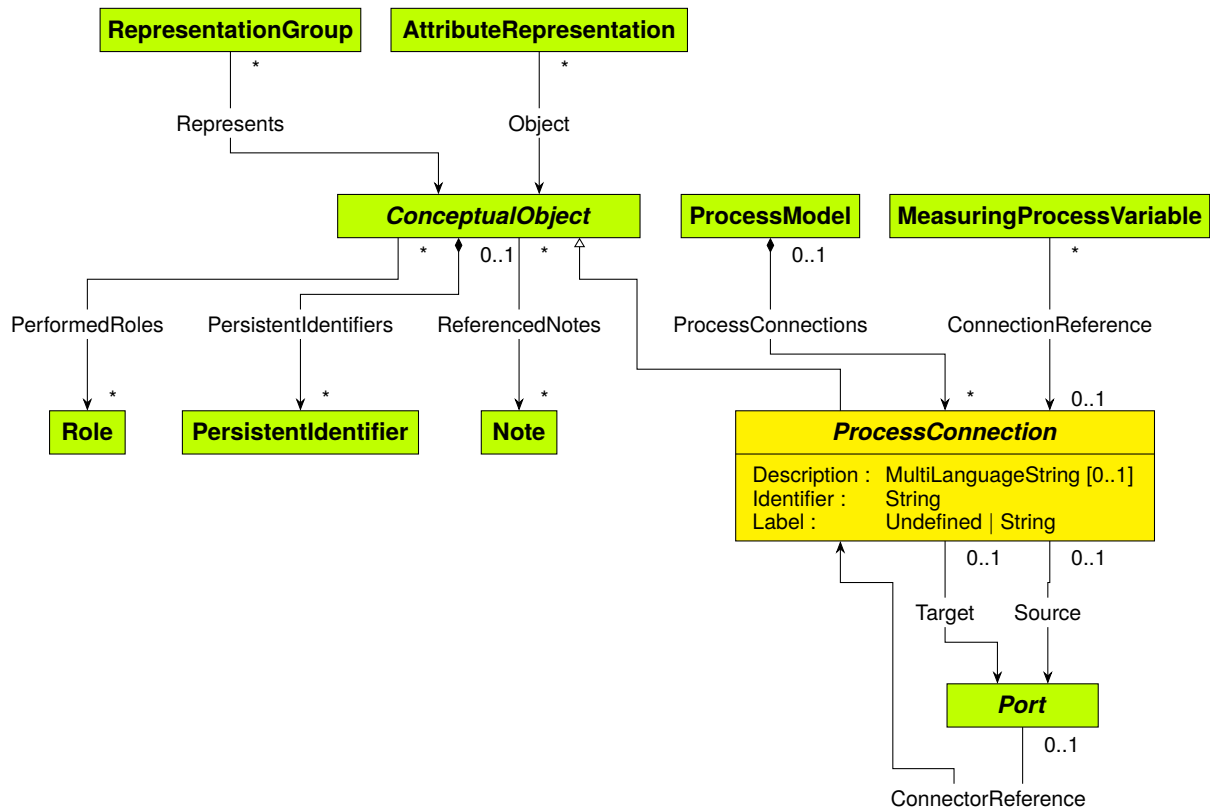
ProcessConnection

Process::Process::ProcessConnection

Overview

Abstract Class

This data structure is the abstract parent for all connections between blocks in the DEXPI Process model.



Supertypes

- *ConceptualObject*

Subtypes

- *EnergyFlow*
- *InformationFlow*
- *Stream*

Members

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>Source</i>	1	<i>Port</i>	no	no	0..1
<i>Target</i>	1	<i>Port</i>	no	no	0..1

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Description</i>	0..1	<i>MultiLanguageString</i>	no	no
<i>Identifier</i>	1	<i>String</i>	no	no
<i>Label</i>	1	<i>Undefined String</i>	no	no

Description

Data Property

Type: *MultiLanguageString*

Multiplicity: 0..1

Ordered: no

Unique: no

Identifier

Data Property

Type: *String*

Multiplicity: 1

Ordered: no

Unique: no

Label

Data Property

Type: *Undefined | String*

Multiplicity: 1

Ordered: no

Unique: no

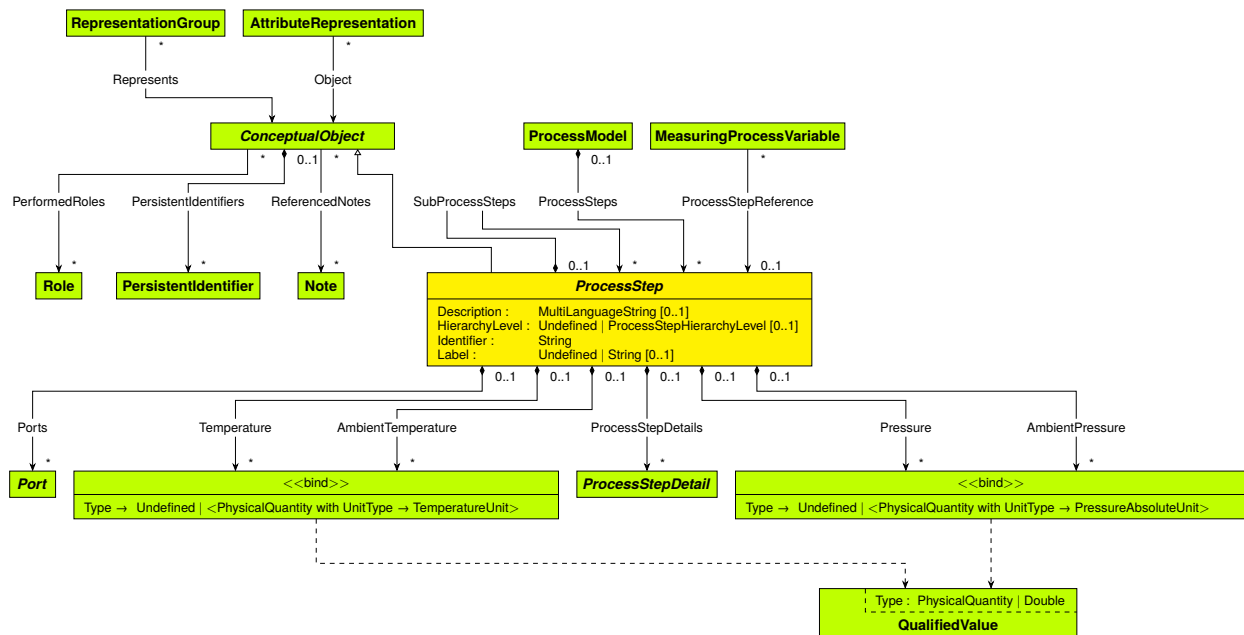
Source

Reference Property

Type: *Port*

Multiplicity: 1**Ordered:** no**Unique:** no**Opposite Multiplicity:** 0..1**Target****Reference Property****Type:** *Port***Multiplicity:** 1**Ordered:** no**Unique:** no**Opposite Multiplicity:** 0..1**ProcessStep***Process::Process::ProcessStep***Overview****Abstract Class**

The *ProcessStep* type is the parent type for all process steps in the model.

**Supertypes**

- *ConceptualObject*

Subtypes

- *Emitting*
- *ExchangingThermalEnergy*
- *Flaring*
- *FormingSolidMaterial*
- *GeneratingFlow*
- *IncreasingParticleSize*
- *Mixing*
- *Packaging*
- *ReactingChemicals*
- *ReducingParticleSize*
- *RemovingThermalEnergy*
- *Separating*
- *Sink*
- *Source*
- *Splitting*
- *SteeringFlow*
- *StoringEnergy*
- *StoringMaterial*
- *SupplyingElectricalEnergy*
- *SupplyingFluids*
- *SupplyingMechanicalEnergy*
- *SupplyingSolids*
- *SupplyingThermalEnergy*
- *TransportingElectricalEnergy*
- *TransportingFluids*
- *TransportingSolids*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>AmbientPressure</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PressureAbsoluteUnit>></i>	no
<i>AmbientTemperature</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → TemperatureUnit>></i>	no
<i>Ports</i>	*	<i>Port</i>	no
<i>Pressure</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PressureAbsoluteUnit>></i>	no
<i>ProcessStepDetails</i>	*	<i>ProcessStepDetail</i>	no
<i>SubProcessSteps</i>	*	<i>ProcessStep</i>	no
<i>Temperature</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → TemperatureUnit>></i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Description</i>	0..1	<i>MultiLanguageString</i>	no	no
<i>HierarchyLevel</i>	0..1	<i>Undefined ProcessStepHierarchyLevel</i>	no	no
<i>Identifier</i>	1	<i>String</i>	no	no
<i>Label</i>	0..1	<i>Undefined String</i>	no	no

AmbientPressure

Composition Property

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → PressureAbsoluteUnit>>*

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Bar,
      Value = 1.01325))
]
```

AmbientTemperature**Composition Property**

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow TemperatureUnit \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = DegreeCelsius,  
      Value = 20.0))  
]
```

Description**Data Property**

Type: *MultiLanguageString*

Multiplicity: 0..1

Ordered: no

Unique: no

HierarchyLevel**Data Property**

Type: $Undefined \mid ProcessStepHierarchyLevel$

Multiplicity: 0..1

Ordered: no

Unique: no

Example

UnitOperation

Identifier**Data Property**

Type: *String*

Multiplicity: 1

Ordered: no

Unique: no

Label**Data Property****Type:** *Undefined* | *String***Multiplicity:** 0..1**Ordered:** no**Unique:** no**Ports****Composition Property****Type:** *Port***Multiplicity:** ***Ordered:** no**Pressure****Composition Property****Type:** $\langle \textit{QualifiedValue}$ with $\textit{Type} \rightarrow \textit{Undefined}$ | $\langle \textit{PhysicalQuantity}$ with $\textit{UnitType} \rightarrow \textit{PressureAbsoluteUnit} \rangle \rangle$ **Multiplicity:** ***Ordered:** no**Example**

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Bar,
      Value = 2.2))
]
```

ProcessStepDetails**Composition Property****Type:** *ProcessStepDetail***Multiplicity:** ***Ordered:** no**SubProcessSteps****Composition Property**The *ProcessSteps* of the *ProcessStep*.**Type:** *ProcessStep*

Multiplicity: *

Ordered: no

Temperature

Composition Property

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow TemperatureUnit \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = DegreeCelsius,  
      Value = 76.0))  
]
```

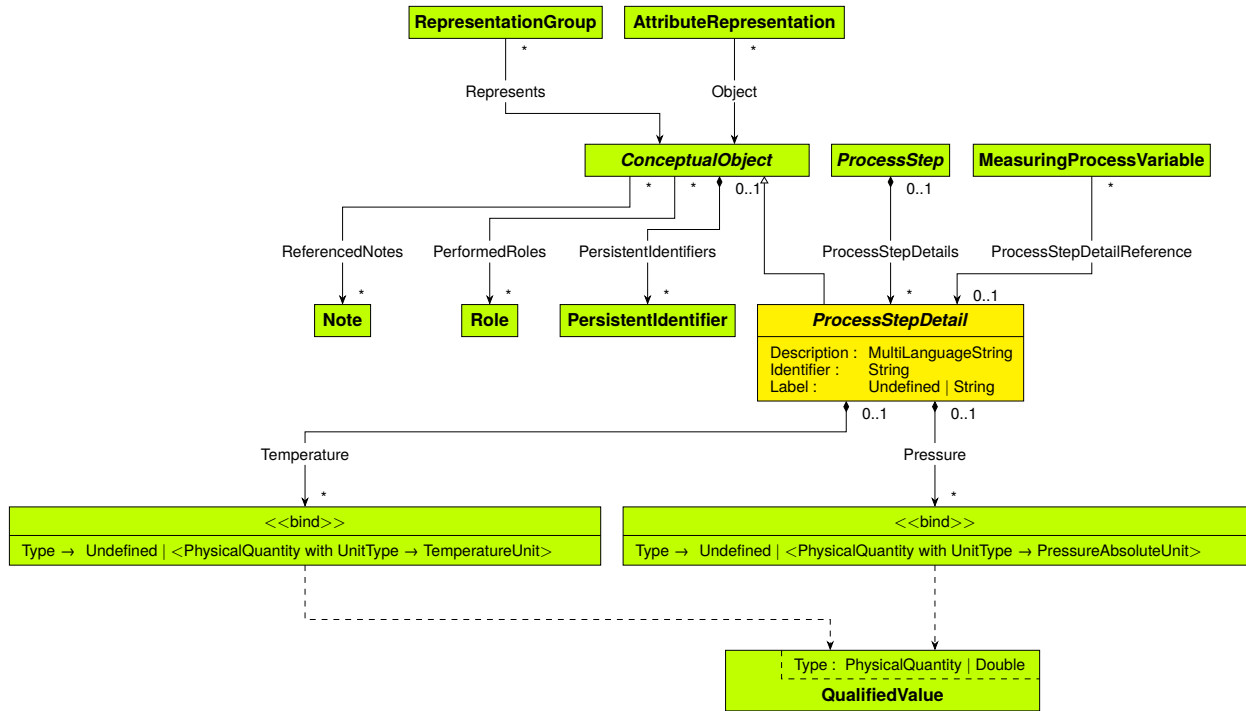
ProcessStepDetail

Process::Process::ProcessStepDetail

Overview

Abstract Class

This type is the parent type for processes that cannot be run independently of another process.



Supertypes

- *ConceptualObject*

Subtypes

- *Agitating*
- *ContactingInPacking*
- *ContactingOnTray*
- *SupplyingThermalEnergyWithBurner*

Members

Composition Properties

Name	Multi- plicity	Type	Or- dered
<i>Pressure</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PressureAbsoluteUnit>></i>	no
<i>Temperature</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → TemperatureUnit>></i>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Description</i>	1	<i>MultiLanguageString</i>	no	no
<i>Identifier</i>	1	<i>String</i>	no	no
<i>Label</i>	1	<i>Undefined</i> <i>String</i>	no	no

Description

Data Property

Type: *MultiLanguageString*

Multiplicity: 1

Ordered: no

Unique: no

Identifier

Data Property

Type: *String*

Multiplicity: 1

Ordered: no

Unique: no

Label

Data Property

Type: *Undefined* | *String*

Multiplicity: 1

Ordered: no

Unique: no

Pressure

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PressureAbsoluteUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  

```



```
[
    QualifiedValue(
        Value = PhysicalQuantity(
            Unit = DegreeCelsius,
            Value = 76.0))
]
```

```

classDiagram
    class RepresentationGroup
    class AttributeRepresentation
    class ConceptualObject
    class ProcessModel
    class MeasuringProcessVariable
    class ProcessStep
    class Role
    class PersistentIdentifier
    class Note
    class ProcessStepDetail
    class GeneratingFlow
    class Port
    class Pumping
    class QualifiedValue

    RepresentationGroup --> ConceptualObject : Represents
    AttributeRepresentation --> ConceptualObject : Represents
    ConceptualObject --> ProcessModel : 0..1
    ConceptualObject --> Role : PerformedRoles
    ConceptualObject --> PersistentIdentifier : PersistentIdentifiers
    ConceptualObject --> Note : ReferenceNotes
    ProcessModel --> ProcessStep : ProcessSteps
    ProcessModel --> ProcessStep : SubProcessSteps
    ProcessModel --> ProcessStep : ProcessStepReference
    ProcessStep --> ProcessStepDetail : ProcessStepDetails
    ProcessStep --> Port : Ports
    ProcessStep --> GeneratingFlow : Flow
    ProcessStep --> Port : PressureDifference
    ProcessStep --> Port : Pressure
    ProcessStep --> Temperature : Temperature
    ProcessStep --> AmbientTemperature : AmbientTemperature
    Pumping --> GeneratingFlow : VolumeFlow
    Pumping --> Head : Head
    Pumping --> QualifiedValue : 0..1
    Pumping --> QualifiedValue : 0..1
    GeneratingFlow --> QualifiedValue : 0..1
    Port --> QualifiedValue : 0..1
    Temperature --> QualifiedValue : 0..1
    AmbientTemperature --> QualifiedValue : 0..1
    
```

Supertypes

- *GeneratingFlow*

Members

Composition Properties

Name	Multi- plicity	Type	Or- dered
<i>Head</i>	*	< <i>QualifiedValue</i> with <i>Type</i> → <i>Undefined</i> < <i>PhysicalQuantity</i> with <i>UnitType</i> → <i>LengthUnit</i> >>	no
<i>Volume- Flow</i>	*	< <i>QualifiedValue</i> with <i>Type</i> → <i>Undefined</i> < <i>PhysicalQuantity</i> with <i>UnitType</i> → <i>VolumeFlowRateUnit</i> >>	no

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Method</i>	0..1	<i>Undefined</i> <i>PumpingMethod</i>	no	no

Head

Composition Property

Type: <*QualifiedValue* with *Type* → *Undefined* | <*PhysicalQuantity* with *UnitType* → *LengthUnit*>>

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Metre,  
      Value = 7.6))  
]
```

Method

Data Property

Type: *Undefined* | *PumpingMethod*

Multiplicity: 0..1

Ordered: no

Unique: no

VolumeFlow

Composition Property

Type: $\langle \text{QualifiedValue with } Type \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{VolumeFlowRateUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = LitrePerSecond,
      Value = 12.0))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.Pumping">
  <Components property="Head">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.LengthUnit.Metre"/>
          </Data>
          <Data property="Value">
            <Double>7.6</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="VolumeFlow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.
↪LitrePerSecond"/>
          </Data>
          <Data property="Value">
            <Double>12.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```

PureMaterialComponent

Process::Process::PureMaterialComponent

Overview

Concrete Class

A *PureMaterialComponent* represents a *MaterialComponent* that is modelled as a pure chemical compound with defined composition and structure. This is used in the *MaterialTemplate* to document the composition vector and define properties for use in the process.

Supertypes

- *MaterialComponent*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>ChEBI_identifier</i>	0..1	<i>Undefined</i> <i>String</i>	no	no
<i>IUPAC_identifier</i>	0..1	<i>Undefined</i> <i>String</i>	no	no

ChEBI_identifier

Data Property

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

IUPAC_identifier

Data Property

Type: *Undefined* | *String*

Multiplicity: 0..1

Ordered: no

Unique: no

Example in DEXPI XML

```
<Object type="Process/Process.PureMaterialComponent"/>
```


Method**Data Property****Type:** *ReactionProcessType***Multiplicity:** 1**Ordered:** no**Unique:** no**Volume****Composition Property****Type:** $\langle \text{QualifiedValue with } Type \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{VolumeUnit} \rangle \rangle$ **Multiplicity:** ***Ordered:** no**Example**

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = CentimetreCubed,  
      Value = 4200.0))  
]
```

Example in DEXPI XML

```
<Object type="Process/Process.ReactingChemicals">  
  <Components property="Volume">  
    <Object type="Core/QualifiedValue">  
      <Data property="Value">  
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
          <Data property="Unit">  
            <DataReference data="Core/PhysicalQuantities.VolumeUnit.CentimetreCubed"/>  
          </Data>  
          <Data property="Value">  
            <Double>4200.0</Double>  
          </Data>  
        </AggregatedDataValue>  
      </Data>  
    </Object>  
  </Components>  
</Object>
```


FeedParticleSize**Composition Property**

Characteristic particle size for the feed.

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow ParticleSizeUnit \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Millimetre,  
      Value = 80.0))  
]
```

Flow**Composition Property**

Mass flow rate through process.

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow MassFlowRateUnit \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = KilogramPerHour,  
      Value = 90.0))  
]
```

Power**Composition Property**

Power demand.

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow PowerUnit \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Kilowatt,
      Value = 80.0))
]
```

ProductParticleSize**Composition Property**

Characteristic particle size for the product.

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → ParticleSizeUnit>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Millimetre,
      Value = 6.0))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.ReducingParticleSize">
  <Components property="FeedParticleSize">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.ParticleSizeUnit.Millimetre"/>
          </Data>
          <Data property="Value">
            <Double>80.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="Flow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
```

(continues on next page)

(continued from previous page)

```

    <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
      <Data property="Unit">
        <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
      </Data>
      <Data property="Value">
        <Double>90.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
</Components>
<Components property="Power">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
        </Data>
        <Data property="Value">
          <Double>80.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
<Components property="ProductParticleSize">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.ParticleSizeUnit.Millimetre"/>
        </Data>
        <Data property="Value">
          <Double>6.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
</Object>

```

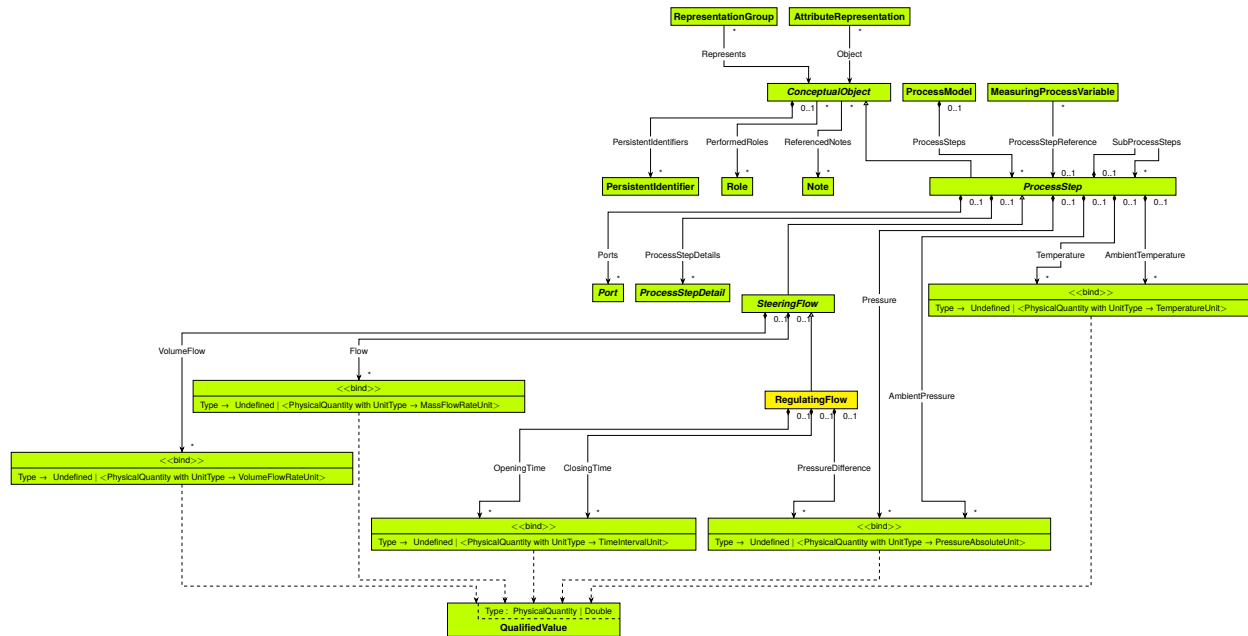
RegulatingFlow

Process::Process::RegulatingFlow

Overview

Concrete Class

A *SteeringFlow* process where a flow of material is manipulated over a continuous range of values. The desired flow rate is supplied in the XG1 *InformationPort*.



Supertypes

- *SteeringFlow*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>ClosingTime</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → TimeIntervalUnit>></i>	no
<i>OpeningTime</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → TimeIntervalUnit>></i>	no
<i>PressureDifference</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PressureAbsoluteUnit>></i>	no

ClosingTime

Composition Property

Closing time.

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → TimeIntervalUnit>>*

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Millisecond,  
      Value = 750.0))  
]
```

OpeningTime**Composition Property**

Opening time.

