

FIRST PAGE LEFT LEFT BLANK

Literate Data Model

Component
An element or building block of the literate data model
<p>Components</p> <p>Components</p> <p>Annotation</p> <p>LiterateDataModel, Subject, Class, Key, AttributeSection, Attribute, Constraint, Method, ParameterAnInputToAMethod</p>
<p>the name of the component, not in camel case</p> <p>(String value O_O)</p> <p>This is a warning with emoji</p>
<p>The name of the component</p> <p>(CamelName value O_O)</p>
<p>(QualifiedCamel value O_O)</p>
<p>a short form of the component's name, used for cross references and improved readability.</p> <p>(CamelName value O_O)</p> <p>"LDM" is the short form of "Literate Data Model".</p> <p>name - how do you say name in english?</p> <p>x.name == y</p> <p>the abbreviated name should be shorter than the actual name</p> <p>len(abbreviatedName) < len(name)</p> <p>Why have an abbreviation longer than the name?</p> <p>Warning</p> <p>Does this annotation find it's way to the Constraint? YES! It's fixed!</p>
<p>A brief, one-line definition or description of the component, suitable for use in a descriptive table of contents. _</p> <p>(OneLiner value O_O)</p>
<p>A more detailed explanation or discussion of the component _</p> <p>(RichText value O_O)</p>
mechanical attributes
<p>Indicates whether this component is an embellishment added during post-parsing processing _</p> <p>(Boolean value O_O)</p> <p>false</p>

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.

Each LDM declares a set of Annotation Types, with defined labels, emojis, and clearly documented purposes. These are *recognized* or *registered* Annotation Types.

AnnotationTypes

AnnotationTypes

[LiterateDataModel](#)

an emoji

([Emoji](#) value O_O)

an emoji

([String](#) value O_O)

the Unicode for the emoji

([String](#) value O_O)

A short label to indicate the purpose of the annotation _

([LowerCamel](#) value O_O)

the plural form of the label

([UpperCamel](#) value O_O)

based on label

the intended reason for the annotation.

([OneLiner](#) value O_O)

A link back to the LiterateDataModel on which this AnnotationType depends.

([LiterateDataModel](#) value M_1)

reverse attribute for Annotation.annotationType from which this was implied.

([Annotation](#) value M_1)

[Annotation.annotationType](#)

<p>A link back to the LiterateDataModel on which this AnnotationType depends.</p> <p>(LiterateDataModel value M_1)</p>
<p>inverse attribute for Annotation.annotationType from which this was implied.</p> <p>(Annotation value M_1)</p> <p>Annotation.annotationType</p>

Annotation	
A note or comment associated with a model element	
Annotations	
Annotations	
Component	
	(Optional AnnotationType value O_O)
An Annotation is considered to <i>recognized</i> if the label is associated with an Annotation Type. otherwise it is <i>ad hoc</i> .	
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.	
A link back to the Component on which this Annotation depends.	(Component value M_1)
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
LiterateDataModels AnnotationType , Subject 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)
the recommended language for expressing derivation, defaults, and constraints
(CodingLanguage value O_O)
OCL
(Optional List of CodingLanguages value O_O)
the recommended language for expressing derivation, defaults, and constraints
(TemplateLanguage value O_O)
Handlebars
(Optional List of TemplateLanguages value O_O)
A list of functions that require sophisticated AI-powered implementation *
(List of String value O_O)
['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.

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

	(UpperCamel value O_O)
Component.name	

The parent subject, if any, under which this subject is nested _	(<i>Optional</i> Subject value O_O)
Subject.inverseOfParentSubject	

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

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

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

[Subject.inverseOfChildSubjects](#)

A link back to the LiterateDataModel on which this Subject depends. (LiterateDataModel value M_1)
Inverse attribute for Subject.parentSubject from which this was implied. (Subject value M_1)
Subject.parentSubject
Inverse attribute for Subject.childSubjects from which this was implied.

	(Subject value M_1)
Subject.childSubjects	
A link back to the LiterateDataModel on which this Subject depends.	(LiterateDataModel value M_1)
Inverse attribute for Subject.parentSubject from which this was implied.	(Subject value M_1)
Subject.parentSubject	
Inverse attribute for Subject.childSubjects from which this was implied.	(Subject value M_1)
Subject.childSubjects	
SubjectArea A main topic or area of focus within the model, containing related subjects and classes	
parentSubject is absent SubjectAreas LiterateModel , Xyz Subject	
A link back to the LiterateModel on which this SubjectArea depends.	(LiterateModel value M_1)
A link back to the Xyz on which this SubjectArea depends.	(Xyz value M_1)
A link back to the LiterateModel on which this SubjectArea depends.	(LiterateModel value M_1)
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

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

([Es](#) value O_O)

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 Classes](#) value O_O)

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

[Class.inverseOfSubtypes](#)

The attributes or properties of the class, in the order in which they should be presented _
(*List of [Attributes](#) value O_O*)

[Attribute.inverseOfAttributes](#)

additional attributes or properties of the class, grouped for clarity and elaboration. _
(*List of [AttributeSections](#) value O_O*)

[AttributeSection.inverseOfAttributeSections](#)

Any constraints, rules, or validations specific to this class _
(*List of [Constraints](#) value O_O*)

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

Any behaviors or operations associated with this class _
(*List of [Methods](#) value O_O*)

[Method.inverseOfMethods](#)

the Classes which are basedOn this Class
(*Optional Set of [Classes](#) value O_O*)

[Class.basedOn](#)

(*Optional Set of [UniqueKeys](#) value O_O*)

[UniqueKey.basedOn](#)

Inverse attribute for `LiterateDataModel.allSubjects` from which this was implied.
(*[LiterateDataModel](#) value M_1*)

[LiterateDataModel.allSubjects](#)

Inverse attribute for `LiterateDataModel.allClasses` from which this was implied.
(*[LiterateDataModel](#) value M_1*)

[LiterateDataModel.allClasses](#)

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

	(Subtyping_value M_1)
Subtyping.classes	
Inverse attribute for Class.basedOn from which this was implied.	(Class_value M_1)
Class.basedOn	
Inverse attribute for Class.subtypes from which this was implied.	(Class_value M_1)
Class.subtypes	
Inverse attribute for Subtyping.classes from which this was implied.	(Subtyping_value M_1)
Subtyping.classes	
Inverse attribute for SimpleDataTypeSubtpeOfDataType.coreClass from which this was implied.	(SimpleDataTypeSubtpeOfDataType_value M_1)
SimpleDataTypeSubtpeOfDataType.coreClass	

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

Subtypings
Subtypings
[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](#)

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)
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)
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	
RAIReferenceTypes	
Class	
CodeType A data type or enumeration used in the model	
CodeTypes	
RAICodeTypes	
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	
RAICodeValues	
CodeType	
A short code or abbreviation for the value _	(NameString 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:	
A link back to the CodeType on which this CodeValue depends.	(CodeType value M_1)

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

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

Key
a list of attributes of a class

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

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

[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.

