FIRST PAGE LEFT LEFT BLANK

Blank First Page

Literate Data Model

Component
An element or building block of the literate data model

LURAL Components
EDPLURAL Components
DENTS Annotation
TYPES LiterateDataModel , Subject , Class , Key , AttributeSection , Attribute , Constraint , Method , ParameterAnInputToAMethod

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

arning This is a warning with emoji

me The name of the component
(CamelName value O_O)

me (QualifiedCamel value O_O)

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

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

EFAULT name - how do you say name in english?
OCL x.name == y
RAINTS the abbreviated name should be shorter than the actual name
OCL len(abbreviatedName) < len(name)
ESSAGE Why have an abbreviation longer than the name?
VERITY Warning
note Does this annotation find it's way to the Constraint? YES! It's fixed!

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

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

mechanical attributes

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

EFAULT 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.

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 *recognized* or *registered* Annotation Types.

LURAL AnnotationTypes
EDPLURAL AnnotationTypes
SEDON LiterateDataModel

oji an emoji
*(**Emoji** value O_O)*

me an emoji
*(**String** value O_O)*

de the Unicode for the emoji
*(**String** value O_O)*

bel A short label to indicate the purpose of the annotation _
*(**LowerCamel** value O_O)*

ral the plural form of the label
*(**UpperCamel** value O_O)*

EFAULT based on label

se the intended reason for the annotation.
*(**OneLiner** value O_O)*

s Model A link back to the LiterateDataModel on which this AnnotationType depends.
*(**LiterateDataModel** value M_1)*

s tionType verse attribute for Annotation.annotationType from which this was implied.
*(**Annotation** value M_1)*

VERSE Annotation.annotationType

s Model A link back to the LiterateDataModel on which this AnnotationType depends.
*(**LiterateDataModel** value M_1)*

s tionType verse attribute for Annotation.annotationType from which this was implied.
*(**Annotation** value M_1)*

INVERSE Annotation.annotationType

Annotation
A note or comment associated with a model element

PLURAL Annotations
UMEDPLURAL Annotations
BASEDON Component

nType ***(Optional AnnotationType value O_O)***

note An Annotation is considered to *recognized* if the label is associated with an Annotation Type. otherwise it is *ad hoc* .

note Should be a Value Type

INVERSE AnnotationType.inverseOfAnnotationType

label A short label to indicate the purpose of the annotation _
(CamelName value O_O)

But any short label is valid.

DEFAULT from annotationType

emoji ***(Optional Emoji value O_O)***

DEFAULT from annotation type

ontent The content or body of the annotation
(RichText value O_O)

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

DEFAULT 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.

utes
onent A link back to the Component on which this Annotation depends.
(Component value M_1)

onent A link back to the Component on which this Annotation depends.
(Component value M_1)

LiterateDataModel
A representation of a domain's entities, attributes, and relationships, along with explanatory text and examples

LURAL LiterateDataModels
DENTS AnnotationType , Subject
YPEOF Component

me (UpperCamel value O_O)
RRIDES Component.name

cts list of all classes in the model, as ordered in the definition of the model.
(List of Classes value O_O)
VERSE Class.inverseOfAllSubjects
VATION gathering s.allSubjects over s in subjectAreas
RAINTS Subject names must be unique across the model.

ses list of all classes in the model, as ordered in the definition of the model.
(List of Classes value O_O)
VERSE Class.inverseOfAllClasses
VATION gathering s.allClasses over s in allSubjects.
RAINTS Class names must be unique across the model.

pes (List of AnnotationTypes value O_O)

Language the recommended lanquage for expressing derivation, defaults, and constraints
(CodingLanguage value O_O)
EFAULT OCL

Languages (Optional List of CodingLanguages value O_O)

teLanguage the recommended lanquage for expressing derivation, defaults, and constraints
(TemplateLanguage value O_O)
EFAULT Handlebars

eLanguages (Optional List of TemplateLanguages value O_O)

ns A list of functions that require sophisticated AI-powered implementation *
(List of String value O_O)
VATION ['aiEnglishPlural()']

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 LiterateDataModel
BTYPEOF Component
UBTYPES SubjectArea

name ***(UpperCamel value O_O)***
VERRIDES Component.name

ubject The parent subject, if any, under which this subject is nested _
(Optional Subject value O_O)
INVERSE Subject.inverseOfParentSubject

asses 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

bjects 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.inverseOfChildSubjects

ataModel A link back to the LiterateDataModel on which this Subject depends.
(LiterateDataModel value M_1)

ntSubject Inverse attribute for Subject.parentSubject from which this was implied.
(Subject value M_1)

INVERSE Subject.parentSubject

dSubjects Inverse attribute for Subject.childSubjects from which this was implied.
(Subject value M_1)

VERSE Subject.childSubjects

Model A link back to the LiterateDataModel on which this Subject depends.
(LiterateDataModel value M_1)

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

VERSE Subject.parentSubject

ubjects Inverse attribute for Subject.childSubjects from which this was implied.
(Subject value M_1)

VERSE Subject.childSubjects

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

WHERE parentSubject is absent
LURAL SubjectAreas
SEDON LiterateModel , Xyz
YPEOF Subject

s
del A link back to the LiterateModel on which this SubjectArea depends.
(LiterateModel value M_1)

s
Xyz A link back to the Xyz on which this SubjectArea depends.
(Xyz value M_1)

s
del A link back to the LiterateModel on which this SubjectArea depends.
(LiterateModel value M_1)

s
Xyz A link back to the Xyz on which this SubjectArea depends.
(Xyz value M_1)

Class

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

PLURAL Classes

PENDENTS Subtyping , Key , AttributeSection , ClassConstraint

BTYPEOF Component

UBTYPES ReferenceType

STRAINTS Within each Class, attribute names must be unique.

lForm 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".

sedOn 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 synonymousle in this metamodel.

INVERSE Class.inverseOfBasedOn

rtypes The parent class

(Es value O_O)

pings 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

types Any subtypes or specializations of this class based on it's subtypings.

(List of Classes value O_O)

ample For instance, using the `Book` example, the subtypes could include `FictionBook` , `Non-fictionBook` , `HardcoverBook` , `PaperbackBook` , `ScienceBook` , and `HistoryBook` .

VERSE Class.inverseOfSubtypes

tes The attributes or properties of the class, in the order in which they should be presented _

(List of Attributes value O_O)

VERSE Attribute.inverseOfAttributes

ns additional attributes or properties of the class, grouped for clarity and elaboration. _

(List of AttributeSections value O_O)

VERSE AttributeSection.inverseOfAttributeSections

nts Any constraints, rules, or validations specific to this class _

(List of Constraints 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 Methods value O_O)

VERSE Method.inverseOfMethods

s

nts the Classes which are basedOn this Class

(Optional Set of Classes value O_O)

VERSE Class.basedOn

eys (Optional Set of UniqueKeys value O_O)

VERSE UniqueKey.basedOn

s

ects Inverse attribute for LiterateDataModel.allSubjects from which this was implied.

(LiterateDataModel value M_1)

VERSE LiterateDataModel.allSubjects

ses Inverse attribute for LiterateDataModel.allClasses from which this was implied.

(LiterateDataModel value M_1)

VERSE LiterateDataModel.allClasses

ses Inverse attribute for Subject.classes from which this was implied.

(Subject value M_1)

INVERSE Subject.classes

sedOn Inverse attribute for Class.basedOn from which this was implied.
(Class value M_1)

INVERSE Class.basedOn

types Inverse attribute for Class.subtypes from which this was implied.
(Class value M_1)

INVERSE Class.subtypes

asses Inverse attribute for Subtyping.classes from which this was implied.
(Subtyping value M_1)

INVERSE Subtyping.classes

eClass Inverse attribute for SimpleDataTypeSubtpeOfDataType.coreClass from which this was implied.
(SimpleDataTypeSubtpeOfDataType value M_1)

INVERSE SimpleDataTypeSubtpeOfDataType.coreClass

ubjects Inverse attribute for LiterateDataModel.allSubjects from which this was implied.
(LiterateDataModel value M_1)

INVERSE LiterateDataModel.allSubjects

lasses Inverse attribute for LiterateDataModel.allClasses from which this was implied.
(LiterateDataModel value M_1)

INVERSE LiterateDataModel.allClasses

asses Inverse attribute for Subject.classes from which this was implied.
(Subject value M_1)

INVERSE Subject.classes

sedOn Inverse attribute for Class.basedOn from which this was implied.
(Class value M_1)

INVERSE Class.basedOn

types Inverse attribute for Class.subtypes from which this was implied.
(Class value M_1)

INVERSE Class.subtypes

asses Inverse attribute for Subtyping.classes from which this was implied.
(Subtyping value M_1)

INVERSE Subtyping.classes

On Inverse attribute for Class.basedOn from which this was implied.
(Class value M_1)

VERSE Class.basedOn

es Inverse attribute for Class.subtypes from which this was implied.
(Class value M_1)

VERSE Class.subtypes

es Inverse attribute for Subtyping.classes from which this was implied.
(Subtyping value M_1)

VERSE Subtyping.classes

ass Inverse attribute for SimpleDataTypeSubtpeOfDataType.coreClass from which this was implied.
(SimpleDataTypeSubtpeOfDataType value M_1)

VERSE SimpleDataTypeSubtpeOfDataType.coreClass

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

PLURAL Subtypings
UMEDPLURAL Subtypings
BASEDON Class

name *(LowerCamel value O_O)*

lusive *(Boolean value O_O)*

DEFAULT true

ustive *(Boolean value O_O)*

DEFAULT true

asses *(List of Classes value O_O)*

DSL : Shown in the DSL as

- Subbtypes: 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

utes

typings 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)

typings 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)

ings Inverse attribute for Class.subtypings from which this was implied.
(Class value M_1)

VERSE Class.subtypings

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

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

LURAL ReferenceTypes
EDPLURAL ReferenceTypes
YPEOF Class

Type CodeType A data type or enumeration used in the model

LURAL CodeTypes
EDPLURAL CodeTypes
DENTS CodeValue

ive the code type was implied by use in an attribute and is only used for that attribute
(Boolean value O_O)

Type CodeValue A possible value for an enumerated data class

LURAL CodeValues
EDPLURAL CodeValues
SEDON CodeType

de A short code or abbreviationi for the value _
(NameString value O_O)

on 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:

s
pe A link back to the CodeType on which this CodeValue depends.
(CodeType value M_1)

s
pe A link back to the CodeType on which this CodeValue depends.
(CodeType value M_1)

utes

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

(CodeType value M_1)

Key

a list of attributes of a class

PLURAL Keys

UMEDPLURAL Keys

BASEDON Class

BTYPEOF Component

SUBTYPES UniqueKey

ibutes the attributes of the base Class.