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → TimeIntervalUnit>>

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Millisecond,  
      Value = 1200.0))  
]
```

PressureDifference**Composition Property**

Pressure difference across process.

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → PressureAbsoluteUnit>>

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Bar,  
      Value = 2.5))  
]
```

Example in DEXPI XML

```
<Object type="Process/Process.RegulatingFlow">
  <Components property="ClosingTime">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.TimeIntervalUnit.Millisecond"/>
          </Data>
          <Data property="Value">
            <Double>750.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="OpeningTime">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.TimeIntervalUnit.Millisecond"/>
          </Data>
          <Data property="Value">
            <Double>1200.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="PressureDifference">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.PressureAbsoluteUnit.Bar"/>
          </Data>
          <Data property="Value">
            <Double>2.5</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```

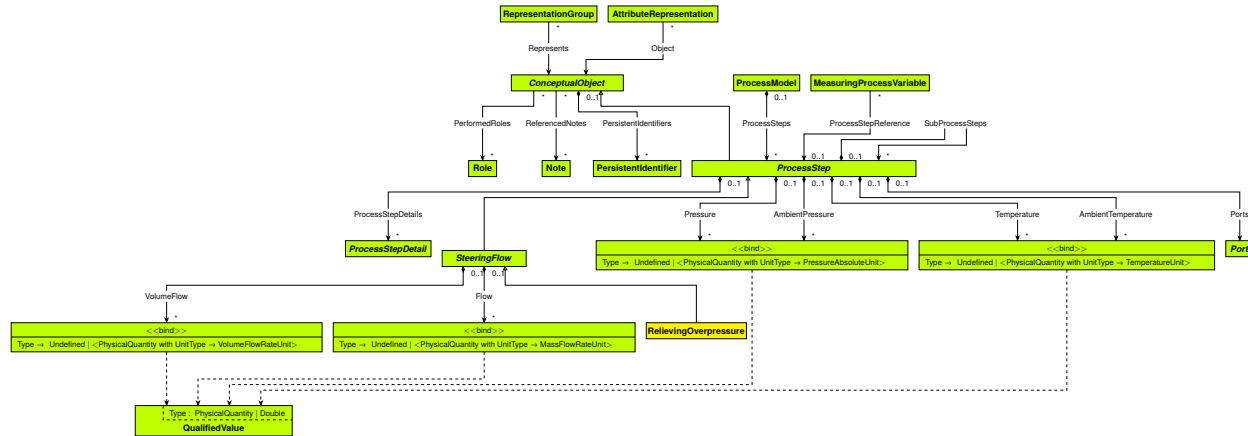
RelievingOverpressure

Process::Process::RelievingOverpressure

Overview

Concrete Class

A *SteeringFlow* process where a flow of material is manipulated to ensure that the pressure in a system shall not exceed a specified value.



Supertypes

- *SteeringFlow*

Example in DEXPI XML

```
<Object type="Process/Process.RelievingOverpressure"/>
```

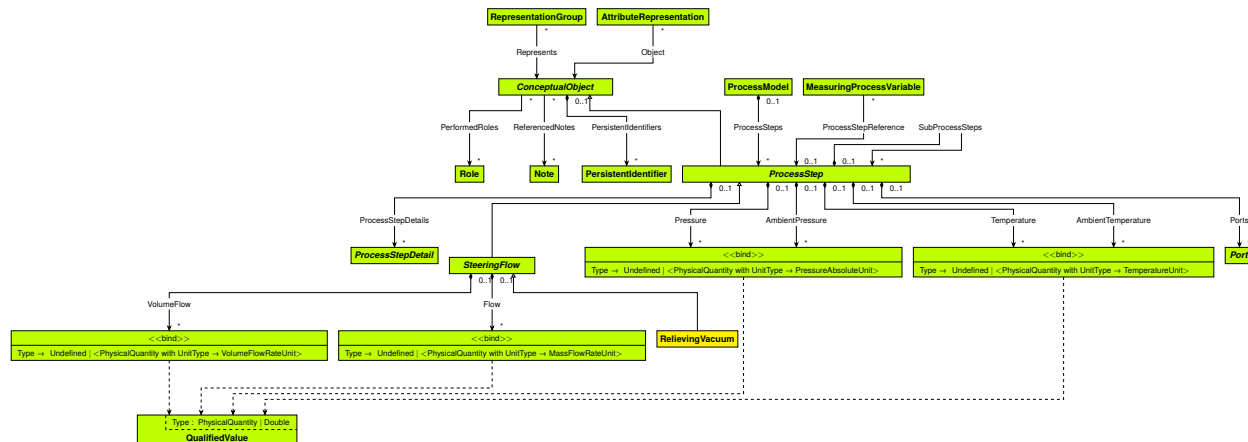
RelievingVacuum

Process::Process::RelievingVacuum

Overview

Concrete Class

A *SteeringFlow* process where a flow of material is manipulated to ensure that the pressure in a system shall not become sub-atmospheric.



Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = CentimetreSquared,
      Value = 7500.0))
]
```

Duty**Composition Property**

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → PowerUnit>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Kilowatt,
      Value = 120.0))
]
```

HeatTransferCoefficient**Composition Property**

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → HeatTransferCoefficientUnit>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KilowattPerMetreSquaredKelvin,
      Value = 25.5))
]
```

HeatTransferResistance

Composition Property

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{HeatTransferResistanceUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = MetreSquaredKelvinPerWatt,  
      Value = 3.92e-05))  
]
```

Method

Data Property

Type: *HeatExchangeMethod*

Multiplicity: 1

Ordered: no

Unique: no

SkinTemperature

Composition Property

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{TemperatureUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = DegreeCelsius,  
      Value = 320.0))  
]
```

TemperatureDifference**Composition Property**

Type: $\langle \text{QualifiedValue with } Type \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{TemperatureUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Kelvin,
      Value = 95.0))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.RemovingThermalEnergy">
  <Components property="Area">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.AreaUnit.CentimetreSquared"/>
          </Data>
          <Data property="Value">
            <Double>7500.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="Duty">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
          </Data>
          <Data property="Value">
            <Double>120.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="HeatTransferCoefficient">
    <Object type="Core/QualifiedValue">
```

(continues on next page)

(continued from previous page)

```

    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.HeatTransferCoefficientUnit.
↪KilowattPerMetreSquaredKelvin"/>
        </Data>
        <Data property="Value">
          <Double>25.5</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
<Components property="HeatTransferResistance">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.HeatTransferResistanceUnit.
↪MetreSquaredKelvinPerWatt"/>
        </Data>
        <Data property="Value">
          <Double>3.92e-05</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
<Components property="SkinTemperature">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.TemperatureUnit.DegreeCelsius"/>
        </Data>
        <Data property="Value">
          <Double>320.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
<Components property="TemperatureDifference">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.TemperatureUnit.Kelvin"/>
        </Data>
        <Data property="Value">
          <Double>95.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>

```

(continues on next page)

Process::Process::RotaryMixing

Concrete Class

- *Mixing*

Composit

Power

Composition Property

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{PowerUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Kilowatt,  
      Value = 190.0))  
]
```

RotationalFrequency

Composition Property

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{RotationalFrequencyUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = ReciprocalMinute,  
      Value = 300.0))  
]
```

Example in DEXPI XML

```
<Object type="Process/Process.RotaryMixing">  
  <Components property="Power">  
    <Object type="Core/QualifiedValue">  
      <Data property="Value">  
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
          <Data property="Unit">  
            <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>  
          </Data>  
          <Data property="Value">  
            <Double>190.0</Double>  
          </Data>  
        </AggregatedDataValue>  
      </Data>  
    </Object>  
  </Components>  
</Object>
```

(continues on next page)

(continued from previous page)

```

    </AggregatedDataValue>
  </Data>
</Object>
</Components>
<Components property="RotationalFrequency">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ReciprocalMinute"/>
        </Data>
        <Data property="Value">
          <Double>300.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
</Object>

```

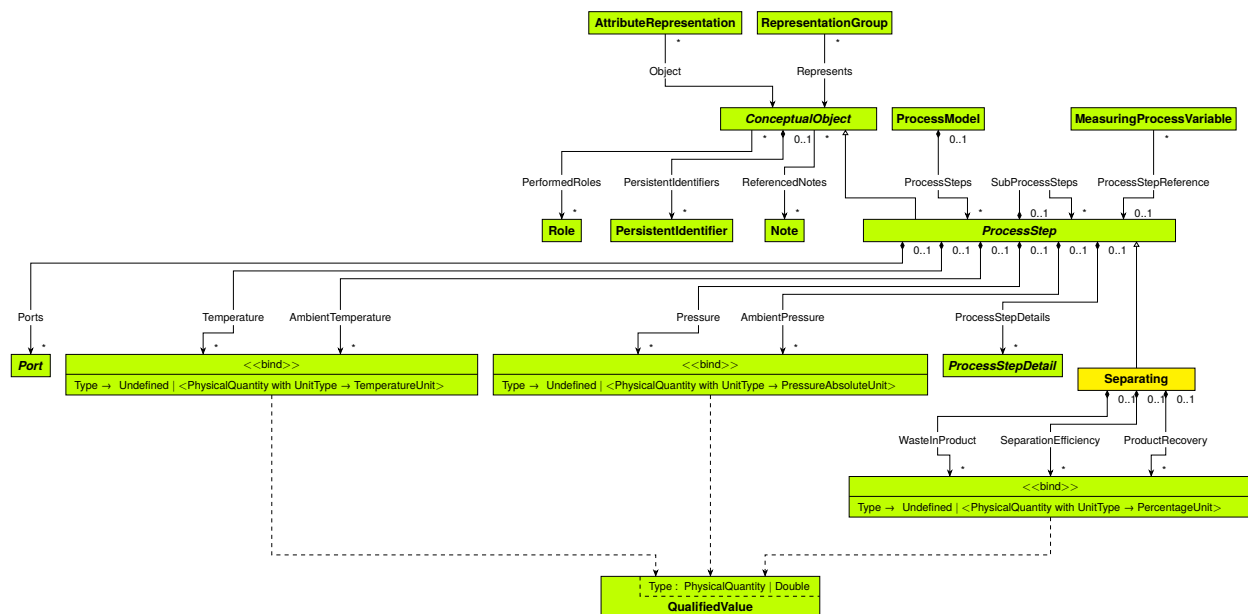
Separating

Process::Process::Separating

Overview

Concrete Class

This is the base type for all processes where one or more feeds are transformed into streams of material, each with different intensive properties and chemical composition.



Supertypes

- *ProcessStep*

Subtypes

- *SeparatingByElectromagneticForce*
- *SeparatingByFlash*
- *SeparatingByPhaseSeparation*
- *SeparatingByPhysicalProcess*
- *SeparatingByThermalProcess*
- *SeparatingMechanically*

Members**Composition Properties**

Name	Multiplicity	Type	Ordered
<i>ProductRecovery</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PercentageUnit>></i>	no
<i>SeparationEfficiency</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PercentageUnit>></i>	no
<i>WasteInProduct</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PercentageUnit>></i>	no

ProductRecovery**Composition Property**

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → PercentageUnit>>*

Multiplicity: *

Ordered: no

Example

```
[
    QualifiedValue(
        Value = PhysicalQuantity(
            Unit = Percent,
            Value = 35.0))
]
```


SeparationEfficiency**Composition Property**

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PercentageUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Percent,
      Value = 98.0))
]
```

WasteInProduct**Composition Property**

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PercentageUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Percent,
      Value = 4.0))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.Separating">
  <Components property="ProductRecovery">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.PercentageUnit.Percent"/>
          </Data>
          <Data property="Value">
            <Double>35.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```

(continues on next page)

(continued from previous page)

```

        </AggregatedDataValue>
    </Data>
</Object>
</Components>
<Components property="SeparationEfficiency">
    <Object type="Core/QualifiedValue">
        <Data property="Value">
            <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
                <Data property="Unit">
                    <DataReference data="Core/PhysicalQuantities.PercentageUnit.Percent"/>
                </Data>
                <Data property="Value">
                    <Double>98.0</Double>
                </Data>
            </AggregatedDataValue>
        </Data>
    </Object>
</Components>
<Components property="WasteInProduct">
    <Object type="Core/QualifiedValue">
        <Data property="Value">
            <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
                <Data property="Unit">
                    <DataReference data="Core/PhysicalQuantities.PercentageUnit.Percent"/>
                </Data>
                <Data property="Value">
                    <Double>4.0</Double>
                </Data>
            </AggregatedDataValue>
        </Data>
    </Object>
</Components>
</Object>

```

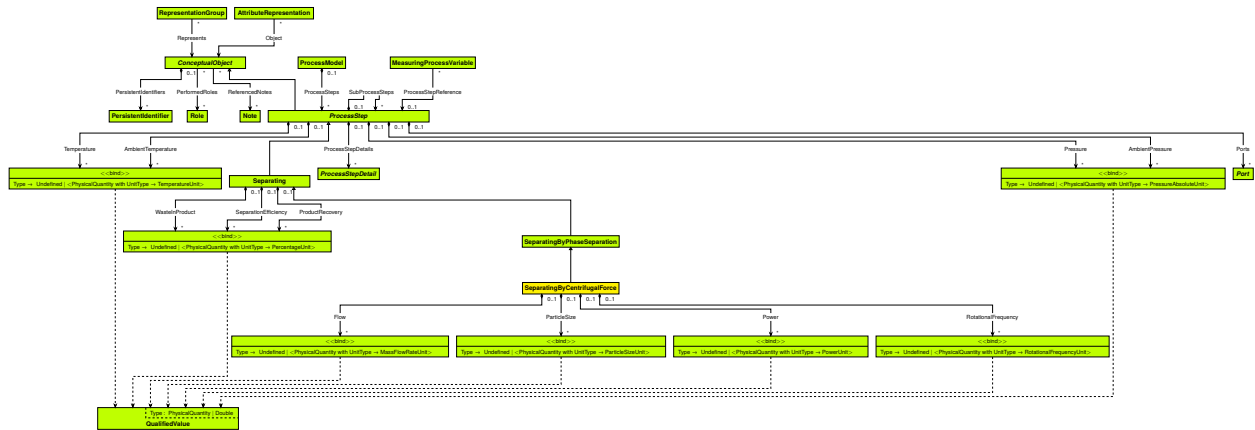
SeparatingByCentrifugalForce

Process::Process::SeparatingByCentrifugalForce

Overview

Concrete Class

This is a *SeparatingByPhaseSeparation* process where a fluid mixture is separated into a heavy phase and light phase through the external application mechanical energy to produce centrifugal forces.



Supertypes

- *SeparatingByPhaseSeparation*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Flow</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MassFlowRateUnit>></i>	no
<i>ParticleSize</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → ParticleSizeUnit>></i>	no
<i>Power</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PowerUnit>></i>	no
<i>RotationalFrequency</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → RotationalFrequencyUnit>></i>	no

Flow

Composition Property

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → MassFlowRateUnit>>*

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KilogramPerHour,
      Value = 67.0))
]
```

ParticleSize**Composition Property**

Type: <*QualifiedValue* with *Type* → *Undefined* | <*PhysicalQuantity* with *UnitType* → *ParticleSizeUnit*>>

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Millimetre,  
      Value = 0.9))  
]
```

Power**Composition Property**

Type: <*QualifiedValue* with *Type* → *Undefined* | <*PhysicalQuantity* with *UnitType* → *PowerUnit*>>

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Kilowatt,  
      Value = 120.0))  
]
```

RotationalFrequency**Composition Property**

Type: <*QualifiedValue* with *Type* → *Undefined* | <*PhysicalQuantity* with *UnitType* → *RotationalFrequencyUnit*>>

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(
```

Unit = ReciprocalMinute,
Value = 300.0))

]

Example in DEXPI XML

```
<Object type="Process/Process.SeparatingByCentrifugalForce">
  <Components property="Flow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
          </Data>
          <Data property="Value">
            <Double>67.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="ParticleSize">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.ParticleSizeUnit.Millimetre"/>
          </Data>
          <Data property="Value">
            <Double>0.9</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="Power">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
          </Data>
          <Data property="Value">
            <Double>120.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="RotationalFrequency">
    <Object type="Core/QualifiedValue">
```

(continues on next page)

(continued from previous page)

```

<Data property="Value">
  <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
    <Data property="Unit">
      <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ReciprocalMinute"/>
    </Data>
    <Data property="Value">
      <Double>300.0</Double>
    </Data>
  </AggregatedDataValue>
</Data>
</Object>
</Components>
</Object>

```

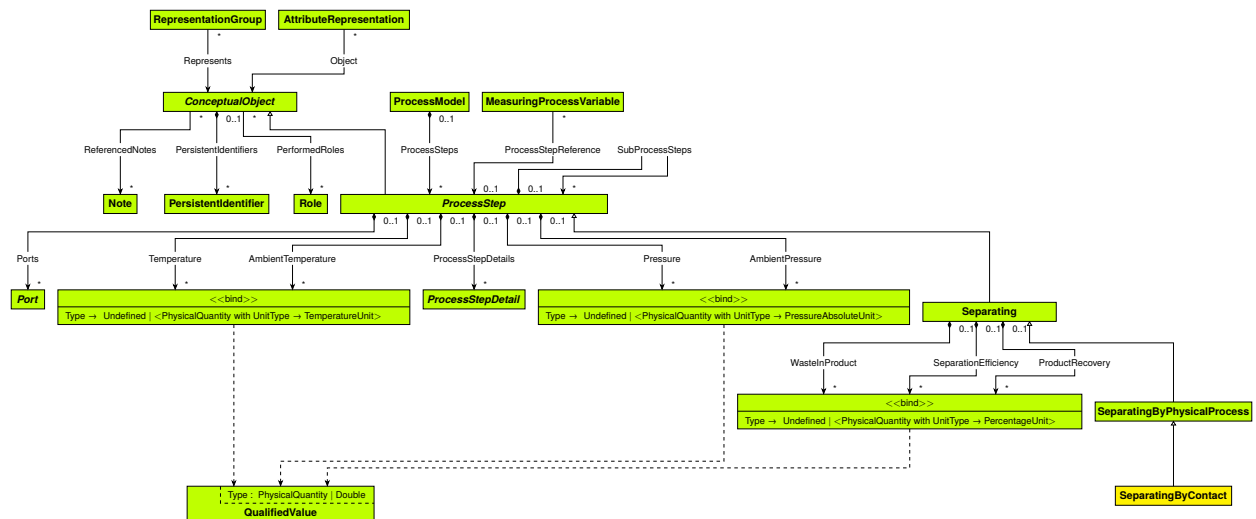
SeparatingByContact

Process::Process::SeparatingByContact

Overview

Concrete Class

This is a *SeparatingByPhysicalProcess* process where materials are brought into contact with each to transfer mass and energy.



Supertypes

- *SeparatingByPhysicalProcess*

Example in DEXPI XML

```
<Object type="Process/Process.SeparatingByContact"/>
```

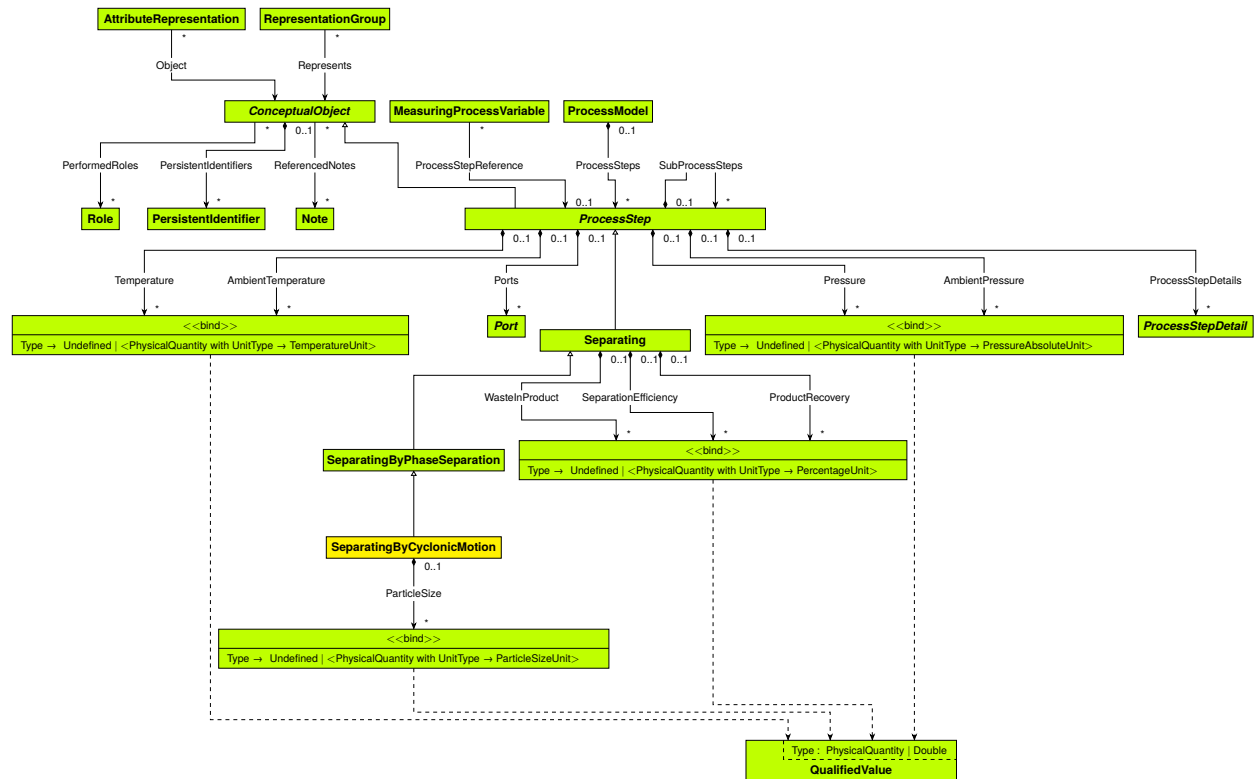
SeparatingByCyclonicMotion

Process::Process::SeparatingByCyclonicMotion

Overview

Concrete Class

This is a *SeparatingByPhaseSeparation* process where a fluid mixture is separated into a heavy phase and light phase through the centrifugal forces generated by the flow of fluids through a cyclone arrangement.



Supertypes

- *SeparatingByPhaseSeparation*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Particle-Size</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → ParticleSizeUnit>></i>	no

ParticleSize

Composition Property

Characteristic particle size for separation.

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{ParticleSizeUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Millimetre,  
      Value = 1.2))  
]
```

Example in DEXPI XML

```
<Object type="Process/Process.SeparatingByCyclonicMotion">  
  <Components property="ParticleSize">  
    <Object type="Core/QualifiedValue">  
      <Data property="Value">  
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
          <Data property="Unit">  
            <DataReference data="Core/PhysicalQuantities.ParticleSizeUnit.Millimetre"/>  
          </Data>  
          <Data property="Value">  
            <Double>1.2</Double>  
          </Data>  
        </AggregatedDataValue>  
      </Data>  
    </Object>  
  </Components>  
</Object>
```

SeparatingByElectromagneticForce

Process::Process::SeparatingByElectromagneticForce

Overview

Concrete Class

This is a *Separating* process where a material mixture is separated by the application of an electrical or magnetic field. This type is parent to *SeparatingByElectrostaticForce* and *SeparatingByMagneticForce*.



- ## Subtypes

- ## Members

Name	Multiplicity	Type	Ordered
<i>Particle-Size</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{ParticleSizeUnit} \rangle \rangle$	no
<i>Power</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle \rangle$	no

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{ParticleSizeUnit} \rangle \rangle$

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Millimetre,  
      Value = 0.4))  
]
```

Power**Composition Property**

Electrical power demand.