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

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

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

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)
<p>If the Attribute Section is required, then each Attribute within the section is optional or required, depending on how it is marked.</p> <ul style="list-style-type: none"> • • But if the Attribute Section is optional each attribute in the section is only required if any attribute in the section is present.
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)
inverse attribute for Class.attributeSections from which this was implied. (Class value M_1) Class.attributeSections

Reverse attribute for Class.attributeSections from which this was implied. (<u>Class</u> value M_1)
<u>Class.attributeSections</u>
A link back to the Class on which this AttributeSection depends. (<u>Class</u> 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 _

([CardinalityCode](#) 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.

([InventedName](#) value O_O)

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

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)
from the data type. Null unless attribute is invertible.
(Optional Attribute value O_O)
(Optional Attribute value O_O)
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)
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 _
(Optional Derivation value O_O)
on insert vs on access?
Any validation rules specific to this attribute _
(List of Constraints value O_O)
from Class.constraints
Inverse attribute for Class.attributes from which this was implied.
(Class value M_1)
Class.attributes
Inverse attribute for Key.keyAttributes from which this was implied.
(Key value M_1)
Key.keyAttributes
A link back to the AttributeSection on which this Attribute depends.

	(AttributeSection value M_1)
Inverse attribute for Class.attributes from which this was implied.	(Class value M_1)
Class.attributes	
Inverse attribute for Key.keyAttributes from which this was implied.	(Key value M_1)
Key.keyAttributes	
Inverse attribute for Class.attributes from which this was implied.	(Class value M_1)
Class.attributes	
Inverse attribute for Key.keyAttributes from which this was implied.	(Key value M_1)
Key.keyAttributes	
A link back to the AttributeSection on which this Attribute depends.	(AttributeSection value M_1)
Derivation A rule or formula for deriving the value of an attribute	
Derivations	
An English language statement of the derivation rule _	(RichText value O_O)
The formal expression of the derivation in a programming language _	(CodeExpression value O_O)
Constraint A rule, condition, or validation that must be satisfied by the model	
Constraints	
Component	
ClassConstraint , AttributeConstraint	
An English language statement of the constraint _	(RichText value O_O)
The formal expression of the constraint in a programming language	(InventedName value O_O)
	(Code value O_O)

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

Message

Messages
RAIMessages

ClassConstraint

ClassConstraints
RAIClassConstraints
[Class](#)
[Constraint](#)

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

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

AttributeConstraint

AttributeConstraints
RAIAttributeConstraints
[Attribute](#)
[Constraint](#)

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

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

CodeExpression
CodeExpressions
CodeExpressions
the programming language (<i>Code value O_O</i>)
<div> <div></div> <div>OCL, Object Constraint Language</div> <div>Java, Java</div> </div>
(<i>String value O_O</i>)
Method
A behavior or operation associated with a class
Methods
Component
The input parameters of the method _ (<i>List of Parameters value O_O</i>)
ParameterAnInputToAMethod.inverseOfParameters
The data type of the value returned by the method _ (<i>DataType value O_O</i>)
Inverse attribute for Class.methods from which this was implied. (<i>Class value M_1</i>)
Class.methods
Inverse attribute for Class.methods from which this was implied. (<i>Class value M_1</i>)
Class.methods
Inverse attribute for Class.methods from which this was implied. (<i>Class value M_1</i>)
Class.methods

ParameterAnInputToAMethod
Parameters Component
The data type of the parameter _ (DataType value O_O)
The cardinality of the parameter (InventedName value O_O)
Inverse attribute for Method.parameters from which this was implied. (Method value M_1) Method.parameters
Inverse attribute for Method.parameters from which this was implied. (Method value M_1) Method.parameters
DataType
DataTypes ⊢ AI DataTypes
SimpleDataTypeSubtpeOfDataType
SimpleDataTypeSubtpeOfDataTypes ⊢ AI SimpleDataTypeSubtpeOfDataTypes
 (Class value O_O) Class.inverseOfCoreClass
ComplexDataType
ComplexDataTypes ⊢ AI ComplexDataTypes
 (AggregatingOperator value O_O)
 (List of DataTypes value O_O)

AggregatingOperator

AggregatingOperators
❗❗❗AggregatingOperators

([Code](#) value O_O)

SetOf
ListOf
Mapping

([Integer](#) value O_O)

([Template](#) value O_O)

Emoji

Emojis
❗❗❗Emojis

String

Strings
❗❗❗Strings

CamelName

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

CamelNames
❗❗❗CamelNames

[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
UpperCamels

RAI [CamelName](#)

LowerCamel a CamelName that begins with a lower case letter

"firstName", "orderTotal", "shippingAddress"

content begins with a lower case letter.

LowerCamels

RAI LowerCamels

[CamelName](#)

QualifiedCamel

an expression consisting of Camel Names separated by periods

QualifiedCamels

RAI QualifiedCamels

[String](#)

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.