(List of Attributes value O_O)

INVERSE Attribute.inverseOfKeyAttributes

STRAINTS each attribute must be a direct or inherited of the base class.

STRAINTS no repetitions allowed in keyAttributes

Issue : introduce PureLists?

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

utes

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

(Class value M_1)

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

(Class value M_1)

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

(Class value M_1)

UniqueKey

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

note order unimportant for Unique Keys.

PLURAL UniqueKeys

UMEDPLURAL UniqueKeys

BTYPEOF Key

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

LURAL AttributeSections
EDPLURAL AttributeSections
SEDON Class
DENTS Attribute
YPEOF Component

nal 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 ot required, depending on how it is marked.

-
- But if the Arrribute Section is optional each attribute in the section is only required if any attribute in the section is ptresent.

s
teSections inverse attribute for Class.attributeSections from which this was implied.
(Class value M_1)
VERSE Class.attributeSections

s
ss A link back to the Class on which this AttributeSection depends.
(Class value M_1)

s
teSections inverse attribute for Class.attributeSections from which this was implied.
(Class value M_1)
VERSE Class.attributeSections

s
teSections inverse attribute for Class.attributeSections from which this was implied.
(Class value M_1)
VERSE Class.attributeSections

s
ss A link back to the Class on which this AttributeSection depends.
(Class value M_1)

Attribute

A property or characteristic of a class

PLURAL Attributes

BASEDON AttributeSection

PENDENTS AttributeConstraint

BTYPEOF Component

name (***LowerCamel*** value O_O)

VERRIDES Component.name

aType The kind of object to which the attribute refers. _

(***DataType*** value O_O)

But,

- ◦ List of Editions
- ◦ Set of Edition
- ◦ ... and more complicated cases.

see the section below on Data Type Specifiers.

tional Indicates whether the attribute must have a value for every instance of the class _

(***Boolean*** value O_O)

DEFAULT *** False

inality The cardinality of the relationship represented by the attribute _

(***CardinalityCode*** value O_O)

DEFAULT

*** 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.

rExample

uthor (***InventedName*** value O_O)

books (***Optional InventedName*** value O_O)

note how this works with optionality

utes

ertible (***Boolean*** value O_O)

ERIVATION 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 arrribute is invertible.

ute ***(Optional Attribute value O_O)***

nal ***(Optional Attribute value O_O)***

ult The rule or formula for calculating the value, if no value is supplied Now running to a second line with the parenthentical on yet a third line
(Optional Derivation 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.

on For derived attributes, the rule or formula for calculating the value _
(Optional Derivation value O_O)

issue on insert vs on access?

nts Any validation rules specific to this attribute _
(List of Constraints value O_O)

note from Class.constraints

tes Inverse attribute for Class.attributes from which this was implied.
(Class value M_1)

VERSE Class.attributes

ributes Inverse attribute for Key.keyAttributes from which this was implied.
(Key value M_1)

VERSE Key.keyAttributes

ction A link back to the AttributeSection on which this Attribute depends.
(AttributeSection value M_1)

tes Inverse attribute for Class.attributes from which this was implied.
(Class value M_1)

VERSE Class.attributes

ributes Inverse attribute for Key.keyAttributes from which this was implied.
(Key value M_1)

VERSE Key.keyAttributes

ibutes Inverse attribute for Class.attributes from which this was implied.
(Class value M_1)

INVERSE Class.attributes

Attributes Inverse attribute for Key.keyAttributes from which this was implied.
(Key value M_1)

INVERSE Key.keyAttributes

Section A link back to the AttributeSection on which this Attribute depends.
(AttributeSection value M_1)

lue Type **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)

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

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

PLURAL Constraints
BTYPEOF Component
SUBTYPES ClassConstraint , AttributeConstraint

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

ssion The formal expression of the constraint in a programming language
(InventedName value O_O)

verity *(Code value O_O)*

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

lue Type **Message**

PLURAL Messages
MEDPLURAL Messages

lue Type **ClassConstraint**

PLURAL ClassConstraints
MEDPLURAL ClassConstraints

SEDON Class
YPEOF Constraint

s
ss
A link back to the Class on which this ClassConstraint depends.
(Class value M_1)

s
ss
A link back to the Class on which this ClassConstraint depends.
(Class value M_1)

Type AttributeConstraint

LURAL AttributeConstraints
EDPLURAL AttributeConstraints
SEDON Attribute
YPEOF Constraint

s
ute
A link back to the Attribute on which this AttributeConstraint depends.
(Attribute value M_1)

s
ute
A link back to the Attribute on which this AttributeConstraint depends.
(Attribute value M_1)

Type CodeExpression

LURAL CodeExpressions
EDPLURAL CodeExpressions

ge
the programming language
(Code value O_O)

```
OCL, Object Constraint Language
Java, Java
```

ion
(String value O_O)

Method
A behavior or operation associated with a class

PLURAL Methods
BTYPEOF Component

neters The input parameters of the method _
(List of Parameters value O_O)

INVERSE ParameterAnInputToAMethod.inverseOfParameters

nType The data type of the value returned by the method _
(DataType value O_O)

utes
thods Inverse attribute for Class.methods from which this was implied.
(Class value M_1)

INVERSE Class.methods

thods Inverse attribute for Class.methods from which this was implied.
(Class value M_1)

INVERSE Class.methods

thods Inverse attribute for Class.methods from which this was implied.
(Class value M_1)

INVERSE Class.methods

ParameterAnInputToAMethod

PLURAL Parameters
BTYPEOF Component

type The data type of the parameter _
(DataType value O_O)

inality The cardinality of the parameter
(InventedName value O_O)

utes
ameters
thod Inverse attribute for Method.parameters from which this was implied.
(Method value M_1)

INVERSE Method.parameters

utes
ameters
thod Inverse attribute for Method.parameters from which this was implied.
(Method value M_1)

INVERSE Method.parameters

Type DataType

LURAL DataTypes
EDPLURAL DataTypes

Type SimpleDataTypeSubtpeOfDataType

LURAL SimpleDataTypeSubtpeOfDataTypes
EDPLURAL SimpleDataTypeSubtpeOfDataTypes

ss (Class value O_O)

VERSE Class.inverseOfCoreClass

Type ComplexDataType

LURAL ComplexDataTypes
EDPLURAL ComplexDataTypes

on (AggregatingOperator value O_O)

es (List of DataTypes value O_O)

Type AggregatingOperator

LURAL AggregatingOperators
EDPLURAL AggregatingOperators

me (Code value O_O)

```
SetOf
ListOf
Mapping
```

ity (Integer value O_O)

ng (Template value O_O)

Type Emoji

LURAL Emojis
EDPLURAL Emojis

Type String

LURAL Strings
EDPLURAL Strings

Type CamelName

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

Plural CamelNames
umedPlural CamelNames
btypeOf String
Subtypes UpperCamel , LowerCamel

String ***(String value O_O)***

straints Must follow the camel case naming convention and not be empty.
example "firstName", "orderDate", "customerID"

elingNote

- *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.

lue Type **UpperCamel** a CamelName that begins with a capital letter
example _ "Customer", "ProductCategory", "PaymentMethod"
Where content begins with an upper case letter.
Plural UpperCamels
umedPlural UpperCamels
btypeOf CamelName

lue Type **LowerCamel** a CamelName that begins with a lower case letter
example "firstName", "orderTotal", "shippingAddress"
Where content begins with a lower case letter.
Plural LowerCamels
umedPlural LowerCamels
btypeOf CamelName

lue Type **QualifiedCamel**
an expression consisting of Camel Names separated by periods

Plural QualifiedCamels
umedPlural QualifiedCamels
btypeOf String
straints

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

ValueTypeRichText
A string with markup for block level formatting.

LURAL ValueTypeRichTexts
EDPLURAL ValueTypeRichTexts
YPEOF String

lue the string content
(String value O_O)

nat the rich text coding language used
(Code value O_O)

```
HTML
MarkDown
```

Type **OneLiner** String with markup for line level formatting.

LURAL OneLiners
EDPLURAL OneLiners
YPEOF RichText

lue the string content
(String value O_O)

RAINTS must not containa line break or new line character
ESSAGE A line can't span two lines

Type **PrimitiveType** A basic, built-in data type

LURAL PrimitiveTypes
EDPLURAL PrimitiveTypes
TYPES String , Integer , Decimal , Boolean , Date , Time , DateTime

Type **String**

LURAL Strings
EDPLURAL Strings
YPEOF PrimitiveType
TYPES CamelName , QualifiedCamel , ValueTypeRichText

Type **Integer**

LURAL Integers
EDPLURAL Integers
YPEOF PrimitiveType

Type **Decimal**

PLURAL Decimals
UMEDPLURAL Decimals
BTYPEOF PrimitiveType

lue Type Boolean
PLURAL Booleans
UMEDPLURAL Booleans
BTYPEOF PrimitiveType

lue Type Date
PLURAL Dates
UMEDPLURAL Dates
BTYPEOF PrimitiveType

lue Type Time
PLURAL Times
UMEDPLURAL Times
BTYPEOF PrimitiveType

lue Type DateTime
PLURAL DateTimes
UMEDPLURAL DateTimes
BTYPEOF PrimitiveType

Component
An element or building block of the literate data model

LURAL Components
EDPLURAL Components
DENTS Annotation
TYPES LiterateDataModel , Subject , Class , Key , AttributeSection , Attribute , Constraint , Method , ParameterAnInputToAMethod

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

arning This is a warning with emoji

me The name of the component
(CamelName value O_O)

me (QualifiedCamel value O_O)

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

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

EFAULT name - how do you say name in english?
OCL x.name == y

RAINTS the abbreviated name should be shorter than the actual name
OCL len(abbreviatedName) < len(name)
ESSAGE Why have an abbreviation longer than the name?
VERITY Warning

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

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

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

mechanical attributes

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

EFAULT 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.

