



FIRST PAGE LEFT LEFT BLANK

Literate Data Model

BLANK

Preliminaries

the basic structure of the model

In Literate Data Modeling, the main components of interest are typically Classes, Attributes, Models, and Subjects. However, to streamline the model and promote reusability, we introduce a supertype called Component. By defining common attributes and behaviors in the Component class, we can inherit them in the subclasses, ensuring consistency and reducing duplication throughout the model.

We present the Component class first because it is a best practice in modeling to introduce supertypes before their subtypes. This approach allows readers to understand the general concepts and shared properties before delving into the specifics of each specialized component.

Preliminaries

Component
An element or building block of the literate data model

PLURAL Components

IMPL PLURAL Components

DEPENDENTS [Annotation](#)

SUBTYPES [LiterateModel](#), [Subject](#), [Class](#), [Key](#), [AttributeSection](#), [Attribute](#), [Constraint](#), [Method](#), [Parameter](#)

Name	the name of the component, not in camel case
	(String value O_O)

warning This is a warning with emoji

name	The name of the component
	(CamelName value O_O)

Name	(QualifiedCamel value O_O)
-------------	--

Name	a short form of the component's name, used for cross references and improved readability.
	(CamelName value O_O)

example "LDM" is the short form of "Literate Data Model".

DEFAULT name - how do you say name in english?

OCL x.name == y

CONSTRAINTS the abbreviated name should be shorter than the actual name

OCL len(abbreviatedName) < len(name)

MESSAGE Why have an abbreviation longer than the name?

SEVERITY Warning

note Does this annotation find it's way to the Constraint? YES! It's fixed!

OneLiner	A brief, one-line definition or description of the component, suitable for use in a descriptive table of contents. _
	(OneLiner value O_O)

Description	A more detailed explanation or discussion of the component _
	(RichText value O_O)

Embellishment	Indicates whether this component is an embellishment added during post-parsing processing _
	(Boolean value O_O)

DEFAULT false

note This attribute is set to true for components that are automatically generated or added during the fleshing out, review, or rendering processes, such as implied attributes or suggested model elements. It helps distinguish embellishments from the core model elements defined in the original LDM source.

	mechanical attributes	
	Indicates whether this component is an embellishment added during post-parsing processing _	
		(<u>Boolean</u> value O_O)
DEFAULT	false	
note	This attribute is set to true for components that are automatically generated or added during the fleshing out, review, or rendering processes, such as implied attributes or suggested model elements. It helps distinguish embellishments from the core model elements defined in the original LDM source.	

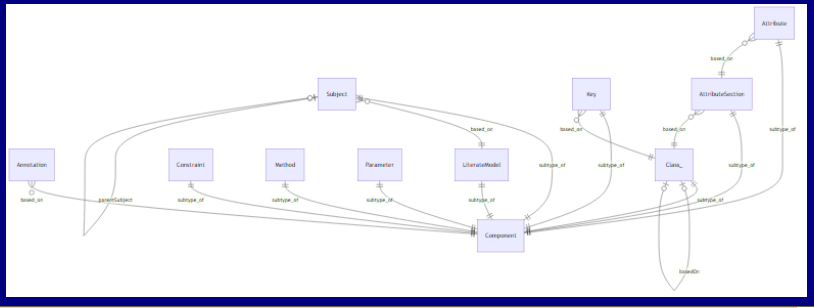
```
Mermaid ER Diagram for Component - Inert

erDiagram
    Annotation }o--|| Component : based_on
    LiterateModel ||--|| Component : subtype_of
    Subject ||--|| Component : subtype_of
    Subject }o--|| LiterateModel : based_on
    Subject |o--o| Subject : parentSubject
    Class_ ||--|| Component : subtype_of
    Class_ |o--o| Class_ : basedOn
    Key ||--|| Component : subtype_of
    Key }o--|| Class_ : based_on
    AttributeSection ||--|| Component : subtype_of
    AttributeSection }o--|| Class_ : based_on
    Attribute ||--|| Component : subtype_of
    Attribute }o--|| AttributeSection : based_on
    Constraint ||--|| Component : subtype_of
    Method ||--|| Component : subtype_of
    Parameter ||--|| Component : subtype_of
```

```
Mermaid ER Diagram for Component - Live!

erDiagram
    Annotation }o--|| Component : based_on
    LiterateModel ||--|| Component : subtype_of
    Subject ||--|| Component : subtype_of
    Subject }o--|| LiterateModel : based_on
    Subject |o--o| Subject : parentSubject
    Class_ ||--|| Component : subtype_of
    Class_ |o--o| Class_ : basedOn
    Key ||--|| Component : subtype_of
    Key }o--|| Class_ : based_on
    AttributeSection ||--|| Component : subtype_of
    AttributeSection }o--|| Class_ : based_on
    Attribute ||--|| Component : subtype_of
    Attribute }o--|| AttributeSection : based_on
    Constraint ||--|| Component : subtype_of
    Method ||--|| Component : subtype_of
    Parameter ||--|| Component : subtype_of
```


Mermaid ER Diagram for Component - PNG for mermaid



	AnnotationType a kind of note, or aside, used to call attention to additional information about some Component.
note	Each LDM declares a set of Annotation Types, with defined labels, emojis, and clearly documented purposes. These are <i>recognized</i> or <i>registered</i> Annotation Types.
PLURAL	AnnotationTypes
DEPENDSON	LiterateModel
emoji	an emoji (Emoji value O_O)
name	an emoji (String value O_O)
unicode	the Unicode for the emoji (String value O_O)
label	A short label to indicate the purpose of the annotation (LowerCamel value O_O)
plural	the plural form of the label (UpperCamel value O_O)
DEFAULT	based on label
reason	the intended reason for the annotation. (OneLiner value O_O)
isCreatedBy	created for AnnotationType
dependsOn	A link back to the LiterateModel on which this AnnotationType depends. (LiterateModel value M_1)
inverseAttribute	inverse attribute for Annotation.annotationType from which this was implied. (Annotation value M_1)
VERSE	Annotation.annotationType

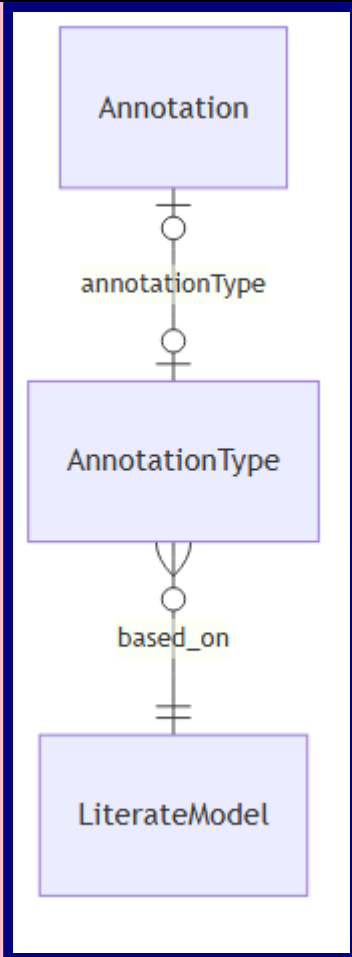
Mermaid ER Diagram for AnnotationType - Inert

```
erDiagram
    AnnotationType }o--|| LiterateModel : based_on
    Annotation |o--o| AnnotationType : annotationType
```

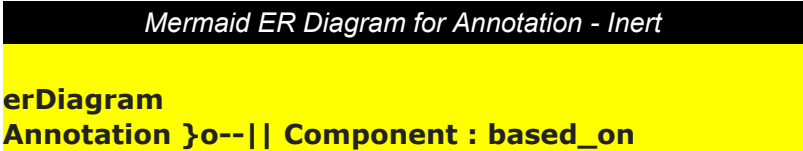
Mermaid ER Diagram for AnnotationType - Live!

```
erDiagram
    AnnotationType }o--|| LiterateModel : based_on
    Annotation |o--o| AnnotationType : annotationType
```

Mermaid ER Diagram for AnnotationType - PNG for mermaid



Annotation	
A note or comment associated with a model element	
Annotations	
Annotations	
Component	
type	(Optional AnnotationType value O_O)
note	An Annotation is considered to <i>recognized</i> if the label is associated with an Annotation Type. otherwise it is <i>ad hoc</i> .
note	Should be a Value Type
AnnotationType.inverseOfAnnotationType	
VERSE	
label	A short label to indicate the purpose of the annotation _
(CamelName value O_O)	
But any short label is valid.	
from annotationType	
emoji	(Optional Emoji value O_O)
from annotation type	
content	The content or body of the annotation
(RichText value O_O)	
embellishment	Indicates whether this annotation is an embellishment added during post-parsing processing _
(Boolean value O_O)	
false	
note	This attribute is set to true for annotations that are automatically generated or added during the fleshing out, review, or rendering processes, such as suggestions, issues, or diagnostic messages. It helps distinguish embellishment annotations from the annotations defined in the original LDM source.
created for Annotation	
based_on	A link back to the Component on which this Annotation depends.
(Component value M_1)	

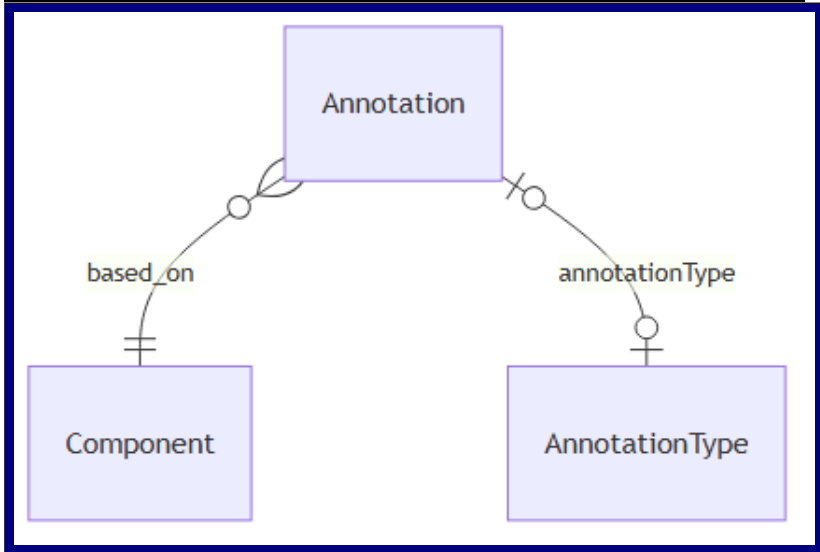


Annotation |o--o| AnnotationType : annotationType

Mermaid ER Diagram for Annotation - Live!