ValueTypeRichTexts

RAI ValueTypeRichTexts

[String](#)

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

RAI OneLiners

[RichText](#)

the string content

([String_value O_O](#))

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

❧ PrimitiveTypes

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

String

Strings

❧ Strings

[PrimitiveType](#)

[CamelName](#), [QualifiedCamel](#), [ValueTypeRichText](#)

Integer

Integers

❧ Integers

[PrimitiveType](#)

Decimal

Decimals

❧ Decimals

[PrimitiveType](#)

Boolean

Booleans

❧ Booleans

[PrimitiveType](#)

Date

Dates

❧ Dates

[PrimitiveType](#)

Time

Times

❧ Times

[PrimitiveType](#)

DateTime

DateTimes

❧ DateTimes

[PrimitiveType](#)

Component An element or building block of the literate data model Components RAIComponents Annotation LiterateDataModel , Subject , Class , Key , AttributeSection , Attribute , Constraint , Method , ParameterAnInputToAMethod
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)
mechanical attributes
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.

AnnotationType	
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</i> or <i>registered</i> Annotation Types.	
AnnotationTypes	
AnnotationTypes	
LiterateDataModel	
an emoji	(Emoji value O_O)
an emoji	(String value O_O)
the Unicode for the emoji	(String value O_O)
A short label to indicate the purpose of the annotation _	(LowerCamel value O_O)
the plural form of the label	(UpperCamel value O_O)
based on label	
the intended reason for the annotation.	(OneLiner value O_O)
A link back to the LiterateDataModel on which this AnnotationType depends.	(LiterateDataModel value M_1)
reverse attribute for Annotation.annotationType from which this was implied.	(Annotation value M_1)
Annotation.annotationType	

<p>A link back to the LiterateDataModel on which this AnnotationType depends.</p> <p>(LiterateDataModel value M_1)</p>
<p>reverse attribute for Annotation.annotationType from which this was implied.</p> <p>(Annotation value M_1)</p> <p>Annotation.annotationType</p>

Annotation A note or comment associated with a model element	
Annotations Annotations Component	
	(<i>Optional</i> AnnotationType value <i>O_O</i>)
An Annotation is considered to <i>recognized</i> if the label is associated with an Annotation Type. otherwise it is <i>ad hoc</i> . Should be a Value Type AnnotationType.inverseOfAnnotationType	
A short label to indicate the purpose of the annotation _	(CamelName value <i>O_O</i>)
But any short label is valid. from annotationType	
	(<i>Optional</i> Emoji value <i>O_O</i>)
from annotation type	
The content or body of the annotation	(RichText value <i>O_O</i>)
Indicates whether this annotation is an embellishment added during post-parsing processing _	(Boolean value <i>O_O</i>)
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.	
A link back to the Component on which this Annotation depends.	(Component value <i>M_1</i>)
A link back to the Component on which this Annotation depends.	(Component value <i>M_1</i>)

LiterateDataModel A representation of a domain's entities, attributes, and relationships, along with explanatory text and examples LiterateDataModels AnnotationType , Subject 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)
the recommended language for expressing derivation, defaults, and constraints (CodingLanguage value O_O) OCL	
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()']	

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.

Subjects

[LiterateDataModel](#)

[Component](#)

[SubjectArea](#)

	(UpperCamel value O_O)
Component.name	

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

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

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.

[Subject.inverseOfChildSubjects](#)

A link back to the LiterateDataModel on which this Subject depends. (LiterateDataModel value M_1)
Inverse attribute for Subject.parentSubject from which this was implied. (Subject value M_1)
Subject.parentSubject
Inverse attribute for Subject.childSubjects from which this was implied.

	(Subject value M_1)
Subject.childSubjects	
A link back to the LiterateDataModel on which this Subject depends.	(LiterateDataModel value M_1)
Inverse attribute for Subject.parentSubject from which this was implied.	(Subject value M_1)
Subject.parentSubject	
Inverse attribute for Subject.childSubjects from which this was implied.	(Subject value M_1)
Subject.childSubjects	
SubjectArea A main topic or area of focus within the model, containing related subjects and classes parentSubject is absent SubjectAreas LiterateModel , Xyz Subject	
A link back to the LiterateModel on which this SubjectArea depends.	(LiterateModel value M_1)
A link back to the Xyz on which this SubjectArea depends.	(Xyz value M_1)
A link back to the LiterateModel on which this SubjectArea depends.	(LiterateModel value M_1)
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

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

([Es](#) value O_O)

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 Classes](#) value O_O)

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

[Class.inverseOfSubtypes](#)

The attributes or properties of the class, in the order in which they should be presented _
(List of [Attributes](#) value O_O)

[Attribute.inverseOfAttributes](#)

additional attributes or properties of the class, grouped for clarity and elaboration. _
(List of [AttributeSections](#) value O_O)

[AttributeSection.inverseOfAttributeSections](#)

Any constraints, rules, or validations specific to this class _
(List of [Constraints](#) value O_O)

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

Any behaviors or operations associated with this class _
(List of [Methods](#) value O_O)

[Method.inverseOfMethods](#)

the Classes which are basedOn this Class
(Optional Set of [Classes](#) value O_O)

[Class.basedOn](#)

(Optional Set of [UniqueKeys](#) value O_O)

[UniqueKey.basedOn](#)

Inverse attribute for `LiterateDataModel.allSubjects` from which this was implied.
([LiterateDataModel](#) value M_1)

[LiterateDataModel.allSubjects](#)

Inverse attribute for `LiterateDataModel.allClasses` from which this was implied.
([LiterateDataModel](#) value M_1)

[LiterateDataModel.allClasses](#)

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

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

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

Subtypings
 Subtypings
[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

- Subtypes: 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](#)

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)
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)
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	
↗ReferenceTypes	
Class	
CodeType A data type or enumeration used in the model	
CodeTypes	
↗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	
↗CodeValues	
CodeType	
A short code or abbreviation for the value _	(NameString 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:	
A link back to the CodeType on which this CodeValue depends.	(CodeType value M_1)

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

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

Key
a list of attributes of a class

Keys
AllKeys
[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.