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → PowerUnit>>

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Kilowatt,  
      Value = 35.0))  
]
```

Example in DEXPI XML

```
<Object type="Process/Process.SeparatingByElectromagneticForce">  
  <Components property="ParticleSize">  
    <Object type="Core/QualifiedValue">  
      <Data property="Value">  
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
          <Data property="Unit">  
            <DataReference data="Core/PhysicalQuantities.ParticleSizeUnit.Millimetre"/>  
          </Data>  
          <Data property="Value">  
            <Double>0.4</Double>  
          </Data>  
        </AggregatedDataValue>  
      </Data>  
    </Object>  
  </Components>  
  <Components property="Power">  
    <Object type="Core/QualifiedValue">  
      <Data property="Value">
```

(continues on next page)

(continued from previous page)

```

<AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
  <Data property="Unit">
    <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
  </Data>
  <Data property="Value">
    <Double>35.0</Double>
  </Data>
</AggregatedDataValue>
</Data>
</Object>
</Components>
</Object>

```

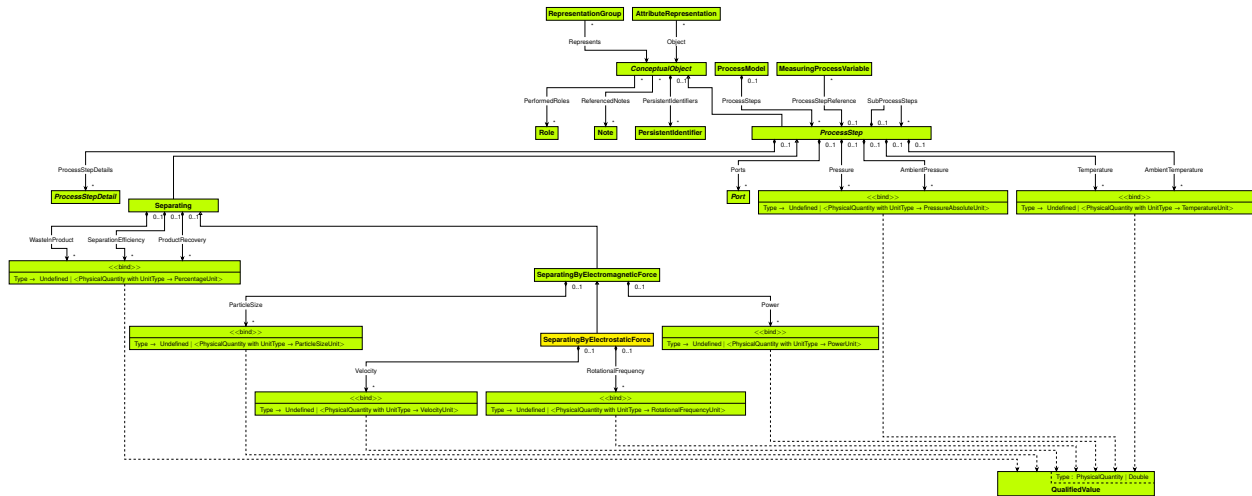
SeparatingByElectrostaticForce

Process::Process::SeparatingByElectrostaticForce

Overview

Concrete Class

This is a *SeparatingByElectromagneticForce* process where a material mixture is separated by the application of an electrostatic field.



Supertypes

- *SeparatingByElectromagneticForce*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>RotationalFrequency</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{RotationalFrequencyUnit} \rangle \rangle$	no
<i>Velocity</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VelocityUnit} \rangle \rangle$	no

RotationalFrequency

Composition Property

The rotational frequency of the process.

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{RotationalFrequencyUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = ReciprocalMinute,  
      Value = 220.0))  
]
```

Velocity

Composition Property

The velocity of the process.

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VelocityUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = MetrePerSecond,  
      Value = 8.0))  
]
```

]

Example in DEXPI XML

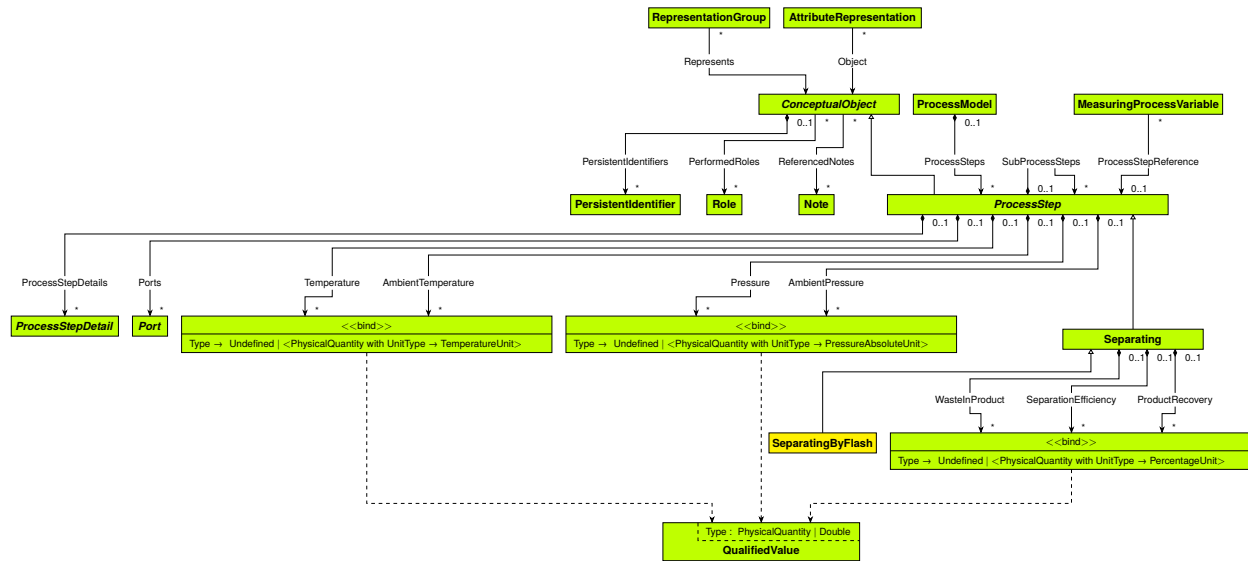
```

<Object type="Process/Process.SeparatingByElectrostaticForce">
  <Components property="RotationalFrequency">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ReciprocalMinute"/>
          </Data>
          <Data property="Value">
            <Double>220.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="Velocity">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.VelocityUnit.MetrePerSecond"/>
          </Data>
          <Data property="Value">
            <Double>8.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>

```

SeparatingByFlash*Process::Process::SeparatingByFlash***Overview****Concrete Class**

This is a *SeparatingByThermalProcess* process where a fluid mixture is separated into a vapour phase and a liquid phase in a vessel, usually after a sudden decrease in pressure. This is a single-stage process, unlike *Distilling*.



Supertypes

- *Separating*

Example in DEXPI XML

```
<Object type="Process/Process.SeparatingByFlash"/>
```

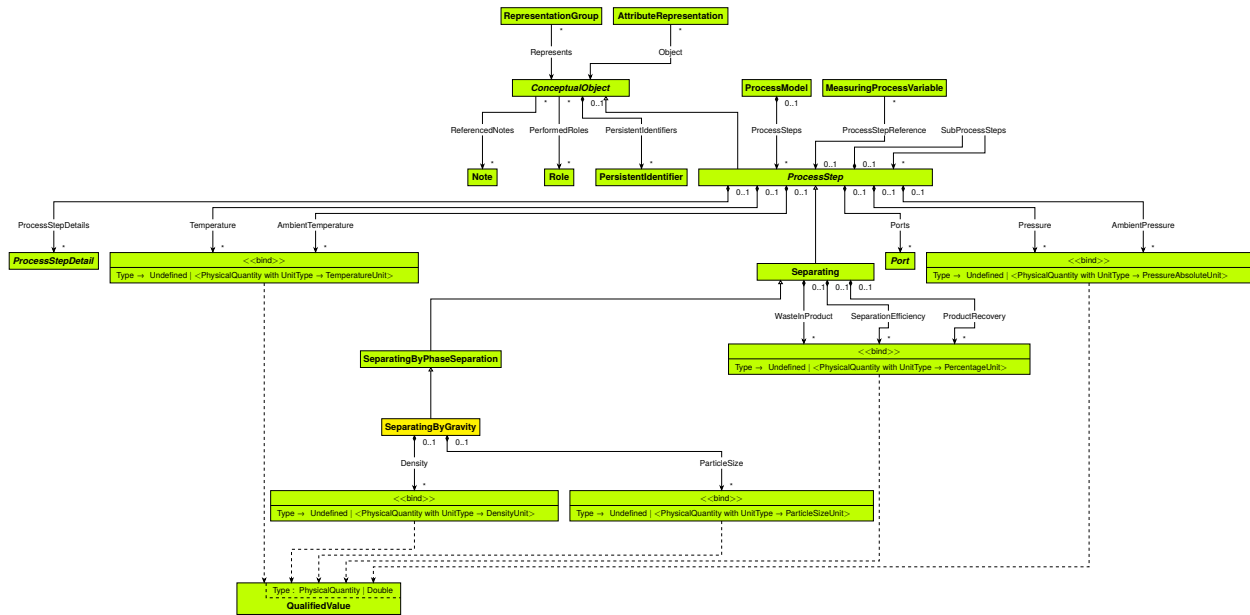
SeparatingByGravity

Process::Process::SeparatingByGravity

Overview

Concrete Class

This *SeparatingByPhaseSeparation* process separates substances based on the influence of gravity on substances with different density or specific gravity.



Supertypes

- *SeparatingByPhaseSeparation*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Density</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → DensityUnit>></i>	no
<i>Particle-Size</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → ParticleSizeUnit>></i>	no

Density

Composition Property

Characteristic density for separation.

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → DensityUnit>>*

Multiplicity: *

Ordered: no

Example

[

```

QualifiedValue(
    Value = PhysicalQuantity(
        Unit = KilogramPerMetreCubed,

```

```

    Value = 1220.0))
]

```

ParticleSize

Composition Property

Characteristic particle size for separation.

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{ParticleSizeUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```

[
    QualifiedValue(
        Value = PhysicalQuantity(
            Unit = Millimetre,
            Value = 0.8))
]

```

Example in DEXPI XML

```

<Object type="Process/Process.SeparatingByGravity">
  <Components property="Density">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.DensityUnit.
↪KilogramPerMetreCubed"/>
          </Data>
          <Data property="Value">
            <Double>1220.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="ParticleSize">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.ParticleSizeUnit.Millimetre"/>
          </Data>
          <Data property="Value">
            <Double>0.8</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>

```

(continues on next page)

(continued from previous page)

```

</AggregatedDataValue>
</Data>
</Object>
</Components>
</Object>

```

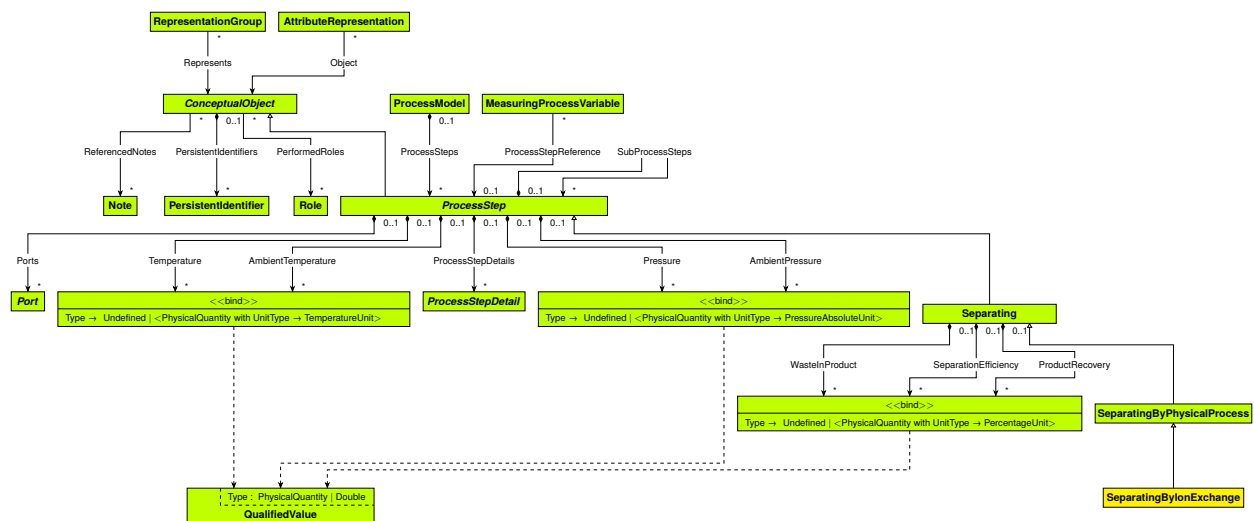
SeparatingByIonExchange

Process::Process::SeparatingByIonExchange

Overview

Concrete Class

This *SeparatingByPhysicalProcess* is a reversible interchange of one kind of ion present on an insoluble solid with another of like charge present in a solution surrounding the solid with the reaction being used especially for softening or demineralizing water, the purification of chemicals, or the separation of substances.



Supertypes

- *SeparatingByPhysicalProcess*

Example in DEXPI XML

```
<Object type="Process/Process.SeparatingByIonExchange"/>
```

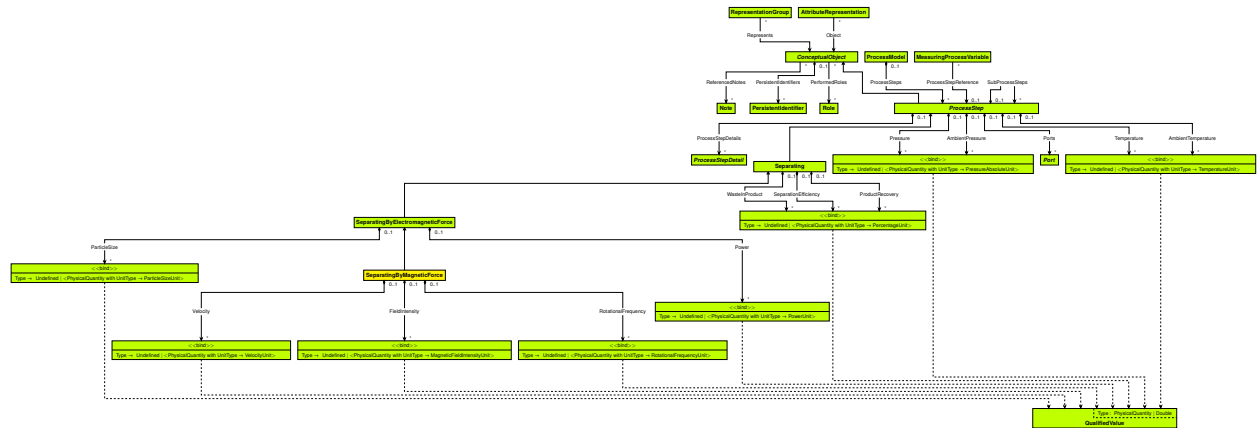
SeparatingByMagneticForce

Process::Process::SeparatingByMagneticForce

Overview

Concrete Class

This is a *SeparatingByElectromagneticForce* process where a material mixture is separated by the application of a magnetic field.



Supertypes

- *SeparatingByElectromagneticForce*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>FieldIntensity</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MagneticFieldIntensityUnit>></i>	no
<i>Rotational-Frequency</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → RotationalFrequencyUnit>></i>	no
<i>Velocity</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → VelocityUnit>></i>	no

FieldIntensity

Composition Property

Intensity of the magnetic field.

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → MagneticFieldIntensityUnit>>*

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KiloamperePerMetre,
      Value = 14.5))
]
```

RotationalFrequency**Composition Property**

The rotational frequency of the process.

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{RotationalFrequencyUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = ReciprocalMinute,
      Value = 380.0))
]
```

Velocity**Composition Property**

The velocity of the process.

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{VelocityUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = MetrePerSecond,
      Value = 9.0))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.SeparatingByMagneticForce">
  <Components property="FieldIntensity">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MagneticFieldIntensityUnit.
↪KiloamperePerMetre"/>
          </DataReference>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```

(continues on next page)

(continued from previous page)

```

        </Data>
        <Data property="Value">
            <Double>14.5</Double>
        </Data>
    </AggregatedDataValue>
</Data>
</Object>
</Components>
<Components property="RotationalFrequency">
    <Object type="Core/QualifiedValue">
        <Data property="Value">
            <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
                <Data property="Unit">
                    <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ReciprocalMinute"/>
                </Data>
                <Data property="Value">
                    <Double>380.0</Double>
                </Data>
            </AggregatedDataValue>
        </Data>
    </Object>
</Components>
<Components property="Velocity">
    <Object type="Core/QualifiedValue">
        <Data property="Value">
            <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
                <Data property="Unit">
                    <DataReference data="Core/PhysicalQuantities.VelocityUnit.MetrePerSecond"/>
                </Data>
                <Data property="Value">
                    <Double>9.0</Double>
                </Data>
            </AggregatedDataValue>
        </Data>
    </Object>
</Components>
</Object>

```

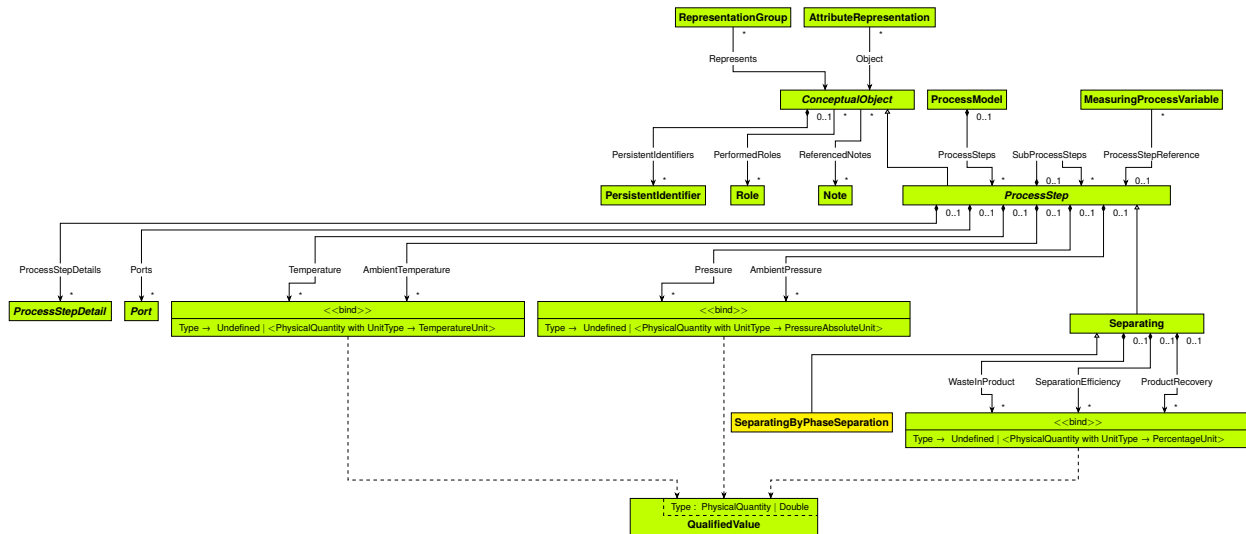
SeparatingByPhaseSeparation

Process::Process::SeparatingByPhaseSeparation

Overview

Concrete Class

This is a *Separating* process where the materials form distinct phases, which can then be separated.



Supertypes

- *Separating*

Subtypes

- *SeparatingByCentrifugalForce*
- *SeparatingByCyclonicMotion*
- *SeparatingByGravity*

Example in DEXPI XML

```
<Object type="Process/Process.SeparatingByPhaseSeparation"/>
```

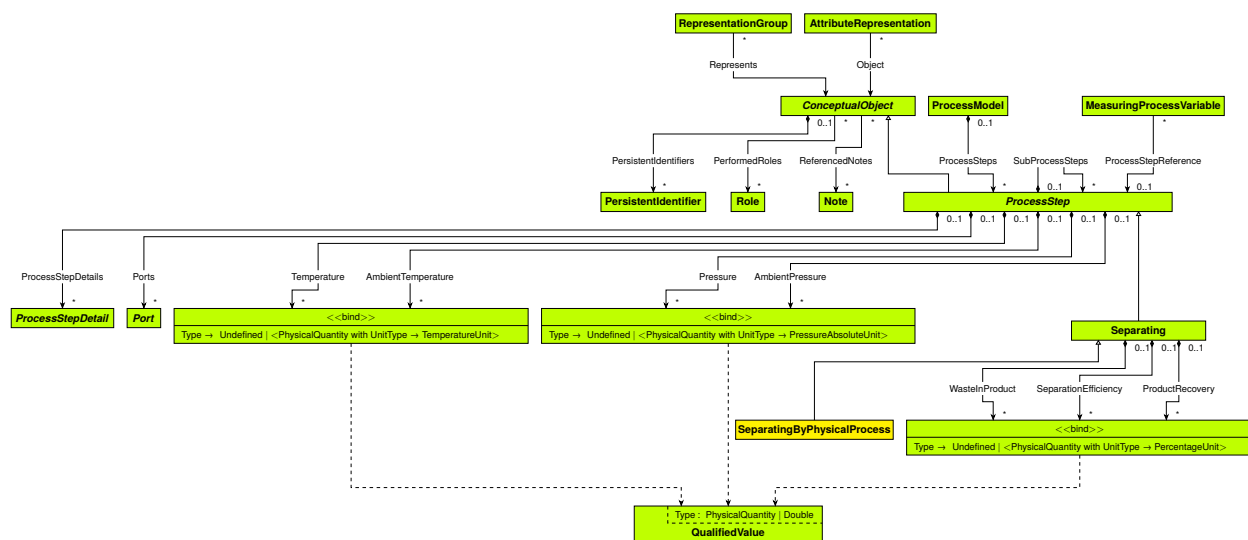
SeparatingByPhysicalProcess

Process::Process::SeparatingByPhysicalProcess

Overview

Concrete Class

This *Separating* process uses a physical phenomenon, such as mass transfer, ion exchange or surface tension to separate materials. Its child types are *SeparatingByContact*, *SeparatingByIonExchange*, *Absorbing*, *Adsorbing* and *SeparatingBySurfaceTension*.



Supertypes

- *Separating*

Subtypes

- *Absorbing*
- *Adsorbing*
- *SeparatingByContact*
- *SeparatingByIonExchange*
- *SeparatingBySurfaceTension*

Example in DEXPI XML

```
<Object type="Process/Process.SeparatingByPhysicalProcess"/>
```

SeparatingBySurfaceTension

Process::Process::SeparatingBySurfaceTension

Overview

Concrete Class

This is a *Separating* process where the materials are separated through a difference in surface forces between different liquid phases or between liquids and solid particles. Froth flotation is a typical process of this type.

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = KilogramPerHour,  
      Value = 12.2))  
]
```

GasFlow**Composition Property**

Flow of frothing gas.

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → MassFlowRateUnit>>

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = KilogramPerHour,  
      Value = 7.8))  
]
```

ParticleSize**Composition Property**

Characteristic particle size for separation.

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → ParticleSizeUnit>>

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Millimetre,  
      Value = 0.8))  
]
```


PulpDensity**Composition Property**

Density of the pulp.

Type: <*QualifiedValue* with *Type* → *Undefined* | <*PhysicalQuantity* with *UnitType* → *PercentageUnit*>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Percent,
      Value = 48.0))
]
```

SurfactantFlow**Composition Property**

Flow of surfactant chemicals.

Type: <*QualifiedValue* with *Type* → *Undefined* | <*PhysicalQuantity* with *UnitType* → *MassFlowRateUnit*>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KilogramPerHour,
      Value = 22.0))
]
```

Volume**Composition Property**

Volume of the process.

Type: <*QualifiedValue* with *Type* → *Undefined* | <*PhysicalQuantity* with *UnitType* → *VolumeUnit*>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = CentimetreCubed,
      Value = 2200.0))
]
```

pH**Composition Property**

pH for the process.

