



FIRST PAGE LEFT LEFT BLANK

Literate Data Model

BLANK

Preliminaries

the basic structure of the model

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

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

Preliminaries

Component
An element or building block of the literate data model

PLURAL	Components
SINGULAR	Component
DEPENDENTS	Annotation
SUBTYPES	LiterateModel , Subject , Class , Key , AttributeSection , Attribute , Constraint , Method , Parameter

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

warning	This is a warning with emoji
----------------	------------------------------

name	The name of the component (CamelName value O_O)
-------------	--

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

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

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

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

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

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

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

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

ent	Indicates whether this component is an embellishment added during post-parsing processing _	(<u>Boolean</u> value 0_0)
-----	---	------------------------------

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

	mechanical attributes	
--	-----------------------	--

ent	Indicates whether this component is an embellishment added during post-parsing processing _	(<u>Boolean</u> value 0_0)
-----	---	------------------------------

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.

Preliminaries

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

IMPLIES AnnotationTypes

BASED ON [LiterateModel](#)

emoji an emoji
([Emoji](#) value O_O)

Name an emoji
([String](#) value O_O)

unicode the Unicode for the emoji
([String](#) value O_O)

label A short label to indicate the purpose of the annotation _
([LowerCamel](#) value O_O)

plural the plural form of the label
([UpperCamel](#) value O_O)

DEFAULT based on label

purpose the intended reason for the annotation.
([OneLiner](#) value O_O)

created by created for AnnotationType

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

AnnotationType inverse attribute for Annotation.annotationType from which this was implied.
([Annotation](#) value M_1)

INVERSE [Annotation.annotationType](#)

AnnotationTypes inverse attribute for LiterateModel.annotationTypes from which this was implied.
([LiterateModel](#) value M_1)

INVERSE [LiterateModel.annotationTypes](#)

Annotation	
A note or comment associated with a model element	
PLURAL	Annotations
EDPLURAL	Annotations
SEDON	Component
pe	(<i>Optional</i> AnnotationType value O_O)
note	An Annotation is considered to <i>recognized</i> if the label is associated with an Annotation Type. otherwise it is <i>ad hoc</i> .
note	Should be a Value Type
VERSE	AnnotationType.inverseOfAnnotationType
bel	A short label to indicate the purpose of the annotation _ (CamelName value O_O)
But any short label is valid.	
DEFAULT	from annotationType
oji	(<i>Optional</i> Emoji value O_O)
DEFAULT	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)
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.
ent	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.

Preliminaries

utes	created for Annotation
onent	A link back to the Component on which this Annotation depends. (<u>Component</u> value <i>M_1</i>)

BLANK

The Model and its Subjects

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

Types	(List of AnnotationTypes value 0_0)
INVERSE	AnnotationType.inverseOfAnnotationTypes
CodingLanguage	the recommended language for expressing derivation, defaults, and constraints (CodingLanguage value 0_0)
DEFAULT	Python
CodingLanguages	(Optional List of CodingLanguages value 0_0)
TemplateLanguage	the recommended language for expressing derivation, defaults, and constraints (TemplateLanguage value 0_0)
DEFAULT	Handlebars
TemplateLanguages	(Optional List of TemplateLanguages value 0_0)
Functions	A list of functions that require sophisticated AI-powered implementation * (List of String value 0_0)
DERIVATION	[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.

LURAL Subjects
SEDON [LiterateModel](#)
YPOF [Component](#)
YPES [SubjectArea](#)

me ([UpperCamel](#) value O_O)
RIDES [Component.name](#)

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

VERSE [Subject.inverseOfParentSubject](#)

es 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?
VERSE [Class.inverseOfClasses](#)

cts 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.

VERSE [Subject.parentSubject](#)

s created for Subject
del A link back to the LiterateModel on which this Subject depends.
([LiterateModel](#) value M_1)

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

The Model and its Subjects

SubjectArea

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

WHERE parentSubject is absent
PLURAL SubjectAreas
BASEDON [LiterateModel](#)
BTYPEOF [Subject](#)

utes created for SubjectArea

Model A link back to the LiterateModel on which this SubjectArea depends.