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

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

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

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

([Es](#) value O_O)

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 Classes](#) value O_O)

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

[Class.inverseOfSubtypes](#)

The attributes or properties of the class, in the order in which they should be presented _
(*List of [Attributes](#) value O_O*)

[Attribute.inverseOfAttributes](#)

additional attributes or properties of the class, grouped for clarity and elaboration. _
(*List of [AttributeSections](#) value O_O*)

[AttributeSection.inverseOfAttributeSections](#)

Any constraints, rules, or validations specific to this class _
(*List of [Constraints](#) value O_O*)

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

Any behaviors or operations associated with this class _
(*List of [Methods](#) value O_O*)

[Method.inverseOfMethods](#)

the Classes which are basedOn this Class
(*Optional Set of [Classes](#) value O_O*)

[Class.basedOn](#)

(*Optional Set of [UniqueKeys](#) value O_O*)
[UniqueKey.basedOn](#)

Inverse attribute for `LiterateDataModel.allSubjects` from which this was implied.
(*[LiterateDataModel](#) value M_1*)

[LiterateDataModel.allSubjects](#)

Inverse attribute for `LiterateDataModel.allClasses` from which this was implied.
(*[LiterateDataModel](#) value M_1*)
[LiterateDataModel.allClasses](#)

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

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

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

Subtypings
 Subtypings
[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

- Subtypes: 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](#)

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)
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)
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	
⚡ReferenceTypes	
Class	
CodeType A data type or enumeration used in the model	
CodeTypes	
⚡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	
⚡CodeValues	
CodeType	
A short code or abbreviation for the value _	(NameString 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:	
A link back to the CodeType on which this CodeValue depends.	(CodeType value M_1)

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

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

Key
a list of attributes of a class

Keys
AllKeys
[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.

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

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

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

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

AttributeSections
RAIAttributeSections
[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.

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**)

inverse attribute for Class.attributeSections from which this was implied.
([Class](#) value **M_1**)
[Class.attributeSections](#)

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 _

([CardinalityCode](#) 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.

([InventedName](#) value O_O)

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

[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)
from the data type. Null unless attribute is invertible.
(Optional Attribute value O_O)
(Optional Attribute value O_O)
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)
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 _
(Optional Derivation value O_O)
on insert vs on access?
Any validation rules specific to this attribute _
(List of Constraints value O_O)
from Class.constraints
Inverse attribute for Class.attributes from which this was implied.
(Class value M_1)
Class.attributes
Inverse attribute for Key.keyAttributes from which this was implied.
(Key value M_1)
Key.keyAttributes
A link back to the AttributeSection on which this Attribute depends.

	(AttributeSection value M_1)
Inverse attribute for Class.attributes from which this was implied.	(Class value M_1)
Class.attributes	
Inverse attribute for Key.keyAttributes from which this was implied.	(Key value M_1)
Key.keyAttributes	
Inverse attribute for Class.attributes from which this was implied.	(Class value M_1)
Class.attributes	
Inverse attribute for Key.keyAttributes from which this was implied.	(Key value M_1)
Key.keyAttributes	
A link back to the AttributeSection on which this Attribute depends.	(AttributeSection value M_1)
Derivation A rule or formula for deriving the value of an attribute	
Derivations	
An English language statement of the derivation rule _	(RichText value O_O)
The formal expression of the derivation in a programming language _	(CodeExpression value O_O)
Constraint A rule, condition, or validation that must be satisfied by the model	
Constraints	
Component	
ClassConstraint , AttributeConstraint	
An English language statement of the constraint _	(RichText value O_O)
The formal expression of the constraint in a programming language	(InventedName value O_O)
	(Code value O_O)

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

Message

Messages
RAILMessages

ClassConstraint

ClassConstraints
RAIClassConstraints
Class
Constraint

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

(Class value M_1)

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

(Class value M_1)

AttributeConstraint

AttributeConstraints

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

(Attribute value M_1)

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

(Attribute value M_1)

CodeExpression
CodeExpressions AI CodeExpressions
the programming language (<i>Code value O_O</i>)
<div> <div></div> <div>OCL, Object Constraint Language Java, Java</div> </div>
(<i>String value O_O</i>)
Method A behavior or operation associated with a class
Methods Component
The input parameters of the method _ (<i>List of Parameters value O_O</i>) ParameterAnInputToAMethod.inverseOfParameters
The data type of the value returned by the method _ (<i>DataType value O_O</i>)
Inverse attribute for Class.methods from which this was implied. (<i>Class value M_1</i>) Class.methods
Inverse attribute for Class.methods from which this was implied. (<i>Class value M_1</i>) Class.methods
Inverse attribute for Class.methods from which this was implied. (<i>Class value M_1</i>) Class.methods

ParameterAnInputToAMethod
Parameters Component
The data type of the parameter _ (DataType value O_O)
The cardinality of the parameter (InventedName value O_O)
Inverse attribute for Method.parameters from which this was implied. (Method value M_1) Method.parameters
Inverse attribute for Method.parameters from which this was implied. (Method value M_1) Method.parameters
DataType
DataTypes ⌞DataTypes
SimpleDataTypeSubtpeOfDataType
SimpleDataTypeSubtpeOfDataTypes ⌞SimpleDataTypeSubtpeOfDataTypes
 (Class value O_O) Class.inverseOfCoreClass
ComplexDataType
ComplexDataTypes ⌞ComplexDataTypes
 (AggregatingOperator value O_O)
 (List of DataTypes value O_O)

AggregatingOperator

AggregatingOperators
RAI AggregatingOperators

([Code](#) value O_O)

SetOf
ListOf
Mapping

([Integer](#) value O_O)

([Template](#) value O_O)

Emoji

Emojis
RAI Emojis

String

Strings
RAI Strings

CamelName

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

CamelNames
RAI CamelNames

[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
UpperCamels

LowerCamelName

LowerCamel a CamelName that begins with a lower case letter

"firstName", "orderTotal", "shippingAddress"

content begins with a lower case letter.

LowerCamels

LowerCamels

[CamelName](#)

QualifiedCamel

an expression consisting of Camel Names separated by periods

QualifiedCamels

QualifiedCamels

[String](#)

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.

ValueTypeRichTexts

ValueTypeRichTexts

[String](#)

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

OneLiners

[RichText](#)

the string content

([String_value](#) O_O)

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

String

Strings
❖ Strings
[PrimitiveType](#)
[CamelName](#), [QualifiedCamel](#), [ValueTypeRichText](#)

Integer

Integers
❖ Integers
[PrimitiveType](#)

Decimal

Decimals
❖ Decimals
[PrimitiveType](#)

Boolean

Booleans
❖ Booleans
[PrimitiveType](#)

Date

Dates
❖ Dates
[PrimitiveType](#)

Time

Times
❖ Times
[PrimitiveType](#)

DateTime

DateTimes
❖ DateTimes
[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 Components Components Annotation LiterateDataModel , Subject , Class , Key , AttributeSection , Attribute , Constraint , Method , ParameterAnInputToAMethod
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)
mechanical attributes
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.

