

FIRST PAGE LEFT LEFT BLANK

## Literate Data Model

## 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

Components

Components

[Annotation](#)

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

the name of the component, not in camel case

( [String](#) value O\_O )

This is a warning with emoji

The name of the component

( [CamelName](#) value O\_O )

( [QualifiedCamel](#) value O\_O )

a short form of the component's name, used for cross references and improved readability.

( [CamelName](#) value O\_O )

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

name - how do you say name in english?

x.name == y

the abbreviated name should be shorter than the actual name

len(abbreviatedName) < len(name)

Why have an abbreviation longer than the name?

Warning

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

A brief, one-line definition or description of the component, suitable for use in a descriptive table of contents. \_

( [OneLiner](#) value O\_O )

A more detailed explanation or discussion of the component \_

( [RichText](#) value O\_O )

Indicates whether this component is an embellishment added during post-parsing processing \_

( [Boolean](#) value O\_O )

false

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.

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

false

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

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

false

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.

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

## Annotation

A note or comment associated with a model element

Annotations

Annotations

[Component](#)

( *Optional* [AnnotationType](#) value O\_O )

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

Should be a Value Type

[AnnotationType.inverseOfAnnotationType](#)

A short label to indicate the purpose of the annotation \_

( [CamelName](#) value O\_O )

But any short label is valid.

from annotationType

( *Optional* [Emoji](#) value O\_O )

from annotation type

The content or body of the annotation

( [RichText](#) value O\_O )

Indicates whether this annotation is an embellishment added during post-parsing processing \_

( [Boolean](#) value O\_O )

false

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.

Indicates whether this annotation is an embellishment added during post-parsing processing \_

( [Boolean](#) value O\_O )

false

This attribute is set to true for annotations that are automatically generated or added during the fleshing out, review, or rendering processes, such as suggestions, issues, or diagnostic messages. It helps distinguish embellishment annotations from the annotations defined in the original LDM source.



created for Annotation
A link back to the Component on which this Annotation depends. ( <a href="#">Component</a> value M_1 )

## The Model and its Subjects

## LiterateModel

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

LiterateModels

[AnnotationType](#), [Subject](#), [SubjectArea](#)  
[Component](#)

( [UpperCamel](#) value O\_O )

[Component.name](#)

list of all classes in the model, as ordered in the definition of the model.

( [List of Classes](#) value O\_O )

[Class.inverseOfAllSubjects](#)

gathering s.allSubjects over s in subjectAreas

Subject names must be unique across the model.

list of all classes in the model, as ordered in the definition of the model.

( [List of Classes](#) value O\_O )

[Class.inverseOfAllClasses](#)

gathering s.allClasses over s in allSubjects.

Class names must be unique across the model.

( [List of AnnotationTypes](#) value O\_O )

[AnnotationType.inverseOfAnnotationTypes](#)

the recommended language for expressing derivation, defaults, and constraints

( [CodingLanguage](#) value O\_O )

Python

[Languages](#)

( [Optional List of CodingLanguages](#) value O\_O )

the recommended language for expressing derivation, defaults, and constraints

( [TemplateLanguage](#) value O\_O )

Handlebars

[Languages](#)

( [Optional List of TemplateLanguages](#) value O\_O )

A list of functions that require sophisticated AI-powered implementation \*

( [List of String](#) value O\_O )

['aiEnglishPlural()']

( <i>List of <a href="#">AnnotationTypes</a> value O_O</i> )
<a href="#">AnnotationType.inverseOfAnnotationTypes</a>
the recommended language for expressing derivation, defaults, and constraints
( <i><a href="#">CodingLanguage</a> value O_O</i> )
Python
<i>ges</i> ( <i>Optional List of <a href="#">CodingLanguages</a> value O_O</i> )
the recommended language for expressing derivation, defaults, and constraints
( <i><a href="#">TemplateLanguage</a> value O_O</i> )
Handlebars
<i>uages</i> ( <i>Optional List of <a href="#">TemplateLanguages</a> value O_O</i> )
A list of functions that require sophisticated AI-powered implementation *
( <i>List of <a href="#">String</a> value O_O</i> )
['aiEnglishPlural()']

<b>Subject</b> 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.