erDiagram
Annotation }o--|| Component : based_on Annotation
|o--o| AnnotationType : annotationType

Mermaid ER Diagram for Annotation - PNG for mermaid



BLANK

The Model and its Subjects

	LiterateModel A representation of a domain's entities, attributes, and relationships, along with explanatory text and examples
PLURAL	LiterateModels
DEPENDENTS	AnnotationType , Subject , SubjectArea
TYPEOF	Component
name	(UpperCamel value O_O)
OVERIDES	Component.name
classes	list of all classes in the model, as ordered in the definition of the model. (List of Classes value O_O)
INVERSE	Class.inverseOfAllSubjects
INVARIATION	gathering s.allSubjects over s in subjectAreas
CONSTRAINTS	Subject names must be unique across the model.
classes	list of all classes in the model, as ordered in the definition of the model. (List of Classes value O_O)
INVERSE	Class.inverseOfAllClasses
INVARIATION	gathering s.allClasses over s in allSubjects.
CONSTRAINTS	Class names must be unique across the model.
AnnotationTypes	(List of AnnotationTypes value O_O)
CodingLanguage	the recommended language for expressing derivation, defaults, and constraints (CodingLanguage value O_O)
DEFAULT	OCL
CodingLanguages	(Optional List of CodingLanguages value O_O)
TemplateLanguage	the recommended language for expressing derivation, defaults, and constraints (TemplateLanguage value O_O)
DEFAULT	Handlebars
TemplateLanguages	(Optional List of TemplateLanguages value O_O)
Functions	A list of functions that require sophisticated AI-powered implementation * (List of String value O_O)
INVARIATION	[aiEnglishPlural()]

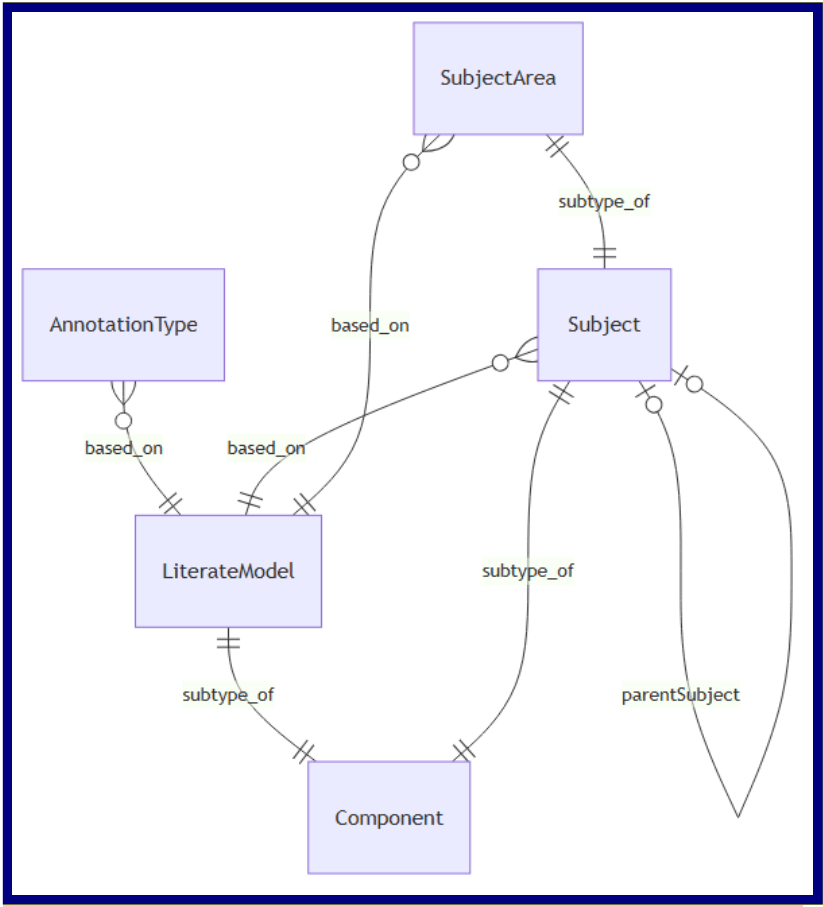
Mermaid ER Diagram for LiterateModel - Inert

```
erDiagram
    AnnotationType }o--|| LiterateModel : based_on
    LiterateModel ||--|| Component : subtype_of
    Subject ||--|| Component : subtype_of
    Subject }o--|| LiterateModel : based_on
    Subject |o--o| Subject : parentSubject
    SubjectArea ||--|| Subject : subtype_of
    SubjectArea }o--|| LiterateModel : based_on
```

Mermaid ER Diagram for LiterateModel - Live!

```
erDiagram
    AnnotationType }o--|| LiterateModel : based_on
    LiterateModel ||--|| Component : subtype_of
    Subject ||--|| Component : subtype_of
    Subject }o--|| LiterateModel : based_on
    Subject |o--o| Subject : parentSubject
    SubjectArea ||--|| Subject : subtype_of
    SubjectArea }o--|| LiterateModel : based_on
```

Mermaid ER Diagram for LiterateModel - PNG for mermaid



The Model and its Subjects

Subject

A specific topic or theme within the model

Subjects are the chapters an sections of the model.

- A subject need not contain any Classes if it's just expository.

PLURAL Subjects
BASEDON [LiterateModel](#)
BTYPEOF [Component](#)
SUBTYPES [SubjectArea](#)

name ([UpperCamel](#) value O_O)

VERRIDES [Component.name](#)

parentSubject The parent subject, if any, under which this subject is nested _
([Optional Subject](#) value O_O)

INVERSE [Subject.inverseOfParentSubject](#)

classes The major classes related to this subject, in the order in which they should be presented _
([List of Classes](#) value O_O)

issue define chapter, section, subsection as levels?

INVERSE [Class.inverseOfClasses](#)

childSubjects Any child subjects nested under this subject, in the order in which they should be presented _
([List of Subjects](#) value O_O)

DSL : the Classes within a Subject are always displayed before the childSubjects.

INVERSE [Subject.parentSubject](#)

created created for Subject

Model A link back to the LiterateModel on which this Subject depends.
([LiterateModel](#) value M_1)

parentSubject Inverse attribute for Subject.parentSubject from which this was implied.
([Subject](#) value M_1)

INVERSE [Subject.parentSubject](#)

Mermaid ER Diagram for Subject - Inert

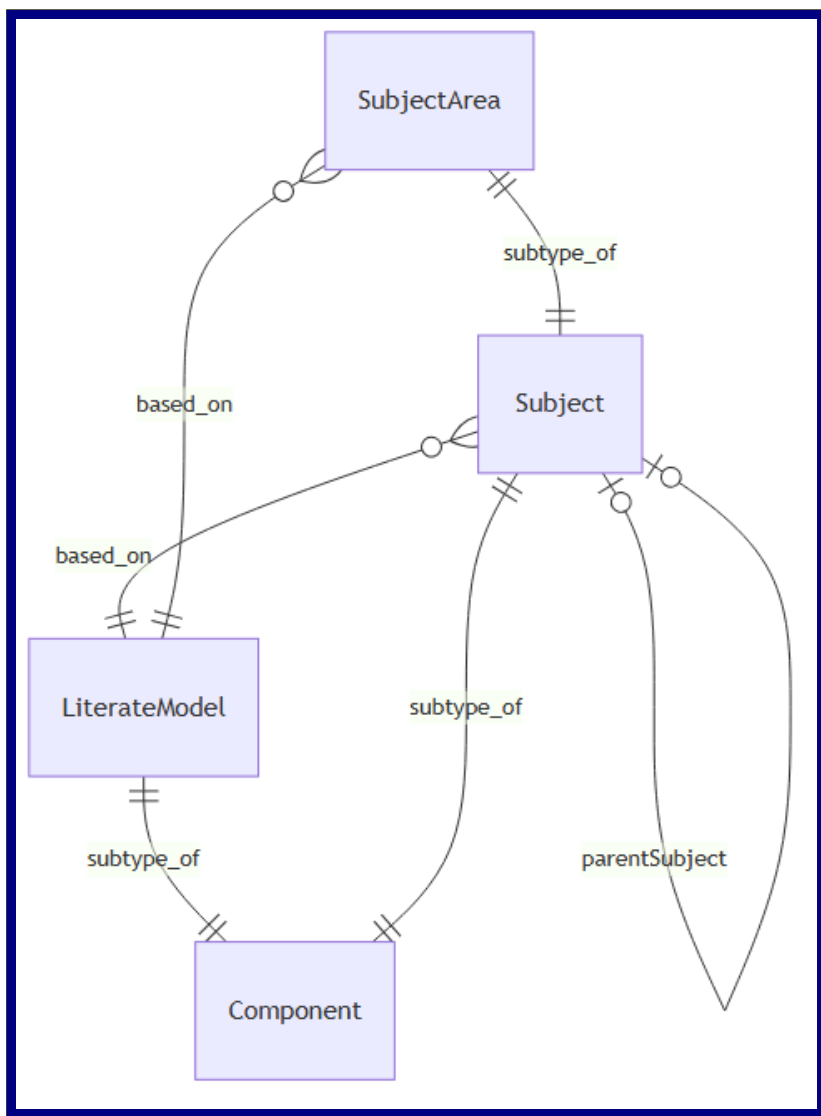
erDiagram

LiterateModel ||--|| Component : subtype_of
Subject ||--|| Component : subtype_of
Subject }o--|| LiterateModel : based_on
Subject |o--o| Subject : parentSubject
SubjectArea ||--|| Subject : subtype_of
SubjectArea }o--|| LiterateModel : based_on

Mermaid ER Diagram for Subject - Live!

```
erDiagram
    LiterateModel ||--|| Component : subtype_of
    Subject ||--|| Component : subtype_of
    Subject }o--|| LiterateModel : based_on
    Subject |o--o| Subject : parentSubject
    SubjectArea ||--|| Subject : subtype_of
    SubjectArea }o--|| LiterateModel : based_on
```

Mermaid ER Diagram for Subject - PNG for mermaid



SubjectArea
A main topic or area of focus within the model, containing related subjects and classes

WHERE parentSubject is absent
LURAL SubjectAreas
BASEDON [LiterateModel](#)
TYPEOF [Subject](#)