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 *recognized* or *registered* Annotation Types.

LURAL AnnotationTypes
EDPLURAL AnnotationTypes
SEDON LiterateDataModel

oji an emoji
(Emoji value O_O)

me an emoji
(String value O_O)

de the Unicode for the emoji
(String value O_O)

bel A short label to indicate the purpose of the annotation _
(LowerCamel value O_O)

ral the plural form of the label
(UpperCamel value O_O)

EFAULT based on label

se the intended reason for the annotation.
(OneLiner value O_O)

s Model A link back to the LiterateDataModel on which this AnnotationType depends.
(LiterateDataModel value M_1)

s tionType verse attribute for Annotation.annotationType from which this was implied.
(Annotation value M_1)

VERSE Annotation.annotationType

s Model A link back to the LiterateDataModel on which this AnnotationType depends.
(LiterateDataModel value M_1)

s tionType verse attribute for Annotation.annotationType from which this was implied.
(Annotation value M_1)

INVERSE Annotation.annotationType

Annotation
A note or comment associated with a model element

PLURAL Annotations
UMEDPLURAL Annotations
BASEDON Component

nType (Optional AnnotationType value O_O)

note An Annotation is considered to *recognized* if the label is associated with an Annotation Type. otherwise it is *ad hoc* .

note Should be a Value Type

INVERSE AnnotationType.inverseOfAnnotationType

label A short label to indicate the purpose of the annotation _
(CamelName value O_O)

But any short label is valid.

DEFAULT from annotationType

emoji (Optional Emoji value O_O)

DEFAULT from annotation type

ontent The content or body of the annotation
(RichText value O_O)

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

DEFAULT 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.

utes
onent A link back to the Component on which this Annotation depends.
(Component value M_1)

onent A link back to the Component on which this Annotation depends.
(Component value M_1)

LiterateDataModel
A representation of a domain's entities, attributes, and relationships, along with explanatory text and examples

LURAL LiterateDataModels
DENTS AnnotationType , Subject
YPEOF Component

me (UpperCamel value O_O)
RRIDES Component.name

cts list of all classes in the model, as ordered in the definition of the model.
(List of Classes value O_O)
VERSE Class.inverseOfAllSubjects
VATION gathering s.allSubjects over s in subjectAreas
RAINTS Subject names must be unique across the model.

ses list of all classes in the model, as ordered in the definition of the model.
(List of Classes value O_O)
VERSE Class.inverseOfAllClasses
VATION gathering s.allClasses over s in allSubjects.
RAINTS Class names must be unique across the model.

pes (List of AnnotationTypes value O_O)

Language the recommended lanquage for expressing derivation, defaults, and constraints
(CodingLanguage value O_O)
EFAULT OCL

Languages (Optional List of CodingLanguages value O_O)

teLanguage the recommended lanquage for expressing derivation, defaults, and constraints
(TemplateLanguage value O_O)
EFAULT Handlebars

eLanguages (Optional List of TemplateLanguages value O_O)

ns A list of functions that require sophisticated AI-powered implementation *
(List of String value O_O)
VATION ['aiEnglishPlural()']

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 LiterateDataModel
BTYPEOF Component
SUBTYPES SubjectArea

name *(UpperCamel value O_O)*
VERRIDES Component.name

ubject The parent subject, if any, under which this subject is nested _
(Optional Subject value O_O)
INVERSE Subject.inverseOfParentSubject

asses 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

bjects 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.inverseOfChildSubjects

ataModel A link back to the LiterateDataModel on which this Subject depends.
(LiterateDataModel value M_1)

entSubject Inverse attribute for Subject.parentSubject from which this was implied.
(Subject value M_1)

INVERSE Subject.parentSubject

dSubjects Inverse attribute for Subject.childSubjects from which this was implied.
(Subject value M_1)

VERSE Subject.childSubjects

Model A link back to the LiterateDataModel on which this Subject depends.
(LiterateDataModel value M_1)

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

VERSE Subject.parentSubject

ubjects Inverse attribute for Subject.childSubjects from which this was implied.
(Subject value M_1)

VERSE Subject.childSubjects

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

WHERE parentSubject is absent
LURAL SubjectAreas
SEDON LiterateModel , Xyz
YPEOF Subject

s
del A link back to the LiterateModel on which this SubjectArea depends.
(LiterateModel value M_1)

s
Xyz A link back to the Xyz on which this SubjectArea depends.
(Xyz value M_1)

s
del A link back to the LiterateModel on which this SubjectArea depends.
(LiterateModel value M_1)

s
Xyz A link back to the Xyz on which this SubjectArea depends.
(Xyz value M_1)

Class

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

PLURAL Classes

PENDENTS Subtyping , Key , AttributeSection , ClassConstraint

BTYPEOF Component

UBTYPES ReferenceType

STRAINTS Within each Class, attribute names must be unique.

lForm 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".

sedOn 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 synonymousle in this metamodel.

INVERSE Class.inverseOfBasedOn

types The parent class

(Es value O_O)

pings 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

types Any subtypes or specializations of this class based on it's subtypings.

(List of Classes value O_O)

ample For instance, using the `Book` example, the subtypes could include `FictionBook` , `Non-fictionBook` , `HardcoverBook` , `PaperbackBook` , `ScienceBook` , and `HistoryBook` .

VERSE Class.inverseOfSubtypes

tes The attributes or properties of the class, in the order in which they should be presented _

(List of Attributes value O_O)

VERSE Attribute.inverseOfAttributes

ns additional attributes or properties of the class, grouped for clarity and elaboration. _

(List of AttributeSections value O_O)

VERSE AttributeSection.inverseOfAttributeSections

nts Any constraints, rules, or validations specific to this class _

(List of Constraints 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 Methods value O_O)

VERSE Method.inverseOfMethods

s

nts the Classes which are basedOn this Class

(Optional Set of Classes value O_O)

VERSE Class.basedOn

eys (Optional Set of UniqueKeys value O_O)

VERSE UniqueKey.basedOn

s

ects Inverse attribute for LiterateDataModel.allSubjects from which this was implied.

(LiterateDataModel value M_1)

VERSE LiterateDataModel.allSubjects

ses Inverse attribute for LiterateDataModel.allClasses from which this was implied.

(LiterateDataModel value M_1)

VERSE LiterateDataModel.allClasses

ses Inverse attribute for Subject.classes from which this was implied.

(Subject value M_1)

INVERSE Subject.classes

sedOn Inverse attribute for Class.basedOn from which this was implied.
(Class value M_1)

INVERSE Class.basedOn

types Inverse attribute for Class.subtypes from which this was implied.
(Class value M_1)

INVERSE Class.subtypes

asses Inverse attribute for Subtyping.classes from which this was implied.
(Subtyping value M_1)

INVERSE Subtyping.classes

eClass Inverse attribute for SimpleDataTypeSubtpeOfDataType.coreClass from which this was implied.
(SimpleDataTypeSubtpeOfDataType value M_1)

INVERSE SimpleDataTypeSubtpeOfDataType.coreClass

ubjects Inverse attribute for LiterateDataModel.allSubjects from which this was implied.
(LiterateDataModel value M_1)

INVERSE LiterateDataModel.allSubjects

lasses Inverse attribute for LiterateDataModel.allClasses from which this was implied.
(LiterateDataModel value M_1)

INVERSE LiterateDataModel.allClasses

asses Inverse attribute for Subject.classes from which this was implied.
(Subject value M_1)

INVERSE Subject.classes

sedOn Inverse attribute for Class.basedOn from which this was implied.
(Class value M_1)

INVERSE Class.basedOn

types Inverse attribute for Class.subtypes from which this was implied.
(Class value M_1)

INVERSE Class.subtypes

asses Inverse attribute for Subtyping.classes from which this was implied.
(Subtyping value M_1)

INVERSE Subtyping.classes

On Inverse attribute for Class.basedOn from which this was implied.
(Class value M_1)

VERSE Class.basedOn

es Inverse attribute for Class.subtypes from which this was implied.
(Class value M_1)

VERSE Class.subtypes

es Inverse attribute for Subtyping.classes from which this was implied.
(Subtyping value M_1)

VERSE Subtyping.classes

ass Inverse attribute for SimpleDataTypeSubtpeOfDataType.coreClass from which this was implied.
(SimpleDataTypeSubtpeOfDataType value M_1)

VERSE SimpleDataTypeSubtpeOfDataType.coreClass

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

PLURAL Subtypings
UMEDPLURAL Subtypings
BASEDON Class

name *(LowerCamel value O_O)*

lusive *(Boolean value O_O)*

DEFAULT true

ustive *(Boolean value O_O)*

DEFAULT true

asses *(List of Classes value O_O)*

DSL : Shown in the DSL as

- Subbtypes: 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

utes

typings 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)

typings 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)

ings Inverse attribute for Class.subtypings from which this was implied.
(Class value M_1)

VERSE Class.subtypings

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

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

LURAL ReferenceTypes
EDPLURAL ReferenceTypes
YPEOF Class

Type **CodeType** A data type or enumeration used in the model

LURAL CodeTypes
EDPLURAL CodeTypes
DENTS CodeValue

ive the code type was implied by use in an attribute and is only used for that attribute
(Boolean value O_O)

Type **CodeValue** A possible value for an enumerated data class

LURAL CodeValues
EDPLURAL CodeValues
SEDON CodeType

de A short code or abbreviationi for the value _
(NameString value O_O)

on 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:

s
ype A link back to the CodeType on which this CodeValue depends.
(CodeType value M_1)

s
ype A link back to the CodeType on which this CodeValue depends.
(CodeType value M_1)

utes

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

(CodeType value M_1)

Key

a list of attributes of a class

PLURAL Keys

UMEDPLURAL Keys

BASEDON Class

BTYPEOF Component

SUBTYPES UniqueKey

ibutes the attributes of the base Class.