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{pHUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = pH,
      Value = 6.1))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.SeparatingBySurfaceTension">
  <Components property="FrotherFlow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
          </Data>
          <Data property="Value">
            <Double>12.2</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="GasFlow">
    <Object type="Core/QualifiedValue">
```

(continues on next page)

(continued from previous page)

```

    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
        </Data>
        <Data property="Value">
          <Double>7.8</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
<Components property="ParticleSize">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.ParticleSizeUnit.Millimetre"/>
        </Data>
        <Data property="Value">
          <Double>0.8</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
<Components property="PulpDensity">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.PercentageUnit.Percent"/>
        </Data>
        <Data property="Value">
          <Double>48.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
<Components property="SurfactantFlow">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
        </Data>
        <Data property="Value">
          <Double>22.0</Double>
        </Data>

```

(continues on next page)

(continued from previous page)

```

        </AggregatedDataValue>
    </Data>
</Object>
</Components>
<Components property="Volume">
    <Object type="Core/QualifiedValue">
        <Data property="Value">
            <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
                <Data property="Unit">
                    <DataReference data="Core/PhysicalQuantities.VolumeUnit.CentimetreCubed"/>
                </Data>
                <Data property="Value">
                    <Double>2200.0</Double>
                </Data>
            </AggregatedDataValue>
        </Data>
    </Object>
</Components>
<Components property="pH">
    <Object type="Core/QualifiedValue">
        <Data property="Value">
            <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
                <Data property="Unit">
                    <DataReference data="Core/PhysicalQuantities.pHUnit.pH"/>
                </Data>
                <Data property="Value">
                    <Double>6.1</Double>
                </Data>
            </AggregatedDataValue>
        </Data>
    </Object>
</Components>
</Object>

```

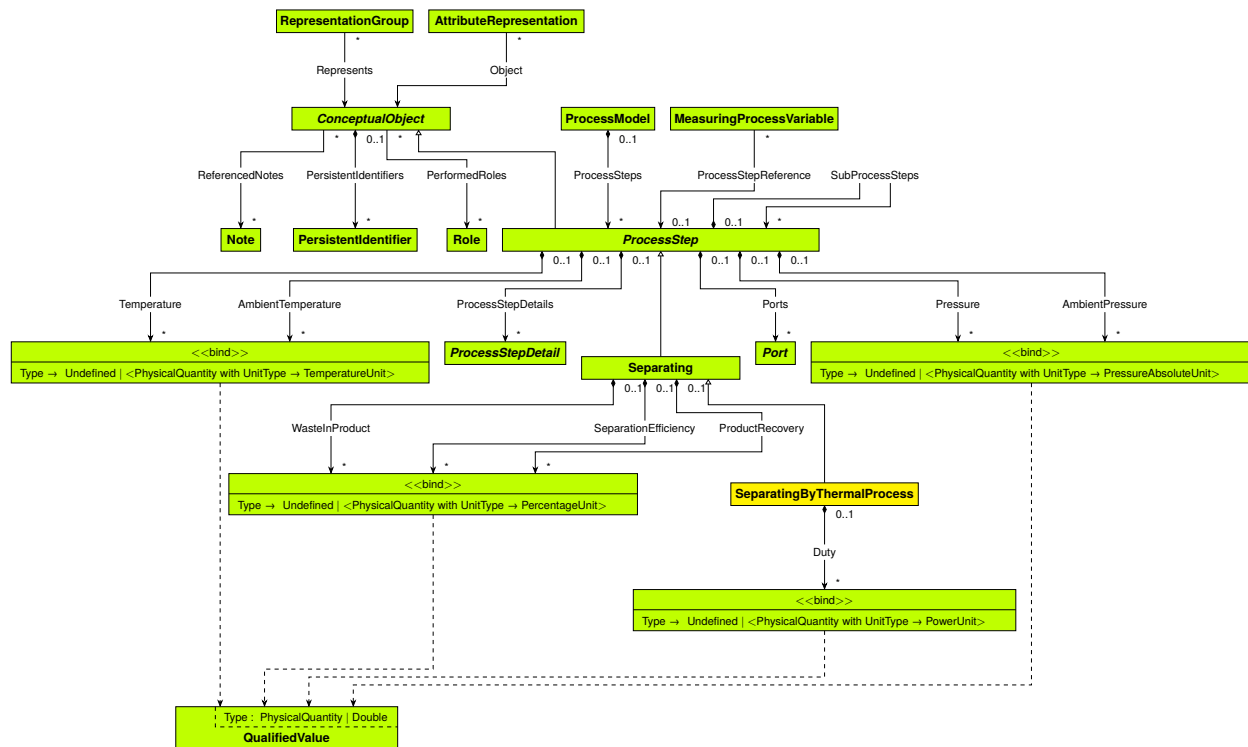
SeparatingByThermalProcess

Process::Process::SeparatingByThermalProcess

Overview

Concrete Class

This is a *Separating* process where the materials are separated through the application of heat or thermal energy.



Supertypes

- *Separating*

Subtypes

- *Distilling*
- *Drying*
- *Evaporating*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Duty</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PowerUnit>></i>	no

Duty

Composition Property

Duty of the process.

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{PowerUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Kilowatt,  
      Value = 90.0))  
]
```

Example in DEXPI XML

```
<Object type="Process/Process.SeparatingByThermalProcess">  
  <Components property="Duty">  
    <Object type="Core/QualifiedValue">  
      <Data property="Value">  
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
          <Data property="Unit">  
            <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>  
          </Data>  
          <Data property="Value">  
            <Double>90.0</Double>  
          </Data>  
        </AggregatedDataValue>  
      </Data>  
    </Object>  
  </Components>  
</Object>
```

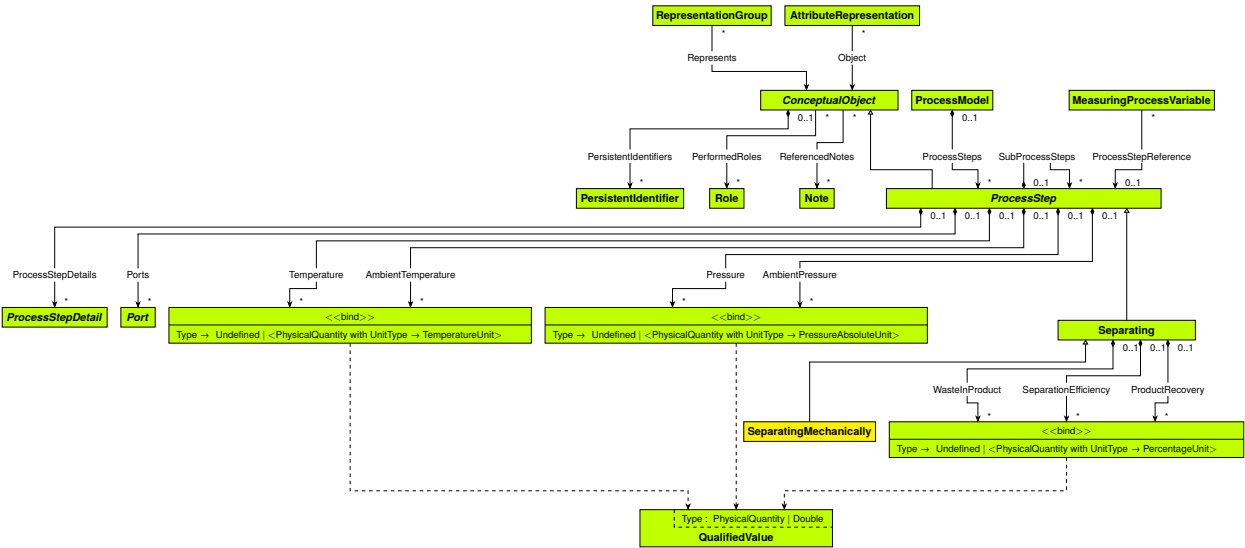
SeparatingMechanically

Process::Process::SeparatingMechanically

Overview

Concrete Class

This is a parent process where the materials are separated by a mechanical operation, such as *Filtering*, *Skimming* or *Sieving*.



Supertypes

- *Separating*

Subtypes

- *Filtering*
- *Sieving*
- *Skimming*

Example in DEXPI XML

```
<Object type="Process/Process.SeparatingMechanically"/>
```

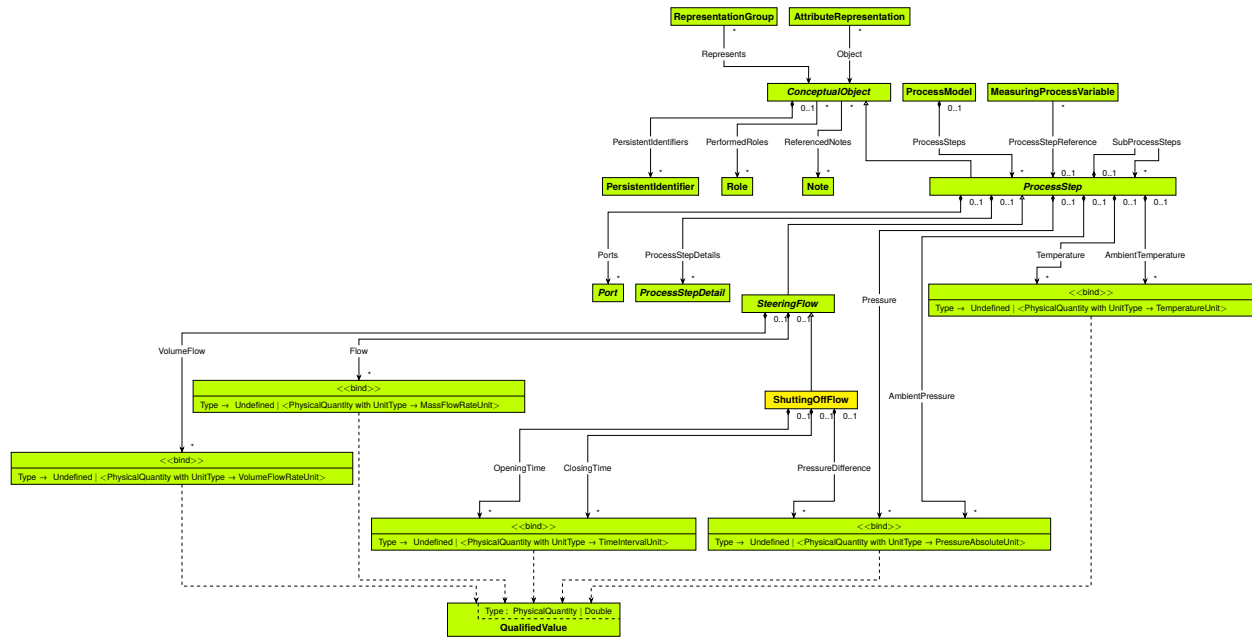
ShuttingOffFlow

Process::Process::ShuttingOffFlow

Overview

Concrete Class

This is *SteeringFlow* process with the purpose of shutting off flow for operational and/or safety reasons. It is usually realized by an on-off valve.



Supertypes

- *SteeringFlow*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>ClosingTime</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → TimeIntervalUnit>></i>	no
<i>OpeningTime</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → TimeIntervalUnit>></i>	no
<i>PressureDifference</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PressureAbsoluteUnit>></i>	no

ClosingTime

Composition Property

Closing time.

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → TimeIntervalUnit>>*

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Millisecond,
      Value = 720.0))
]
```

OpeningTime**Composition Property**

Opening time.

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → TimeIntervalUnit>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Millisecond,
      Value = 1100.0))
]
```

PressureDifference**Composition Property**

Pressure difference across process.

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → PressureAbsoluteUnit>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Bar,
      Value = 3.4))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.ShuttingOffFlow">
  <Components property="ClosingTime">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.TimeIntervalUnit.Millisecond"/>
          </Data>
          <Data property="Value">
            <Double>720.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="OpeningTime">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.TimeIntervalUnit.Millisecond"/>
          </Data>
          <Data property="Value">
            <Double>1100.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="PressureDifference">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.PressureAbsoluteUnit.Bar"/>
          </Data>
          <Data property="Value">
            <Double>3.4</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```

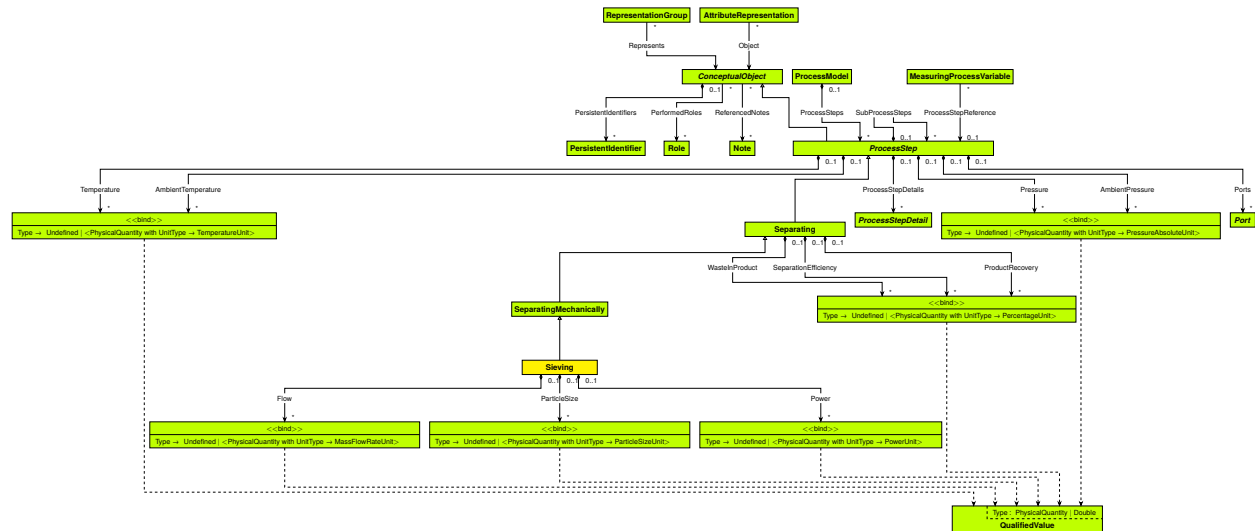
Sieving

Process::Process::Sieving

Overview

Concrete Class

This *SeparatingMechanically* processs for solids is based on particle size. Material is passed over a series of screens with a specified aperture size. Smaller particles pass through the screens, while larger particles remain on the screen.



Supertypes

- *SeparatingMechanically*

Members

Composition Properties

Name	Multi- plicity	Type	Or- dered
<i>Flow</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MassFlowRateUnit>></i>	no
<i>Particle- Size</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → ParticleSizeUnit>></i>	no
<i>Power</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PowerUnit>></i>	no

Flow

Composition Property

Mass flow rate through process.

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → MassFlowRateUnit>>*

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = KilogramPerHour,  
      Value = 80.0))  
]
```

ParticleSize**Composition Property**

Characteristic particle size for separation.

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → ParticleSizeUnit>>

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Millimetre,  
      Value = 2.5))  
]
```

Power**Composition Property**

Type: <QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → PowerUnit>>

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Kilowatt,  
      Value = 50.0))  
]
```

Example in DEXPI XML

```

<Object type="Process/Process.Sieving">
  <Components property="Flow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
          </Data>
          <Data property="Value">
            <Double>80.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="ParticleSize">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.ParticleSizeUnit.Millimetre"/>
          </Data>
          <Data property="Value">
            <Double>2.5</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="Power">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
          </Data>
          <Data property="Value">
            <Double>50.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>

```


Example in DEXPI XML

```
<Object type="Process/Process.Sink"/>
```

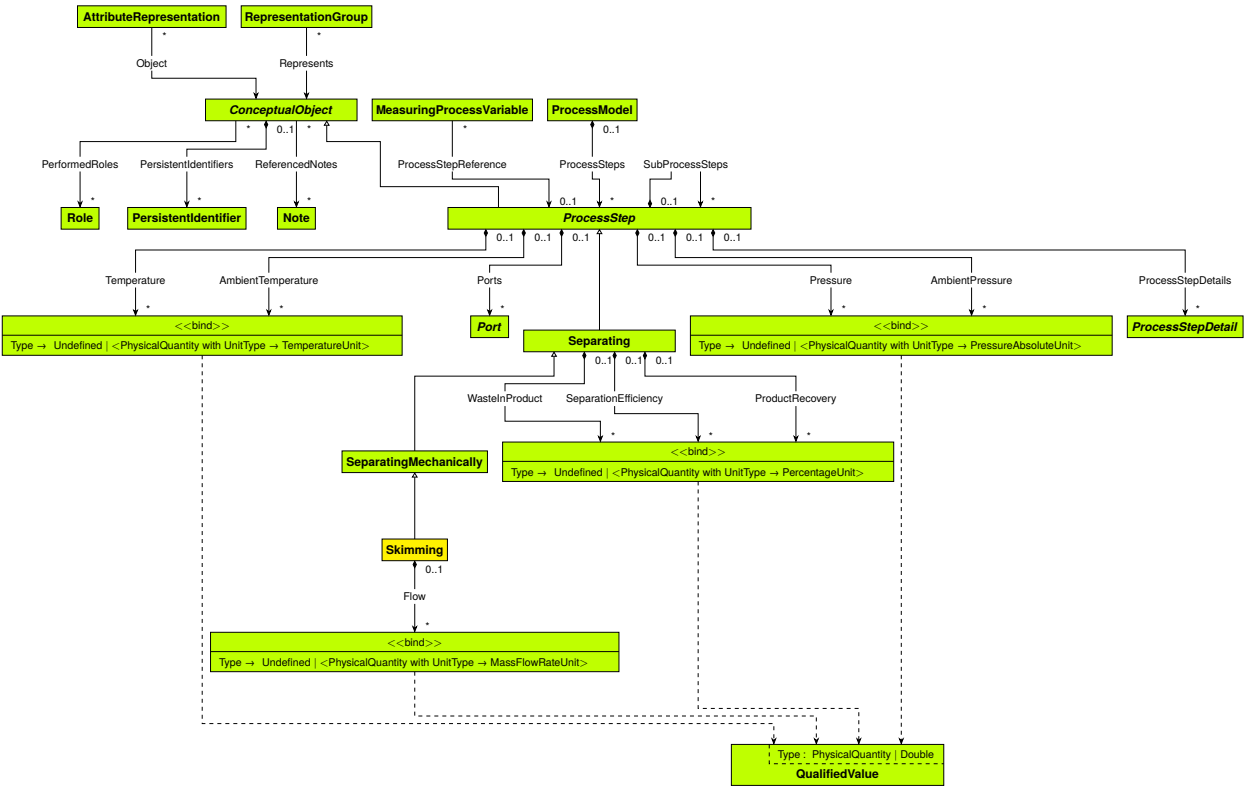
Skimming

Process::Process::Skimming

Overview

Concrete Class

This is a *SeparatingMechanically* process in which a lighter immiscible fluid, such as oil, is separated from a heavier fluid, such as water, by mechanical wiping of the surface of the heavier fluid.



Supertypes

- SeparatingMechanically*

Members

Composition Properties

Name	Multi- plicity	Type	Or- dered
Flow	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MassFlowRateUnit>></i>	no

Flow

Composition Property

Mass flow rate through process.

Type: $\langle \text{QualifiedValue with } Type \rightarrow Undefined \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{MassFlowRateUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = KilogramPerMinute,  
      Value = 82.0))  
]
```

Example in DEXPI XML

```
<Object type="Process/Process.Skimming">  
  <Components property="Flow">  
    <Object type="Core/QualifiedValue">  
      <Data property="Value">  
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
          <Data property="Unit">  
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.  
↪KilogramPerMinute"/>  
          </Data>  
          <Data property="Value">  
            <Double>82.0</Double>  
          </Data>  
        </AggregatedDataValue>  
      </Data>  
    </Object>  
  </Components>  
</Object>
```

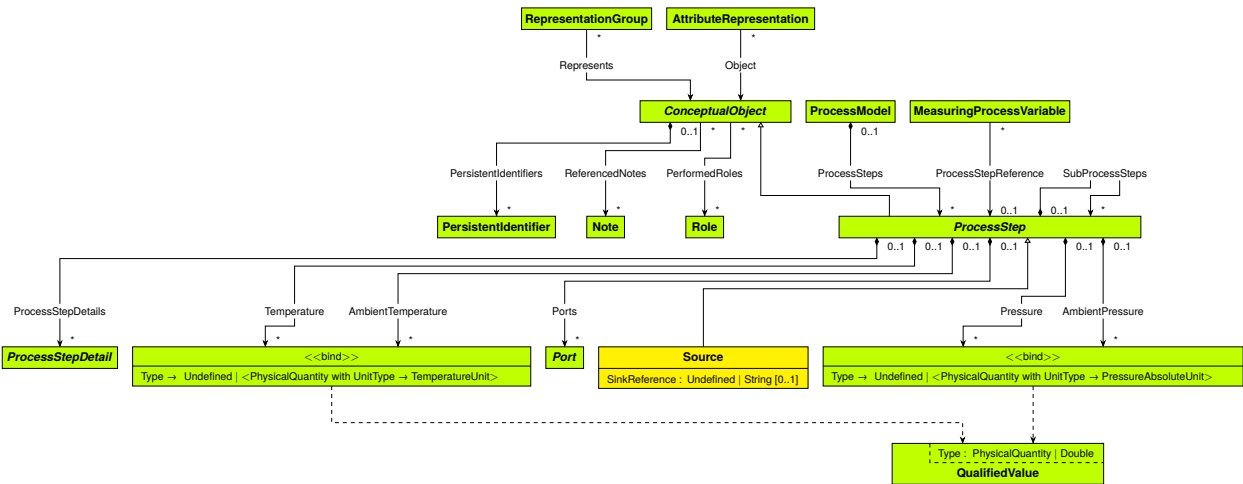
Source

Process::Process::Source

Overview

Concrete Class

This type is used to represent a flow of material into the process. It can be used to represent an incoming off-page connector in a PFD or BFD.



Supertypes

- *ProcessStep*

Members

Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>SinkReference</i>	0..1	<i>Undefined String</i>	no	no

SinkReference

Data Property

Reference to *Sink* object that represents the same flow.

Type: *Undefined | String*

Multiplicity: 0..1

Ordered: no

Unique: no

Example in DEXPI XML

```
<Object type="Process/Process.Source"/>
```

Splitting

Process::Process::Splitting

Overview

Abstract Class

This type is used to represent the splitting of a material or energy flow. This type is abstract, and the child types *SplittingMaterial* and *SplittingEnergy* are used in practice.

This type is used to represent the splitting of an energy flow. The intensive properties of the flow remain the same, but the total amount of flow (power) is divided among the outlet ports.