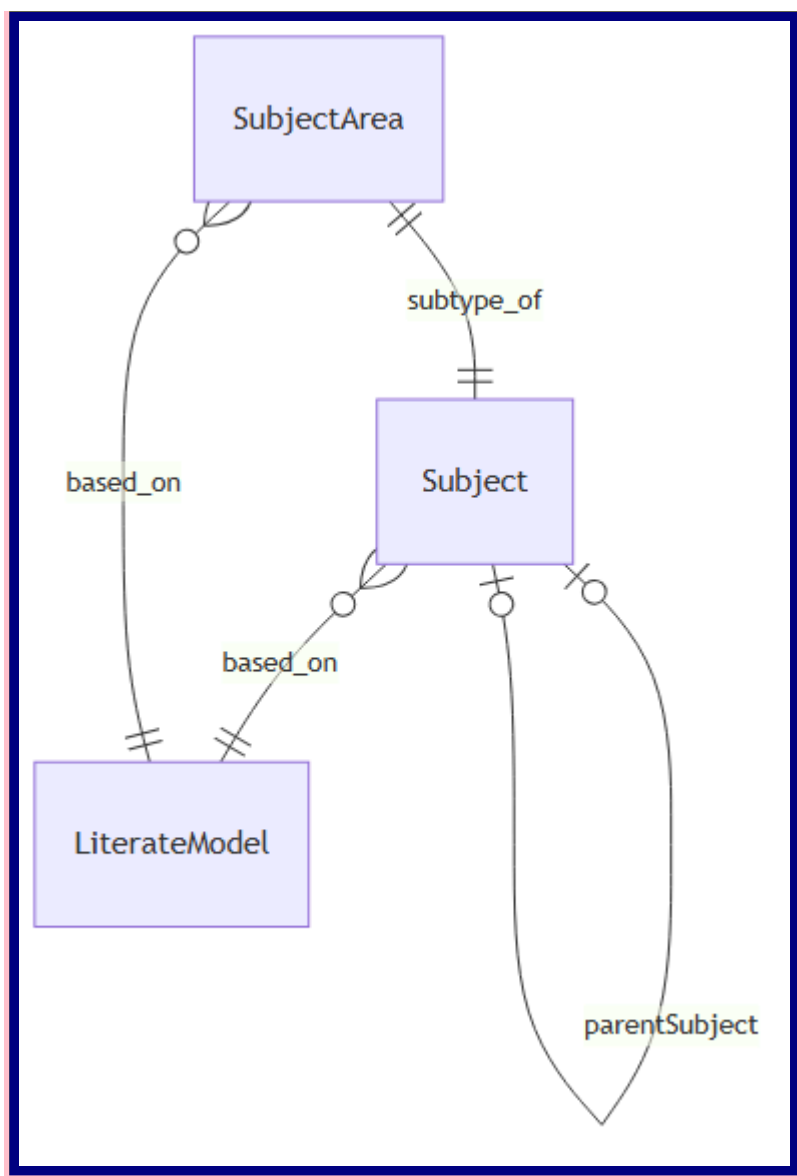
created for SubjectArea
A link back to the LiterateModel on which this SubjectArea depends. (LiterateModel value <i>M_1</i>)

Mermaid ER Diagram for SubjectArea - Inert

```
erDiagram
    Subject }o--|| LiterateModel : based_on
    Subject |o--o| Subject : parentSubject
    SubjectArea ||--|| Subject : subtype_of
    SubjectArea }o--|| LiterateModel : based_on
```

```
erDiagram
    Subject }o--|| LiterateModel : based_on
    Subject |o--o| Subject : parentSubject
    SubjectArea ||--|| Subject : subtype_of
    SubjectArea }o--|| LiterateModel : based_on
```

Mermaid ER Diagram for SubjectArea - PNG for mermaid



Classes

Classes

Class

A key entity or object type in the model, often corresponding to a real-world concept

PLURAL	Classes
DEPENDENTS	Subtyping , Key , AttributeSection , ClassConstraint
BTYEOF	Component
SUBTYPES	ReferenceType
STRAINTS	Within each Class, attribute names must be unique.

Form the normal English plural form of the name of the Class

([UpperCamel](#) value O_O)

Might be Books for the Book class or other regular plurals.

- But also might be People for Person.

note When inputting a model, you will rarely need to specify the plural form. The input program will just look it up.

DEFAULT the regular plural, formed by adding "s" or "es".

basedOn the Class or Classes on which this class is dependent

([Set of](#) [Class](#) value O_O)

This is solely based on **Existence Dependency**. A true dependent entity cannot logically exist without the related parent entity. For instance, an Order Item cannot exist without an Order. If removing the parent entity logically implies removing the dependent entity, then it is a dependent entity.

note that basedOn and dependentOf are being used synonymously in this metamodel.

INVERSE [Class.inverseOfBasedOn](#)

types The parent class or classes from which this class inherits attributes

([List of](#) [Classes](#) value O_O)

INVERSE [Class.inverseOfSupertypes](#)

typings the criteria, or dimensions, by which the class can be divided into subtypes

([List of](#) [Subtypings](#) value O_O)

example in a library model, the `Book` class could have subtypings based on genre (e.g., Fiction, Non-fiction), format (e.g., Hardcover, Paperback), or subject (e.g., Science, History).

INVERSE [Subtyping.inverseOfSubtypings](#)

es	Any subtypes or specializations of this class based on it's subtypings. (List of <u>Classes</u> value O_O)
ample	For instance, using the Book example, the subtypes could include FictionBook , Non-fictionBook , HardcoverBook , PaperbackBook , ScienceBook , and HistoryBook .
VERSE	<u>Class.inverseOfSubtypes</u>
es	The attributes or properties of the class, in the order in which they should be presented _ (List of <u>Attributes</u> value O_O)
VERSE	<u>Attribute.inverseOfAttributes</u>
ns	additional attributes or properties of the class, grouped for clarity and elaboration. _ (List of <u>AttributeSections</u> value O_O)
VERSE	<u>AttributeSection.inverseOfAttributeSections</u>
nts	Any constraints, rules, or validations specific to this class _ (List of <u>Constraints</u> value O_O)
note	Constraints may be expressed on either the Class or the Attribute. Always?
ds	Any behaviors or operations associated with this class _ (List of <u>Methods</u> value O_O)
VERSE	<u>Method.inverseOfMethods</u>
s	
nts	the Classes which are basedOn this Class (Optional Set of <u>Classes</u> value O_O)
VERSE	<u>Class.basedOn</u>
ys	(Optional Set of <u>UniqueKeys</u> value O_O)
VERSE	<u>UniqueKey.basedOn</u>
ects	Inverse attribute for LiterateModel.allSubjects from which this was implied. (<u>LiterateModel</u> value M_1)
VERSE	<u>LiterateModel.allSubjects</u>
ses	Inverse attribute for LiterateModel.allClasses from which this was implied. (<u>LiterateModel</u> value M_1)
VERSE	<u>LiterateModel.allClasses</u>
es	Inverse attribute for Subject.classes from which this was implied. (<u>Subject</u> value M_1)

INVERSE	Subject.classes
basedOn	Inverse attribute for Class.basedOn from which this was implied. (Class value M_1)
INVERSE	Class.basedOn
supertypes	Inverse attribute for Class.supertypes from which this was implied. (Class value M_1)
INVERSE	Class.supertypes
subtypes	Inverse attribute for Class.subtypes from which this was implied. (Class value M_1)
INVERSE	Class.subtypes
classes	Inverse attribute for Subtyping.classes from which this was implied. (Subtyping value M_1)
INVERSE	Subtyping.classes
coreClass	Inverse attribute for SimpleDataTypeSubtpeOfDataType.coreClass from which this was implied. (SimpleDataTypeSubtpeOfDataType value M_1)
INVERSE	SimpleDataTypeSubtpeOfDataType.coreClass

Mermaid ER Diagram for Class_ - Inert

erDiagram

```

Class_ ||--|| Component : subtype_of
Class_ |o--o| Class_ : basedOn
Subtyping }o--|| Class_ : based_on
ReferenceType ||--|| Class_ : subtype_of
Key ||--|| Component : subtype_of
Key }o--|| Class_ : based_on
AttributeSection ||--|| Component : subtype_of
AttributeSection }o--|| Class_ : based_on
ClassConstraint }o--|| Class_ : based_on

```

Mermaid ER Diagram for Class_ - Live!

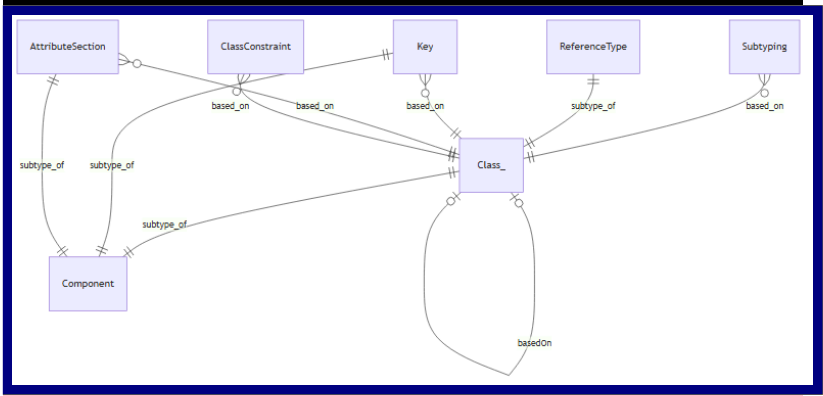
```

erDiagram
Class_ ||--|| Component : subtype_of
Class_ |o--o| Class_ : basedOn
Subtyping }o--|| Class_ : based_on
ReferenceType ||--|| Class_ : subtype_of
Key ||--|| Component : subtype_of
Key }o--|| Class_ : based_on
AttributeSection ||--|| Component : subtype_of
AttributeSection }o--|| Class_ : based_on

```

ClassConstraint }o--|| Class_ : based_on

Mermaid ER Diagram for Class_ - PNG for mermaid



Classes

Subtyping
a way in which subtypes of a Class may be classified

PLURAL Subtypings
MEDPLURAL Subtypings
BASEDON [Class](#)

name ([LowerCamel](#) value O_O)

exclusive ([Boolean](#) value O_O)

DEFAULT true

exclusive ([Boolean](#) value O_O)

DEFAULT true

classes (List of [Classes](#) value O_O)

DSL : Shown in the DSL as

- Subtypes: byBrand - Brand1, Brand2,... (non exclusive, exhaustive)
- on the super class. And as
 - Subtype of: SuperClass byBrand
- on the subclass.

note every class can have an unnamed subtyping.
INVERSE [Class.inverseOfClasses](#)

notes created for Subtyping

subtypings Inverse attribute for Class.subtypings from which this was implied.
([Class](#) value M_1)