(List of Attributes value O_O)

INVERSE Attribute.inverseOfKeyAttributes

STRAINTS each attribute must be a direct or inherited of the base class.

STRAINTS no repetitions allowed in keyAttributes

Issue : introduce PureLists?

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

utes

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

(Class value M_1)

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

(Class value M_1)

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

(Class value M_1)

UniqueKey

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

note order unimportant for Unique Keys.

PLURAL UniqueKeys

UMEDPLURAL UniqueKeys

BTYPEOF Key

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

LURAL Classes
DENTS Subtyping , Key , AttributeSection , ClassConstraint
YPEOF Component
TYPES ReferenceType
RAINTS Within each Class, attribute names must be unique.

rm 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.

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

On 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 synonymousle in this metamodel.

VERSE Class.inverseOfBasedOn

es The parent class
(Es value O_O)

gs the criteria, or dimensions, by which the class can be divided into subtypes
(List of Subtypings value O_O)

ample 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).

VERSE Subtyping.inverseOfSubtypings

es Any subtypes or specializations of this class based on it's subtypings.
(List of Classes value O_O)

example For instance, using the `Book` example, the subtypes could include `FictionBook` , `Non-fictionBook` , `HardcoverBook` , `PaperbackBook` , `ScienceBook` , and `HistoryBook` .

INVERSE Class.inverseOfSubtypes

ibutes The attributes or properties of the class, in the order in which they should be presented _

(List of Attributes value O_O)

INVERSE Attribute.inverseOfAttributes

ctions additional attributes or properties of the class, grouped for clarity and elaboration. _

(List of AttributeSections value O_O)

INVERSE AttributeSection.inverseOfAttributeSections

raints Any constraints, rules, or validations specific to this class _

(List of Constraints value O_O)

note Constraints may be expressed on either the Class or the Attribute. Always?

thods Any behaviors or operations associated with this class _

(List of Methods value O_O)

INVERSE Method.inverseOfMethods

utes

dents the Classes which are basedOn this Class

(Optional Set of Classes value O_O)

INVERSE Class.basedOn

eKeys ***(Optional Set of UniqueKeys value O_O)***

INVERSE UniqueKey.basedOn

utes

ubjects Inverse attribute for LiterateDataModel.allSubjects from which this was implied.

(LiterateDataModel value M_1)

INVERSE LiterateDataModel.allSubjects

lasses Inverse attribute for LiterateDataModel.allClasses from which this was implied.

(LiterateDataModel value M_1)

INVERSE LiterateDataModel.allClasses

asses Inverse attribute for Subject.classes from which this was implied.

(Subject value M_1)

VERSE Subject.classes

On Inverse attribute for Class.basedOn from which this was implied.
(Class value M_1)

VERSE Class.basedOn

es Inverse attribute for Class.subtypes from which this was implied.
(Class value M_1)

VERSE Class.subtypes

es Inverse attribute for Subtyping.classes from which this was implied.
(Subtyping value M_1)

VERSE Subtyping.classes

ass Inverse attribute for SimpleDataTypeSubtpeOfDataType.coreClass from which this was implied.
(SimpleDataTypeSubtpeOfDataType value M_1)

VERSE SimpleDataTypeSubtpeOfDataType.coreClass

ects Inverse attribute for LiterateDataModel.allSubjects from which this was implied.
(LiterateDataModel value M_1)

VERSE LiterateDataModel.allSubjects

ses Inverse attribute for LiterateDataModel.allClasses from which this was implied.
(LiterateDataModel value M_1)

VERSE LiterateDataModel.allClasses

es Inverse attribute for Subject.classes from which this was implied.
(Subject value M_1)

VERSE Subject.classes

On Inverse attribute for Class.basedOn from which this was implied.
(Class value M_1)

VERSE Class.basedOn

es Inverse attribute for Class.subtypes from which this was implied.
(Class value M_1)

VERSE Class.subtypes

es Inverse attribute for Subtyping.classes from which this was implied.
(Subtyping value M_1)

VERSE Subtyping.classes

sedOn	Inverse attribute for Class.basedOn from which this was implied. ***(Class value M_1)***
INVERSE	Class.basedOn
types	Inverse attribute for Class.subtypes from which this was implied. ***(Class value M_1)***
INVERSE	Class.subtypes
asses	Inverse attribute for Subtyping.classes from which this was implied. ***(Subtyping value M_1)***
INVERSE	Subtyping.classes
eClass	Inverse attribute for SimpleDataTypeSubtpeOfDataType.coreClass from which this was implied. ***(SimpleDataTypeSubtpeOfDataType value M_1)***
INVERSE	SimpleDataTypeSubtpeOfDataType.coreClass

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

LURAL Subtypings
EDPLURAL Subtypings
SEDON Class

me (LowerCamel value O_O)

ive (Boolean value O_O)

EFAULT true

ive (Boolean value O_O)

EFAULT true

ses (List of Classes value O_O)

DSL : Shown in the DSL as

- Subbtypes: 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.

VERSE Class.inverseOfClasses

s
ings Inverse attribute for Class.subtypings from which this was implied.
(Class value M_1)

VERSE Class.subtypings

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

ings Inverse attribute for Class.subtypings from which this was implied.
(Class value M_1)

VERSE Class.subtypings

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

typings 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)

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

PLURAL ReferenceTypes
UMEDPLURAL ReferenceTypes
BTYPEOF Class

lue Type **CodeType** A data type or enumeration used in the model

PLURAL CodeTypes
UMEDPLURAL CodeTypes
PENDENTS CodeValue

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

(Boolean value O_O)

lue Type **CodeValue** A possible value for an enumerated data class

PLURAL CodeValues
UMEDPLURAL CodeValues
BASEDON CodeType

code A short code or abbreviationi for the value _

(NameString value O_O)

iption 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:

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

(CodeType value M_1)

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

(CodeType value M_1)

s
ype

A link back to the CodeType on which this CodeValue depends.
(CodeType value M_1)

Key
a list of attributes of a class

LURAL Keys
EDPLURAL Keys
SEDON Class
YPEOF Component
TYPES UniqueKey

tes

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

VERSE Attribute.inverseOfKeyAttributes
RAINTS each attribute must be a direct or inherited of the base class.
RAINTS no repetitions allowed in keyAttributes

Issue : introduce PureLists?

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

s
ss

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

ss

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

ss

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

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

note order unimportant for Unique Keys.

LURAL UniqueKeys
EDPLURAL UniqueKeys
YPEOF Key

AttributeSection

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

PLURAL AttributeSections
UMEDPLURAL AttributeSections
BASEDON Class
PENDENTS Attribute
BTYPEOF Component

tional 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 ot required, depending on how it is marked.

-
- But if the Arrribute Section is optional each attribute in the section is only required if any attribute in the section is ptresent.

utes
ibuteSections verse attribute for Class.attributeSections from which this was implied.
(Class value M_1)
INVERSE Class.attributeSections

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

utes
ibuteSections verse attribute for Class.attributeSections from which this was implied.
(Class value M_1)
INVERSE Class.attributeSections

utes
ibuteSections verse attribute for Class.attributeSections from which this was implied.
(Class value M_1)
INVERSE Class.attributeSections

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

Attribute
A property or characteristic of a class

LURAL Attributes
SEDON AttributeSection
DENTS AttributeConstraint
YPEOF Component

me (LowerCamel value O_O)
RRIDES Component.name

ype The kind of object to which the attribute refers. _
(DataType value O_O)

But,

- ◦ 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)

EFAULT *** False

lity The cardinality of the relationship represented by the attribute _
(CardinalityCode value O_O)

EFAULT
*** 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.

xample

nor (InventedName value O_O)

ks (Optional InventedName value O_O)

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.

Class	the class which contains, or would contain the inverse attribute (Optional Class value O_O)
ERIVATION	from the data type. Null unless arrribute is invertible.
ribute	(Optional Attribute value O_O)
tional	(Optional Attribute value O_O)
efault	The rule or formula for calculating the value, if no value is supplied Now running to a second line with the parenthentical on yet a third line (Optional Derivation 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.
vation	For derived attributes, the rule or formula for calculating the value _ (Optional Derivation value O_O)
issue	on insert vs on access?
raints	Any validation rules specific to this attribute _ (List of Constraints value O_O)
note	from Class.constraints
king rrides utes	
ibutes	Inverse attribute for Class.attributes from which this was implied. (Class value M_1)
INVERSE	Class.attributes
Attributes	Inverse attribute for Key.keyAttributes from which this was implied. (Key value M_1)
INVERSE	Key.keyAttributes
Section	A link back to the AttributeSection on which this Attribute depends. (AttributeSection value M_1)
ibutes	Inverse attribute for Class.attributes from which this was implied. (Class value M_1)
INVERSE	Class.attributes
Attributes	Inverse attribute for Key.keyAttributes from which this was implied. (Key value M_1)
INVERSE	Key.keyAttributes

tes Inverse attribute for Class.attributes from which this was implied.
(Class value M_1)

VERSE Class.attributes

ributes Inverse attribute for Key.keyAttributes from which this was implied.
(Key value M_1)

VERSE Key.keyAttributes

tion A link back to the AttributeSection on which this Attribute depends.
(AttributeSection value M_1)

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

LURAL Derivations

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

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

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

LURAL Constraints
YPEOF Component
TYPES ClassConstraint , AttributeConstraint

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

ion The formal expression of the constraint in a programming language
(InventedName value O_O)

ity (Code value O_O)