Value = 75.0))

]

Example in DEXPI XML

```
<Object type="Process/Process.SplittingEnergy">
  <Components property="Capacity">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassUnit.Kilogram"/>
          </Data>
          <Data property="Value">
            <Double>75.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```

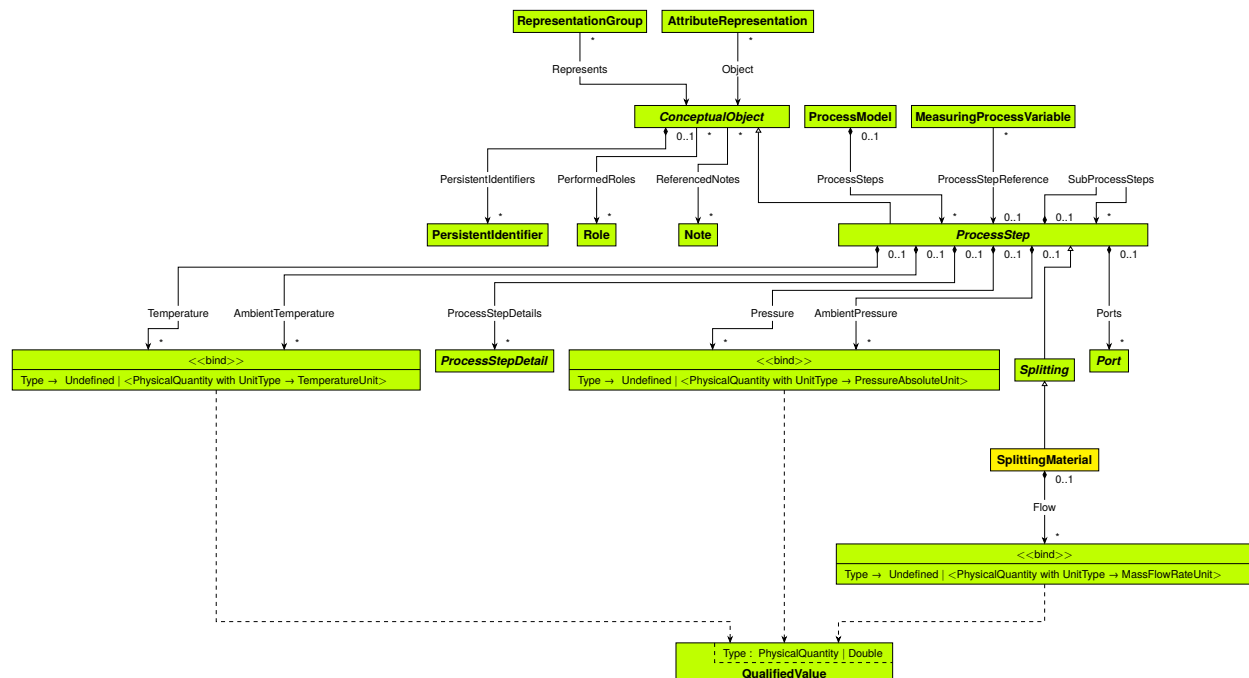
SplittingMaterial

Process::Process::SplittingMaterial

Overview

Concrete Class

This process splits an incoming flow of material into two or more outgoing flows of material. The intensive properties of the material flows remain the same.



Supertypes

- *Splitting*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Flow</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MassFlowRateUnit>></i>	no

Flow

Composition Property

Mass flow rate through process.

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → MassFlowRateUnit>>*

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KilogramPerHour,
      Value = 32.0))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.SplittingMaterial">
  <Components property="Flow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
          </Data>
          <Data property="Value">
            <Double>32.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```

(continues on next page)

(continued from previous page)

```
</Components>
</Object>
```

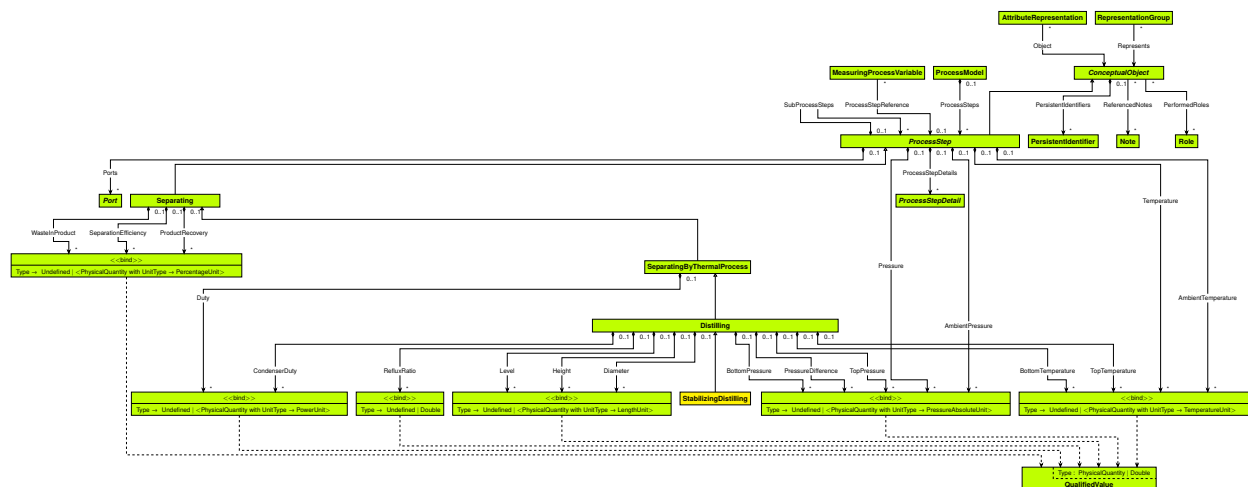
StabilizingDistilling

Process::Process::StabilizingDistilling

Overview

Concrete Class

This *Distilling* process has the purpose of producing a stabilized liquid product by removing the volatile chemical compounds that would otherwise make the product unstable for transport or storage.



Supertypes

- *Distilling*

Example in DEXPI XML

```
<Object type="Process/Process.StabilizingDistilling"/>
```

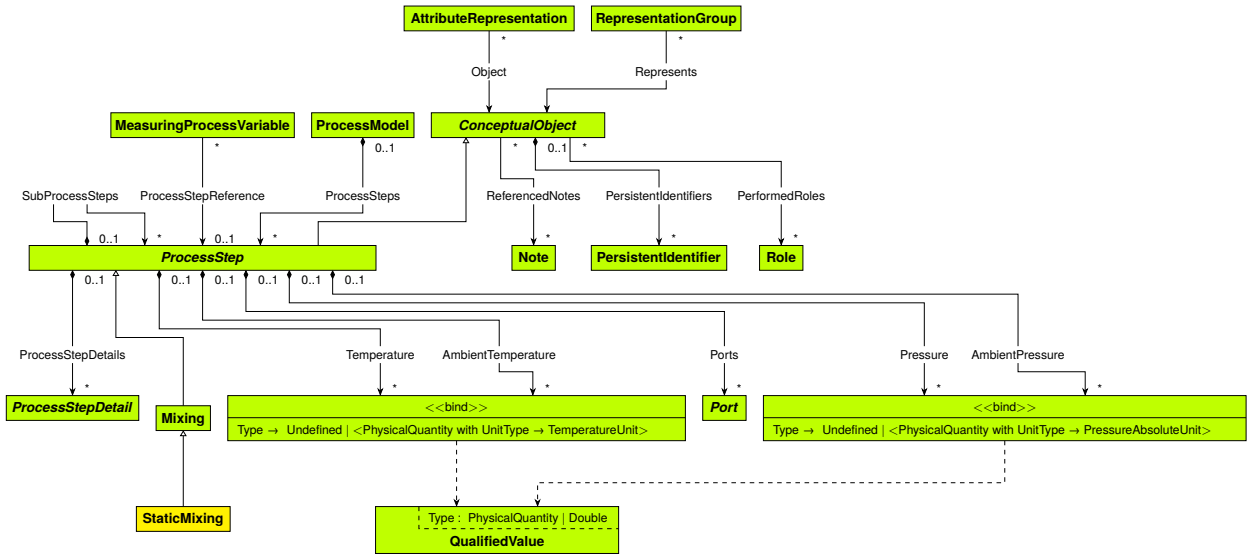
StaticMixing

Process::Process::StaticMixing

Overview

Concrete Class

This *Mixing* process uses a stationary element in the flow of material to produce the necessary homogeneity in the material that comes out of the process. It is realized by a *StaticMixer*.



Supertypes

- *Mixing*

Example in DEXPI XML

```
<Object type="Process/Process.StaticMixing"/>
```

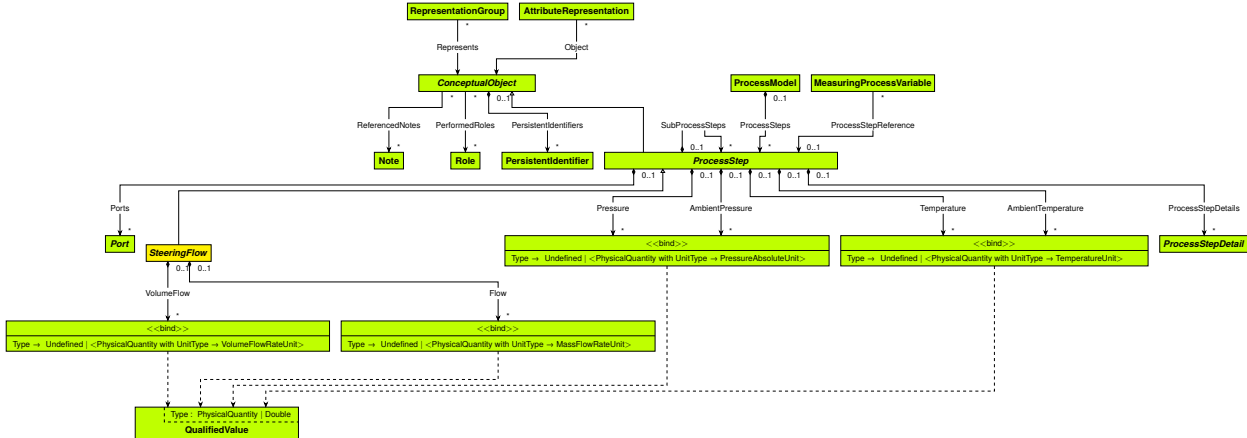
SteeringFlow

Process::Process::SteeringFlow

Overview

Abstract Class

The base type for processes that manipulate the rate of flow of material into and out of the process step. This process is usually realized using a valve. Child types of this type have been specified to support design of the process control and safety systems.



Supertypes

- *ProcessStep*

Subtypes

- *BlowingDown*
- *Draining*
- *FeedingMaterial*
- *LimitingFlow*
- *PreventingBackflow*
- *RegulatingFlow*
- *RelievingOverpressure*
- *RelievingVacuum*
- *RelievingVacuumAndOverpressure*
- *ShuttingOffFlow*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Flow</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MassFlowRateUnit>></i>	no
<i>Volume-Flow</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → VolumeFlowRateUnit>></i>	no

Flow

Composition Property

Mass flow rate through process.

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → MassFlowRateUnit>>*

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = KilogramPerHour,  
      Value = 40.0))  
]
```


Subtypes

- *StoringInBattery*

Members

Composition Properties

Name	Multi- plicity	Type	Or- dered
<i>ChargeCurrent</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{ElectricCurrentUnit} \rangle \rangle$	no
<i>DischargeCurrent</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{ElectricCurrentUnit} \rangle \rangle$	no
<i>Voltage</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VoltageUnit} \rangle \rangle$	no

ChargeCurrent

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{ElectricCurrentUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Ampere,  
      Value = 12.0))  
]
```

DischargeCurrent

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{ElectricCurrentUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Ampere,  

```

Value = 24.0))

]

Voltage

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VoltageUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

[

```
QualifiedValue(
  Value = PhysicalQuantity(
    Unit = Volt,
    Value = 18.0))
```

]

Example in DEXPI XML

```
<Object type="Process/Process.StoringElectricalEnergy">
  <Components property="ChargeCurrent">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.ElectricCurrentUnit.Ampere"/>
          </Data>
          <Data property="Value">
            <Double>12.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="DischargeCurrent">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.ElectricCurrentUnit.Ampere"/>
          </Data>
          <Data property="Value">
            <Double>24.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```

(continues on next page)

(continued from previous page)

```
</Object>
</Components>
<Components property="Voltage">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.VoltageUnit.Volt"/>
        </Data>
        <Data property="Value">
          <Double>18.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
</Object>
```

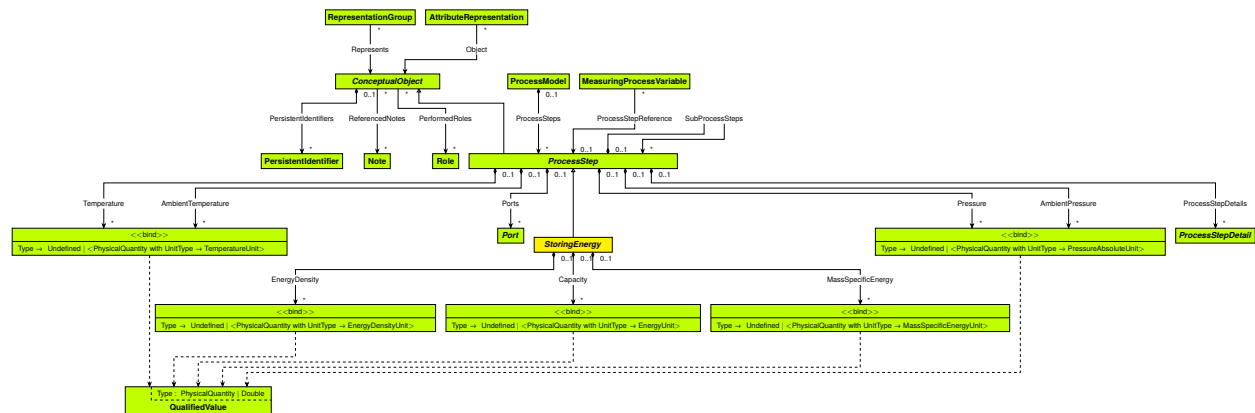
Storing Energy

Process::Process::StoringEnergy

Overview

Abstract Class

This process step stores electrical energy.



Supertypes

- *ProcessStep*

Subtypes

- *StoringElectricalEnergy*
- *StoringThermalEnergy*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Capacity</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{EnergyUnit} \rangle \rangle$	no
<i>EnergyDensity</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{EnergyDensityUnit} \rangle \rangle$	no
<i>MassSpecificEnergy</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassSpecificEnergyUnit} \rangle \rangle$	no

Capacity

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{EnergyUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Kilojoule,
      Value = 300.0))
]
```

EnergyDensity

Composition Property

The energy density of the *StoringEnergy*.

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{EnergyDensityUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

MassSpecificEnergy

Composition Property

The mass specific energy of the *StoringEnergy*.

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassSpecificEnergyUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Level**Composition Property**

Type: $\langle \text{QualifiedValue with } Type \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{LengthUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Centimetre,
      Value = 250.0))
]
```

Example in DEXPI XML

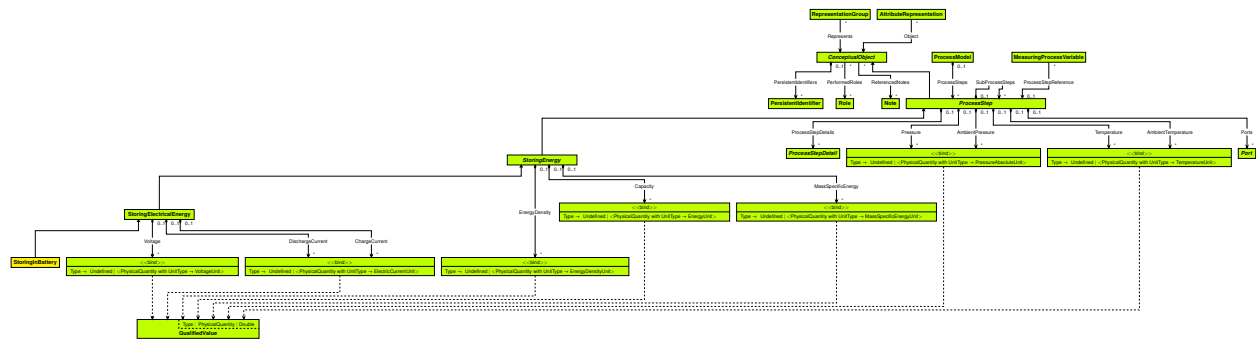
```
<Object type="Process/Process.StoringFluids">
  <Components property="Level">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
          </Data>
          <Data property="Value">
            <Double>250.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```

StoringInBattery

Process::Process::StoringInBattery

Overview**Concrete Class**

This *StoringElectricalEnergy* process step stores electrical energy in a battery. It can be connected to a charging flow of electricity and can also be used as a supplier of electrical energy to other process steps.



Supertypes

- *StoringElectricalEnergy*

Example in DEXPI XML

```
<Object type="Process/Process.StoringInBattery"/>
```

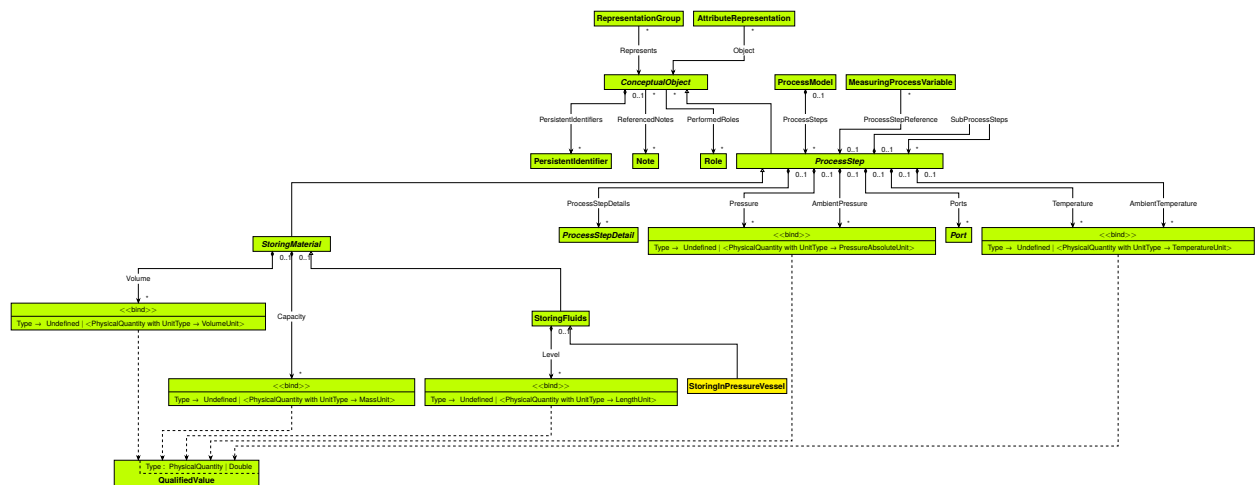
StoringInPressureVessel

Process::Process::StoringInPressureVessel

Overview

Concrete Class

This *StoringFluids* process accumulates fluid materials in a *PressureVessel*. This process assumes atmospheric storage of the material at pressures above or below atmospheric pressure. Use *StoringInTank* where fluid is stored at atmospheric conditions. It can be connected to a filling flow of material and an optional emptying flow of material. In this way it can be used as supplier of material for other process steps.



Supertypes

- *StoringFluids*

Example in DEXPI XML

```
<Object type="Process/Process.StoringInPressureVessel"/>
```

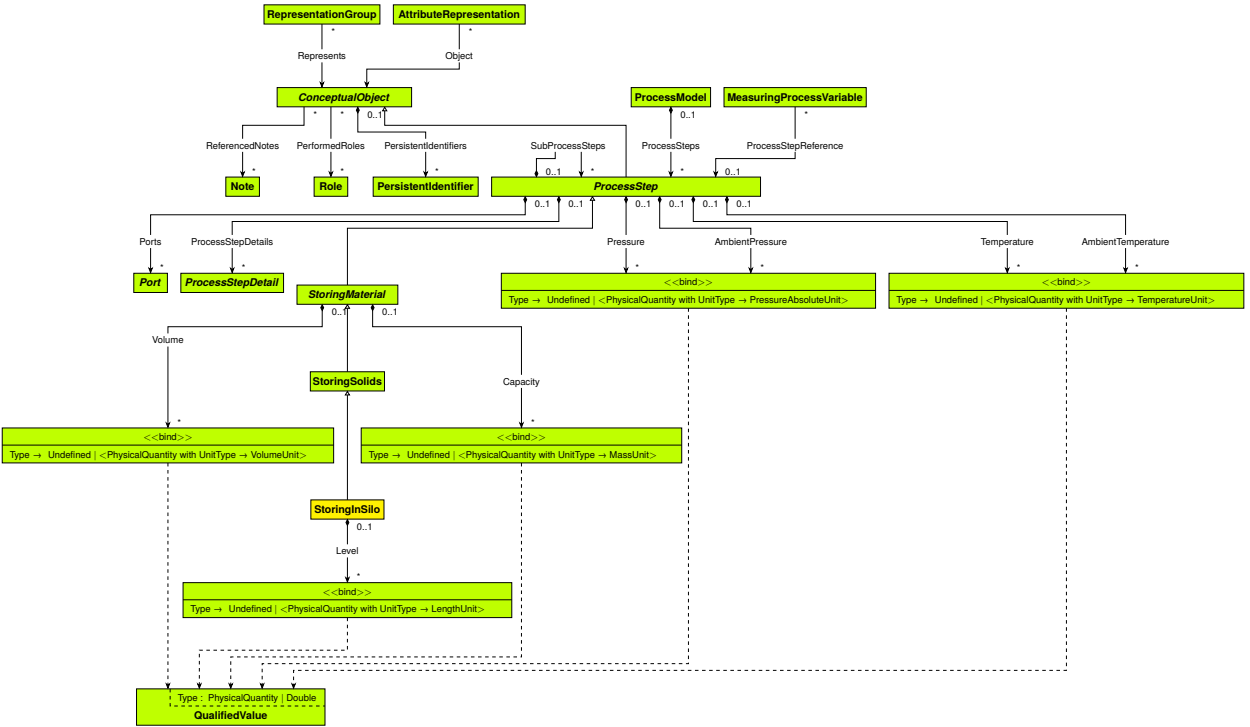
StoringInSilo

Process::Process::StoringInSilo

Overview

Concrete Class

This *StoringSolids* process is realized by a *Silo*.