INVERSE [Class.subtypings](#)

Class A link back to the Class on which this Subtyping depends.
([Class](#) value M_1)

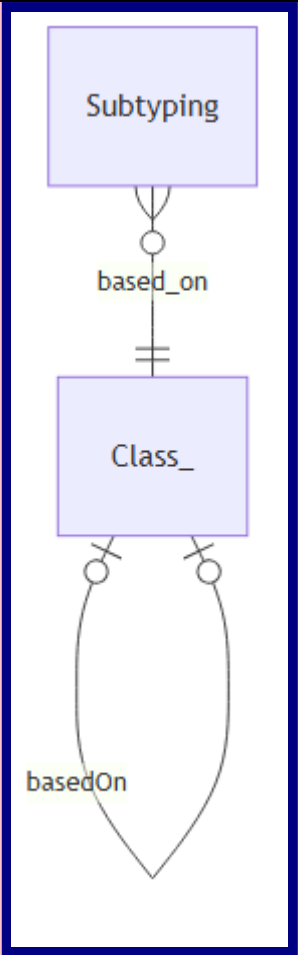
Mermaid ER Diagram for Subtyping - Inert

erDiagram
Class_ ||--o| Class_ : basedOn
Subtyping }o--|| Class_ : based_on

Mermaid ER Diagram for Subtyping - Live!

```
erDiagram
    Class_ ||--o| Class_ : basedOn Subtyping
    Class_ ||--o| Class_ : based_on
```

Mermaid ER Diagram for Subtyping - PNG for mermaid



Classes

ReferenceType

A class that is presumed to be used as a reference, rather than a value

PLURAL ReferenceTypes
MEDPLURAL ReferenceTypes
BTYPEOF [Class](#)

Mermaid ER Diagram for ReferenceType - Inert

erDiagram

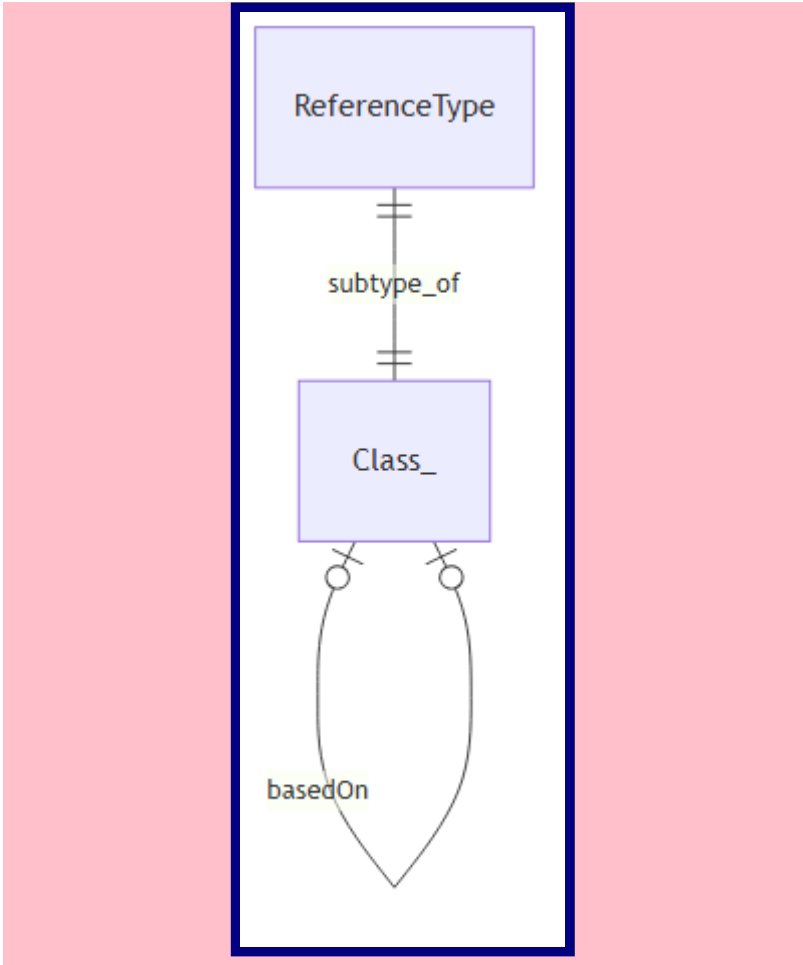
Class_ ||o--o| Class_ : basedOn

ReferenceType ||--|| Class_ : subtype_of

Mermaid ER Diagram for ReferenceType - Live!

**erDiagram Class_ ||o--o| Class_ : basedOn ReferenceType ||--||
Class_ : subtype_of**

Mermaid ER Diagram for ReferenceType - PNG for mermaid



CodeType

A data type or enumeration used in the model

LURAL CodeTypes
DPLURALCodeTypes
DENTS [CodeValue](#)

ve the code type was implied by use in an attribute and is only used for that attribute

([Boolean](#) value 0_0)

Mermaid ER Diagram for CodeType - Inert

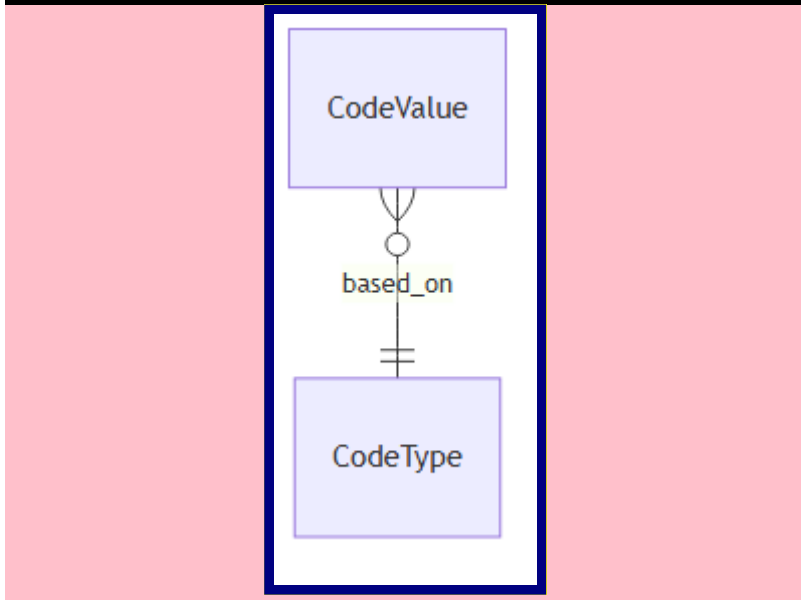
erDiagram

CodeValue }o--|| CodeType : based_on

Mermaid ER Diagram for CodeType - Live!

```
erDiagram CodeValue }o--|| CodeType : based_on
```

Mermaid ER Diagram for CodeType - PNG for mermaid



CodeValue

A possible value for an enumerated data class

PLURAL CodeValues

IMMEDPLURALCodeValues

BASEDON [CodeType](#)

code A short code or abbreviation for the value _

([String](#) value O_O)

description an explanation of what the code means

([RichText](#) value O_O)

note Often, a CodeType will be assigned to just one attribute in the model. In such cases, there's no need to declare a new Code Type and invent a name for it. Instead:

notes created for CodeValue

CodeType A link back to the CodeType on which this CodeValue depends.

([CodeType](#) value M_1)

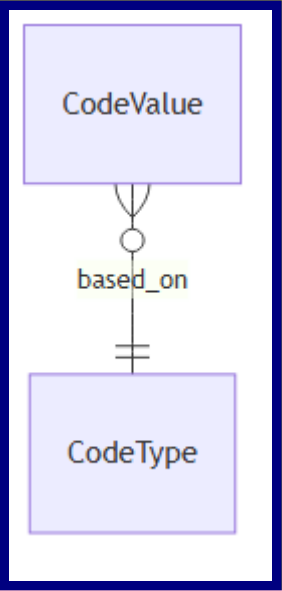
Mermaid ER Diagram for CodeValue - Inert

erDiagram
CodeValue }o--|| CodeType : based_on

Mermaid ER Diagram for CodeValue - Live!

erDiagram CodeValue }o--|| CodeType : based_on

Mermaid ER Diagram for CodeValue - PNG for mermaid



Classes

Key
a list of attributes of a class

PLURAL Keys
IMEDPLURAL Keys
BASED ON [Class](#)
BTYPED OF [Component](#)
SUBTYPES [UniqueKey](#)

Attributes	the attributes of the base Class.
	(List of Attributes value O_0)

INVERSE [Attribute.inverseOfKeyAttributes](#)
CONSTRAINTS each attribute must be a direct or inherited of the base class.
CONSTRAINTS no repetitions allowed in keyAttributes

👉 **Issue** : introduce PureLists?

issue need ascending descending to support index keys or ordering keys.

Attributes	created for Key
-------------------	-----------------

Class	A link back to the Class on which this Key depends.
	(Class value M_1)