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

Type Message

LURAL Messages
EDPLURAL Messages

Type ClassConstraint

LURAL ClassConstraints
EDPLURAL ClassConstraints

BASEDON Class
BTYPEOF Constraint

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

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

lue Type AttributeConstraint

PLURAL AttributeConstraints
UMEDPLURAL AttributeConstraints
BASEDON Attribute
BTYPEOF Constraint

utes
ribute
A link back to the Attribute on which this AttributeConstraint depends.
(Attribute value M_1)

utes
ribute
A link back to the Attribute on which this AttributeConstraint depends.
(Attribute value M_1)

lue Type CodeExpression

PLURAL CodeExpressions
UMEDPLURAL CodeExpressions

guage
the programming language
(Code value O_O)

```
OCL, Object Constraint Language
Java, Java
```

ession
(String value O_O)

Method
A behavior or operation associated with a class

LURAL Methods
YPEOF Component

ers The input parameters of the method _
(List of Parameters value O_O)
VERSE ParameterAnInputToAMethod.inverseOfParameters

pe The data type of the value returned by the method _
(DataType value O_O)

s
ds Inverse attribute for Class.methods from which this was implied.
(Class value M_1)
VERSE Class.methods

ds Inverse attribute for Class.methods from which this was implied.
(Class value M_1)
VERSE Class.methods

ds Inverse attribute for Class.methods from which this was implied.
(Class value M_1)
VERSE Class.methods

ParameterAnInputToAMethod

LURAL Parameters
YPEOF Component

pe The data type of the parameter _
(DataType value O_O)

ity The cardinality of the parameter
(InventedName value O_O)

s
eters Inverse attribute for Method.parameters from which this was implied.
d (Method value M_1)
VERSE Method.parameters

s
eters Inverse attribute for Method.parameters from which this was implied.
d (Method value M_1)
VERSE Method.parameters

lue Type DataType

PLURAL DataTypes
UMEDPLURAL DataTypes

lue Type SimpleDataTypeSubtpeOfDataType

PLURAL SimpleDataTypeSubtpeOfDataTypes
UMEDPLURAL SimpleDataTypeSubtpeOfDataTypes

Class (Class value O_O)

INVERSE Class.inverseOfCoreClass

lue Type ComplexDataType

PLURAL ComplexDataTypes
UMEDPLURAL ComplexDataTypes

gation (AggregatingOperator value O_O)

Types (List of DataTypes value O_O)

lue Type AggregatingOperator

PLURAL AggregatingOperators
UMEDPLURAL AggregatingOperators

name (Code value O_O)

```
SetOf
ListOf
Mapping
```

arity (Integer value O_O)

elling (Template value O_O)

lue Type Emoji

PLURAL Emojis
UMEDPLURAL Emojis

lue Type String

PLURAL Strings
UMEDPLURAL Strings

lue Type CamelName

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

LURAL CamelNames
EDPLURAL CamelNames
YPEOF String
TYPES UpperCamel , LowerCamel

ng (String value O_O)

RAINTS Must follow the camel case naming convention and not be empty.
xample "firstName", "orderDate", "customerID"

gNote

- *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.

Type UpperCamel a CamelName that begins with a capital letter

ample _ "Customer", "ProductCategory", "PaymentMethod"
WHERE content begins with an upper case letter.
LURAL UpperCamels
EDPLURAL UpperCamels
YPEOF CamelName

Type LowerCamel a CamelName that begins with a lower case letter

ample "firstName", "orderTotal", "shippingAddress"
WHERE content begins with a lower case letter.
LURAL LowerCamels
EDPLURAL LowerCamels
YPEOF CamelName

Type QualifiedCamel
an expression consisting of Camel Names separated by periods

LURAL QualifiedCamels
EDPLURAL QualifiedCamels
YPEOF String
RAINTS

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

ValueTypeRichText
A string with markup for block level formatting.

PLURAL ValueTypeRichTexts
UMEDPLURAL ValueTypeRichTexts
BTYPEOF String

value the string content
(String value O_O)

ormat the rich text coding language used
(Code value O_O)

```
HTML
MarkDown
```

lue Type **OneLiner** String with markup for line level formatting.

PLURAL OneLiners
UMEDPLURAL OneLiners
BTYPEOF RichText

value the string content
(String value O_O)

STRAINTS must not containa line break or new line character
MESSAGE A line can't span two lines

lue Type **PrimitiveType** A basic, built-in data type

PLURAL PrimitiveTypes
UMEDPLURAL PrimitiveTypes
UBTYPES String , Integer , Decimal , Boolean , Date , Time , DateTime

lue Type **String**

PLURAL Strings
UMEDPLURAL Strings
BTYPEOF PrimitiveType
UBTYPES CamelName , QualifiedCamel , ValueTypeRichText

lue Type **Integer**

PLURAL Integers
UMEDPLURAL Integers
BTYPEOF PrimitiveType

lue Type **Decimal**

LURAL Decimals
EDPLURAL Decimals
YPEOF PrimitiveType

Type **Boolean**

LURAL Booleans
EDPLURAL Booleans
YPEOF PrimitiveType

Type **Date**

LURAL Dates
EDPLURAL Dates
YPEOF PrimitiveType

Type **Time**

LURAL Times
EDPLURAL Times
YPEOF PrimitiveType

Type **DateTime**

LURAL DateTimes
EDPLURAL DateTimes
YPEOF PrimitiveType

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.

Component
An element or building block of the literate data model

PLURAL Components
UMEDPLURAL Components
PENDENTS Annotation
SUBTYPES LiterateDataModel , Subject , Class , Key , AttributeSection , Attribute , Constraint , Method , ParameterAnInputToAMethod

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

STRAINTS 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!

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

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

y mechanical attributes

nment 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.

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 *recognized* or *registered* Annotation Types.

PLURAL AnnotationTypes
UMEDPLURAL AnnotationTypes
BASEDON LiteralDataModel

emoji
an emoji
(Emoji value O_O)

Name
an emoji
(String value O_O)

icode
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

rpose
the intended reason for the annotation.
(OneLiner value O_O)

ataModel
A link back to the LiterateDataModel on which this AnnotationType depends.
(LiterateDataModel value M_1)

otationType
Inverse attribute for Annotation.annotationType from which this was implied.
(Annotation value M_1)

INVERSE Annotation.annotationType

ataModel
A link back to the LiterateDataModel on which this AnnotationType depends.
(LiterateDataModel value M_1)

otationType
Inverse attribute for Annotation.annotationType from which this was implied.
(Annotation value M_1)

VERSE Annotation.annotationType

Annotation
A note or comment associated with a model element

LURAL Annotations
EDPLURAL Annotations
SEDON Component

ype *(Optional AnnotationType value O_O)*

note An Annotation is considered to *recognized* if the label is associated with an Annotation Type. otherwise it is *ad hoc* .

note Should be a Value Type

VERSE AnnotationType.inverseOfAnnotationType

bel A short label to indicate the purpose of the annotation _
(CamelName value O_O)

But any short label is valid.

EFAULT from annotationType

oji *(Optional Emoji value O_O)*

EFAULT from annotation type

ent The content or body of the annotation
(RichText value O_O)

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

EFAULT 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.

ent A link back to the Component on which this Annotation depends.
(Component value M_1)

ent A link back to the Component on which this Annotation depends.
(Component value M_1)

The Model and its Subjects

LiterateDataModel
A representation of a domain's entities, attributes, and relationships, along with explanatory text and examples

LURAL LiterateDataModels
DENTS AnnotationType , Subject
YPEOF Component

me *(UpperCamel value O_O)*
RRIDES Component.name

cts list of all classes in the model, as ordered in the definition of the model.
(List of Classes value O_O)
VERSE Class.inverseOfAllSubjects
VATION gathering s.allSubjects over s in subjectAreas
RAINTS Subject names must be unique across the model.

ses list of all classes in the model, as ordered in the definition of the model.
(List of Classes value O_O)
VERSE Class.inverseOfAllClasses
VATION gathering s.allClasses over s in allSubjects.
RAINTS Class names must be unique across the model.

pes *(List of AnnotationTypes value O_O)*

Language the recommended lanquage for expressing derivation, defaults, and constraints
(CodingLanguage value O_O)
EFAULT OCL

Languages *(Optional List of CodingLanguages value O_O)*

teLanguage the recommended lanquage for expressing derivation, defaults, and constraints
(TemplateLanguage value O_O)
EFAULT Handlebars

eLanguages *(Optional List of TemplateLanguages value O_O)*

ns A list of functions that require sophisticated AI-powered implementation *
(List of String value O_O)
VATION ['aiEnglishPlural()']

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 LiterateDataModel
BTYPEOF Component
SUBTYPES SubjectArea

name (*UpperCamel* value O_O)
VERRIDES Component.name

ubject The parent subject, if any, under which this subject is nested _
(*Optional Subject* value O_O)
INVERSE Subject.inverseOfParentSubject

asses 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

bjects 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.inverseOfChildSubjects

utes
ataModel A link back to the LiterateDataModel on which this Subject depends.
(*LiterateDataModel* value M_1)

ntSubject Inverse attribute for Subject.parentSubject from which this was implied.
(*Subject* value M_1)
INVERSE Subject.parentSubject

dSubjects Inverse attribute for Subject.childSubjects from which this was implied.
(*Subject* value M_1)

VERSE Subject.childSubjects

Model A link back to the LiterateDataModel on which this Subject depends.
(LiterateDataModel value M_1)

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

VERSE Subject.parentSubject

ubjects Inverse attribute for Subject.childSubjects from which this was implied.
(Subject value M_1)

VERSE Subject.childSubjects

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

WHERE parentSubject is absent
LURAL SubjectAreas
SEDON LiterateModel , Xyz
YPEOF Subject

s
del A link back to the LiterateModel on which this SubjectArea depends.
(LiterateModel value M_1)

s
Xyz A link back to the Xyz on which this SubjectArea depends.
(Xyz value M_1)

s
del A link back to the LiterateModel on which this SubjectArea depends.
(LiterateModel value M_1)

s
Xyz A link back to the Xyz on which this SubjectArea depends.
(Xyz value M_1)

Class

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

PLURAL Classes

PENDENTS Subtyping , Key , AttributeSection , ClassConstraint

BTYPEOF Component

UBTYPES ReferenceType

STRAINTS Within each Class, attribute names must be unique.

lForm 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".

sedOn 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 synonymousle in this metamodel.