([LiterateModel](#) value *M_1*)

Classes

Classes

Class

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

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

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

([UpperCamel](#) value O_O)

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

- But also might be People for Person.

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

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

basedOn the Class or Classes on which this class is dependent

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

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

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

INVERSE [Class.inverseOfBasedOn](#)

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

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

INVERSE [Class.inverseOfSupertypes](#)

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

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

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

INVERSE [Subtyping.inverseOfSubtypings](#)

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

Classes

Classes	Inverse attribute for LiterateModel.allClasses from which this was implied. (LiterateModel value M_1)
INVERSE	LiterateModel.allClasses
Classes	Inverse attribute for Subject.classes from which this was implied. (Subject value M_1)
INVERSE	Subject.classes
basedOn	Inverse attribute for Class.basedOn from which this was implied. (Class value M_1)
INVERSE	Class.basedOn
Supertypes	Inverse attribute for Class.supertypes from which this was implied. (Class value M_1)
INVERSE	Class.supertypes
Subtypes	Inverse attribute for Class.subtypes from which this was implied. (Class value M_1)
INVERSE	Class.subtypes
Classes	Inverse attribute for Subtyping.classes from which this was implied. (Subtyping value M_1)
INVERSE	Subtyping.classes
inverseClass	Inverse attribute for Attribute.inverseClass from which this was implied. (Attribute value M_1)
INVERSE	Attribute.inverseClass
coreClass	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

SINGULAR Subtypings
EDPLURALSubtypings
BASED ON [Class](#)

NAME ([LowerCamel](#) value O_O)

IS ([Boolean](#) value O_O)

DEFAULT true

IS ([Boolean](#) value O_O)

DEFAULT true

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

DSL : Shown in the DSL as

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

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

VERSE

CLASS created for Subtyping

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

VERSE [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

SINGULAR ReferenceTypes
EDPLURALReferenceTypes
TYPE OF [Class](#)

CodeType

A data type or enumeration used in the model

PLURAL CodeTypes

IMPLIES CodeTypes

DEPENDENTS [CodeValue](#)

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

([Boolean](#) value O_O)

CodeValue

A possible value for an enumerated data class

PLURAL CodeValues

IMPLIES CodeValues

BASED ON [CodeType](#)

code A short code or abbreviation for the value _

([String](#) value O_O)

Description an explanation of what the code means

([RichText](#) value O_O)

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

Attributes created for CodeValue

CodeType 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
ADPLURAL	Keys
DEPENDSON	Class
TYPEOF	Component
TYPES	UniqueKey
Notes	the attributes of the base Class. (<i>List of Attributes value O_O</i>)
INVERSE	Attribute.inverseOfKeyAttributes
CONSTRAINTS	each attribute must be a direct or inherited of the base class.
CONSTRAINTS	no repetitions allowed in keyAttributes 👉 Issue : introduce PureLists?
Issues	need ascending descending to support index keys or ordering keys.
STATUS	created for Key
CLASS	A link back to the Class on which this Key depends. (<i>Class value M_1</i>)
	UniqueKey a list of attributes on which instances of the base class may be keyed.
note	order unimportant for Unique Keys.
PLURAL	UniqueKeys
ADPLURAL	UniqueKeys
TYPEOF	Key

Attributes

AttributeSection

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

LURAL AttributeSections

ADPLURAL 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 or required, depending on how it is marked.

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

s created for AttributeSection

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

([Class](#) value M_1)

VERSE [Class.attributeSections](#)

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

([Class](#) value M_1)

Attributes

Attribute

A property or characteristic of a class

PLURAL Attributes
BASED ON [AttributeSection](#)
DEPENDENTS [AttributeConstraint](#)
BTYPED OF [Component](#)

name ([LowerCamel](#) value O_O)

OVERRIDES [Component.name](#)

DataType 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.