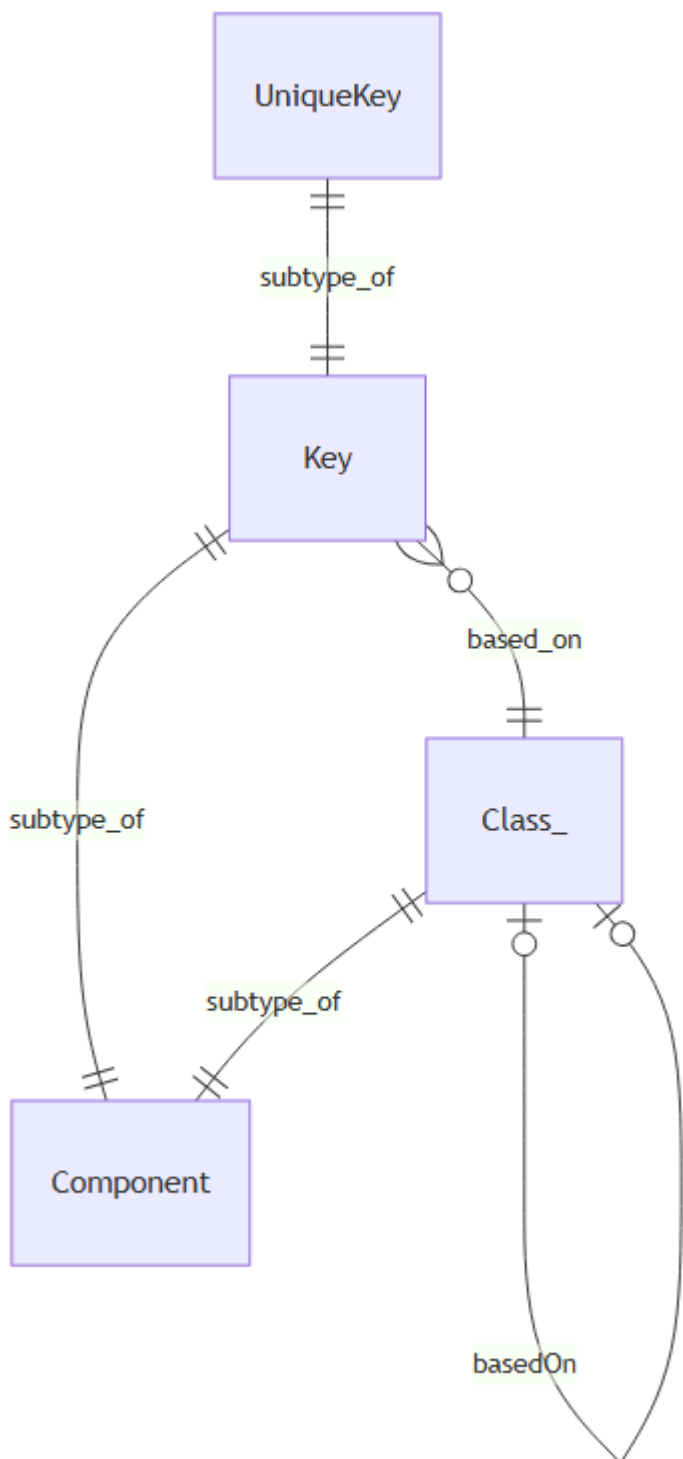
Mermaid ER Diagram for Key - Inert

```
erDiagram
    Class_ ||--|| Component : subtype_of
    Class_ |o--o| Class_ : basedOn
    Key ||--|| Component : subtype_of
    Key }o--|| Class_ : based_on
    UniqueKey ||--|| Key : subtype_of
```

Mermaid ER Diagram for Key - Live!

```
erDiagram
    Class_ ||--|| Component : subtype_of
    Class_ |o--o| Class_ : basedOn
    Key ||--|| Component : subtype_of
    Key }o--|| Class_ : based_on
    UniqueKey ||--|| Key : subtype_of
```

Mermaid ER Diagram for Key - PNG for mermaid



UniqueKey

a list of attributes on which instances of the base class may be keyed.

note order unimportant for Unique Keys.

PLURAL UniqueKeys

MEDPLURAL UniqueKeys

BTYPEOF [Key](#)

Mermaid ER Diagram for UniqueKey - Inert

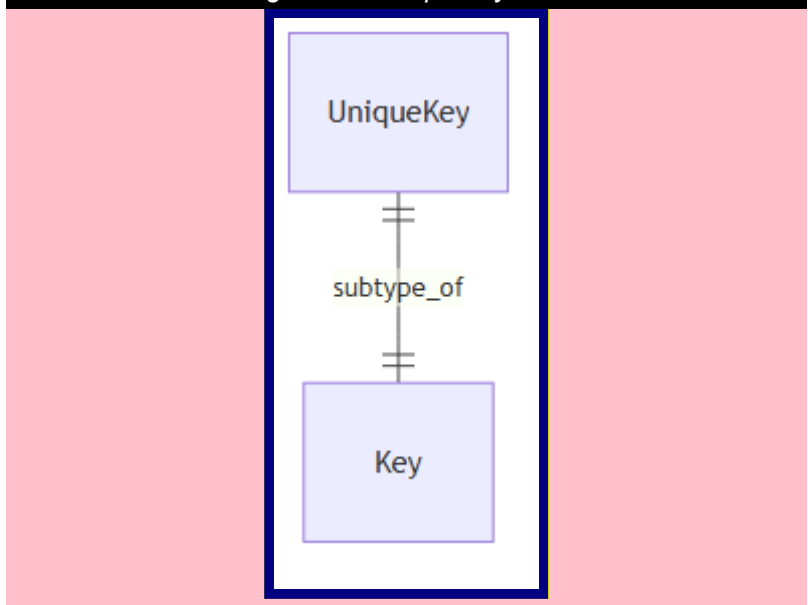
erDiagram

UniqueKey ||--|| Key : subtype_of

Mermaid ER Diagram for UniqueKey - Live!

erDiagram UniqueKey ||--|| Key : subtype_of

Mermaid ER Diagram for UniqueKey - PNG for mermaid



BLANK

Attributes

AttributeSection

a group of attributes for a class that merit a shared explanation.

LURAL AttributeSections
ADPLURAL AttributeSections
SEDON [Class](#)
DENTS [Attribute](#)
YPOF [Component](#)

whether the attributes in this section, taken together, are optional.

([Boolean](#) value O_O)

If the Attribute Section is required, then each Attribute within the sectional is optional or required, depending on how it is marked.

- But if the Attribute Section is optional each attribute in the section is only required if any attribute in the section is present.

created for AttributeSection

AttributeSections reverse attribute for Class.attributeSections from which this was implied.

([Class](#) value M_1)

VERSE [Class.attributeSections](#)

ss A link back to the Class on which this AttributeSection depends.

([Class](#) value M_1)

Mermaid ER Diagram for AttributeSection - Inert

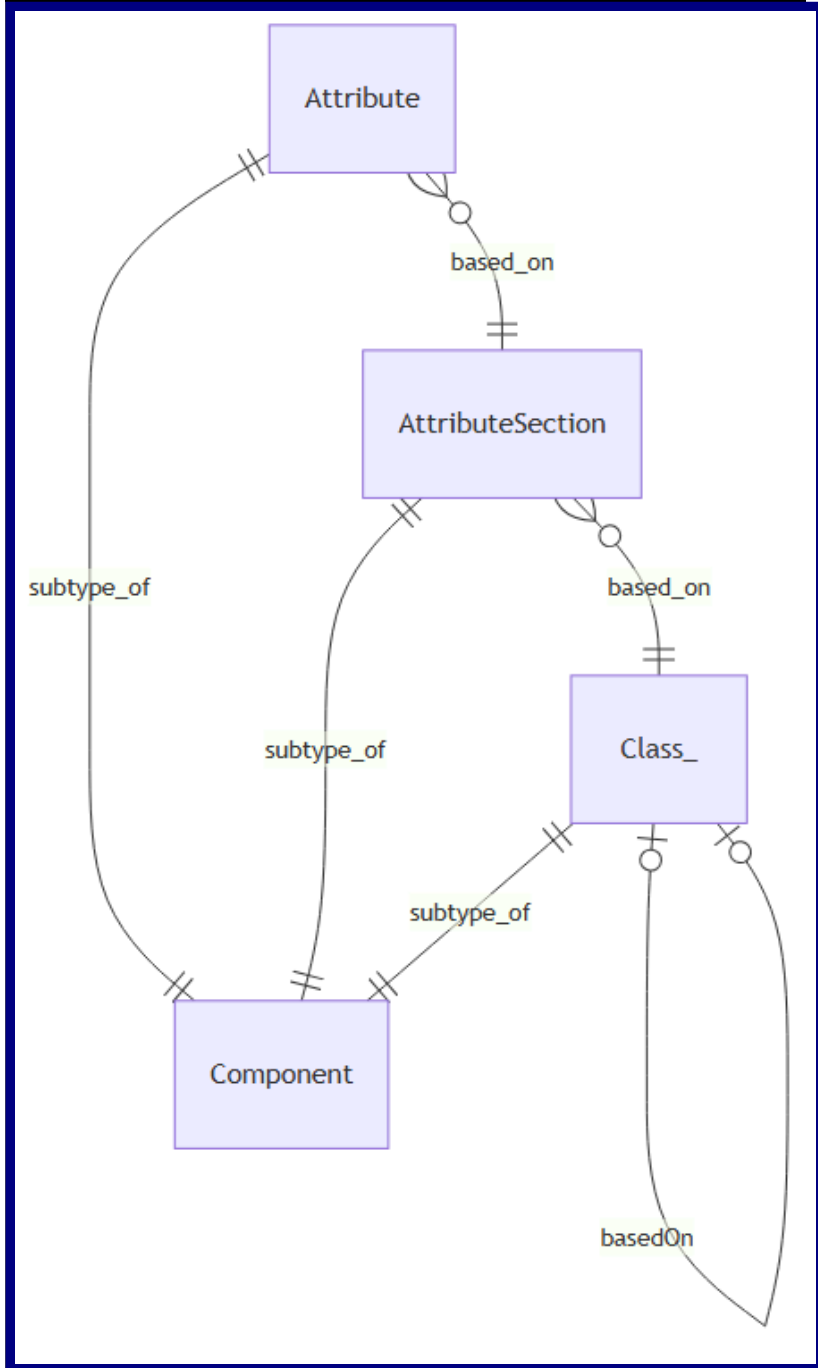
erDiagram

Class_ ||--|| Component : subtype_of
Class_ |o--o| Class_ : basedOn
AttributeSection ||--|| Component : subtype_of
AttributeSection }o--|| Class_ : based_on
Attribute ||--|| Component : subtype_of
Attribute }o--|| AttributeSection : based_on

Mermaid ER Diagram for AttributeSection - Live!

erDiagram Class_ ||--|| Component : subtype_of Class_ |o--o| Class_ : basedOn AttributeSection ||--|| Component : subtype_of AttributeSection }o--|| Class_ : based_on Attribute ||--|| Component : subtype_of Attribute }o--|| AttributeSection : based_on

Mermaid ER Diagram for AttributeSection - PNG for mermaid



	Attribute	
	A property or characteristic of a class	
LURAL	Attributes	
SEDON	AttributeSection	
DENTS	AttributeConstraint	
YPEOF	Component	
me		(LowerCamel value O_O)
RIDES	Component.name	
pe	The kind of object to which the attribute refers. _	(DataType value O_O)
	But,	
	<ul style="list-style-type: none">• ◦ List of Editions• ◦ Set of Edition• ◦ ... and more complicated cases.	
see	the section below on Data Type Specifiers.	
nal	Indicates whether the attribute must have a value for every instance of the class _	(Boolean value O_O)
FAULT	*** False	
ity	The cardinality of the relationship represented by the attribute _	(Cardinality value O_O)
FAULT	*** For a singular attribute, the default cardinality is N:1. If the attribute is 1:1, it must be stated explicitly. For a collective attribute, the default is 1:N. If the attribute is N:M, it must be stated explicitly.	
note	how this works with optionality	
s ble		(Boolean value O_O)
VATION	true if the data type is a class or a simple collection of members of a class.	
SS	the class which contains, or would contain the inverse attribute	(Optional Class value O_O)
VATION	from the data type. Null unless attribute is invertible.	
ute		(Optional Attribute value O_O)