INVERSE Class.inverseOfBasedOn

types The parent class

(Es value O_O)

pings 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

types Any subtypes or specializations of this class based on it's subtypings.

(List of Classes value O_O)

ample For instance, using the `Book` example, the subtypes could include `FictionBook` , `Non-fictionBook` , `HardcoverBook` , `PaperbackBook` , `ScienceBook` , and `HistoryBook` .

VERSE Class.inverseOfSubtypes

tes The attributes or properties of the class, in the order in which they should be presented _

(List of Attributes value O_O)

VERSE Attribute.inverseOfAttributes

ns additional attributes or properties of the class, grouped for clarity and elaboration. _

(List of AttributeSections value O_O)

VERSE AttributeSection.inverseOfAttributeSections

nts Any constraints, rules, or validations specific to this class _

(List of Constraints 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 Methods value O_O)

VERSE Method.inverseOfMethods

s

nts the Classes which are basedOn this Class

(Optional Set of Classes value O_O)

VERSE Class.basedOn

eys (Optional Set of UniqueKeys value O_O)

VERSE UniqueKey.basedOn

s

ects Inverse attribute for LiterateDataModel.allSubjects from which this was implied.

(LiterateDataModel value M_1)

VERSE LiterateDataModel.allSubjects

ses Inverse attribute for LiterateDataModel.allClasses from which this was implied.

(LiterateDataModel value M_1)

VERSE LiterateDataModel.allClasses

ses Inverse attribute for Subject.classes from which this was implied.

(Subject value M_1)

INVERSE Subject.classes

sedOn Inverse attribute for Class.basedOn from which this was implied.
(Class value M_1)

INVERSE Class.basedOn

types Inverse attribute for Class.subtypes from which this was implied.
(Class value M_1)

INVERSE Class.subtypes

asses Inverse attribute for Subtyping.classes from which this was implied.
(Subtyping value M_1)

INVERSE Subtyping.classes

eClass Inverse attribute for SimpleDataTypeSubtpeOfDataType.coreClass from which this was implied.
(SimpleDataTypeSubtpeOfDataType value M_1)

INVERSE SimpleDataTypeSubtpeOfDataType.coreClass

ubjects Inverse attribute for LiterateDataModel.allSubjects from which this was implied.
(LiterateDataModel value M_1)

INVERSE LiterateDataModel.allSubjects

lasses Inverse attribute for LiterateDataModel.allClasses from which this was implied.
(LiterateDataModel value M_1)

INVERSE LiterateDataModel.allClasses

asses Inverse attribute for Subject.classes from which this was implied.
(Subject value M_1)

INVERSE Subject.classes

sedOn Inverse attribute for Class.basedOn from which this was implied.
(Class value M_1)

INVERSE Class.basedOn

types Inverse attribute for Class.subtypes from which this was implied.
(Class value M_1)

INVERSE Class.subtypes

asses Inverse attribute for Subtyping.classes from which this was implied.
(Subtyping value M_1)

INVERSE Subtyping.classes

On Inverse attribute for Class.basedOn from which this was implied.
(Class value M_1)

VERSE Class.basedOn

es Inverse attribute for Class.subtypes from which this was implied.
(Class value M_1)

VERSE Class.subtypes

es Inverse attribute for Subtyping.classes from which this was implied.
(Subtyping value M_1)

VERSE Subtyping.classes

ass Inverse attribute for SimpleDataTypeSubtpeOfDataType.coreClass from which this was implied.
(SimpleDataTypeSubtpeOfDataType value M_1)

VERSE SimpleDataTypeSubtpeOfDataType.coreClass

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

PLURAL Subtypings
UMEDPLURAL Subtypings
BASEDON Class

name *(LowerCamel value O_O)*

lusive *(Boolean value O_O)*

DEFAULT true

ustive *(Boolean value O_O)*

DEFAULT true

asses *(List of Classes value O_O)*

DSL : Shown in the DSL as

- Subbtypes: 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

utes

typings 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)

typings 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)

ings Inverse attribute for Class.subtypings from which this was implied.
(Class value M_1)

VERSE Class.subtypings

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

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

LURAL ReferenceTypes
EDPLURAL ReferenceTypes
YPEOF Class

Type **CodeType**
A data type or enumeration used in the model

LURAL CodeTypes
EDPLURAL CodeTypes
DENTS CodeValue

ive the code type was implied by use in an attribute and is only used for that attribute
(Boolean value O_O)

Type **CodeValue**
A possible value for an enumerated data class

LURAL CodeValues
EDPLURAL CodeValues
SEDON CodeType

de A short code or abbreviationi for the value _
(NameString value O_O)

on 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:

s
pe A link back to the CodeType on which this CodeValue depends.
(CodeType value M_1)

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

(CodeType value M_1)

utes
eType

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

(CodeType value M_1)

Key

a list of attributes of a class

PLURAL Keys
UMEDPLURAL Keys
BASEDON Class
BTYPEOF Component
SUBTYPES UniqueKey

ibutes

the attributes of the base Class.

(List of Attributes value O_O)

INVERSE Attribute.inverseOfKeyAttributes
STRAINTS each attribute must be a direct or inherited of the base class.
STRAINTS no repetitions allowed in keyAttributes

Issue : introduce PureLists?

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

utes
Class

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

(Class value M_1)

Class

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

(Class value M_1)

Class

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

(Class value M_1)

UniqueKey

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

note order unimportant for Unique Keys.
PLURAL UniqueKeys
UMEDPLURAL UniqueKeys
BTYPEOF Key

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

LURAL Classes
DENTS Subtyping , Key , AttributeSection , ClassConstraint
YPEOF Component
TYPES ReferenceType
RAINTS Within each Class, attribute names must be unique.

rm 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.

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

On 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 synonymousle in this metamodel.

VERSE Class.inverseOfBasedOn

es The parent class
(Es value O_O)

gs the criteria, or dimensions, by which the class can be divided into subtypes
(List of Subtypings value O_O)

ample 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).

VERSE Subtyping.inverseOfSubtypings

es Any subtypes or specializations of this class based on it's subtypings.
(List of Classes value O_O)

example For instance, using the `Book` example, the subtypes could include `FictionBook` , `Non-fictionBook` , `HardcoverBook` , `PaperbackBook` , `ScienceBook` , and `HistoryBook` .

INVERSE Class.inverseOfSubtypes

ibutes The attributes or properties of the class, in the order in which they should be presented _

(List of Attributes value O_O)

INVERSE Attribute.inverseOfAttributes

ctions additional attributes or properties of the class, grouped for clarity and elaboration. _

(List of AttributeSections value O_O)

INVERSE AttributeSection.inverseOfAttributeSections

raints Any constraints, rules, or validations specific to this class _

(List of Constraints value O_O)

note Constraints may be expressed on either the Class or the Attribute. Always?

thods Any behaviors or operations associated with this class _

(List of Methods value O_O)

INVERSE Method.inverseOfMethods

utes

dents the Classes which are basedOn this Class

(Optional Set of Classes value O_O)

INVERSE Class.basedOn

eKeys ***(Optional Set of UniqueKeys value O_O)***

INVERSE UniqueKey.basedOn

utes

ubjects Inverse attribute for LiterateDataModel.allSubjects from which this was implied.

(LiterateDataModel value M_1)

INVERSE LiterateDataModel.allSubjects

lasses Inverse attribute for LiterateDataModel.allClasses from which this was implied.

(LiterateDataModel value M_1)

INVERSE LiterateDataModel.allClasses

asses Inverse attribute for Subject.classes from which this was implied.

(Subject value M_1)

VERSE Subject.classes

On Inverse attribute for Class.basedOn from which this was implied.
(Class value M_1)

VERSE Class.basedOn

es Inverse attribute for Class.subtypes from which this was implied.
(Class value M_1)

VERSE Class.subtypes

es Inverse attribute for Subtyping.classes from which this was implied.
(Subtyping value M_1)

VERSE Subtyping.classes

ass Inverse attribute for SimpleDataTypeSubtpeOfDataType.coreClass from which this was implied.
(SimpleDataTypeSubtpeOfDataType value M_1)

VERSE SimpleDataTypeSubtpeOfDataType.coreClass

ects Inverse attribute for LiterateDataModel.allSubjects from which this was implied.
(LiterateDataModel value M_1)

VERSE LiterateDataModel.allSubjects

ses Inverse attribute for LiterateDataModel.allClasses from which this was implied.
(LiterateDataModel value M_1)

VERSE LiterateDataModel.allClasses

es Inverse attribute for Subject.classes from which this was implied.
(Subject value M_1)

VERSE Subject.classes

On Inverse attribute for Class.basedOn from which this was implied.
(Class value M_1)

VERSE Class.basedOn

es Inverse attribute for Class.subtypes from which this was implied.
(Class value M_1)

VERSE Class.subtypes

es Inverse attribute for Subtyping.classes from which this was implied.
(Subtyping value M_1)

VERSE Subtyping.classes

sedOn	Inverse attribute for Class.basedOn from which this was implied. ***(Class value M_1)***
INVERSE	Class.basedOn
types	Inverse attribute for Class.subtypes from which this was implied. ***(Class value M_1)***
INVERSE	Class.subtypes
asses	Inverse attribute for Subtyping.classes from which this was implied. ***(Subtyping value M_1)***
INVERSE	Subtyping.classes
eClass	Inverse attribute for SimpleDataTypeSubtpeOfDataType.coreClass from which this was implied. ***(SimpleDataTypeSubtpeOfDataType value M_1)***
INVERSE	SimpleDataTypeSubtpeOfDataType.coreClass

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

LURAL Subtypings
EDPLURAL Subtypings
SEDON Class

me (LowerCamel value O_O)

ive (Boolean value O_O)

EFAULT true

ive (Boolean value O_O)

EFAULT true

ses (List of Classes value O_O)

DSL : Shown in the DSL as

- Subbtypes: 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.