Optional Indicates whether the attribute must have a value for every instance of the class _
([Boolean](#) value O_O)

DEFAULT *** False

Cardinality The cardinality of the relationship represented by the attribute _
([Cardinality](#) 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.

note how this works with optionality

Invertible ([Boolean](#) value O_O)

DERIVATION 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)

INVERSE [Class.inverseOfInverseClass](#)

DERIVATION from the data type. Null unless attribute is invertible.

Attribute ([Optional](#) [Attribute](#) value O_O)

VERSE [Attribute.inverseOfInverseAttribute](#)

nal ([Optional \[Attribute\]\(#\) value O_O](#))

VERSE [Attribute.inverseOfInverselsOptional](#)

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

es the higher level attribute which this one overrides - for type or ...
([Attribute value O_O](#))

VERSE [Attribute.inverseOfOverrides](#)

nal Indicates whether the attribute must have a value for every instance of the class _
([Boolean value O_O](#))

DEFAULT *** False

ity The cardinality of the relationship represented by the attribute _
([Cardinality 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.

note how this works with optionality

s ble ([Boolean value O_O](#))

VATION true if the data type is a class or a simple collection of members of a class.

ss the class which contains, or would contain the inverse attribute
([Optional \[Class\]\(#\) value O_O](#))

Attributes

INVERSE	Class.inverseOfInverseClass	
DERIVATION	from the data type. Null unless attribute is invertible.	
Attribute		(Optional Attribute value 0_0)
INVERSE	Attribute.inverseOfInverseAttribute	
Optional		(Optional Attribute value 0_0)
INVERSE	Attribute.inverseOfInverselsOptional	
Default	The rule or formula for calculating the value, if no value is supplied Now running to a second line with the parenthetical on yet a third line	(Optional Derivation value 0_0)
note	even when an Attribute has a default derivation, there's no guarantee that every instance will have an assigned value. Example needed.	
Derivation	For derived attributes, the rule or formula for calculating the value _	(Optional Derivation value 0_0)
issue	on insert vs on access?	
Constraints	Any validation rules specific to this attribute _	(List of Constraints value 0_0)
note	from Class.constraints	
Overrides	the higher level attribute which this one overrides - for type or ...	(Attribute value 0_0)
INVERSE	Attribute.inverseOfOverrides	
Attributes	created for Attribute	
Attributes	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)
inverseAttribute	inverse attribute for Attribute.inverseAttribute from which this was implied.	(Attribute value M_1)

VERSE	Attribute.inverseAttribute	
IsOptional	Inverse attribute for Attribute.inverselsOptional from which this was implied.	(Attribute value M_1)
VERSE	Attribute.inverselsOptional	
Is	Inverse attribute for Attribute.overrides from which this was implied.	(Attribute value M_1)
VERSE	Attribute.overrides	
	Derivation	
	A rule or formula for deriving the value of an attribute	
PLURAL	Derivations	
Definition	An English language statement of the derivation rule _	(RichText value O_0)
Formal	The formal expression of the derivation in a programming language _	(CodeExpression value O_0)
	Constraint	
	A rule, condition, or validation that must be satisfied by the model	
PLURAL	Constraints	
TYPEOF	Component	
TYPES	ClassConstraint , AttributeConstraint	
Definition	An English language statement of the constraint _	(RichText value O_0)
Formal	The formal expression of the constraint in a programming language, for example: OCL or Python. _	(CodeExpression value O_0)
Formality		(Code value O_0)
	<div>Warning, nothing fatal; just a caution</div> <div>Error, serious. Fix now</div>	
	ClassConstraint	
PLURAL	ClassConstraints	
ADPLURAL	ClassConstraints	
BASED ON	Class	
TYPEOF	Constraint	

Attributes

Attributes	created for ClassConstraint
Class	A link back to the Class on which this ClassConstraint depends. (<u>Class</u> value M_1)
	AttributeConstraint

PLURAL AttributeConstraints
IMEDPLURAL AttributeConstraints
BASEDON [Attribute](#)
BTYPEOF [Constraint](#)

Attributes	created for AttributeConstraint
Attribute	A link back to the Attribute on which this AttributeConstraint depends. (<u>Attribute</u> value M_1)