Attributes

Optional	(<i>Optional</i> <u>Attribute</u> value O_O)
Default	<div>The rule or formula for calculating the value, if no value is supplied Now running to a second line with the parenthetical on yet a third line</div> (<i>Optional</i> <u>Derivation</u> value O_O)
note	even when an Attribute has a default derivation, there's no guarantee that every instance will have an assigned value. Example needed.
Derivation	<div>For derived attributes, the rule or formula for calculating the value _</div> (<i>Optional</i> <u>Derivation</u> value O_O)
issue	on insert vs on access?
Constraints	<div>Any validation rules specific to this attribute _</div> (<i>List of</i> <u>Constraints</u> value O_O)
note	from Class.constraints
Linking overrides attributes	
Attributes	<div>created for Attribute</div> <div>Inverse attribute for Class.attributes from which this was implied.</div> (<u>Class</u> value M_1)
INVERSE	<u>Class.attributes</u>
Attributes	<div>Inverse attribute for Key.keyAttributes from which this was implied.</div> (<u>Key</u> value M_1)
INVERSE	<u>Key.keyAttributes</u>
Section	<div>A link back to the AttributeSection on which this Attribute depends.</div> (<u>AttributeSection</u> value M_1)

Mermaid ER Diagram for Attribute - Inert

erDiagram

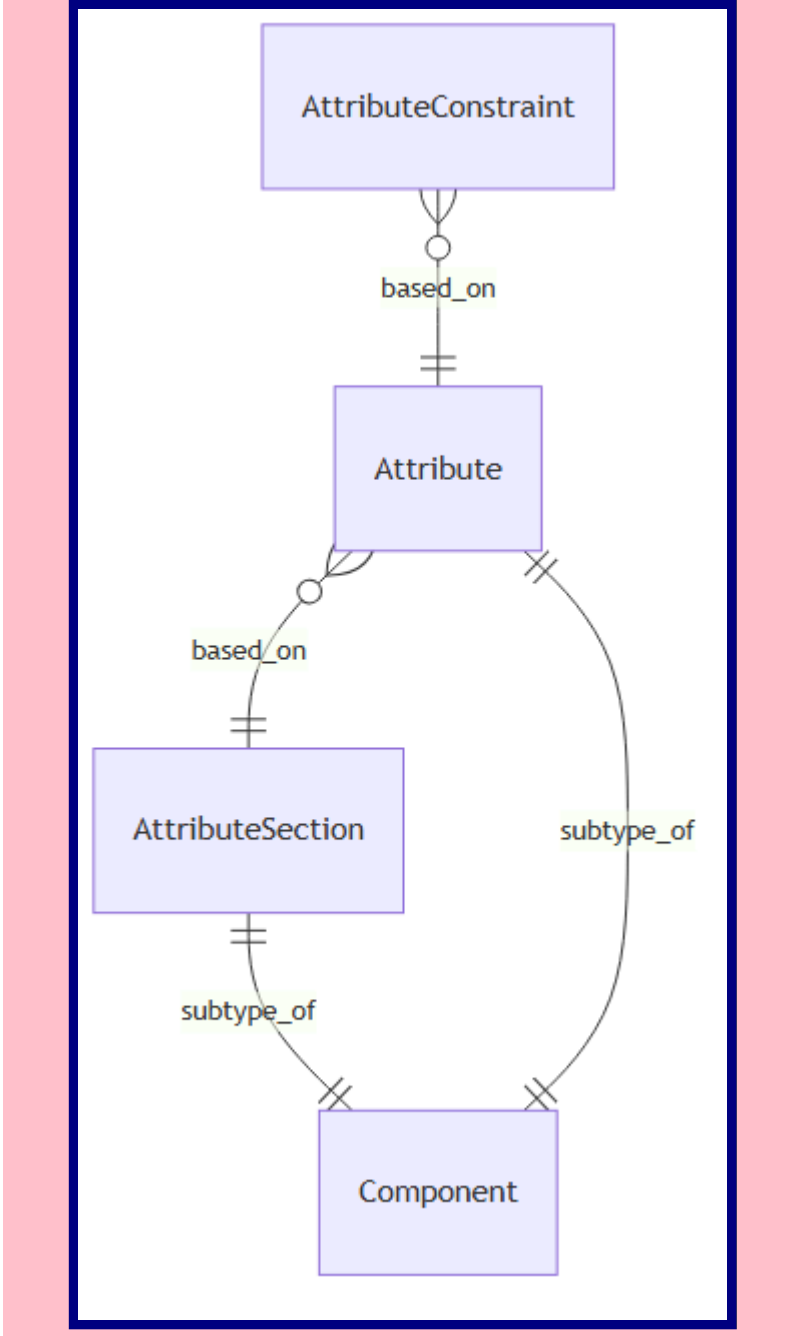
AttributeSection ||--|| Component : subtype_of
Attribute ||--|| Component : subtype_of
Attribute }o--|| AttributeSection : based_on
AttributeConstraint }o--|| Attribute : based_on

Mermaid ER Diagram for Attribute - Live!

erDiagram AttributeSection ||--|| Component : subtype_of Attribute
||--|| Component : subtype_of Attribute }o--|| AttributeSection :

based_on AttributeConstraint }o--|| Attribute : based_on

Mermaid ER Diagram for Attribute - PNG for mermaid



Attributes

Derivation
A rule or formula for deriving the value of an attribute

PLURAL Derivations

ement	An English language statement of the derivation rule _ (RichText value O_O)
-------	--

ession	The formal expression of the derivation in a programming language _ (CodeExpression value O_O)
--------	---

Constraint
A rule, condition, or validation that must be satisfied by the model

PLURAL Constraints

BTYPOF [Component](#)

SUBTYPES [ClassConstraint](#), [AttributeConstraint](#)

ement	An English language statement of the constraint _ (RichText value O_O)
-------	---

ession	The formal expression of the constraint in a programming language, for example: OCL or Python. _ (CodeExpression value O_O)
--------	--

erity	(Code value O_O)
-------	--------------------

Warning, nothing fatal; just a caution
Error, serious. Fix now

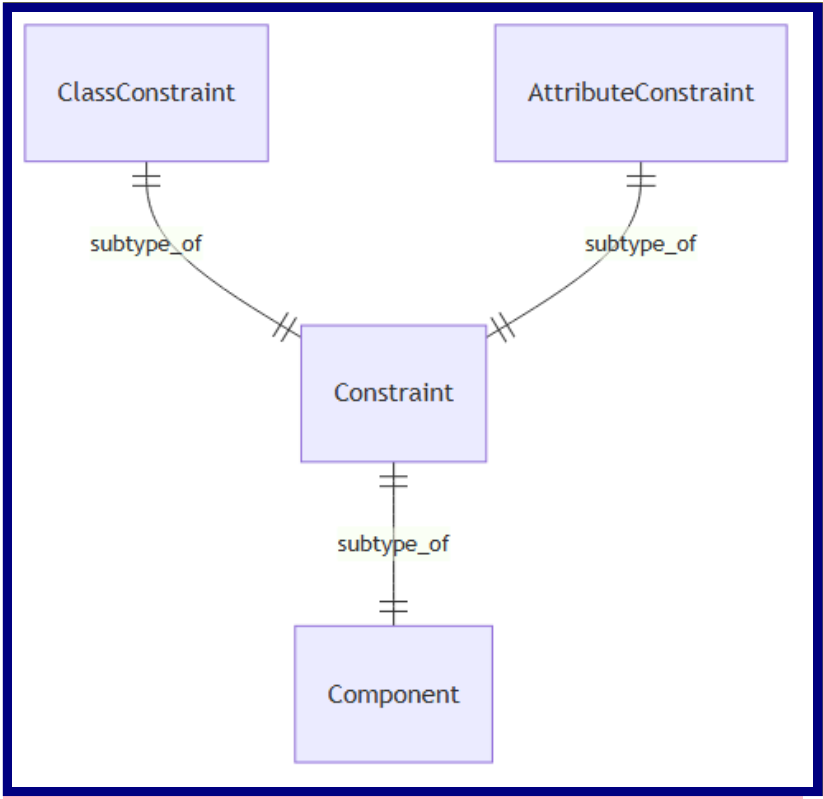
Mermaid ER Diagram for Constraint - Inert

```
erDiagram
    Constraint ||--|| Component : subtype_of
    Constraint ||--|| ClassConstraint : subtype_of
    Constraint ||--|| AttributeConstraint : subtype_of
```

Mermaid ER Diagram for Constraint - Live!

```
erDiagram
    Constraint ||--|| Component : subtype_of
    Constraint ||--|| ClassConstraint : subtype_of
    Constraint ||--|| AttributeConstraint : subtype_of
```

Mermaid ER Diagram for Constraint - PNG for mermaid



ClassConstraint

LURAL ClassConstraints
EDPLURAL ClassConstraints
SEDON [Class](#)
TYPEOF [Constraint](#)

created for ClassConstraint

A link back to the Class on which this ClassConstraint depends.