VERSE Class.inverseOfClasses

s
ings Inverse attribute for Class.subtypings from which this was implied.
(Class value M_1)

VERSE Class.subtypings

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

ings Inverse attribute for Class.subtypings from which this was implied.
(Class value M_1)

VERSE Class.subtypings

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

typings 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)

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

PLURAL ReferenceTypes
UMEDPLURAL ReferenceTypes
BTYPEOF Class

lue Type **CodeType**
A data type or enumeration used in the model

PLURAL CodeTypes
UMEDPLURAL CodeTypes
PENDENTS CodeValue

aptive the code type was implied by use in an attribute and is only used for that attribute
(Boolean value O_O)

lue Type **CodeValue**
A possible value for an enumerated data class

PLURAL CodeValues
UMEDPLURAL CodeValues
BASEDON CodeType

code A short code or abbreviationi for the value _
(NameString value O_O)

iption 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:

utes
eType A link back to the CodeType on which this CodeValue depends.
(CodeType value M_1)

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

(CodeType value M_1)

s
ype
A link back to the CodeType on which this CodeValue depends.
(CodeType value M_1)

Key
a list of attributes of a class

LURAL Keys
EDPLURAL Keys
SEDON Class
YPEOF Component
TYPES UniqueKey

tes
the attributes of the base Class.
(List of Attributes value O_O)

VERSE Attribute.inverseOfKeyAttributes
RAINTS each attribute must be a direct or inherited of the base class.
RAINTS no repetitions allowed in keyAttributes

Issue : introduce PureLists?

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

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

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

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

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

note order unimportant for Unique Keys.
LURAL UniqueKeys
EDPLURAL UniqueKeys
YPEOF Key

Classes

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

LURAL Classes
DENTS Subtyping , Key , AttributeSection , ClassConstraint
YPEOF Component
TYPES ReferenceType
RAINTS Within each Class, attribute names must be unique.

rm 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.

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

On 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 synonymousle in this metamodel.

VERSE Class.inverseOfBasedOn

es The parent class
(Es value O_O)

gs the criteria, or dimensions, by which the class can be divided into subtypes
(List of Subtypings value O_O)

ample 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).

VERSE Subtyping.inverseOfSubtypings

es Any subtypes or specializations of this class based on it's subtypings.
(List of Classes value O_O)

example For instance, using the `Book` example, the subtypes could include `FictionBook` , `Non-fictionBook` , `HardcoverBook` , `PaperbackBook` , `ScienceBook` , and `HistoryBook` .

INVERSE Class.inverseOfSubtypes

ibutes The attributes or properties of the class, in the order in which they should be presented _

(List of Attributes value O_O)

INVERSE Attribute.inverseOfAttributes

ctions additional attributes or properties of the class, grouped for clarity and elaboration. _

(List of AttributeSections value O_O)

INVERSE AttributeSection.inverseOfAttributeSections

raints Any constraints, rules, or validations specific to this class _

(List of Constraints value O_O)

note Constraints may be expressed on either the Class or the Attribute. Always?

thods Any behaviors or operations associated with this class _

(List of Methods value O_O)

INVERSE Method.inverseOfMethods

utes

dents the Classes which are basedOn this Class

(Optional Set of Classes value O_O)

INVERSE Class.basedOn

eKeys ***(Optional Set of UniqueKeys value O_O)***

INVERSE UniqueKey.basedOn

utes

ubjects Inverse attribute for LiterateDataModel.allSubjects from which this was implied.

(LiterateDataModel value M_1)

INVERSE LiterateDataModel.allSubjects

lasses Inverse attribute for LiterateDataModel.allClasses from which this was implied.

(LiterateDataModel value M_1)

INVERSE LiterateDataModel.allClasses

asses Inverse attribute for Subject.classes from which this was implied.

(Subject value M_1)

VERSE Subject.classes

On Inverse attribute for Class.basedOn from which this was implied.
(Class value M_1)

VERSE Class.basedOn

es Inverse attribute for Class.subtypes from which this was implied.
(Class value M_1)

VERSE Class.subtypes

es Inverse attribute for Subtyping.classes from which this was implied.
(Subtyping value M_1)

VERSE Subtyping.classes

ass Inverse attribute for SimpleDataTypeSubtpeOfDataType.coreClass from which this was implied.
(SimpleDataTypeSubtpeOfDataType value M_1)

VERSE SimpleDataTypeSubtpeOfDataType.coreClass

ects Inverse attribute for LiterateDataModel.allSubjects from which this was implied.
(LiterateDataModel value M_1)

VERSE LiterateDataModel.allSubjects

ses Inverse attribute for LiterateDataModel.allClasses from which this was implied.
(LiterateDataModel value M_1)

VERSE LiterateDataModel.allClasses

es Inverse attribute for Subject.classes from which this was implied.
(Subject value M_1)

VERSE Subject.classes

On Inverse attribute for Class.basedOn from which this was implied.
(Class value M_1)

VERSE Class.basedOn

es Inverse attribute for Class.subtypes from which this was implied.
(Class value M_1)

VERSE Class.subtypes

es Inverse attribute for Subtyping.classes from which this was implied.
(Subtyping value M_1)

VERSE Subtyping.classes

sedOn	Inverse attribute for Class.basedOn from which this was implied. *(Class value M_1)*
INVERSE	Class.basedOn
types	Inverse attribute for Class.subtypes from which this was implied. *(Class value M_1)*
INVERSE	Class.subtypes
asses	Inverse attribute for Subtyping.classes from which this was implied. *(Subtyping value M_1)*
INVERSE	Subtyping.classes
eClass	Inverse attribute for SimpleDataTypeSubtpeOfDataType.coreClass from which this was implied. *(SimpleDataTypeSubtpeOfDataType value M_1)*
INVERSE	SimpleDataTypeSubtpeOfDataType.coreClass

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

LURAL Subtypings
EDPLURAL Subtypings
SEDON Class

me (LowerCamel value O_O)

ive (Boolean value O_O)

EFAULT true

ive (Boolean value O_O)

EFAULT true

ses (List of Classes value O_O)

DSL : Shown in the DSL as

- Subbtypes: 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.

VERSE Class.inverseOfClasses

s
ings Inverse attribute for Class.subtypings from which this was implied.
(Class value M_1)

VERSE Class.subtypings

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

ings Inverse attribute for Class.subtypings from which this was implied.
(Class value M_1)

VERSE Class.subtypings

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

typings 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)

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

PLURAL ReferenceTypes
UMEDPLURAL ReferenceTypes
BTYPEOF Class

lue Type **CodeType**
A data type or enumeration used in the model

PLURAL CodeTypes
UMEDPLURAL CodeTypes
PENDENTS CodeValue

aptive the code type was implied by use in an attribute and is only used for that attribute
(Boolean value O_O)

lue Type **CodeValue**
A possible value for an enumerated data class

PLURAL CodeValues
UMEDPLURAL CodeValues
BASEDON CodeType

code A short code or abbreviationi for the value _
(NameString value O_O)

iption 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:

utes
eType A link back to the CodeType on which this CodeValue depends.
(CodeType value M_1)

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

(CodeType value M_1)

ype
A link back to the CodeType on which this CodeValue depends.
(CodeType value M_1)

Key
a list of attributes of a class

LURAL Keys
EDPLURAL Keys
SEDON Class
YPEOF Component
TYPES UniqueKey

tes
the attributes of the base Class.
(List of Attributes value O_O)

VERSE Attribute.inverseOfKeyAttributes
RAINTS each attribute must be a direct or inherited of the base class.
RAINTS no repetitions allowed in keyAttributes

Issue : introduce PureLists?

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

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

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

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

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

note order unimportant for Unique Keys.
LURAL UniqueKeys
EDPLURAL UniqueKeys
YPEOF Key

Attributes

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

LURAL AttributeSections
EDPLURAL AttributeSections
SEDON Class
DENTS Attribute
YPEOF Component

nal 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 ot required, depending on how it is marked.

-
- But if the Arrribute Section is optional each attribute in the section is only required if any attribute in the section is ptresent.

teSections nverse 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)

teSections nverse attribute for Class.attributeSections from which this was implied.
(Class value M_1)
VERSE Class.attributeSections

teSections nverse 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)

Attribute
A property or characteristic of a class

PLURAL Attributes
BASEDON AttributeSection
PENDENTS AttributeConstraint
BTYPEOF Component

name (*LowerCamel* value O_O)
VERRIDES Component.name

aType The kind of object to which the attribute refers. _
(*DataType* value O_O)

But,

- ◦ List of Editions
- ◦ Set of Edition
- ◦ ... and more complicated cases.

see the section below on Data Type Specifiers.

tional Indicates whether the attribute must have a value for every instance of the class _
(*Boolean* value O_O)

DEFAULT *** False

inality The cardinality of the relationship represented by the attribute _
(*CardinalityCode* value O_O)

DEFAULT
*** 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.

rExample

uthor (*InventedName* value O_O)

books (Optional *InventedName* value O_O)

note how this works with optionality

utes
ertible (*Boolean* value O_O)

ERIVATION 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 arrribute is invertible.

ute ***(Optional Attribute value O_O)***

nal ***(Optional Attribute value O_O)***

ult The rule or formula for calculating the value, if no value is supplied Now running to a second line with the parenthentical on yet a third line
(Optional Derivation 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.

on For derived attributes, the rule or formula for calculating the value _
(Optional Derivation value O_O)

issue on insert vs on access?

nts Any validation rules specific to this attribute _
(List of Constraints value O_O)

note from Class.constraints

tes Inverse attribute for Class.attributes from which this was implied.
(Class value M_1)