AnnotationType

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 *recognized* or *registered* Annotation Types.

AnnotationTypes

AnnotationTypes

[LiterateDataModel](#)

an emoji

([Emoji](#) value O_O)

an emoji

([String](#) value O_O)

the Unicode for the emoji

([String](#) value O_O)

A short label to indicate the purpose of the annotation _

([LowerCamel](#) value O_O)

the plural form of the label

([UpperCamel](#) value O_O)

based on label

the intended reason for the annotation.

([OneLiner](#) value O_O)

A link back to the LiterateDataModel on which this AnnotationType depends.

([LiterateDataModel](#) value M_1)

reverse attribute for Annotation.annotationType from which this was implied.

([Annotation](#) value M_1)

[Annotation.annotationType](#)

<p>A link back to the LiterateDataModel on which this AnnotationType depends.</p> <p>(LiterateDataModel value M_1)</p>
<p>inverse attribute for Annotation.annotationType from which this was implied.</p> <p>(Annotation value M_1)</p> <p>Annotation.annotationType</p>

Annotation	
A note or comment associated with a model element	
Annotations	
Annotations	
Component	
	(Optional AnnotationType value O_O)
An Annotation is considered to <i>recognized</i> if the label is associated with an Annotation Type. otherwise it is <i>ad hoc</i> .	
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.	
A link back to the Component on which this Annotation depends.	(Component value M_1)
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

LiterateDataModels

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

[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)

the recommended language for expressing derivation, defaults, and constraints

([CodingLanguage](#) value O_O)

OCL

ges

([Optional List of CodingLanguages](#) value O_O)

the recommended language for expressing derivation, defaults, and constraints

([TemplateLanguage](#) value O_O)

Handlebars

pages

([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

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.

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

	(UpperCamel value O_O)
Component.name	

The parent subject, if any, under which this subject is nested _	(<i>Optional</i> Subject value O_O)
Subject.inverseOfParentSubject	

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

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

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

[Subject.inverseOfChildSubjects](#)

A link back to the LiterateDataModel on which this Subject depends. (LiterateDataModel value M_1)
Inverse attribute for Subject.parentSubject from which this was implied. (Subject value M_1)
Subject.parentSubject
Inverse attribute for Subject.childSubjects from which this was implied.

	(Subject value M_1)
Subject.childSubjects	
A link back to the LiterateDataModel on which this Subject depends.	(LiterateDataModel value M_1)
Inverse attribute for Subject.parentSubject from which this was implied.	(Subject value M_1)
Subject.parentSubject	
Inverse attribute for Subject.childSubjects from which this was implied.	(Subject value M_1)
Subject.childSubjects	
SubjectArea A main topic or area of focus within the model, containing related subjects and classes	
parentSubject is absent SubjectAreas LiterateModel , Xyz Subject	
A link back to the LiterateModel on which this SubjectArea depends.	(LiterateModel value M_1)
A link back to the Xyz on which this SubjectArea depends.	(Xyz value M_1)
A link back to the LiterateModel on which this SubjectArea depends.	(LiterateModel value M_1)
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

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

([Es](#) value O_O)

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 Classes](#) value O_O)

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

[Class.inverseOfSubtypes](#)

The attributes or properties of the class, in the order in which they should be presented _
(*List of [Attributes](#) value O_O*)

[Attribute.inverseOfAttributes](#)

additional attributes or properties of the class, grouped for clarity and elaboration. _
(*List of [AttributeSections](#) value O_O*)

[AttributeSection.inverseOfAttributeSections](#)

Any constraints, rules, or validations specific to this class _
(*List of [Constraints](#) value O_O*)

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

Any behaviors or operations associated with this class _
(*List of [Methods](#) value O_O*)

[Method.inverseOfMethods](#)

the Classes which are basedOn this Class
(*Optional Set of [Classes](#) value O_O*)

[Class.basedOn](#)

(*Optional Set of [UniqueKeys](#) value O_O*)

[UniqueKey.basedOn](#)

Inverse attribute for `LiterateDataModel.allSubjects` from which this was implied.
(*[LiterateDataModel](#) value M_1*)

[LiterateDataModel.allSubjects](#)

Inverse attribute for `LiterateDataModel.allClasses` from which this was implied.
(*[LiterateDataModel](#) value M_1*)

[LiterateDataModel.allClasses](#)

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

	(Subtyping_value M_1)
Subtyping.classes	
Inverse attribute for Class.basedOn from which this was implied.	(Class_value M_1)
Class.basedOn	
Inverse attribute for Class.subtypes from which this was implied.	(Class_value M_1)
Class.subtypes	
Inverse attribute for Subtyping.classes from which this was implied.	(Subtyping_value M_1)
Subtyping.classes	
Inverse attribute for SimpleDataTypeSubtpeOfDataType.coreClass from which this was implied.	(SimpleDataTypeSubtpeOfDataType_value M_1)
SimpleDataTypeSubtpeOfDataType.coreClass	

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

Subtypings
Subtypings
[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](#)

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)
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)
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 RAIReferenceTypes Class	
CodeType A data type or enumeration used in the model	
CodeTypes RAICodeTypes 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 RAICodeValues CodeType	
A short code or abbreviation for the value _	(NameString 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:	
A link back to the CodeType on which this CodeValue depends.	