([Class](#) value M_1)

Mermaid ER Diagram for ClassConstraint - Inert

erDiagram

Class_ ||o--o| Class_ : basedOn

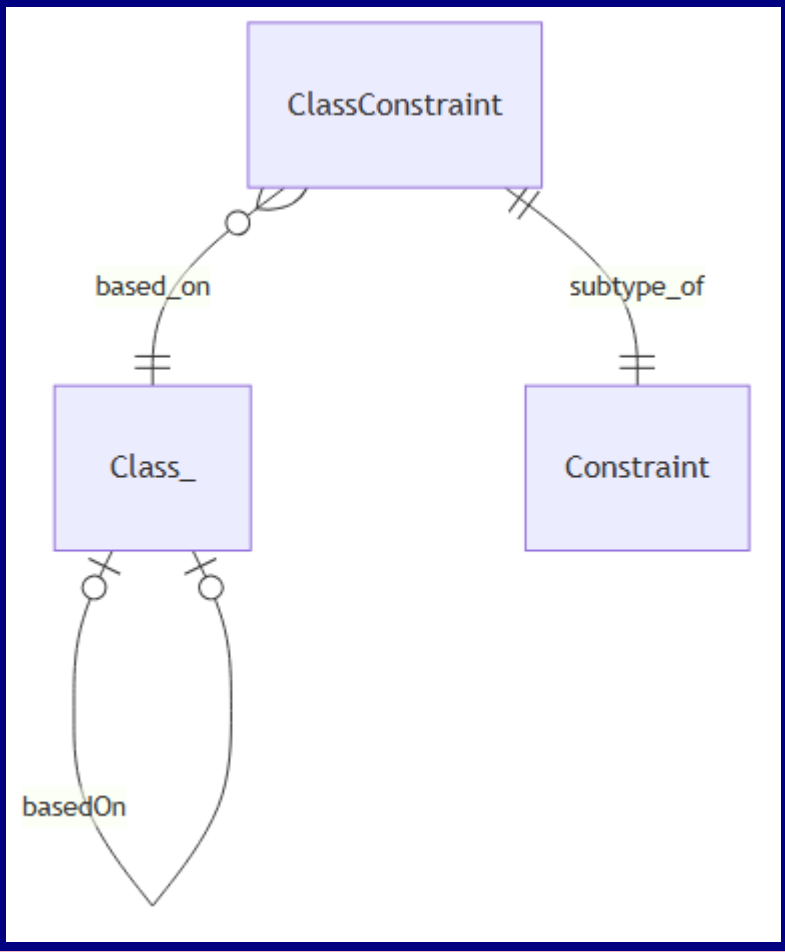
ClassConstraint ||--|| Constraint : subtype_of

ClassConstraint }o--|| Class_ : based_on

Mermaid ER Diagram for ClassConstraint - Live!

```
erDiagram
    Class_ ||--o| Class_ : basedOn
    ClassConstraint ||--|| Class_ : subtype_of
    Constraint ||--o| Class_ : based_on
```

Mermaid ER Diagram for ClassConstraint - PNG for mermaid



AttributeConstraint

PLURAL AttributeConstraints
 MEDPLURAL AttributeConstraints
 BASED ON [Attribute](#)
 BTYPE OF [Constraint](#)

utes created for AttributeConstraint

A link back to the Attribute on which this AttributeConstraint depends.
([Attribute value M_1](#))

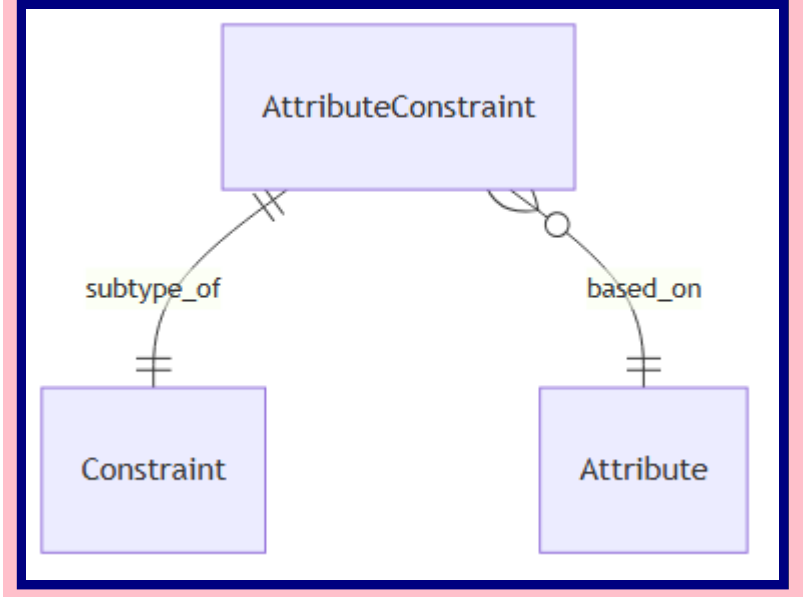
Mermaid ER Diagram for AttributeConstraint - Inert

erDiagram
AttributeConstraint ||--|| Constraint : subtype_of
AttributeConstraint }o--|| Attribute : based_on

Mermaid ER Diagram for AttributeConstraint - Live!

erDiagram AttributeConstraint ||--|| Constraint : subtype_of
AttributeConstraint }o--|| Attribute : based_on

Mermaid ER Diagram for AttributeConstraint - PNG for mermaid



Methods

	Method	
	A behavior or operation associated with a class	
LURAL	Methods	
TYPEOF	Component	
ers	The input parameters of the method _	(<i>List of Parameters value 0_0</i>)
VERSE	Parameter.inverseOfParameters	
pe	The data type of the value returned by the method _	(<i>DataType value 0_0</i>)
s	created for Method	
ds	Inverse attribute for Class.methods from which this was implied.	(<i>Class value M_1</i>)
VERSE	Class.methods	

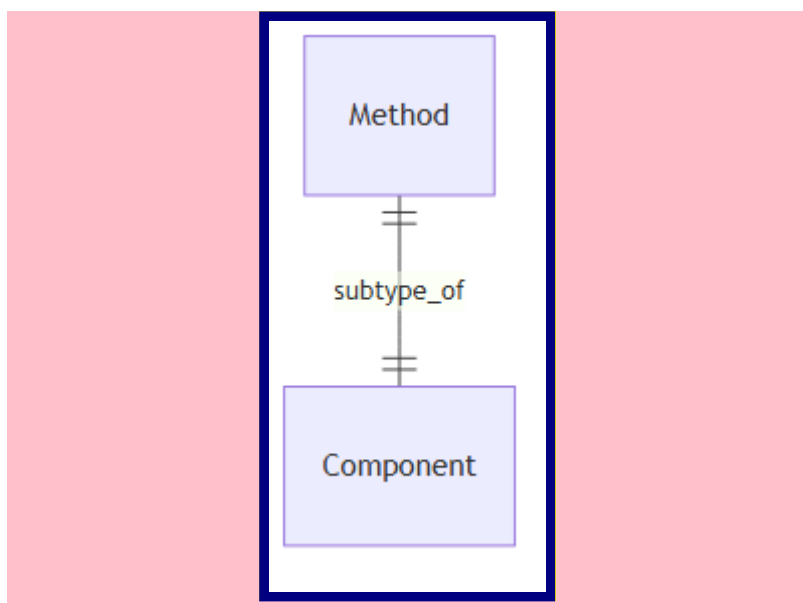
Mermaid ER Diagram for Method - Inert

erDiagram
Method ||--|| Component : subtype_of

Mermaid ER Diagram for Method - Live!

erDiagram Method ||--|| Component : subtype_of

Mermaid ER Diagram for Method - PNG for mermaid



	Parameter
	An input to a method
LURAL	Parameters
TYPEOF	Component
type	The data type of the parameter (DataType value O_O)
ity	The cardinality of the parameter. e.g., optional, required. (Cardinality value O_O)
s	created for Parameter
ters	Inverse attribute for Method.parameters from which this was implied. (Method value M_1)
VERSE	Method.parameters

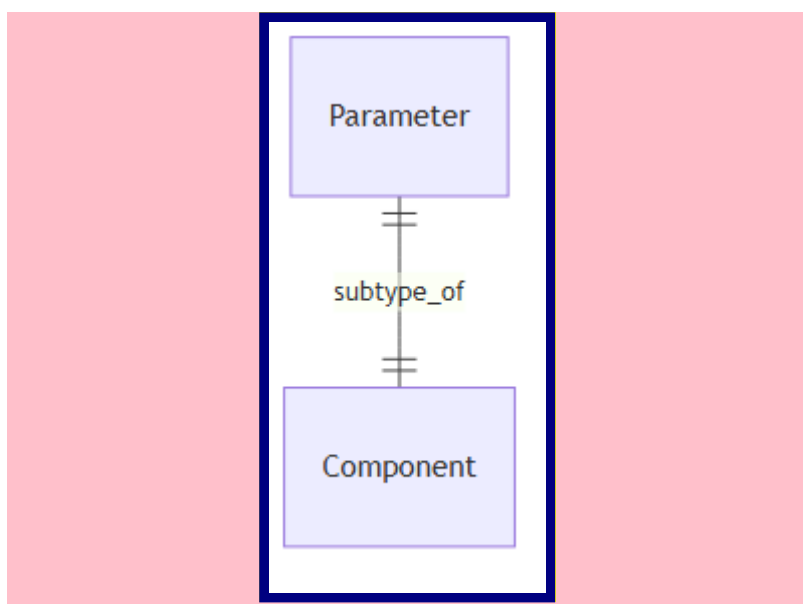
Mermaid ER Diagram for Parameter - Inert

erDiagram
Parameter ||--|| Component : subtype_of

Mermaid ER Diagram for Parameter - Live!

erDiagram Parameter ||--|| Component : subtype_of

Mermaid ER Diagram for Parameter - PNG for mermaid



BLANK

Trivial Data Types

Message

LURAL Messages
DPLURAL Messages

Message is trivial; no diagram

CodeExpression

LURAL CodeExpressions
DPLURAL CodeExpressions

ge

the programming language

([Code](#) value O_O)

OCI, Object Constraint Language
Java, Java

on

([String](#) value O_O)

CodeExpression is trivial; no diagram

DataType

LURAL DataTypes
DPLURAL DataTypes

DataType is trivial; no diagram

SimpleDataTypeSubtpeOfDataType

LURAL SimpleDataTypeSubtpeOfDataTypes
DPLURAL SimpleDataTypeSubtpeOfDataTypes

ss

([Class](#) value O_O)

VERSE [Class.inverseOfCoreClass](#)

SimpleDataTypeSubtpeOfDataType is trivial; no diagram

ComplexDataType

LURAL ComplexDataTypes
DPLURAL ComplexDataTypes

on

([AggregatingOperator](#) value O_O)

es

(List of [DataTypes](#) value O_O)

ComplexDataType is trivial; no diagram

AggregatingOperator

PLURAL AggregatingOperators
IMEDPLURALAggregatingOperators

name (Code value O_O)

SetOf
ListOf
Mapping

arity (Integer value O_O)

elling (Template value O_O)

AggregatingOperator is trivial; no diagram

BLANK

Trivial Low level Data Types

insert Camel Case.md

Emoji

SINGULAR Emojis
MULTIPLURAL Emojis

Emoji is trivial; no diagram

String

SINGULAR Strings
MULTIPLURAL Strings

String is trivial; no diagram

CamelName

A short string without punctuation or spaces, suitable for names, labels, or identifiers and presented in camel case.

SINGULAR CamelNames
MULTIPLURAL CamelNames
TYPEOF [String](#)
TYPES [UpperCamel](#), [LowerCamel](#)

Example `(String value O_O)`

Constraints Must follow the camel case naming convention and not be empty.