BLANK

Methods

	Method	
	A behavior or operation associated with a class	
LURAL	Methods	
TYPEOF	Component	
ers	The input parameters of the method _	(<i>List of Parameters value O_O</i>)
VERSE	Parameter.inverseOfParameters	
pe	The data type of the value returned by the method _	(<i>DataType value O_O</i>)
s	created for Method	
ds	Inverse attribute for Class.methods from which this was implied.	(<i>Class value M_1</i>)
VERSE	Class.methods	
	Parameter	
	An input to a method	
LURAL	Parameters	
TYPEOF	Component	
pe	The data type of the parameter	(<i>DataType value O_O</i>)
ity	The cardinality of the parameter. e.g., optional, required.	(<i>Cardinality value O_O</i>)
s	created for Parameter	
ters	Inverse attribute for Method.parameters from which this was implied.	(<i>Method value M_1</i>)
VERSE	Method.parameters	

Trivial Data Types

Message

LURAL Messages
DPLURAL Messages

CodeExpression

LURAL CodeExpressions
DPLURAL CodeExpressions

ge

the programming language

([Code](#) value O_O)

OCIL, Object Constraint Language
Java, Java
Python, Python

on

([String](#) value O_O)

DataType

LURAL DataTypes
DPLURAL DataTypes

SimpleDataTypeSubtpeOfDataType

LURAL SimpleDataTypeSubtpeOfDataTypes
DPLURAL SimpleDataTypeSubtpeOfDataTypes

ss

([Class](#) value O_O)

VERSE [Class.inverseOfCoreClass](#)

ComplexDataType

LURAL ComplexDataTypes
DPLURAL ComplexDataTypes

on

([AggregatingOperator](#) value O_O)

es

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

AggregatingOperator

LURAL AggregatingOperators
DPLURAL AggregatingOperators

me

([Code](#) value O_O)

Trivial Data Types

SetOf
ListOf
Mapping

arity

(Integer value O_O)

elling

(Template value O_O)

BLANK

Trivial Low level Data Types

insert Camel Case.md

Emoji

LURAL Emojis
DPLURAL Emojis

String

LURAL Strings
DPLURAL Strings

CamelName

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

LURAL CamelNames
DPLURAL CamelNames
TYPEOF [String](#)
TYPES [UpperCamel](#), [LowerCamel](#)

ng ([String](#) value 0_0)

RAINTS Must follow the camel case naming convention and not be empty.
ample "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.

UpperCamel

a CamelName that begins with a capital letter

ample _ "Customer", "ProductCategory", "PaymentMethod"

WHERE content begins with an upper case letter.
LURAL UpperCamels
DPLURAL UpperCamels
TYPEOF [CamelName](#)

LowerCamel

a CamelName that begins with a lower case letter

ample "firstName", "orderTotal", "shippingAddress"

WHERE content begins with a lower case letter.
LURAL LowerCamels
DPLURAL LowerCamels

BTYPOF [CamelName](#)

QualifiedCamel

an expression consisting of Camel Names separated by periods

PLURAL QualifiedCamels

IMEDPLURAL QualifiedCamels

BTYPOF [String](#)

STRAINTS

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

RichText

A string with markup for block level formatting.

PLURAL RichTexts

IMEDPLURAL RichTexts

BTYPOF [String](#)

SUBTYPES [OneLiner](#)

value the string content

([String](#) value O_O)

format the rich text coding language used

([Code](#) value O_O)

HTML
MarkDown

OneLiner

String with markup for line level formatting.