Subjects  
[LiterateModel](#)  
[Component](#)  
[SubjectArea](#)

( <a href="#">UpperCamel</a> value O_O )
<a href="#">Component.name</a>

The parent subject, if any, under which this subject is nested _ ( <a href="#">Optional</a> <a href="#">Subject</a> value O_O )
<a href="#">Subject.inverseOfParentSubject</a>

The major classes related to this subject, in the order in which they should be presented _ ( <a href="#">List of</a> <a href="#">Classes</a> value O_O )
define chapter, section, subsection as levels? <a href="#">Class.inverseOfClasses</a>

Any child subjects nested under this subject, in the order in which they should be presented _ ( <a href="#">List of</a> <a href="#">Subjects</a> value O_O )
--

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

[Subject.parentSubject](#)

created for Subject
A link back to the LiterateModel on which this Subject depends. ( <a href="#">LiterateModel</a> value M_1 )
Inverse attribute for Subject.parentSubject from which this was implied. ( <a href="#">Subject</a> value M_1 )
<a href="#">Subject.parentSubject</a>

### SubjectArea

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

parentSubject is absent

SubjectAreas

[LiterateModel](#)

[Subject](#)

created for SubjectArea

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

( [LiterateModel](#) value *M\_1* )

<b>Classes</b>
----------------

## Class

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

Classes

[Subtyping](#), [Key](#), [AttributeSection](#), [ClassConstraint](#)

[Component](#)

[ReferenceType](#)

Within each Class, attribute names must be unique.

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.

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

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

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.

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

[Class.inverseOfBasedOn](#)

The parent class or classes from which this class inherits attributes

( [List of Classes](#) value O\_O )

[Class.inverseOfSupertypes](#)

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

( [List of Subtypings](#) value O\_O )

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).

[Subtyping.inverseOfSubtypings](#)

Any subtypes or specializations of this class based on its subtypings.



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

Inverse attribute for LiterateModel.allClasses from which this was implied.	( <a href="#">LiterateModel</a> value M_1 )
<a href="#">LiterateModel.allClasses</a>	
Inverse attribute for Subject.classes from which this was implied.	( <a href="#">Subject</a> value M_1 )
<a href="#">Subject.classes</a>	
Inverse attribute for Class.basedOn from which this was implied.	( <a href="#">Class</a> value M_1 )
<a href="#">Class.basedOn</a>	
Inverse attribute for Class.supertypes from which this was implied.	( <a href="#">Class</a> value M_1 )
<a href="#">Class.supertypes</a>	
Inverse attribute for Class.subtypes from which this was implied.	( <a href="#">Class</a> value M_1 )
<a href="#">Class.subtypes</a>	
Inverse attribute for Subtyping.classes from which this was implied.	( <a href="#">Subtyping</a> value M_1 )
<a href="#">Subtyping.classes</a>	
Inverse attribute for Attribute.inverseClass from which this was implied.	( <a href="#">Attribute</a> value M_1 )
<a href="#">Attribute.inverseClass</a>	
Inverse attribute for SimpleDataTypeSubtpeOfDataType.coreClass from which this was implied.	( <a href="#">SimpleDataTypeSubtpeOfDataType</a> value M_1 )
<a href="#">SimpleDataTypeSubtpeOfDataType.coreClass</a>	

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

Subtypings  
RAISubtypings  
[Class](#)

( [LowerCamel](#) value O\_O )

( [Boolean](#) value O\_O )

true

( [Boolean](#) value O\_O )

true

( [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.

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

created for Subtyping

Inverse attribute for Class.subtypings from which this was implied.  
( [Class](#) value M\_1 )

[Class.subtypings](#)

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

ReferenceTypes  
RAISubtypings  
[Class](#)

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

CodeTypes  
RAI CodeTypes  
[CodeValue](#)

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

CodeValues  
RAI CodeValues  
[CodeType](#)

A short code or abbreviation for the value \_  
( [String](#) value O\_O )

an explanation of what the code means  
( [RichText](#) value O\_O )

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:

created for CodeValue

A link back to the CodeType on which this CodeValue depends.  
( [CodeType](#) value M\_1 )

**Key**  
a list of attributes of a class

Keys  
Keys  
[Class](#)  
[Component](#)  
[UniqueKey](#)

the attributes of the base Class.  
( *List of [Attributes](#) value O\_O* )

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

👍 **Issue** : introduce PureLists?

need ascending descending to support index keys or ordering keys.

created for Key

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.

order unimportant for Unique Keys.  
UniqueKeys  
UniqueKeys  
[Key](#)

Attributes
------------

### AttributeSection

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

AttributeSections

AttributeSections

[Class](#)

[Attribute](#)

[Component](#)

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

( [Boolean](#) value *O\_O* )

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

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

created for AttributeSection

inverse attribute for Class.attributeSections from which this was implied.