Supertypes

- *StoringSolids*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Level</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → LengthUnit>></i>	no

Level

Composition Property

Type: $\langle \text{QualifiedValue with } Type \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{LengthUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Metre,  
      Value = 3.0))  
]
```

Example in DEXPI XML

```
<Object type="Process/Process.StoringInSilo">  
  <Components property="Level">  
    <Object type="Core/QualifiedValue">  
      <Data property="Value">  
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
          <Data property="Unit">  
            <DataReference data="Core/PhysicalQuantities.LengthUnit.Metre"/>  
          </Data>  
          <Data property="Value">  
            <Double>3.0</Double>  
          </Data>  
        </AggregatedDataValue>  
      </Data>  
    </Object>  
  </Components>  
</Object>
```

StoringInTank

Process::Process::StoringInTank

Overview

Concrete Class

This *StoringFluids* process accumulates fluid, solid or mixed materials in a *Tank*. This process assumes atmospheric storage of the material. Use *StoringInPressureVessel* where fluid or gas is to be stored at above-atmospheric or sub-atmospheric conditions. It can be connected to a filling flow of material and an optional emptying flow of material. In this way it can be used as supplier of material for other process steps.

Supertypes

- *ProcessStep*

Subtypes

- *StoringFluids*
- *StoringSolids*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Capacity</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassUnit} \rangle \rangle$	no
<i>Volume</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VolumeUnit} \rangle \rangle$	no

Capacity

Composition Property

Storage capacity of the storage process.

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Kilogram,  
      Value = 6500.0))  
]
```

Volume

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VolumeUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = MetreCubed,
      Value = 8.0))
]
```

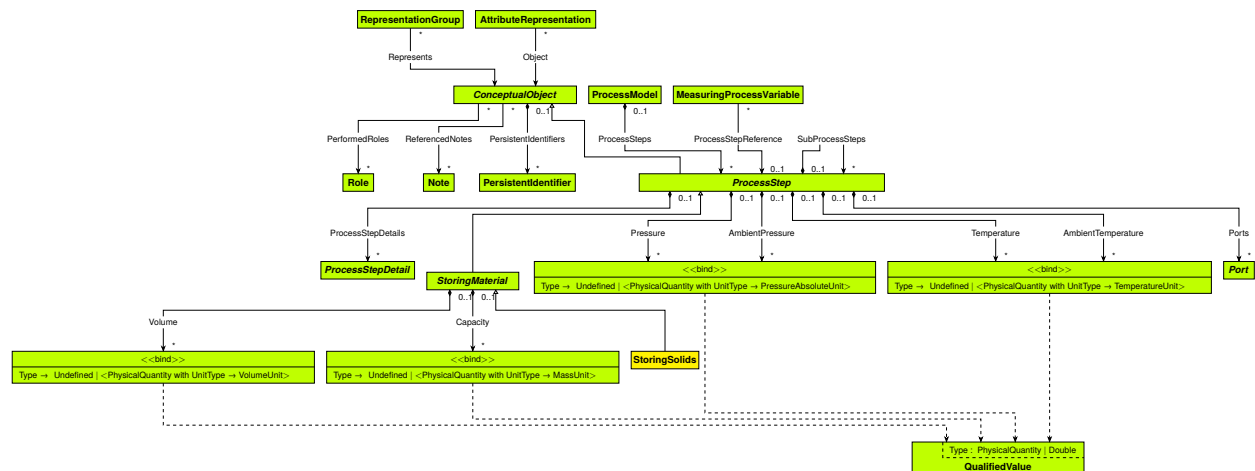
StoringSolids

Process::Process::StoringSolids

Overview

Concrete Class

This *StoringMaterial* process accumulates solid material. It can be connected to a filling flow of material and an optional emptying flow of material. In this way it can be used as supplier of material for other process steps. The principle and construction of the storage is unspecified.



Supertypes

- *StoringMaterial*

Subtypes

- *StoringInSilo*

Example in DEXPI XML

```
<Object type="Process/Process.StoringSolids"/>
```

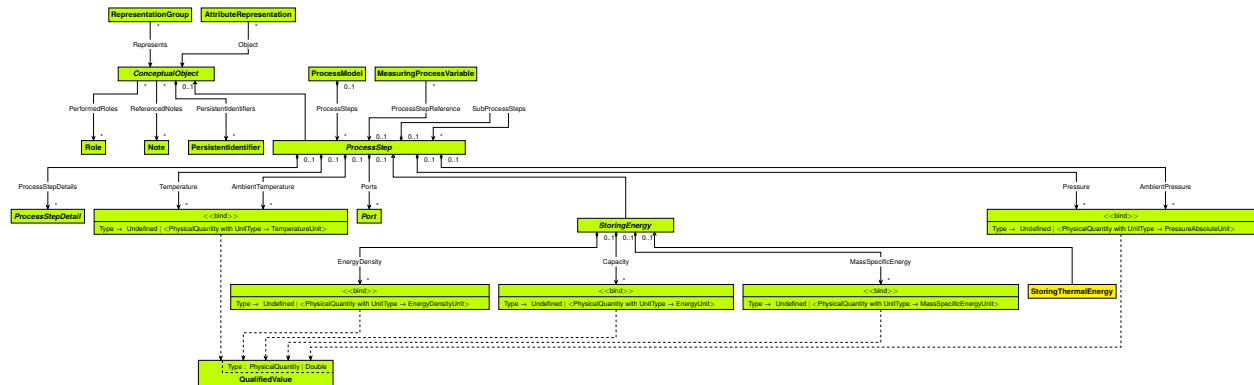
StoringThermalEnergy

Process::Process::StoringThermalEnergy

Overview

Concrete Class

This process accumulates and stores thermal energy for use in other processes. A calorifier is an example of this process in HVAC systems.



Supertypes

- *StoringEnergy*

Example in DEXPI XML

```
<Object type="Process/Process.StoringThermalEnergy"/>
```

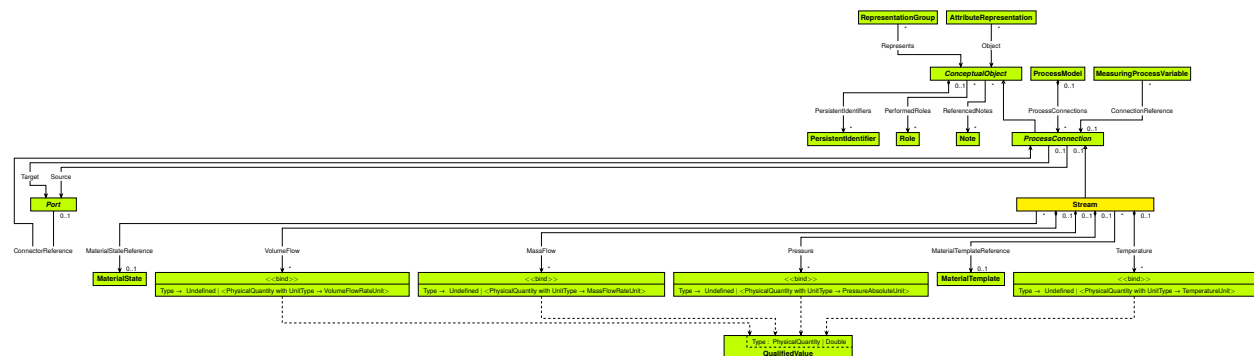
Stream

Process::Process::Stream

Overview

Concrete Class

This data structure defines the properties of a flow of fluid or solid material between two *MaterialPort* objects.



Supertypes

- *ProcessConnection*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Mass-Flow</i>	*	< <i>QualifiedValue</i> with <i>Type</i> → <i>Undefined</i> < <i>PhysicalQuantity</i> with <i>UnitType</i> → <i>MassFlowRateUnit</i> >>	no
<i>Pressure</i>	*	< <i>QualifiedValue</i> with <i>Type</i> → <i>Undefined</i> < <i>PhysicalQuantity</i> with <i>UnitType</i> → <i>PressureAbsoluteUnit</i> >>	no
<i>Temperature</i>	*	< <i>QualifiedValue</i> with <i>Type</i> → <i>Undefined</i> < <i>PhysicalQuantity</i> with <i>UnitType</i> → <i>TemperatureUnit</i> >>	no
<i>Volume-Flow</i>	*	< <i>QualifiedValue</i> with <i>Type</i> → <i>Undefined</i> < <i>PhysicalQuantity</i> with <i>UnitType</i> → <i>VolumeFlowRateUnit</i> >>	no

Reference Properties

Name	Multiplicity	Type	Ordered	Unique	Opposite Multiplicity
<i>MaterialStateReference</i>	0..1	<i>MaterialState</i>	no	no	*
<i>MaterialTemplateReference</i>	0..1	<i>MaterialTemplate</i>	no	no	*

MassFlow

Composition Property

Type: <*QualifiedValue* with *Type* → *Undefined* | <*PhysicalQuantity* with *UnitType* → *MassFlowRateUnit*>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KilogramPerHour,
      Value = 45.0))
]
```

MaterialStateReference

Reference Property

Type: *MaterialState*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

MaterialTemplateReference

Reference Property

Type: *MaterialTemplate*

Multiplicity: 0..1

Ordered: no

Unique: no

Opposite Multiplicity: *

Pressure

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PressureAbsoluteUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = Bar,  
      Value = 3.0))  
]
```

Temperature

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{TemperatureUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = DegreeCelsius,  
      Value = 52.0))  
]
```


]

VolumeFlow**Composition Property****Type:** $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VolumeFlowRateUnit} \rangle \rangle$ **Multiplicity:** ***Ordered:** no**Example**

[

```

QualifiedValue(
  Value = PhysicalQuantity(
    Unit = LitrePerSecond,
    Value = 2.4))

```

]

Example in DEXPI XML

```

<Object type="Process/Process.Stream">
  <Components property="MassFlow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour"
↪"/>
          </Data>
          <Data property="Value">
            <Double>45.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="Pressure">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.PressureAbsoluteUnit.Bar"/>
          </Data>
          <Data property="Value">
            <Double>3.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>

```

(continues on next page)

(continued from previous page)

```

</Components>
<Components property="Temperature">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.TemperatureUnit.DegreeCelsius"/>
        </Data>
        <Data property="Value">
          <Double>52.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
<Components property="VolumeFlow">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.
↪ LitrePerSecond"/>
        </Data>
        <Data property="Value">
          <Double>2.4</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
</Object>

```

StrippingDistilling

Process::Process::StrippingDistilling

Overview

Concrete Class

This *Distilling* process has the purpose of removing volatile components from a fluid.

- ## Concrete Class

Process::Process::SupplyingElectricalEnergy

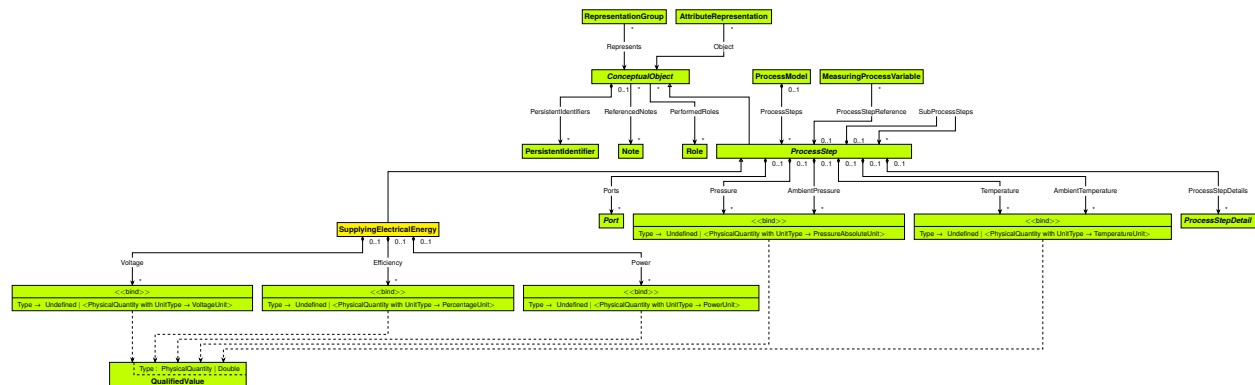
Example in DEXPI XML

- ## Supertypes



SupplyingElectricalEnergy

Process::Process::SupplyingElectricalEnergy



Subtypes

- *GeneratingACPower*
- *GeneratingDCPower*
- *GeneratingInFuelCell*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Efficiency</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PercentageUnit} \rangle \rangle$	no
<i>Power</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle \rangle$	no
<i>Voltage</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VoltageUnit} \rangle \rangle$	no

Efficiency

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PercentageUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
    QualifiedValue(  
        Value = PhysicalQuantity(  
            Unit = Percent,  
            Value = 84.0))  
]
```

Power

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Kilowatt,
      Value = 800.0))
]
```

Voltage**Composition Property**

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VoltageUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Volt,
      Value = 230.0))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.SupplyingElectricalEnergy">
  <Components property="Efficiency">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.PercentageUnit.Percent"/>
          </Data>
          <Data property="Value">
            <Double>84.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="Power">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
```

(continues on next page)

(continued from previous page)

```

    <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
  </Data>
  <Data property="Value">
    <Double>800.0</Double>
  </Data>
</AggregatedDataValue>
</Data>
</Object>
</Components>
<Components property="Voltage">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.VoltageUnit.Volt"/>
        </Data>
        <Data property="Value">
          <Double>230.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
</Object>

```

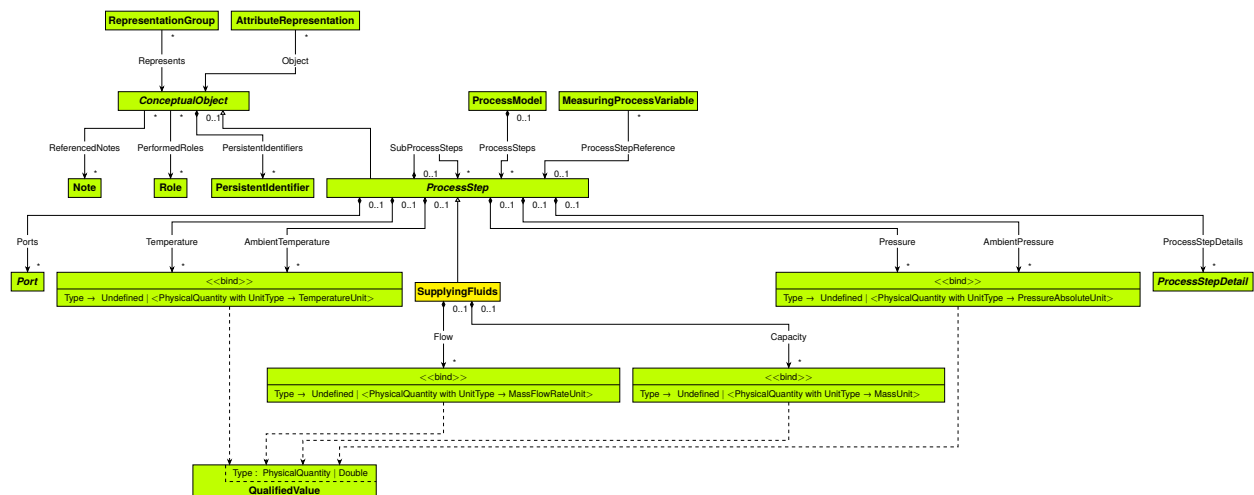
SupplyingFluids

Process::Process::SupplyingFluids

Overview

Concrete Class

This process is a source of fluid materials (liquids, gases or multiphase mixtures) for use in other processes.



Supertypes

- *ProcessStep*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Capacity</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassUnit} \rangle \rangle$	no
<i>Flow</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$	no

Capacity

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Kilogram,
      Value = 25.0))
]
```

Flow

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KilogramPerHour,
      Value = 2.4))
]
```

]

Example in DEXPI XML

```

<Object type="Process/Process.SupplyingFluids">
  <Components property="Capacity">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassUnit.Kilogram"/>
          </Data>
          <Data property="Value">
            <Double>25.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="Flow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
→"/>
          </Data>
          <Data property="Value">
            <Double>2.4</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>

```

SupplyingMechanicalEnergy

Process::Process::SupplyingMechanicalEnergy

Overview

Concrete Class

This process is a source of mechanical energy for use in other processes. It is the parent type for the *DrivingByEngine*, *DrivingByMotor* and *DrivingByTurbine* types.

Unit = Percent,
Value = 72.0))

]

RotationalFrequency

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{RotationalFrequencyUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

[

*QualifiedValue(
Value = PhysicalQuantity(
Unit = ReciprocalMinute,
Value = 900.0))*

]

ShaftPower

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PowerUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

[

*QualifiedValue(
Value = PhysicalQuantity(
Unit = Kilowatt,
Value = 200.0))*

]

Example in DEXPI XML

```
<Object type="Process/Process.SupplyingMechanicalEnergy">
  <Components property="Efficiency">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
```

(continues on next page)

(continued from previous page)

```

        <DataReference data="Core/PhysicalQuantities.PercentageUnit.Percent"/>
      </Data>
      <Data property="Value">
        <Double>72.0</Double>
      </Data>
    </AggregatedDataValue>
  </Data>
</Object>
</Components>
<Components property="RotationalFrequency">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.RotationalFrequencyUnit.
↪ReciprocalMinute"/>
        </Data>
        <Data property="Value">
          <Double>900.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
<Components property="ShaftPower">
  <Object type="Core/QualifiedValue">
    <Data property="Value">
      <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
        <Data property="Unit">
          <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
        </Data>
        <Data property="Value">
          <Double>200.0</Double>
        </Data>
      </AggregatedDataValue>
    </Data>
  </Object>
</Components>
</Object>

```

SupplyingSolids

Process::Process::SupplyingSolids

Overview

Concrete Class

This process is a source of solid material for use in other processes.

- *ProcessStep*

Composition Properties

Name	Multiplicity	Type	Ordered
Capacity	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \text{PhysicalQuantity with UnitType} \rightarrow \text{MassUnit} \rangle$	no
Flow	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle$	no

Composition Property

Multiplicity: *

Ordered: no

```
[
    QualifiedValue(
        Value = PhysicalQuantity(
            Unit = Kilogram,
            Value = 30.0))
]
```

Flow**Composition Property**

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KilogramPerHour,
      Value = 180.0))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.SupplyingSolids">
  <Components property="Capacity">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassUnit.Kilogram"/>
          </Data>
          <Data property="Value">
            <Double>30.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="Flow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
          </Data>
          <Data property="Value">
            <Double>180.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```


Data Properties

Name	Multiplicity	Type	Ordered	Unique
<i>Method</i>	0..1	<i>Undefined HeatExchangeMethod</i>	no	no

Area

Composition Property

Type: <*QualifiedValue* with *Type* → *Undefined* | <*PhysicalQuantity* with *UnitType* → *AreaUnit*>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = CentimetreSquared,
      Value = 700.0))
]
```

Duty

Composition Property

Type: <*QualifiedValue* with *Type* → *Undefined* | <*PhysicalQuantity* with *UnitType* → *PowerUnit*>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Kilowatt,
      Value = 60.0))
]
```

Flow

Composition Property

Type: <*QualifiedValue* with *Type* → *Undefined* | <*PhysicalQuantity* with *UnitType* → *MassFlowRateUnit*>>

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = KilogramPerHour,  
      Value = 120.0))  
]
```

HeatTransferCoefficient**Composition Property**

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{HeatTransferCoefficientUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = KilowattPerMetreSquaredKelvin,  
      Value = 45.0))  
]
```

HeatTransferResistance**Composition Property**

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{HeatTransferResistanceUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[  
  QualifiedValue(  
    Value = PhysicalQuantity(  
      Unit = MetreSquaredKelvinPerWatt,  
      Value = 2.2e-05))  
]
```


Method**Data Property****Type:** *Undefined* | *HeatExchangeMethod***Multiplicity:** 0..1**Ordered:** no**Unique:** no**SkinTemperature****Composition Property****Type:** $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{TemperatureUnit} \rangle \rangle$ **Multiplicity:** ***Ordered:** no**Example**

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = DegreeCelsius,
      Value = 40.0))
]
```

TemperatureDifference**Composition Property****Type:** $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{TemperatureUnit} \rangle \rangle$ **Multiplicity:** ***Ordered:** no**Example**

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Kelvin,
      Value = 22.5))
]
```

Example in DEXPI XML

```

<Object type="Process/Process.SupplyingThermalEnergy">
  <Components property="Area">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.AreaUnit.CentimetreSquared"/>
          </Data>
          <Data property="Value">
            <Double>700.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="Duty">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
          </Data>
          <Data property="Value">
            <Double>60.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="Flow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
          </Data>
          <Data property="Value">
            <Double>120.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="HeatTransferCoefficient">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.HeatTransferCoefficientUnit.
↪KilowattPerMetreSquaredKelvin"/>

```

(continues on next page)

(continued from previous page)

```

        </Data>
        <Data property="Value">
            <Double>45.0</Double>
        </Data>
    </AggregatedDataValue>
</Data>
</Object>
</Components>
<Components property="HeatTransferResistance">
    <Object type="Core/QualifiedValue">
        <Data property="Value">
            <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
                <Data property="Unit">
                    <DataReference data="Core/PhysicalQuantities.HeatTransferResistanceUnit.
↪MetreSquaredKelvinPerWatt"/>
                </Data>
                <Data property="Value">
                    <Double>2.2e-05</Double>
                </Data>
            </AggregatedDataValue>
        </Data>
    </Object>
</Components>
<Components property="SkinTemperature">
    <Object type="Core/QualifiedValue">
        <Data property="Value">
            <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
                <Data property="Unit">
                    <DataReference data="Core/PhysicalQuantities.TemperatureUnit.DegreeCelsius"/>
                </Data>
                <Data property="Value">
                    <Double>40.0</Double>
                </Data>
            </AggregatedDataValue>
        </Data>
    </Object>
</Components>
<Components property="TemperatureDifference">
    <Object type="Core/QualifiedValue">
        <Data property="Value">
            <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
                <Data property="Unit">
                    <DataReference data="Core/PhysicalQuantities.TemperatureUnit.Kelvin"/>
                </Data>
                <Data property="Value">
                    <Double>22.5</Double>
                </Data>
            </AggregatedDataValue>
        </Data>
    </Object>
</Components>
</Object>

```

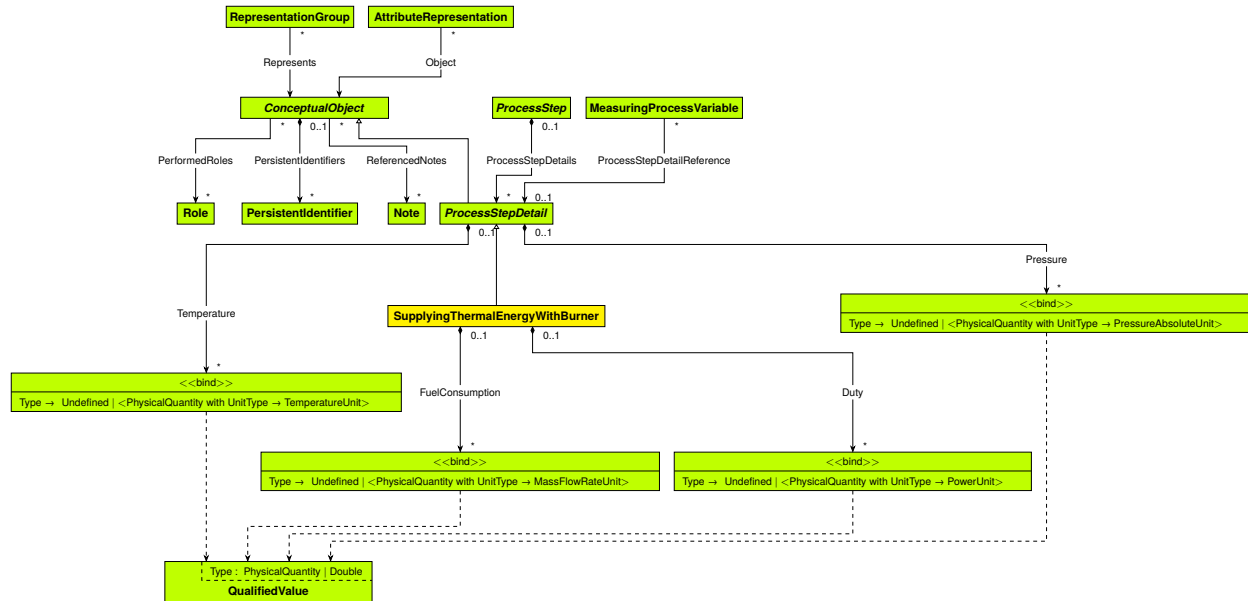
SupplyingThermalEnergyWithBurner

Process::Process::SupplyingThermalEnergyWithBurner

Overview

Concrete Class

This *ProcessStepDetail* is used to specify a combustion process as a sub-process of some other process, such as a sub-type of *SupplyingThermalEnergy*. This process is usually realized by one or more *Burner* components.