	(CodeType value M_1)
A link back to the CodeType on which this CodeValue depends.	(CodeType value M_1)
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_0)
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.	
A link back to the Class on which this Key depends.	(Class value M_1)
A link back to the Class on which this Key depends.	(Class value M_1)
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

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

([Es](#) value O_O)

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 Classes](#) value O_O)

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

[Class.inverseOfSubtypes](#)

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

(*List of [Attributes](#) value O_O*)

[Attribute.inverseOfAttributes](#)

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

(*List of [AttributeSections](#) value O_O*)

[AttributeSection.inverseOfAttributeSections](#)

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

(*List of [Constraints](#) value O_O*)

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

Any behaviors or operations associated with this class _

(*List of [Methods](#) value O_O*)

[Method.inverseOfMethods](#)

the Classes which are basedOn this Class

(*Optional Set of [Classes](#) value O_O*)

[Class.basedOn](#)

(*Optional Set of [UniqueKeys](#) value O_O*)

[UniqueKey.basedOn](#)

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

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

[LiterateDataModel.allSubjects](#)

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

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

[LiterateDataModel.allClasses](#)

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

	(Subtyping_value M_1)
Subtyping.classes	
Inverse attribute for Class.basedOn from which this was implied.	(Class_value M_1)
Class.basedOn	
Inverse attribute for Class.subtypes from which this was implied.	(Class_value M_1)
Class.subtypes	
Inverse attribute for Subtyping.classes from which this was implied.	(Subtyping_value M_1)
Subtyping.classes	
Inverse attribute for SimpleDataTypeSubtpeOfDataType.coreClass from which this was implied.	(SimpleDataTypeSubtpeOfDataType_value M_1)
SimpleDataTypeSubtpeOfDataType.coreClass	

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

Subtypings
Subtypings
[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](#)

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)
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)
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 RAIReferenceTypes Class	
CodeType A data type or enumeration used in the model	
CodeTypes RAICodeTypes 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 RAICodeValues CodeType	
A short code or abbreviation for the value _	(NameString 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:	
A link back to the CodeType on which this CodeValue depends.	

	(CodeType value M_1)
--	--

--

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

--

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_0)

[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.

--

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

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

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

Classes

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

([Es](#) value O_O)

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 Classes](#) value O_O)

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

[Class.inverseOfSubtypes](#)

The attributes or properties of the class, in the order in which they should be presented _
(List of [Attributes](#) value O_O)

[Attribute.inverseOfAttributes](#)

additional attributes or properties of the class, grouped for clarity and elaboration. _
(List of [AttributeSections](#) value O_O)

[AttributeSection.inverseOfAttributeSections](#)

Any constraints, rules, or validations specific to this class _
(List of [Constraints](#) value O_O)

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

Any behaviors or operations associated with this class _
(List of [Methods](#) value O_O)

[Method.inverseOfMethods](#)

the Classes which are basedOn this Class
(Optional Set of [Classes](#) value O_O)

[Class.basedOn](#)

(Optional Set of [UniqueKeys](#) value O_O)
[UniqueKey.basedOn](#)

Inverse attribute for `LiterateDataModel.allSubjects` from which this was implied.
([LiterateDataModel](#) value M_1)

[LiterateDataModel.allSubjects](#)

Inverse attribute for `LiterateDataModel.allClasses` from which this was implied.
([LiterateDataModel](#) value M_1)
[LiterateDataModel.allClasses](#)

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

	(<u>Subtyping</u> .value M_1)
<u>Subtyping.classes</u>	
Inverse attribute for Class.basedOn from which this was implied.	(<u>Class</u> .value M_1)
<u>Class.basedOn</u>	
Inverse attribute for Class.subtypes from which this was implied.	(<u>Class</u> .value M_1)
<u>Class.subtypes</u>	
Inverse attribute for Subtyping.classes from which this was implied.	(<u>Subtyping</u> .value M_1)
<u>Subtyping.classes</u>	
Inverse attribute for SimpleDataTypeSubtpeOfDataType.coreClass from which this was implied.	(<u>SimpleDataTypeSubtpeOfDataType</u> .value M_1)
<u>SimpleDataTypeSubtpeOfDataType.coreClass</u>	

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

Subtypings
 Subtypings
[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

- Subtypes: 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](#)

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)
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)
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 rAIReferenceTypes Class	
CodeType A data type or enumeration used in the model	
CodeTypes rAICodeTypes 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 rAICodeValues CodeType	
A short code or abbreviation for the value _	(NameString 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:	
A link back to the CodeType on which this CodeValue depends.	

([CodeType](#) value M_1)

--

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

--

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.

--

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

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

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

<p>UniqueKey a list of attributes on which instances of the base class may be keyed.</p>

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_0)
<p>If the Attribute Section is required, then each Attribute within the section is optional or required, depending on how it is marked.</p> <ul style="list-style-type: none"> • • But if the Attribute Section is optional each attribute in the section is only required if any attribute in the section is present.
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)
inverse attribute for Class.attributeSections from which this was implied. (Class value M_1) Class.attributeSections

inverse attribute for Class.attributeSections from which this was implied.	(<u>Class</u> value M_1)
<u>Class.attributeSections</u>	
A link back to the Class on which this AttributeSection depends.	(<u>Class</u> value M_1)

Attribute A property or characteristic of a class Attributes AttributeSection AttributeConstraint Component
<div>(LowerCamel value O_O)</div> Component.name
The kind of object to which the attribute refers. _ <div>(DataType value O_O)</div>
But, <ul style="list-style-type: none"> ◦ List of Editions ◦ Set of Edition ◦ ... and more complicated cases. the section below on Data Type Specifiers.
<div>Indicates whether the attribute must have a value for every instance of the class _</div> <div>(Boolean value O_O)</div> *** False
<div>The cardinality of the relationship represented by the attribute _</div> <div>(CardinalityCode value O_O)</div> *** 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.
<div>(InventedName value O_O)</div>
<div>(Optional InventedName value O_O)</div> 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)
from the data type. Null unless attribute is invertible.
(Optional Attribute value O_O)
(Optional Attribute value O_O)
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)
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 _
(Optional Derivation value O_O)
on insert vs on access?
Any validation rules specific to this attribute _
(List of Constraints value O_O)
from Class.constraints
Inverse attribute for Class.attributes from which this was implied.
(Class value M_1)
Class.attributes
Inverse attribute for Key.keyAttributes from which this was implied.
(Key value M_1)
Key.keyAttributes
A link back to the AttributeSection on which this Attribute depends.