Example "firstName", "orderDate", "customerID"

Example Note

- *CamelName* is presented here, just after its first usage by another class (Component), to provide context and understanding before it is used further in the model.

CamelName is trivial; no diagram

UpperCamel

a CamelName that begins with a capital letter

Example _ "Customer", "ProductCategory", "PaymentMethod"

WHERE content begins with an upper case letter.

SINGULAR UpperCamels
MULTIPLURAL UpperCamels
TYPEOF [CamelName](#)

UpperCamel is trivial; no diagram

LowerCamel
a CamelName that begins with a lower case letter

example "firstName", "orderTotal", "shippingAddress"
WHERE content begins with a lower case letter.
PLURAL LowerCamels
MEDPLURAL LowerCamels
BTYPEOF [CamelName](#)

LowerCamel is trivial; no diagram

QualifiedCamel
an expression consisting of Camel Names separated by periods

PLURAL QualifiedCamels
MEDPLURAL QualifiedCamels
BTYPEOF [String](#)
CONSTRAINTS

content consists of CamelNames, separated by periods. Each of the camel names must be Upper Camel except, possibly, the first.

QualifiedCamel is trivial; no diagram

RichText
A string with markup for block level formatting.

PLURAL RichTexts
MEDPLURAL RichTexts
BTYPEOF [String](#)
SUBTYPES [OneLiner](#)

value the string content
([String](#) value O_O)

format the rich text coding language used
([Code](#) value O_O)

HTML
MarkDown

RichText is trivial; no diagram

OneLiner
String with markup for line level formatting.

SINGULAR OneLiners
PLURAL OneLiners
TYPEOF [RichText](#)

VALUE the string content
([String](#) value 0_0)

OVERIDES [RichText.value](#)
CONSTRAINTS must not contain a line break or new line character
MESSAGE A line can't span two lines

OneLiner is trivial; no diagram

PrimitiveType
A basic, built-in data type

SINGULAR PrimitiveTypes
PLURAL PrimitiveTypes
TYPES [String](#), [Integer](#), [Decimal](#), [Boolean](#), [Date](#), [Time](#), [DateTime](#)

PrimitiveType is trivial; no diagram

String

SINGULAR Strings
PLURAL Strings
TYPEOF [PrimitiveType](#)
TYPES [CamelName](#), [QualifiedCamel](#), [RichText](#)

String is trivial; no diagram

Integer

SINGULAR Integers
PLURAL Integers
TYPEOF [PrimitiveType](#)

Integer is trivial; no diagram

Decimal

SINGULAR Decimals
PLURAL Decimals
TYPEOF [PrimitiveType](#)

Decimal is trivial; no diagram

Boolean

PLURAL Booleans

IMEDPLURAL Booleans

BTYEOF [PrimitiveType](#)

Boolean is trivial; no diagram

Date

PLURAL Dates

IMEDPLURAL Dates

BTYEOF [PrimitiveType](#)

Date is trivial; no diagram

Time

PLURAL Times

IMEDPLURAL Times

BTYEOF [PrimitiveType](#)

Time is trivial; no diagram

DateTime

PLURAL DateTimes

IMEDPLURAL DateTimes

BTYEOF [PrimitiveType](#)

DateTime is trivial; no diagram

CodingLanguage

PLURAL CodingLanguages

IMEDPLURAL CodingLanguages

CodingLanguage is trivial; no diagram

Cardinality

PLURAL Cardinalitys

IMEDPLURAL Cardinalitys

Cardinality is trivial; no diagram

TemplateLanguage

LURAL TemplateLanguages
DPLURALTemplateLanguages

TemplateLanguage is trivial; no diagram

Template

LURAL Templates
DPLURALTemplates

Template is trivial; no diagram

Code

LURAL Codes
DPLURALCodes

Code is trivial; no diagram

Annotation Types Used

These are the recognized Annotation Types for the LDM model.

And this is how you register the AnnotationTyped for a model. By including this sort of array in the DSL document for the model.

PlantUML Diagram - Inert

@startjson

```
[
{
  "label": "Error",
  "emoji": "✖",
  "emojiName": "cross_mark",
  "emojiUnicode": "U+274C",
  "purpose": "Indicates a critical error or failure in
the model."
},
{
  "label": "Warning",
  "emoji": "⚠",
  "emojiName": "warning",
  "emojiUnicode": "U+26A0",
  "purpose": "Indicates a potential issue or warning
in the model."
},
{
  "label": "Note",
  "emoji": "📘",
  "emojiName": "blue_book",
  "emojiUnicode": "U+1F4D8",
  "purpose": "Provides additional context,
explanations, or clarifications for the annotated
element."
},
{
  "label": "Issue",
  "emoji": "⚠",
  "emojiName": "warning",
```

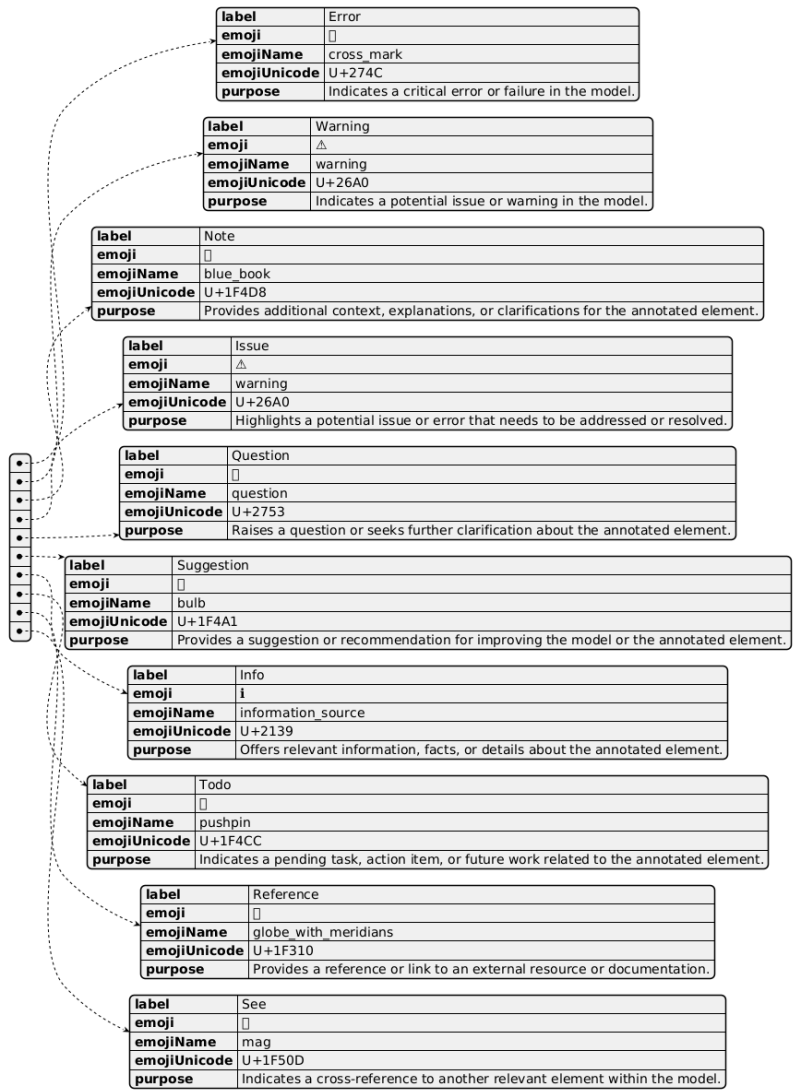
```

"emojiUnicode": "U+26A0",
"purpose": "Highlights a potential issue or error
that needs to be addressed or resolved."
},
{
"label": "Question",
"emoji": "?",
"emojiName": "question",
"emojiUnicode": "U+2753",
"purpose": "Raises a question or seeks further
clarification about the annotated element."
},
{
"label": "Suggestion",
"emoji": "💡",
"emojiName": "bulb",
"emojiUnicode": "U+1F4A1",
"purpose": "Provides a suggestion or
recommendation for improving the model or the
annotated element."
},
{
"label": "Info",
"emoji": "i",
"emojiName": "information_source",
"emojiUnicode": "U+2139",
"purpose": "Offers relevant information, facts, or
details about the annotated element."
},
{
"label": "Todo",
"emoji": "📌",
"emojiName": "pushpin",
"emojiUnicode": "U+1F4CC",
"purpose": "Indicates a pending task, action item,
or future work related to the annotated element."
},
{
"label": "Reference",
"emoji": "🌐",
"emojiName": "globe_with_meridians",

```

```
"emojiUnicode": "U+1F310",  
"purpose": "Provides a reference or link to an  
external resource or documentation."  
},  
{  
"label": "See",  
"emoji": "Q",  
"emojiName": "mag",  
"emojiUnicode": "U+1F50D",  
"purpose": "Indicates a cross-reference to another  
relevant element within the model."  
}  
]  
@endjson
```

PlantUML Diagram - PNG for puml



Annotation types as CSV

label,emoji,emojiName,emojiUnicode,purpose

Error,✖,cross_mark,U+274C,Indicates a critical error or failure in the model.

Warning,⚠,warning,U+26A0,Indicates a potential issue or warning in the model.

Note,📘,blue_book,U+1F4D8,"Provides additional context, explanations, or clarifications for the annotated element."

Issue,⚠,warning,U+26A0,Highlights a potential issue or error that needs to be addressed or resolved.

Question,❓,question,U+2753,Raises a question or seeks further clarification about the annotated element.

Suggestion,💡,bulb,U+1F4A1,Provides a suggestion or recommendation for improving the model or the annotated element.

Info,ℹ,information_source,U+2139,"Offers relevant information, facts, or details about the annotated element."

Todo,📌,pushpin,U+1F4CC,"Indicates a pending task, action item, or future work related to the annotated element."

Reference,🌐,globe_with_meridians,U+1F310,Provides a reference or link to an external resource or documentation.

See,🔍,mag,U+1F50D,Indicates a cross-reference to another relevant element within the model.

	label	emoji	emojiName	emojiUnicode	purpose
0	Error	✖	cross_mark	U+274C	Indicates a critical error or failure in the model.
1	Warning	⚠	warning	U+26A0	Indicates a potential issue or warning in the model.
2	Note	📘	blue_book	U+1F4D8	Provides additional context, explanations, or clarifications for the annotated element.
					Highlights a potential issue

Appendices

various sidebars to include Insert More Sidebars.md Insert Overrides.md insert
LDM Intro.md Insert OCL.md Insert Camel Case.md

== content to add