Supertypes

- *ProcessStepDetail*

Members

Composition Properties

Name	Multi- plicity	Type	Or- dered
<i>Duty</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PowerUnit>></i>	no
<i>FuelConsumption</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → MassFlowRateUnit>></i>	no

Duty

Composition Property

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → PowerUnit>>*

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Kilowatt,
      Value = 65.0))
]
```

FuelConsumption**Composition Property**

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KilogramPerHour,
      Value = 18.0))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.SupplyingThermalEnergyWithBurner">
  <Components property="Duty">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
          </Data>
          <Data property="Value">
            <Double>65.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="FuelConsumption">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
```

(continues on next page)

(continued from previous page)

```
↪ ">
    <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
    </Data>
    <Data property="Value">
      <Double>18.0</Double>
    </Data>
    </AggregatedDataValue>
  </Data>
</Object>
</Components>
</Object>
```

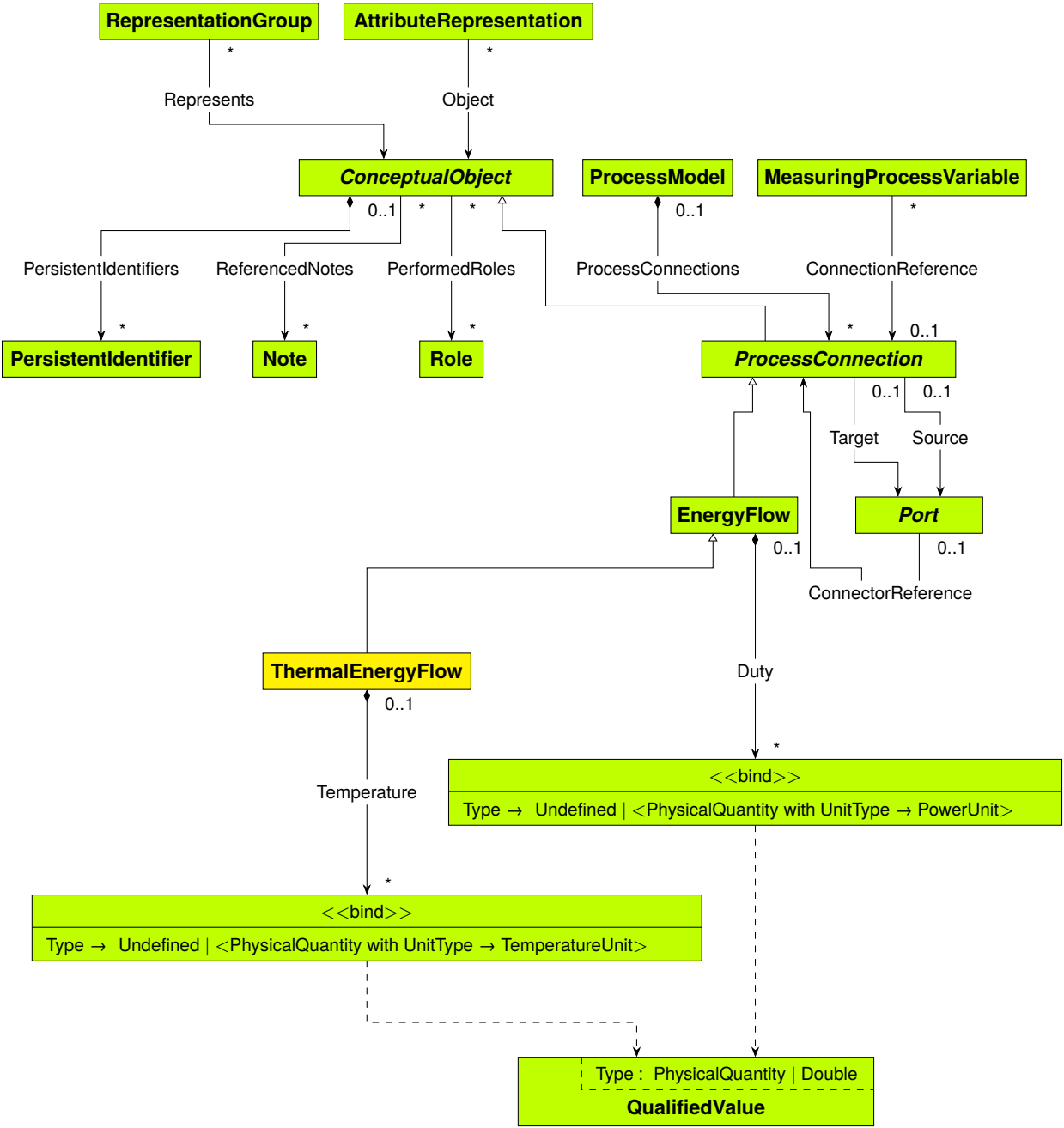
ThermalEnergyFlow

Process::Process::ThermalEnergyFlow

Overview

Concrete Class

This data structure defines the properties of a flow of heat or thermal energy between two *ThermalEnergyPort* objects.



Supertypes

- *EnergyFlow*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Temperature</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → TemperatureUnit>></i>	no

Temperature

Composition Property

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → TemperatureUnit>>*

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = DegreeCelsius,
      Value = 120.0))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.ThermalEnergyFlow">
  <Components property="Temperature">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.TemperatureUnit.DegreeCelsius"/>
          </Data>
          <Data property="Value">
            <Double>120.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```


ThermalEnergyPort

Process::Process::ThermalEnergyPort

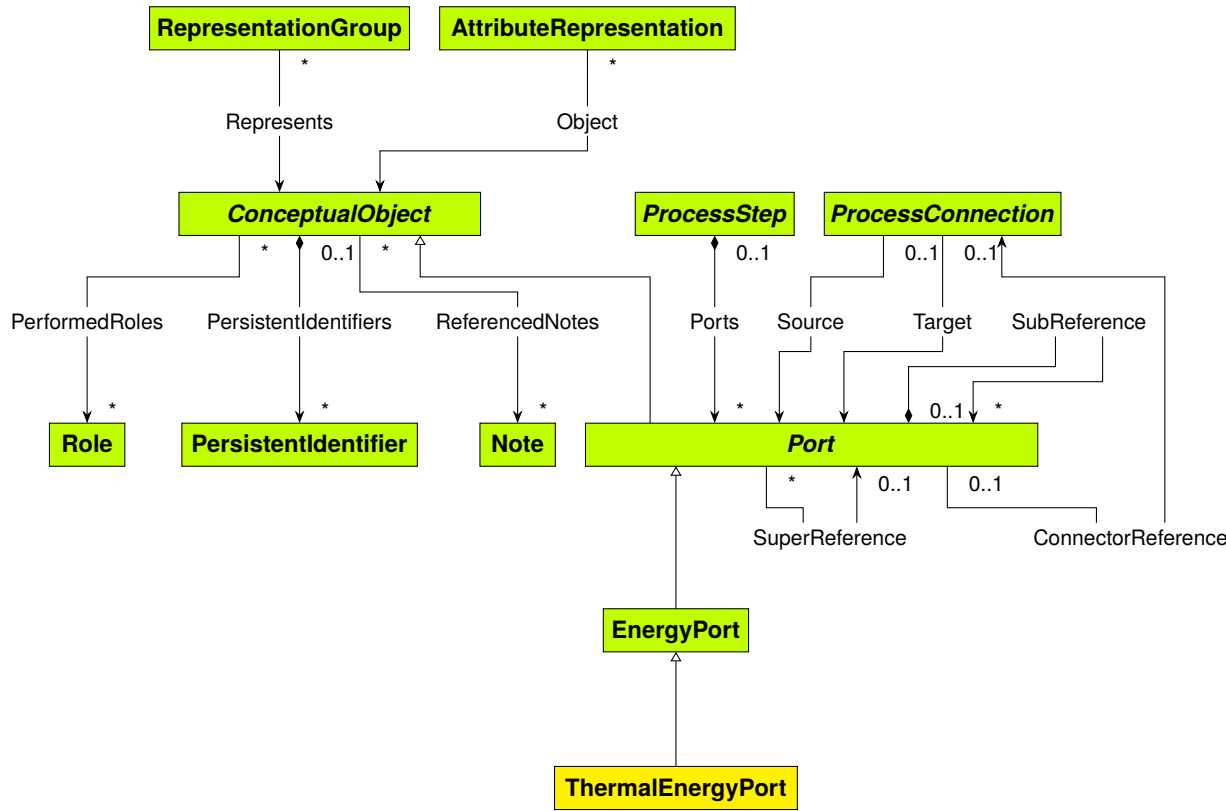
Overview

Concrete Class

This is a *Port* that exchanges heat or thermal energy between processes.

Roles referenced via *PerformedRoles* should have one of the following *Names*:

- Input
- Output



Supertypes

- *EnergyPort*

Example in DEXPI XML

```
<Object type="Process/Process.ThermalEnergyPort"/>
```

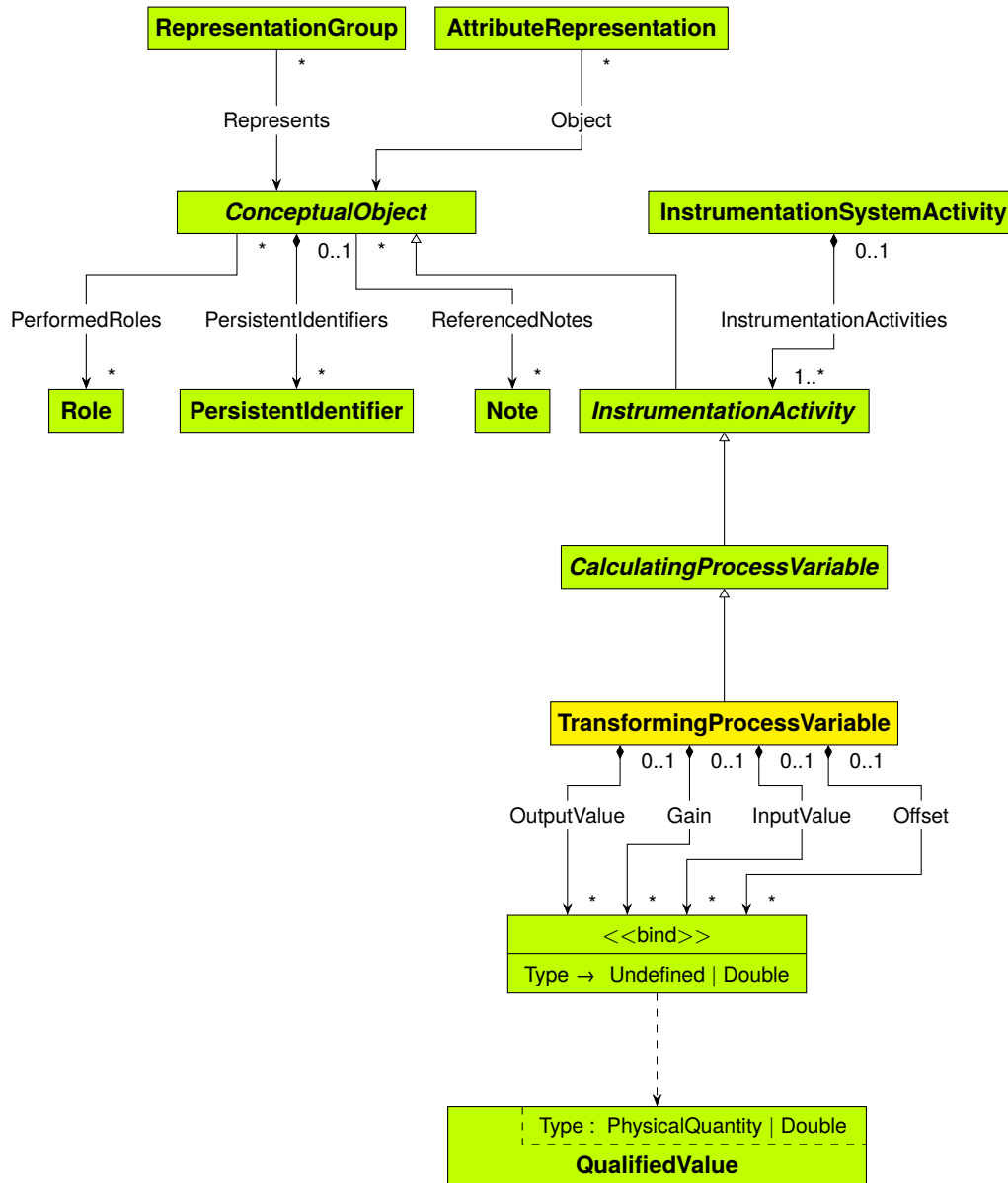
TransformingProcessVariable

Process::Process::TransformingProcessVariable

Overview

Concrete Class

This process performs a specified mathematical transformation on one or more input variables to calculate one or more output variables. The input and output variables are *InformationPort* objects owned by the type.



Supertypes

- *CalculatingProcessVariable*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Gain</i>	*	<i><QualifiedValue with Type → Undefined Double></i>	no
<i>InputValue</i>	*	<i><QualifiedValue with Type → Undefined Double></i>	no
<i>Offset</i>	*	<i><QualifiedValue with Type → Undefined Double></i>	no
<i>OutputValue</i>	*	<i><QualifiedValue with Type → Undefined Double></i>	no

Gain

Composition Property

Type: *<QualifiedValue with Type → Undefined | Double>*

Multiplicity: *

Ordered: no

InputValue

Composition Property

Type: *<QualifiedValue with Type → Undefined | Double>*

Multiplicity: *

Ordered: no

Offset

Composition Property

Type: *<QualifiedValue with Type → Undefined | Double>*

Multiplicity: *

Ordered: no

OutputValue

Composition Property

Type: *<QualifiedValue with Type → Undefined | Double>*

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Process/Process.TransformingProcessVariable"/>
```


Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Kilowatt,
      Value = 300.0))
]
```

Current**Composition Property**

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{ElectricCurrentUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Frequency**Composition Property**

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{ElectricalFrequencyUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

NumberOfPhases**Composition Property**

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \text{Integer} \rangle$

Multiplicity: *

Ordered: no

Voltage**Composition Property**

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VoltageUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Process/Process.TransportingElectricalEnergy">
  <Components property="Capacity">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
```

(continues on next page)

(continued from previous page)

```
<Data property="Unit">
  <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
</Data>
<Data property="Value">
  <Double>300.0</Double>
</Data>
</AggregatedDataValue>
</Data>
</Object>
</Components>
</Object>
```

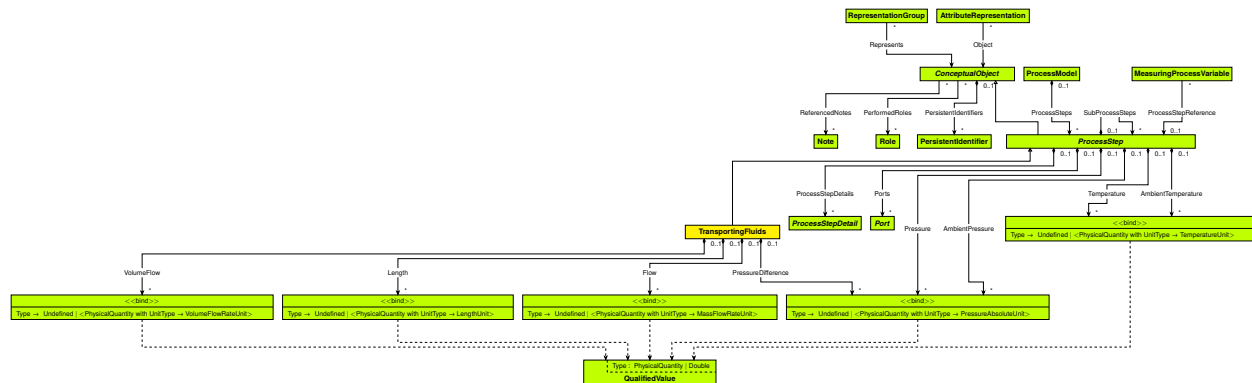
Transporting Fluids

Process::Process::TransportingFluids

Overview

Concrete Class

This process moves fluids: gas, liquid or multiphase flows, between processes. It is also used to model distribution processes, such as steam supply systems, gas reticulation networks, cooling water systems and flare systems. Three child types are used to specify the realisation of this process, as the realization has implications for process design. The child types are *TransportingFluidsInChannel*, *TransportingFluidsInHose* and *TransportingFluidsInPipe*.



Supertypes

- *ProcessStep*

Subtypes

- *TransportingFluidsInChannel*
- *TransportingFluidsInHose*
- *TransportingFluidsInPipe*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Flow</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$	no
<i>Length</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle \rangle$	no
<i>PressureDifference</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PressureAbsoluteUnit} \rangle \rangle$	no
<i>VolumeFlow</i>	*	$\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VolumeFlowRateUnit} \rangle \rangle$	no

Flow

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{MassFlowRateUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KilogramPerHour,
      Value = 165.0))
]
```

Length

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Metre,
      Value = 24.0))
]
```

]

PressureDifference

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{PressureAbsoluteUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

[

```
QualifiedValue(  
  Value = PhysicalQuantity(  
    Unit = Bar,  
    Value = 3.8))
```

]

VolumeFlow

Composition Property

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{VolumeFlowRateUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

[

```
QualifiedValue(  
  Value = PhysicalQuantity(  
    Unit = LitrePerSecond,  
    Value = 20.0))
```

]

Example in DEXPI XML

```
<Object type="Process/Process.TransportingFluids">  
  <Components property="Flow">  
    <Object type="Core/QualifiedValue">  
      <Data property="Value">  
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">  
          <Data property="Unit">  
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour  
↪"/>  
          </Data>
```

(continues on next page)

(continued from previous page)

```

        <Data property="Value">
            <Double>165.0</Double>
        </Data>
    </AggregatedDataValue>
</Data>
</Object>
</Components>
<Components property="Length">
    <Object type="Core/QualifiedValue">
        <Data property="Value">
            <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
                <Data property="Unit">
                    <DataReference data="Core/PhysicalQuantities.LengthUnit.Metre"/>
                </Data>
                <Data property="Value">
                    <Double>24.0</Double>
                </Data>
            </AggregatedDataValue>
        </Data>
    </Object>
</Components>
<Components property="PressureDifference">
    <Object type="Core/QualifiedValue">
        <Data property="Value">
            <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
                <Data property="Unit">
                    <DataReference data="Core/PhysicalQuantities.PressureAbsoluteUnit.Bar"/>
                </Data>
                <Data property="Value">
                    <Double>3.8</Double>
                </Data>
            </AggregatedDataValue>
        </Data>
    </Object>
</Components>
<Components property="VolumeFlow">
    <Object type="Core/QualifiedValue">
        <Data property="Value">
            <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
                <Data property="Unit">
                    <DataReference data="Core/PhysicalQuantities.VolumeFlowRateUnit.
↪LitrePerSecond"/>
                </Data>
                <Data property="Value">
                    <Double>20.0</Double>
                </Data>
            </AggregatedDataValue>
        </Data>
    </Object>
</Components>
</Object>

```


Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Centimetre,
      Value = 80.0))
]
```

Width**Composition Property**

Type: $\langle \text{QualifiedValue with Type} \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with UnitType} \rightarrow \text{LengthUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Centimetre,
      Value = 190.0))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.TransportingFluidsInChannel">
  <Components property="Depth">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
          </Data>
          <Data property="Value">
            <Double>80.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="Width">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
```

(continues on next page)

(continued from previous page)

```

    <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
  </Data>
  <Data property="Value">
    <Double>190.0</Double>
  </Data>
</AggregatedDataValue>
</Data>
</Object>
</Components>
</Object>

```

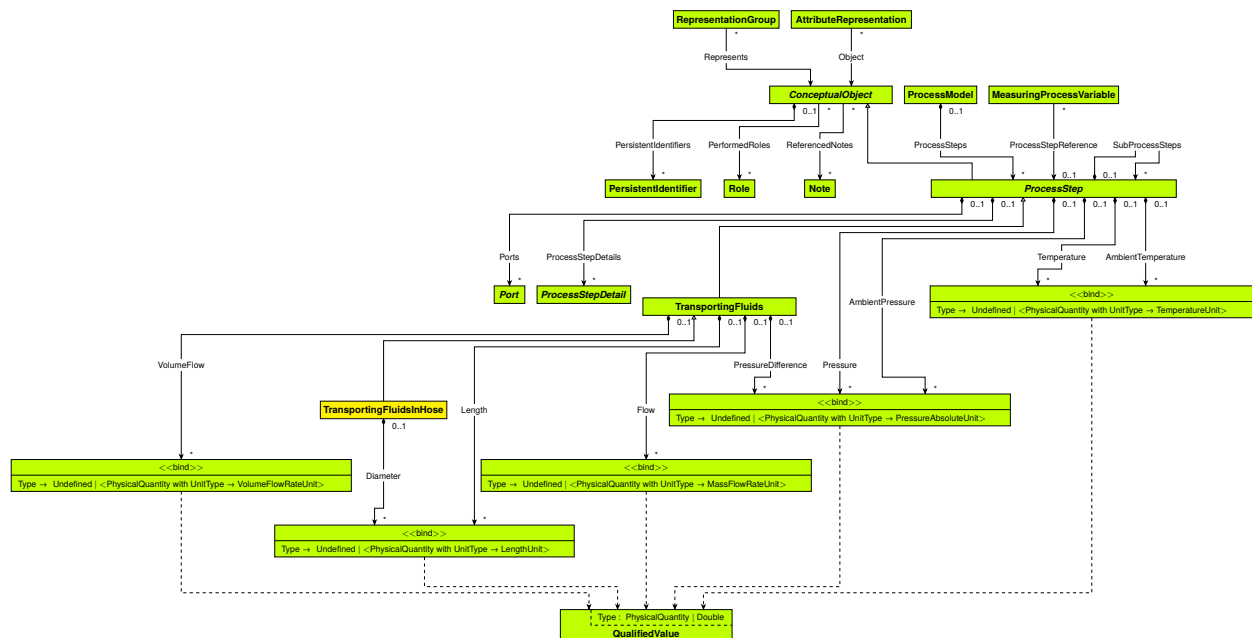
TransportingFluidsInHose

Process::Process::TransportingFluidsInHose

Overview

Concrete Class

This *TransportingFluids* process is realized by a flexible connector or hose. This type can be used to model hose or flexible piping systems.



Supertypes

- *TransportingFluids*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Diameter</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → LengthUnit>></i>	no

Diameter

Composition Property

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → LengthUnit>>*

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Centimetre,
      Value = 7.5))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.TransportingFluidsInHose">
  <Components property="Diameter">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
          </Data>
          <Data property="Value">
            <Double>7.5</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```

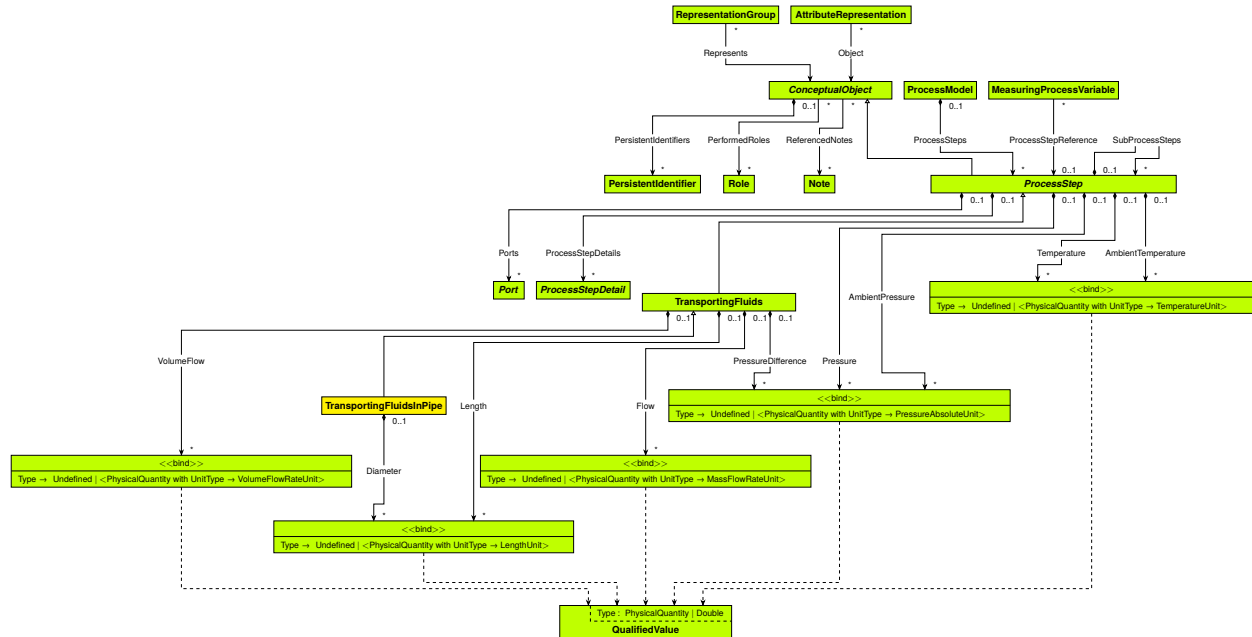
TransportingFluidsInPipe

Process::Process::TransportingFluidsInPipe

Overview

Concrete Class

This *TransportingFluids* process is realized by a rigid connector, duct or pipe. This type can be used to pipeline systems, piping-based fluid distribution systems and ducting systems.



Supertypes

- *TransportingFluids*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Diameter</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → LengthUnit>></i>	no

Diameter

Composition Property

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → LengthUnit>>*

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Centimetre,
      Value = 12.0))
]
```