PLURAL OneLiners

IMEDPLURAL OneLiners

BTYPOF [RichText](#)

value the string content

([String](#) value O_O)

VERRIDES [RichText.value](#)

STRAINTS must not contain a line break or new line character

MESSAGE A line can't span two lines

PrimitiveType

A basic, built-in data type

PLURAL PrimitiveTypes

PrimitiveTypes

String, Integer, Decimal, Boolean, Date, Time, DateTime

String

Strings

Strings

PrimitiveType

CamelName, QualifiedCamel, RichText

Integer

Integers

Integers

PrimitiveType

Decimal

Decimals

Decimals

PrimitiveType

Boolean

Booleans

Booleans

PrimitiveType

Date

Dates

Dates

PrimitiveType

Time

Times

Times

PrimitiveType

DateTime

DateTimes

DateTimes

PrimitiveType

CodingLanguage

CodingLanguages

CodingLanguages

Trivial Low level Data Types

Cardinality

PLURAL Cardinalitys
IMEDPLURALCardinalitys

TemplateLanguage

PLURAL TemplateLanguages
IMEDPLURALTemplateLanguages

Template

PLURAL Templates
IMEDPLURALTemplates

Code

PLURAL Codes
IMEDPLURALCodes

Annotation Types Used

These are the recognized Annotation Types for the LDM model.

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

PlantUML Diagram - Inert

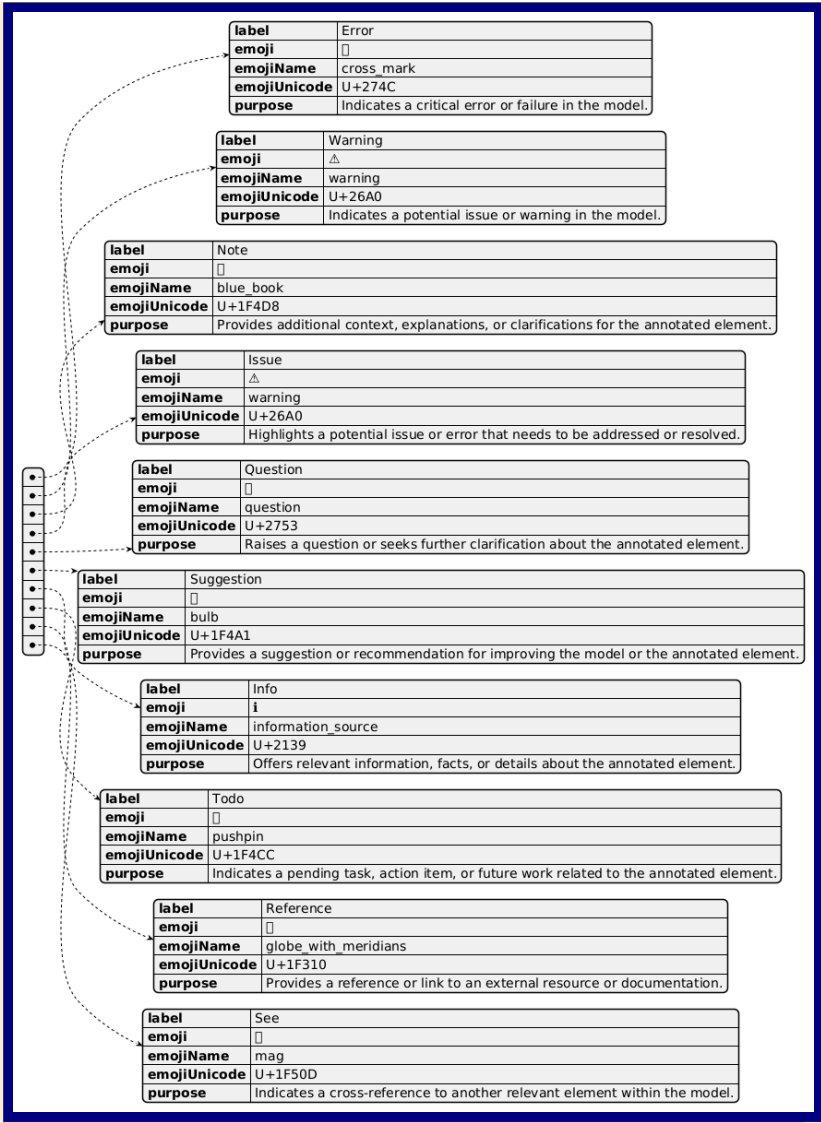
@starttjson

```
[
{
  "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."  
}  
]  
@endjson
```

PlantUML Diagram - PNG for puml



Annotation types as CSV

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

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

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

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

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

BLANK

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