( [Class](#) value *M\_1* )

[Class.attributeSections](#)

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

( [Class](#) value *M\_1* )

## Attribute

A property or characteristic of a class

Attributes

[AttributeSection](#)

[AttributeConstraint](#)

[Component](#)

( [LowerCamel](#) value O\_O )

[Component.name](#)

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

( [DataType](#) value O\_O )

But,

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

[the section below on Data Type Specifiers.](#)

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

( [Boolean](#) value O\_O )

\*\*\* False

The cardinality of the relationship represented by the attribute \_

( [Cardinality](#) value O\_O )

\*\*\* 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.

[how this works with optionality](#)

( [Boolean](#) value O\_O )

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

the class which contains, or would contain the inverse attribute

( [Optional](#) [Class](#) value O\_O )

[Class.inverseOfInverseClass](#)



from the data type. Null unless attribute is invertible.

( <i>Optional</i> <u>Attribute</u> value O_O )
<u>Attribute.inverseOfInverseAttribute</u>

( <i>Optional</i> <u>Attribute</u> value O_O )
<u>Attribute.inverseOfInverselsOptional</u>

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 ( <i>Optional</i> <u>Derivation</u> value O_O )
--

even when an Attribute has a default derivation, there's no guarantee that  
every instance will have an assigned value. Example needed.

For derived attributes, the rule or formula for calculating the value _ ( <i>Optional</i> <u>Derivation</u> value O_O )
on insert vs on access?

Any validation rules specific to this attribute _ ( <i>List of</i> <u>Constraints</u> value O_O )
from Class.constraints

the higher level attribute which this one overrides - for type or ... ( <u>Attribute</u> value O_O )
<u>Attribute.inverseOfOverrides</u>

Indicates whether the attribute must have a value for every instance of the class _ ( <u>Boolean</u> value O_O )
*** False

The cardinality of the relationship represented by the attribute _ ( <u>Cardinality</u> value O_O )
*** 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. how this works with optionality

( <u>Boolean</u> value O_O )

true if the data type is a class or a simple collection of members of a class.
the class which contains, or would contain the inverse attribute ( <i>Optional</i> <u>Class</u> value <i>O_O</i> )
<u>Class.inverseOfInverseClass</u> from the data type. Null unless attribute is invertible.
( <i>Optional</i> <u>Attribute</u> value <i>O_O</i> )
<u>Attribute.inverseOfInverseAttribute</u>
( <i>Optional</i> <u>Attribute</u> value <i>O_O</i> )
<u>Attribute.inverseOfInverselsOptional</u>
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 ( <i>Optional</i> <u>Derivation</u> value <i>O_O</i> )
even when an Attribute has a default derivation, there's no guarantee that every instance will have an assigned value. Example needed.
For derived attributes, the rule or formula for calculating the value _ ( <i>Optional</i> <u>Derivation</u> value <i>O_O</i> )
on insert vs on access?
Any validation rules specific to this attribute _ ( <i>List of</i> <u>Constraints</u> value <i>O_O</i> )
from Class.constraints
the higher level attribute which this one overrides - for type or ... ( <u>Attribute</u> value <i>O_O</i> )
<u>Attribute.inverseOfOverrides</u>
created for Attribute
Inverse attribute for Class.attributes from which this was implied. ( <u>Class</u> value <i>M_1</i> )
<u>Class.attributes</u>
Inverse attribute for Key.keyAttributes from which this was implied. ( <u>Key</u> value <i>M_1</i> )
<u>Key.keyAttributes</u>

A link back to the AttributeSection on which this Attribute depends.	( <a href="#">AttributeSection</a> value M_1 )
Inverse attribute for Attribute.inverseAttribute from which this was implied.	( <a href="#">Attribute</a> value M_1 )
<a href="#">Attribute.inverseAttribute</a>	
Inverse attribute for Attribute.inverselsOptional from which this was implied.	( <a href="#">Attribute</a> value M_1 )
<a href="#">Attribute.inverselsOptional</a>	
Inverse attribute for Attribute.overrides from which this was implied.	( <a href="#">Attribute</a> value M_1 )
<a href="#">Attribute.overrides</a>	
<b>Derivation</b>	
A rule or formula for deriving the value of an attribute	
Derivations	
An English language statement of the derivation rule _	( <a href="#">RichText</a> value O_0 )
The formal expression of the derivation in a programming language _	( <a href="#">CodeExpression</a> value O_0 )
<b>Constraint</b>	
A rule, condition, or validation that must be satisfied by the model	
Constraints	
<a href="#">Component</a>	
<a href="#">ClassConstraint</a> , <a href="#">AttributeConstraint</a>	
An English language statement of the constraint _	( <a href="#">RichText</a> value O_0 )
The formal expression of the constraint in a programming language, for example: OCL or Python. _	( <a href="#">CodeExpression</a> value O_0 )
	( <a href="#">Code</a> value O_0 )