	(AttributeSection value M_1)
Inverse attribute for Class.attributes from which this was implied.	(Class value M_1)
Class.attributes	
Inverse attribute for Key.keyAttributes from which this was implied.	(Key value M_1)
Key.keyAttributes	
Inverse attribute for Class.attributes from which this was implied.	(Class value M_1)
Class.attributes	
Inverse attribute for Key.keyAttributes from which this was implied.	(Key value M_1)
Key.keyAttributes	
A link back to the AttributeSection on which this Attribute depends.	(AttributeSection value M_1)
Derivation	
A rule or formula for deriving the value of an attribute	
Derivations	
An English language statement of the derivation rule _	(RichText value O_0)
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	
Constraints	
Component	
ClassConstraint , AttributeConstraint	
An English language statement of the constraint _	(RichText value O_0)
The formal expression of the constraint in a programming language	(InventedName value O_0)
	(Code value O_0)

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

Message
Messages
RAIMessages

ClassConstraint
ClassConstraints
RAIClassConstraints
Class
Constraint

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

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

AttributeConstraint
AttributeConstraints
RAIAttributeConstraints
Attribute
Constraint

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

--

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

CodeExpression

CodeExpressions
2A1CodeExpressions

the programming language
([Code](#) value O_O)

OCL, Object Constraint Language
Java, Java

([String](#) value O_O)

Methods

Method A behavior or operation associated with a class Methods Component
The input parameters of the method _ (<i>List of Parameters value O_O</i>) ParameterAnInputToAMethod.inverseOfParameters
The data type of the value returned by the method _ (<i>DataType value O_O</i>)
Inverse attribute for Class.methods from which this was implied. (<i>Class value M_1</i>) Class.methods
Inverse attribute for Class.methods from which this was implied. (<i>Class value M_1</i>) Class.methods
Inverse attribute for Class.methods from which this was implied. (<i>Class value M_1</i>) Class.methods

ParameterAnInputToAMethod
Parameters Component
The data type of the parameter _ (DataType value O_O)
The cardinality of the parameter (InventedName value O_O)
Inverse attribute for Method.parameters from which this was implied. (Method value M_1) Method.parameters
Inverse attribute for Method.parameters from which this was implied. (Method value M_1) Method.parameters

Data Types

DataType
DataTypes RAIDatatypes
SimpleDataTypeSubtpeOfDataType
SimpleDataTypeSubtpeOfDataTypes RAISimpleDataTypeSubtpeOfDataTypes
<div>(Class value O_O)</div> <div>Class.inverseOfCoreClass</div>
ComplexDataType
ComplexDataTypes RAIDComplexDataTypes
<div>(AggregatingOperator value O_O)</div>
<div>(List of DataTypes value O_O)</div>
AggregatingOperator
AggregatingOperators RAIAggregatingOperators
<div>(Code value O_O)</div>
<div> <div> <div></div> <div>SetOf</div> <div>ListOf</div> <div>Mapping</div> </div> </div>
<div>(Integer value O_O)</div>
<div>(Template value O_O)</div>

Low level Data Types

insert Camel Case.md

Emoji

Emojis
✂️Emojis

String

Strings
✂️Strings

CamelName

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

CamelNames
✂️CamelNames
[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
✂️UpperCamels
[CamelName](#)

LowerCamel

a CamelName that begins with a lower case letter

"firstName", "orderTotal", "shippingAddress"
content begins with a lower case letter.
LowerCamels
✂️LowerCamels
[CamelName](#)

QualifiedCamel
an expression consisting of Camel Names separated by periods

QualifiedCamels
QualifiedCamels
[String](#)

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.

ValueTypeRichTexts
valueTypeRichTexts
[String](#)

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
OneLiners
[RichText](#)

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

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

String

Strings
RAIStrings
[PrimitiveType](#)
[CamelName](#), [QualifiedCamel](#), [ValueTypeRichText](#)

Integer

Integers
RAIIntegers
[PrimitiveType](#)

Decimal

Decimals
RAIDecimals
[PrimitiveType](#)

Boolean

Booleans
RAIBooleans
[PrimitiveType](#)

Date

Dates
RAIDates
[PrimitiveType](#)

Time

Times
RAITimes
[PrimitiveType](#)

DateTime

DateTimes
RAIDateTimes
[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": "💡",
"emojiName": "bulb",
"emojiUnicode": "U+1F4A1",
"purpose": "Provides a suggestion or recommendation for improving the model or the annotated element."
},
{
"label": "Info",
"emoji": "📘",
"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
```

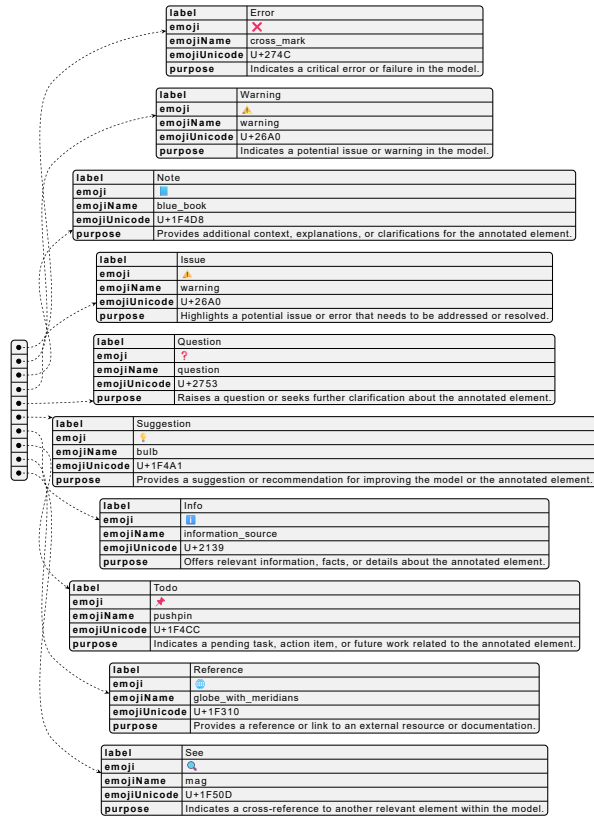