Example in DEXPI XML

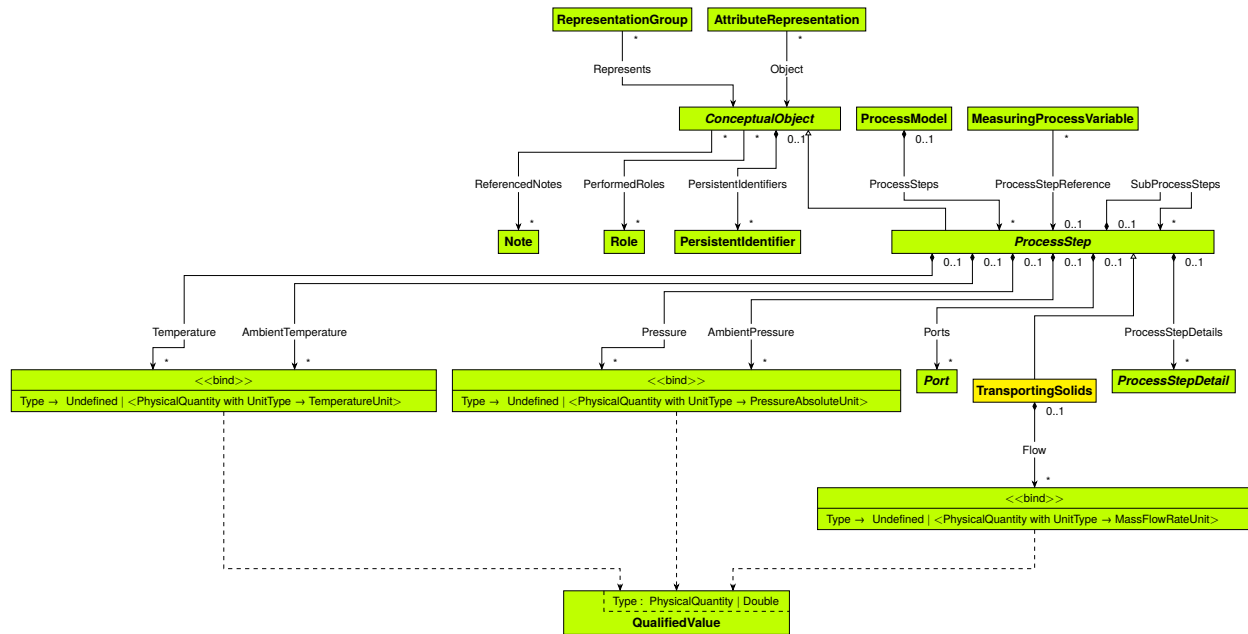
```
<Object type="Process/Process.TransportingFluidsInPipe">
  <Components property="Diameter">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.LengthUnit.Centimetre"/>
          </Data>
          <Data property="Value">
            <Double>12.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```

TransportingSolids

Process::Process::TransportingSolids

Overview**Concrete Class**

This process moves solid material between processes. Two child types are used to specify the realisation of this process, as the realization has implications for process design. The child types are *TransportingSolidsContinuously* and *TransportingSolidsDiscontinuously*.



Supertypes

- *ProcessStep*

Subtypes

- *TransportingSolidsContinuously*
- *TransportingSolidsDiscontinuously*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Flow</i>	*	< <i>QualifiedValue</i> with <i>Type</i> → <i>Undefined</i> < <i>PhysicalQuantity</i> with <i>UnitType</i> → <i>MassFlowRateUnit</i> >>	no

Flow

Composition Property

Type: <*QualifiedValue* with *Type* → *Undefined* | <*PhysicalQuantity* with *UnitType* → *MassFlowRateUnit*>>

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = KilogramPerHour,
      Value = 300.0))
]
```

Example in DEXPI XML

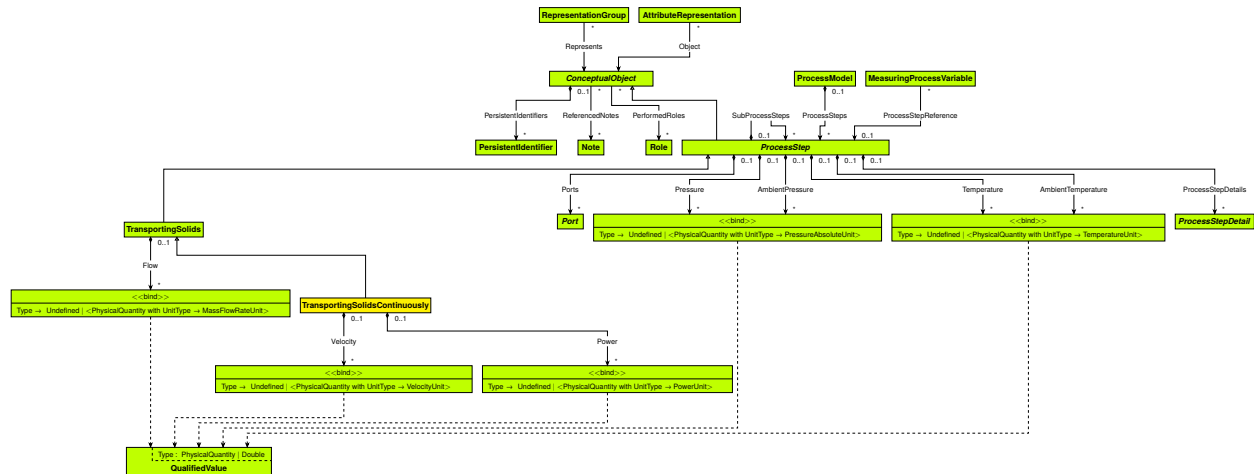
```
<Object type="Process/Process.TransportingSolids">
  <Components property="Flow">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassFlowRateUnit.KilogramPerHour
↪"/>
          </Data>
          <Data property="Value">
            <Double>300.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```

TransportingSolidsContinuously

Process::Process::TransportingSolidsContinuously

Overview**Concrete Class**

This process moves solid material between processes using a continuous operation, such as a conveyor of various types.



Supertypes

- *TransportingSolids*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Power</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → PowerUnit>></i>	no
<i>Velocity</i>	*	<i><QualifiedValue with Type → Undefined <PhysicalQuantity with UnitType → VelocityUnit>></i>	no

Power

Composition Property

Type: *<QualifiedValue with Type → Undefined | <PhysicalQuantity with UnitType → PowerUnit>>*

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Kilowatt,
      Value = 30.0))
]
```

Velocity

Composition Property

Type: $\langle \text{QualifiedValue with } Type \rightarrow \text{Undefined} \mid \langle \text{PhysicalQuantity with } UnitType \rightarrow \text{VelocityUnit} \rangle \rangle$

Multiplicity: *

Ordered: no

Example

```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = MetrePerSecond,
      Value = 2.0))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.TransportingSolidsContinuously">
  <Components property="Power">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.PowerUnit.Kilowatt"/>
          </Data>
          <Data property="Value">
            <Double>30.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
  <Components property="Velocity">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.VelocityUnit.MetrePerSecond"/>
          </Data>
          <Data property="Value">
            <Double>2.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```


Multiplicity: ***Ordered:** no**Example**

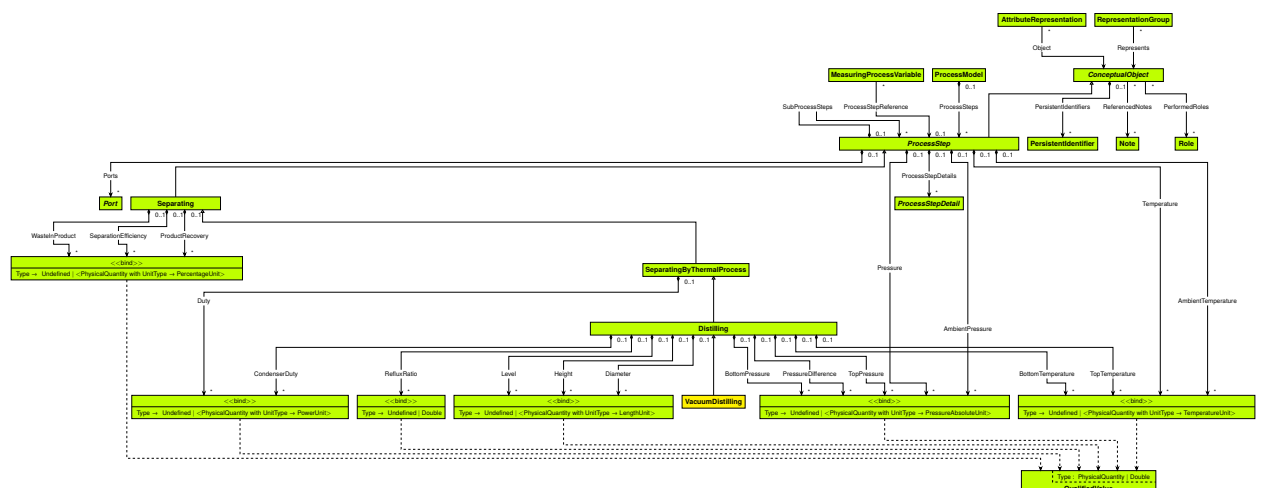
```
[
  QualifiedValue(
    Value = PhysicalQuantity(
      Unit = Kilogram,
      Value = 12.0))
]
```

Example in DEXPI XML

```
<Object type="Process/Process.TransportingSolidsDiscontinuously">
  <Components property="BatchSize">
    <Object type="Core/QualifiedValue">
      <Data property="Value">
        <AggregatedDataValue type="Core/PhysicalQuantities.PhysicalQuantity">
          <Data property="Unit">
            <DataReference data="Core/PhysicalQuantities.MassUnit.Kilogram"/>
          </Data>
          <Data property="Value">
            <Double>12.0</Double>
          </Data>
        </AggregatedDataValue>
      </Data>
    </Object>
  </Components>
</Object>
```

VacuumDistilling*Process::Process::VacuumDistilling***Overview****Concrete Class**

This is a *Distilling* process that is conducted at sub-atmospheric pressures.



Supertypes

- *Distilling*

Example in DEXPI XML

```
<Object type="Process/Process.VacuumDistilling"/>
```

Process::Process

Overview

Package

Members

Classes

Name	Type
<i>Absorbing</i>	Concrete Class
<i>Adsorbing</i>	Concrete Class
<i>Agglomerating</i>	Concrete Class
<i>Agitating</i>	Concrete Class
<i>BlowingDown</i>	Concrete Class
<i>Boiling</i>	Concrete Class
<i>CalculatingProcessVariable</i>	Abstract Class
<i>CalculatingRatio</i>	Concrete Class
<i>CalculatingSplitRange</i>	Concrete Class
<i>Coalescing</i>	Concrete Class
<i>Composition</i>	Concrete Class
<i>Compressing</i>	Concrete Class
<i>ContactingInPacking</i>	Concrete Class
<i>ContactingOnTray</i>	Concrete Class
<i>ControllingProcessVariable</i>	Concrete Class
<i>ConveyingSignal</i>	Concrete Class
<i>Cooling</i>	Concrete Class
<i>Crushing</i>	Concrete Class

continues on next page

Table 10 – continued from previous page

Name	Type
<i>Crystallizing</i>	Concrete Class
<i>CustomMaterialComponent</i>	Concrete Class
<i>Cutting</i>	Concrete Class
<i>Distilling</i>	Concrete Class
<i>Draining</i>	Concrete Class
<i>DrivingByEngine</i>	Concrete Class
<i>DrivingByMotor</i>	Concrete Class
<i>DrivingByTurbine</i>	Concrete Class
<i>Drying</i>	Concrete Class
<i>ElectricalEnergyFlow</i>	Concrete Class
<i>ElectricalEnergyPort</i>	Concrete Class
<i>Emitting</i>	Concrete Class
<i>EnergyFlow</i>	Concrete Class
<i>EnergyPort</i>	Concrete Class
<i>Evaporating</i>	Concrete Class
<i>ExchangingThermalEnergy</i>	Concrete Class
<i>Extruding</i>	Concrete Class
<i>FeedingMaterial</i>	Concrete Class
<i>Filtering</i>	Concrete Class
<i>Flaring</i>	Concrete Class
<i>Flocculating</i>	Concrete Class
<i>FormingSolidMaterial</i>	Concrete Class
<i>GeneratingACPower</i>	Concrete Class
<i>GeneratingDCPower</i>	Concrete Class
<i>GeneratingFlow</i>	Concrete Class
<i>GeneratingInFuelCell</i>	Concrete Class
<i>GeneratingSteam</i>	Concrete Class
<i>Grinding</i>	Concrete Class
<i>HeatingElectrical</i>	Concrete Class
<i>HeatingInFurnace</i>	Concrete Class
<i>Humidifying</i>	Concrete Class
<i>IncreasingParticleSize</i>	Concrete Class
<i>InformationFlow</i>	Concrete Class
<i>InformationPort</i>	Concrete Class
<i>InformationVariant</i>	Concrete Class
<i>InstrumentationActivity</i>	Abstract Class
<i>InstrumentationSystemActivity</i>	Concrete Class
<i>Kneading</i>	Concrete Class
<i>LimitingFlow</i>	Concrete Class
<i>ListOfMaterialComponents</i>	Concrete Class
<i>MaterialComponent</i>	Abstract Class
<i>MaterialPort</i>	Concrete Class
<i>MaterialState</i>	Concrete Class
<i>MaterialStateType</i>	Concrete Class
<i>MaterialTemplate</i>	Concrete Class
<i>MeasuringProcessVariable</i>	Concrete Class
<i>MechanicalEnergyFlow</i>	Concrete Class
<i>MechanicalEnergyPort</i>	Concrete Class
<i>Milling</i>	Concrete Class
<i>Mixing</i>	Concrete Class

continues on next page

Table 10 – continued from previous page

Name	Type
<i>MixingSimple</i>	Concrete Class
<i>Packaging</i>	Concrete Class
<i>Pelletizing</i>	Concrete Class
<i>Port</i>	Abstract Class
<i>PreventingBackflow</i>	Concrete Class
<i>ProcessConnection</i>	Abstract Class
<i>ProcessStep</i>	Abstract Class
<i>ProcessStepDetail</i>	Abstract Class
<i>Pumping</i>	Concrete Class
<i>PureMaterialComponent</i>	Concrete Class
<i>ReactingChemicals</i>	Concrete Class
<i>ReducingParticleSize</i>	Concrete Class
<i>RegulatingFlow</i>	Concrete Class
<i>RelievingOverpressure</i>	Concrete Class
<i>RelievingVacuum</i>	Concrete Class
<i>RelievingVacuumAndOverpressure</i>	Concrete Class
<i>RemovingThermalEnergy</i>	Concrete Class
<i>RotaryMixing</i>	Concrete Class
<i>Separating</i>	Concrete Class
<i>SeparatingByCentrifugalForce</i>	Concrete Class
<i>SeparatingByContact</i>	Concrete Class
<i>SeparatingByCyclonicMotion</i>	Concrete Class
<i>SeparatingByElectromagneticForce</i>	Concrete Class
<i>SeparatingByElectrostaticForce</i>	Concrete Class
<i>SeparatingByFlash</i>	Concrete Class
<i>SeparatingByGravity</i>	Concrete Class
<i>SeparatingByIonExchange</i>	Concrete Class
<i>SeparatingByMagneticForce</i>	Concrete Class
<i>SeparatingByPhaseSeparation</i>	Concrete Class
<i>SeparatingByPhysicalProcess</i>	Concrete Class
<i>SeparatingBySurfaceTension</i>	Concrete Class
<i>SeparatingByThermalProcess</i>	Concrete Class
<i>SeparatingMechanically</i>	Concrete Class
<i>ShuttingOffFlow</i>	Concrete Class
<i>Sieving</i>	Concrete Class
<i>Sink</i>	Concrete Class
<i>Skimming</i>	Concrete Class
<i>Source</i>	Concrete Class
<i>Splitting</i>	Abstract Class
<i>SplittingEnergy</i>	Concrete Class
<i>SplittingMaterial</i>	Concrete Class
<i>StabilizingDistilling</i>	Concrete Class
<i>StaticMixing</i>	Concrete Class
<i>SteeringFlow</i>	Abstract Class
<i>StoringElectricalEnergy</i>	Concrete Class
<i>StoringEnergy</i>	Abstract Class
<i>StoringFluids</i>	Concrete Class
<i>StoringInBattery</i>	Concrete Class
<i>StoringInPressureVessel</i>	Concrete Class
<i>StoringInSilo</i>	Concrete Class

continues on next page

Table 10 – continued from previous page

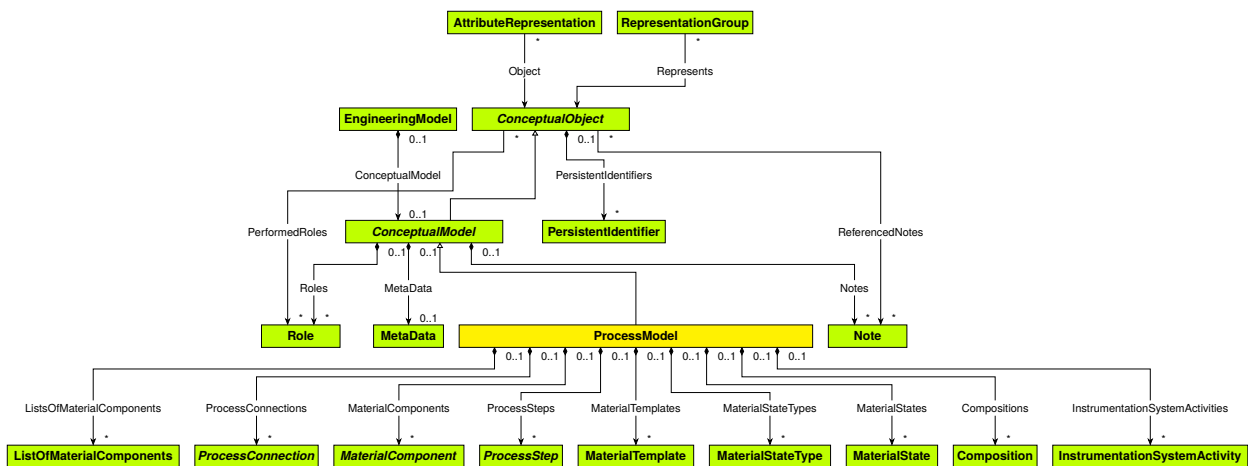
Name	Type
<i>StoringInTank</i>	Concrete Class
<i>StoringMaterial</i>	Abstract Class
<i>StoringSolids</i>	Concrete Class
<i>StoringThermalEnergy</i>	Concrete Class
<i>Stream</i>	Concrete Class
<i>StrippingDistilling</i>	Concrete Class
<i>SupplyingElectricalEnergy</i>	Concrete Class
<i>SupplyingFluids</i>	Concrete Class
<i>SupplyingMechanicalEnergy</i>	Concrete Class
<i>SupplyingSolids</i>	Concrete Class
<i>SupplyingThermalEnergy</i>	Concrete Class
<i>SupplyingThermalEnergyWithBurner</i>	Concrete Class
<i>ThermalEnergyFlow</i>	Concrete Class
<i>ThermalEnergyPort</i>	Concrete Class
<i>TransformingProcessVariable</i>	Concrete Class
<i>TransportingElectricalEnergy</i>	Concrete Class
<i>TransportingFluids</i>	Concrete Class
<i>TransportingFluidsInChannel</i>	Concrete Class
<i>TransportingFluidsInHose</i>	Concrete Class
<i>TransportingFluidsInPipe</i>	Concrete Class
<i>TransportingSolids</i>	Concrete Class
<i>TransportingSolidsContinuously</i>	Concrete Class
<i>TransportingSolidsDiscontinuously</i>	Concrete Class
<i>VacuumDistilling</i>	Concrete Class

3.3.3 ProcessModel

Process::ProcessModel

Overview

Concrete Class



Supertypes

- *ConceptualModel*

Members

Composition Properties

Name	Multiplicity	Type	Ordered
<i>Compositions</i>	*	<i>Composition</i>	no
<i>InstrumentationSystemActivities</i>	*	<i>InstrumentationSystemActivity</i>	no
<i>ListsOfMaterialComponents</i>	*	<i>ListOfMaterialComponents</i>	no
<i>MaterialComponents</i>	*	<i>MaterialComponent</i>	no
<i>MaterialStateTypes</i>	*	<i>MaterialStateType</i>	no
<i>MaterialStates</i>	*	<i>MaterialState</i>	no
<i>MaterialTemplates</i>	*	<i>MaterialTemplate</i>	no
<i>ProcessConnections</i>	*	<i>ProcessConnection</i>	no
<i>ProcessSteps</i>	*	<i>ProcessStep</i>	no

Compositions

Composition Property

The *Compositions* of the *ProcessModel*.

Type: *Composition*

Multiplicity: *

Ordered: no

InstrumentationSystemActivities

Composition Property

The *InstrumentationSystemActivities* of the *ProcessModel*.

Type: *InstrumentationSystemActivity*

Multiplicity: *

Ordered: no

ListsOfMaterialComponents

Composition Property

The *ListsOfMaterialComponents* of the *ProcessModel*.

Type: *ListOfMaterialComponents*

Multiplicity: *

Ordered: no

MaterialComponents**Composition Property**

The *MaterialComponents* of the *ProcessModel*.

Type: *MaterialComponent*

Multiplicity: *

Ordered: no

MaterialStateTypes**Composition Property**

The *MaterialStateTypes* of the *ProcessModel*.

Type: *MaterialStateType*

Multiplicity: *

Ordered: no

MaterialStates**Composition Property**

The *MaterialStates* of the *ProcessModel*.

Type: *MaterialState*

Multiplicity: *

Ordered: no

MaterialTemplates**Composition Property**

The *MaterialTemplates* of the *ProcessModel*.

Type: *MaterialTemplate*

Multiplicity: *

Ordered: no

ProcessConnections**Composition Property**

The *ProcessConnections* of the *ProcessModel*.

Type: *ProcessConnection*

Multiplicity: *

Ordered: no

ProcessSteps**Composition Property**

The *ProcessSteps* of the *ProcessModel*.

Type: *ProcessStep*

Multiplicity: *

Ordered: no

Example in DEXPI XML

```
<Object type="Process/ProcessModel"/>
```

Process

3.3.4 Overview

Model

URI: <https://data.dexpi.org/models/2.0.0/Process.xml>

[Process.xml](#)

Types for the representation of chemical processes and associated graphical representations.

3.3.5 Members

Packages

Name
<i>Enumerations</i>
<i>Process</i>

Classes

Name	Type
<i>ProcessModel</i>	Concrete Class

4.1 Glossary

4.1.1 Dimension of a Physical Quantity Type

A physical quantity type such as *mass*, *length* or *area* is characterized by its dimension (for example, see https://en.wikipedia.org/wiki/Physical_quantity). It is possible to express the dimension of any physical quantity type using the dimensions of some base quantity types. The choice of the base quantity types is arbitrary to a certain extent. The base quantity types that underly the International System of Units are listed below.

Base Quantity Type	Dimension Symbol
amount of substance	N
electric current	I
length	L
luminous intensity	J
mass	M
thermodynamic temperature	Θ (Greek capital letter Theta)
time	T

Using the symbols given in the table, the dimension of *length* is L, and the dimension of *area* is L·L, or L².

4.1.2 Unified Modeling Language

The Unified Modeling Language (UML) is a general-purpose modeling language for the *analysis, design, and implementation of software-based systems*. Within this specification, we use the notation [UML:x] to refer to section x of *OMG® Unified Modeling Language® (OMG UML®), Version 2.5.1* (see <https://www.omg.org/spec/UML/2.5.1>).

4.2 Reference P&ID

The DEXPI Reference P&ID serves to demonstrate the features of the DEXPI Plant Model. A graphical representation of the P&ID is shown below.