VERSE Class.attributes

ributes Inverse attribute for Key.keyAttributes from which this was implied.
(Key value M_1)

VERSE Key.keyAttributes

ction A link back to the AttributeSection on which this Attribute depends.
(AttributeSection value M_1)

tes Inverse attribute for Class.attributes from which this was implied.
(Class value M_1)

VERSE Class.attributes

ributes Inverse attribute for Key.keyAttributes from which this was implied.
(Key value M_1)

VERSE Key.keyAttributes

Attributes

ibutes Inverse attribute for Class.attributes from which this was implied.
(Class value M_1)

INVERSE Class.attributes

Attributes Inverse attribute for Key.keyAttributes from which this was implied.
(Key value M_1)

INVERSE Key.keyAttributes

Section A link back to the AttributeSection on which this Attribute depends.
(AttributeSection value M_1)

lue Type **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)

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

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

PLURAL Constraints
BTYPEOF Component
UBTYPES ClassConstraint , AttributeConstraint

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

ssion The formal expression of the constraint in a programming language
(InventedName value O_O)

verity *(Code value O_O)*

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

lue Type **Message**

PLURAL Messages
MEDPLURAL Messages

lue Type **ClassConstraint**

LURAL ClassConstraints
EDPLURAL ClassConstraints
SEDON Class
YPEOF Constraint

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

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

Type AttributeConstraint

LURAL AttributeConstraints
EDPLURAL AttributeConstraints
SEDON Attribute
YPEOF Constraint

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

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

Type CodeExpression

LURAL CodeExpressions
EDPLURAL CodeExpressions

ge
the programming language
(Code value O_O)

```
OCL, Object Constraint Language
Java, Java
```

on
(String value O_O)

Methods

Method
A behavior or operation associated with a class

LURAL Methods
YPEOF Component

ers The input parameters of the method _
(List of Parameters value O_O)
VERSE ParameterAnInputToAMethod.inverseOfParameters

ype The data type of the value returned by the method _
(DataType value O_O)

s
ds Inverse attribute for Class.methods from which this was implied.
(Class value M_1)
VERSE Class.methods

ds Inverse attribute for Class.methods from which this was implied.
(Class value M_1)
VERSE Class.methods

ds Inverse attribute for Class.methods from which this was implied.
(Class value M_1)
VERSE Class.methods

Methods

ParameterAnInputToAMethod

PLURAL Parameters

BTYPEOF Component

type The data type of the parameter _

(DataType value O_O)

inality The cardinality of the parameter

(InventedName value O_O)

utes
ameters Inverse attribute for Method.parameters from which this was implied.
thod

(Method value M_1)

INVERSE Method.parameters

utes
ameters Inverse attribute for Method.parameters from which this was implied.
thod

(Method value M_1)

INVERSE Method.parameters

BLANK

Data Types

Type DataType

LURAL DataTypes
EDPLURAL DataTypes

Type SimpleDataTypeSubtpeOfDataType

LURAL SimpleDataTypeSubtpeOfDataTypes
EDPLURAL SimpleDataTypeSubtpeOfDataTypes

ss (Class value O_O)

VERSE Class.inverseOfCoreClass

Type ComplexDataType

LURAL ComplexDataTypes
EDPLURAL ComplexDataTypes

on (AggregatingOperator value O_O)

es (List of DataTypes value O_O)

Type AggregatingOperator

LURAL AggregatingOperators
EDPLURAL AggregatingOperators

me (Code value O_O)

```
SetOf
ListOf
Mapping
```

ity (Integer value O_O)

ng (Template value O_O)

Low level Data Types

insert Camel Case.md

Type **Emoji**

LURAL Emojis
EDPLURAL Emojis

Type **String**

LURAL Strings
EDPLURAL Strings

Type **CamelName**

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

LURAL CamelNames
EDPLURAL CamelNames
YPEOF String
TYPES UpperCamel , LowerCamel

ng ***(String value O_O)***

RAINTS Must follow the camel case naming convention and not be empty.
xample "firstName", "orderDate", "customerID"

ngNote

- *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.

Type **UpperCamel**
a CamelName that begins with a capital letter

ample _ "Customer", "ProductCategory", "PaymentMethod"
WHERE content begins with an upper case letter.
LURAL UpperCamels
EDPLURAL UpperCamels
YPEOF CamelName

Type **LowerCamel**
a CamelName that begins with a lower case letter

ample "firstName", "orderTotal", "shippingAddress"
WHERE content begins with a lower case letter.
LURAL LowerCamels
EDPLURAL LowerCamels

BTYPEOF CamelName

lue Type **QualifiedCamel**
an expression consisting of Camel Names separated by periods

PLURAL QualifiedCamels
UMEDPLURAL QualifiedCamels
BTYPEOF String
STRAINTS
content consists of CamelNames, separated by periods. Each of the camel names must be Upper Camel except, possibly, the first.

ValueTypeRichText
A string with markup for block level formatting.

PLURAL ValueTypeRichTexts
UMEDPLURAL ValueTypeRichTexts
BTYPEOF String

value the string content
(String value O_O)

ormat the rich text coding language used
(Code value O_O)

```
HTML
MarkDown
```

lue Type **OneLiner**
String with markup for line level formatting.

PLURAL OneLiners
UMEDPLURAL OneLiners
BTYPEOF RichText

value the string content
(String value O_O)

STRAINTS must not containa line break or new line character
MESSAGE A line can't span two lines

lue Type **PrimitiveType**
A basic, built-in data type

PLURAL PrimitiveTypes
UMEDPLURAL PrimitiveTypes
UBTYPES String , Integer , Decimal , Boolean , Date , Time , DateTime

Type String

LURAL Strings
EDPLURAL Strings
YPEOF PrimitiveType
TYPES CamelName , QualifiedCamel , ValueTypeRichText

Type Integer

LURAL Integers
EDPLURAL Integers
YPEOF PrimitiveType

Type Decimal

LURAL Decimals
EDPLURAL Decimals
YPEOF PrimitiveType

Type Boolean

LURAL Booleans
EDPLURAL Booleans
YPEOF PrimitiveType

Type Date

LURAL Dates
EDPLURAL Dates
YPEOF PrimitiveType

Type Time

LURAL Times
EDPLURAL Times
YPEOF PrimitiveType

Type DateTime

LURAL DateTimes
EDPLURAL DateTimes
YPEOF PrimitiveType

Annotation Types Used

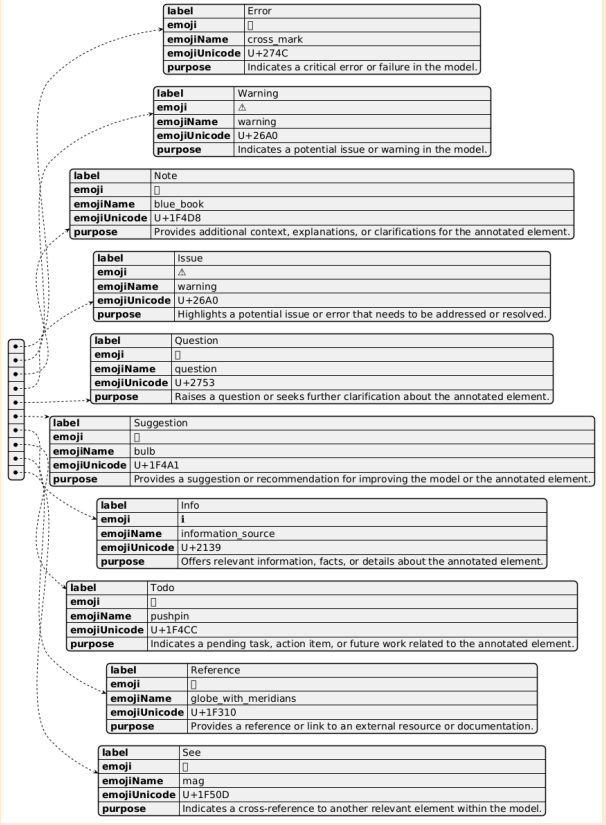
These are the recognized Annotation Types for the LDM model.

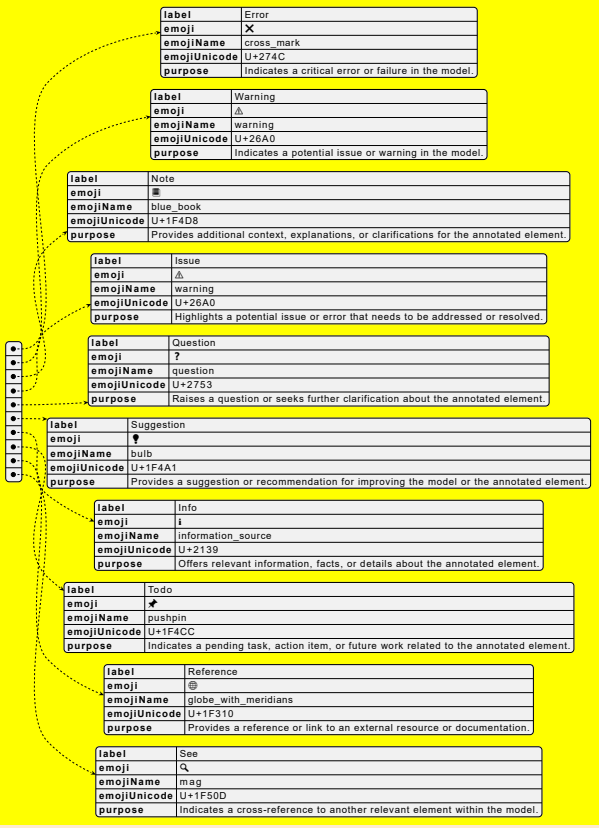
And this is how you register the AnnotationTyped for a model. By including this sort of array in the DSL document for the model.

```
@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": "💡",
```

Annotation Types Used




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 🔍
emojiName mag
emojiUnicode U+1F50D
purpose Indicates a cross-reference to another relevant element within the model.

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

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