```

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


```


•	<table><tr><td>label</td><td>Error</td></tr><tr><td>emoji</td><td>❌</td></tr><tr><td>emojiName</td><td>cross_mark</td></tr><tr><td>emojiUnicode</td><td>U+274C</td></tr><tr><td>purpose</td><td>Indicates a critical error or failure in the model.</td></tr></table>	label	Error	emoji	❌	emojiName	cross_mark	emojiUnicode	U+274C	purpose	Indicates a critical error or failure in the model.
label	Error										
emoji	❌										
emojiName	cross_mark										
emojiUnicode	U+274C										
purpose	Indicates a critical error or failure in the model.										
•	<table><tr><td>label</td><td>Warning</td></tr><tr><td>emoji</td><td>⚠️</td></tr><tr><td>emojiName</td><td>warning</td></tr><tr><td>emojiUnicode</td><td>U+26A0</td></tr><tr><td>purpose</td><td>Indicates a potential issue or warning in the model.</td></tr></table>	label	Warning	emoji	⚠️	emojiName	warning	emojiUnicode	U+26A0	purpose	Indicates a potential issue or warning in the model.
label	Warning										
emoji	⚠️										
emojiName	warning										
emojiUnicode	U+26A0										
purpose	Indicates a potential issue or warning in the model.										
•	<table><tr><td>label</td><td>Note</td></tr><tr><td>emoji</td><td>📖</td></tr><tr><td>emojiName</td><td>blue_book</td></tr><tr><td>emojiUnicode</td><td>U+1F4D8</td></tr><tr><td>purpose</td><td>Provides additional context, explanations, or clarifications for the annotated element.</td></tr></table>	label	Note	emoji	📖	emojiName	blue_book	emojiUnicode	U+1F4D8	purpose	Provides additional context, explanations, or clarifications for the annotated element.
label	Note										
emoji	📖										
emojiName	blue_book										
emojiUnicode	U+1F4D8										
purpose	Provides additional context, explanations, or clarifications for the annotated element.										
•	<table><tr><td>label</td><td>Issue</td></tr><tr><td>emoji</td><td>⚠️</td></tr><tr><td>emojiName</td><td>warning</td></tr><tr><td>emojiUnicode</td><td>U+26A0</td></tr><tr><td>purpose</td><td>Highlights a potential issue or error that needs to be addressed or resolved.</td></tr></table>	label	Issue	emoji	⚠️	emojiName	warning	emojiUnicode	U+26A0	purpose	Highlights a potential issue or error that needs to be addressed or resolved.
label	Issue										
emoji	⚠️										
emojiName	warning										
emojiUnicode	U+26A0										
purpose	Highlights a potential issue or error that needs to be addressed or resolved.										
•	<table><tr><td>label</td><td>Question</td></tr><tr><td>emoji</td><td>❓</td></tr><tr><td>emojiName</td><td>question</td></tr><tr><td>emojiUnicode</td><td>U+2753</td></tr><tr><td>purpose</td><td>Raises a question or seeks further clarification about the annotated element.</td></tr></table>	label	Question	emoji	❓	emojiName	question	emojiUnicode	U+2753	purpose	Raises a question or seeks further clarification about the annotated element.
label	Question										
emoji	❓										
emojiName	question										
emojiUnicode	U+2753										
purpose	Raises a question or seeks further clarification about the annotated element.										
•	<table><tr><td>label</td><td>Suggestion</td></tr><tr><td>emoji</td><td>💡</td></tr><tr><td>emojiName</td><td>bulb</td></tr><tr><td>emojiUnicode</td><td>U+1F4A1</td></tr><tr><td>purpose</td><td>Provides a suggestion or recommendation for improving the model or the annotated element.</td></tr></table>	label	Suggestion	emoji	💡	emojiName	bulb	emojiUnicode	U+1F4A1	purpose	Provides a suggestion or recommendation for improving the model or 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.										
•	<table><tr><td>label</td><td>Info</td></tr><tr><td>emoji</td><td>ℹ️</td></tr><tr><td>emojiName</td><td>information_source</td></tr><tr><td>emojiUnicode</td><td>U+2139</td></tr><tr><td>purpose</td><td>Offers relevant information, facts, or details about the annotated element.</td></tr></table>	label	Info	emoji	ℹ️	emojiName	information_source	emojiUnicode	U+2139	purpose	Offers relevant information, facts, or details about the annotated element.
label	Info										
emoji	ℹ️										
emojiName	information_source										
emojiUnicode	U+2139										
purpose	Offers relevant information, facts, or details about the annotated element.										
•	<table><tr><td>label</td><td>Todo</td></tr><tr><td>emoji</td><td>📌</td></tr><tr><td>emojiName</td><td>pushpin</td></tr><tr><td>emojiUnicode</td><td>U+1F4CC</td></tr><tr><td>purpose</td><td>Indicates a pending task, action item, or future work related to the annotated element.</td></tr></table>	label	Todo	emoji	📌	emojiName	pushpin	emojiUnicode	U+1F4CC	purpose	Indicates a pending task, action item, or future work related to 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.										
•	<table><tr><td>label</td><td>Reference</td></tr><tr><td>emoji</td><td>🌐</td></tr><tr><td>emojiName</td><td>globe_with_meridians</td></tr><tr><td>emojiUnicode</td><td>U+1F310</td></tr><tr><td>purpose</td><td>Provides a reference or link to an external resource or documentation.</td></tr></table>	label	Reference	emoji	🌐	emojiName	globe_with_meridians	emojiUnicode	U+1F310	purpose	Provides a reference or link to an external resource or documentation.
label	Reference										
emoji	🌐										
emojiName	globe_with_meridians										
emojiUnicode	U+1F310										
purpose	Provides a reference or link to an external resource or documentation.										
•	<table><tr><td>label</td><td>See</td></tr><tr><td>emoji</td><td>🔍</td></tr><tr><td>emojiName</td><td>mag</td></tr><tr><td>emojiUnicode</td><td>U+1F50D</td></tr><tr><td>purpose</td><td>Indicates a cross-reference to another relevant element within the model.</td></tr></table>	label	See	emoji	🔍	emojiName	mag	emojiUnicode	U+1F50D	purpose	Indicates a cross-reference to another relevant element within the model.
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, ,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