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

<b>ClassConstraint</b>
ClassConstraints

RAIClassConstraints

[Class](#)

[Constraint](#)

created for ClassConstraint
-----------------------------

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

( [Class](#) value M\_1 )

AttributeConstraint
---------------------

AttributeConstraints

RAIAttributeConstraints

[Attribute](#)

[Constraint](#)

created for AttributeConstraint
---------------------------------

A link back to the Attribute on which this AttributeConstraint depends.
---

( [Attribute](#) value M\_1 )

## Methods

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

Trivial Data Types
--------------------

Message
Messages
RAIMessages
CodeExpression
CodeExpressions
RAICodeExpressions
the programming language
( <a href="#">Code</a> value O_O )
<div> <div></div> <div> OCL, Object Constraint Language  Java, Java  Python, Python </div> </div>
( <a href="#">String</a> value O_O )
DataType
DataTypes
RAIDataTypes
SimpleDataTypeSubtpeOfDataType
SimpleDataTypeSubtpeOfDataTypes
RAISimpleDataTypeSubtpeOfDataTypes
( <a href="#">Class</a> value O_O )
<a href="#">Class.inverseOfCoreClass</a>
ComplexDataType
ComplexDataTypes
RAIComplexDataTypes
( <a href="#">AggregatingOperator</a> value O_O )
( List of <a href="#">DataTypes</a> value O_O )
AggregatingOperator
AggregatingOperators
RAIAggregatingOperators
( <a href="#">Code</a> value O_O )



SetOf  
ListOf  
Mapping

	( <i>Integer</i> value O_O )
	( <i>Template</i> value O_O )

## Trivial Low level Data Types

insert Camel Case.md

Emoji

Emojis  
rALEmojis

String

Strings  
rALStrings

CamelName

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

CamelNames  
rALCamelNames  
[String](#)  
[UpperCamel](#), [LowerCamel](#)

( [String\\_value O\\_O](#) )

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

- *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

\_ "Customer", "ProductCategory", "PaymentMethod"  
content begins with an upper case letter.

UpperCamels  
rALUpperCamels  
[CamelName](#)

LowerCamel

a CamelName that begins with a lower case letter

"firstName", "orderTotal", "shippingAddress"  
content begins with a lower case letter.

LowerCamels  
rALLowerCamels  
[CamelName](#)

### QualifiedCamel

an expression consisting of Camel Names separated by periods

QualifiedCamels

RAIQualifiedCamels

[String](#)

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.

RichTexts

RAIRichTexts

[String](#)

[OneLiner](#)

the string content

( [String\\_value](#) O\_O )

the rich text coding language used

( [Code\\_value](#) O\_O )

HTML  
MarkDown

### OneLiner

String with markup for line level formatting.

OneLiners

RAIOneLiners

[RichText](#)

the string content

( [String\\_value](#) O\_O )

[RichText.value](#)

must not contain a line break or new line character

A line can't span two lines

### PrimitiveType

A basic, built-in data type

PrimitiveTypes

RAIPrimitiveTypes

[String](#), [Integer](#), [Decimal](#), [Boolean](#), [Date](#), [Time](#), [DateTime](#)

String
Strings
RAIStrings
<a href="#">PrimitiveType</a>
<a href="#">CamelName</a> , <a href="#">QualifiedCamel</a> , <a href="#">RichText</a>
Integer
Integers
RAIIntegers
<a href="#">PrimitiveType</a>
Decimal
Decimals
RAIDecimals
<a href="#">PrimitiveType</a>
Boolean
Booleans
RAIBooleans
<a href="#">PrimitiveType</a>
Date
Dates
RAIDates
<a href="#">PrimitiveType</a>
Time
Times
RAITimes
<a href="#">PrimitiveType</a>
DateTime
DateTimes
RAIDateTimes
<a href="#">PrimitiveType</a>
CodingLanguage
CodingLanguages
RAICodingLanguages
Cardinality
Cardinalitys

RAICardinalitys

TemplateLanguage

TemplateLanguages

RAITemplateLanguages

Template

Templates

RAITemplates

Code

Codes

RAICodes

## Annotation Types Used

These are the recognized Annotation Types for the LDM model.

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

*PlantUML Diagram - Inert*

**@startjson**

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

```


"purpose": "Highlights a potential issue or error that needs to be addressed or resolved."

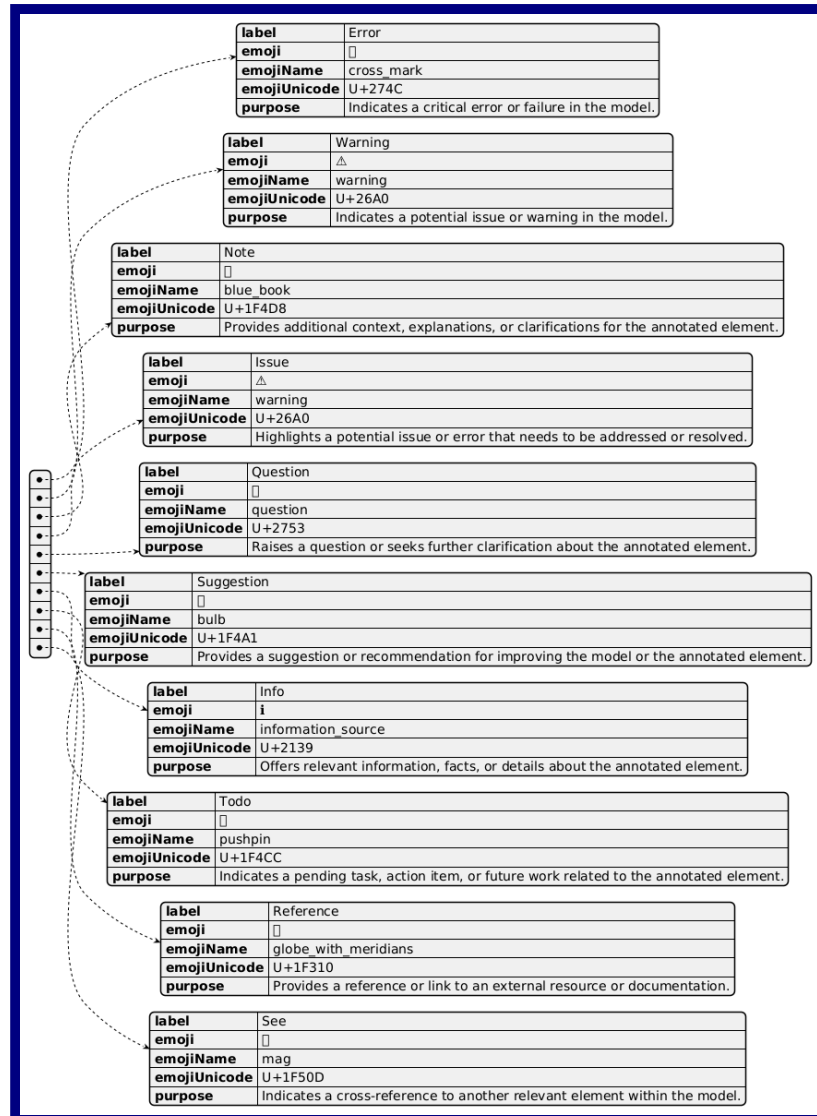

},
{
  "label": "Question",
  "emoji": "❓",
  "emojiName": "question",
  "emojiUnicode": "U+2753",
  "purpose": "Raises a question or seeks further clarification about the annotated element."
},
{
  "label": "Suggestion",
  "emoji": "💡",
  "emojiName": "bulb",
  "emojiUnicode": "U+1F4A1",
  "purpose": "Provides a suggestion or recommendation for improving the model or the annotated element."
},
{
  "label": "Info",
  "emoji": "ℹ️",
  "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

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.
3	Issue	⚠	warning	U+26A0	Highlights a potential issue or error that needs to be addressed or resolved.
4	Question	❓	question	U+2753	Raises a question or seeks further clarification about the annotated element.
5	Suggestion	💡	bulb	U+1F4A1	Provides a suggestion or recommendation for improving the model or the annotated element.
6	Info	📘	information_source	U+2139	Offers relevant information, facts, or details about the annotated element.
7	Todo	📌	pushpin	U+1F4CC	Indicates a pending task, action item, or future work related to the annotated element.
8	Reference	🌐	globe_with_meridians	U+1F310	Provides a reference or link to an external resource or documentation.
9	See	🔍	mag	U+1F50D	Indicates a cross-reference to another relevant element within the model.

## 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