FIRST PAGE LEFT LEFT BLANK

Literate Data Model

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)

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

(Annotation value M_1)

Annotation.annotationType

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

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

Annotation.annotationType

Annotation
A note or comment associated with a model element

Annotations
RALAnnotations
Component

(Optional AnnotationType value O_O)

An Annotation is considered to *recognized* if the label is associated with an Annotation Type. otherwise it is *ad hoc* .
Should be a Value Type
AnnotationType.inverseOfAnnotationType

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

But any short label is valid.

from annotationType

(Optional Emoji value O_O)

from annotation type

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

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

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

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 lanquage for expressing derivation, defaults, and constraints
(CodingLanguage value O_O)

OCL

ges (Optional List of CodingLanguages value O_O)

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

Handlebars

uages (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 synonymousle 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 it's 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
RALSubtypings
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
RALReferenceTypes
Class

CodeType A data type or enumeration used in the model

CodeTypes
RALCodeTypes
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
RALCodeValues
CodeType

A short code or abbreviationi 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
RALKeys
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
RAL
Key

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

AttributeSections
RALAttributeSections
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 sectional is optional ot required, depending on how it is marked.

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

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

bverse attribute for Class.attributeSections from which this was implied. ***(Class value M_1)***

Class.attributeSections

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

Attribute
A property or characteristic of a class

Attributes
AttributeSection
AttributeConstraint
Component

(LowerCamel value O_O)

Component.name

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

But,

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

the section below on Data Type Specifiers.

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

*** False

The cardinality of the relationship represented by the attribute _
(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 arrribute 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 parenthentical 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
RALMessages

ClassConstraint

ClassConstraints
RALClassConstraints
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
RALAttributeConstraints
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

RALCodeExpressions

the programming language *(Code value O_O)*

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

(String value O_O)

Method

A behavior or operation associated with a class

Methods

Component

The input parameters of the method _

(List of Parameters value O_O)

ParameterAnInputToAMethod.inverseOfParameters

The data type of the value returned by the method _

(DataType value O_O)

Inverse attribute for Class.methods from which this was implied.

(Class value M_1)

Class.methods

Inverse attribute for Class.methods from which this was implied.

(Class value M_1)

Class.methods

Inverse attribute for Class.methods from which this was implied.

(Class value M_1)

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
RALDataTypes

SimpleDataTypeSubtpeOfDataType

SimpleDataTypeSubtpeOfDataTypes
RALSimpleDataTypeSubtpeOfDataTypes

(Class value O_O)

Class.inverseOfCoreClass

ComplexDataType

ComplexDataTypes
RALComplexDataTypes

(AggregatingOperator value O_O)

(List of DataTypes value O_O)

AggregatingOperator

AggregatingOperators
RALAggregatingOperators

(Code value O_O)

```
SetOf
ListOf
Mapping
```

(Integer value O_O)

(Template value O_O)

Emoji

Emojis
RALEmojis

String

Strings
RALStrings

CamelName

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

CamelNames
RALCamelNames
String
UpperCamel , LowerCamel

(String value O_O)

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

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

UpperCamel a CamelName that begins with a capital letter

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

RALCamelName

LowerCamel a CamelName that begins with a lower case letter

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

QualifiedCamel
an expression consisting of Camel Names separated by periods

QualifiedCamels
RALQualifiedCamels
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
RALValueTypeRichTexts
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
RALOneLiners
RichText

the string content *(String value O_O)*

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

PrimitiveType A basic, built-in data type

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

String

Strings
RALStrings
PrimitiveType
CamelName , QualifiedCamel , ValueTypeRichText

Integer

Integers
RALIntegers
PrimitiveType

Decimal

Decimals
RALDecimals
PrimitiveType

Boolean

Booleans
RALBooleans
PrimitiveType

Date

Dates
RALDates
PrimitiveType

Time

Times
RALTimes
PrimitiveType

DateTime

DateTimes
RALDateTimes
PrimitiveType

Component
An element or building block of the literate data model

Components
RALComponents
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

RALAnnotationTypes

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)

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

(Annotation value M_1)

Annotation.annotationType

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

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

Annotation.annotationType

Annotation
A note or comment associated with a model element

Annotations
RALAnnotations
Component

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

An Annotation is considered to *recognized* if the label is associated with an Annotation Type. otherwise it is *ad hoc* .
Should be a Value Type
AnnotationType.inverseOfAnnotationType

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

But any short label is valid.

from annotationType

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

from annotation type

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

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

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

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)

ige recommended lanquage for expressing derivation, defaults, and constraints
(CodingLanguage value O_O)

OCL

ges *(Optional List of CodingLanguages value O_O)*

tharecommended lanquage for expressing derivation, defaults, and constraints
(TemplateLanguage value O_O)

Handlebars

uages *(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 synonymousle 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 it’s 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
RALSubtypings
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
RALReferenceTypes
Class

CodeType A data type or enumeration used in the model

CodeTypes
RALCodeTypes
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
RALCodeValues
CodeType

A short code or abbreviationi 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
RALKeys
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
RAL
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 synonymousle 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 it’s 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
RALSubtypings
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
RALReferenceTypes
Class

CodeType A data type or enumeration used in the model

CodeTypes
RALCodeTypes
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
RALCodeValues
CodeType

A short code or abbreviationi 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
RALKeys
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
RAL
Key

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

AttributeSections
RALAttributeSections
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 sectional is optional ot required, depending on how it is marked.

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

bverse 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)

bverse 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.

(Class value M_1)

Class.attributeSections

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

(Class value M_1)

Attribute
A property or characteristic of a class

Attributes
AttributeSection
AttributeConstraint
Component

(LowerCamel value O_O)

Component.name

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

But,

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

the section below on Data Type Specifiers.

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

*** False

The cardinality of the relationship represented by the attribute _
(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 arrribute 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 parenthentical 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
RALMessages

ClassConstraint

ClassConstraints
RALClassConstraints
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
RALAttributeConstraints
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
RALCodeExpressions

the programming language *(Code value O_O)*

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

(String value O_O)

Method
A behavior or operation associated with a class

Methods
Component

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

ParameterAnInputToAMethod.inverseOfParameters

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

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

Class.methods

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

Class.methods

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

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
RALDataTypes

SimpleDataTypeSubtpeOfDataType

SimpleDataTypeSubtpeOfDataTypes
RALSimpleDataTypeSubtpeOfDataTypes

(Class value O_O)

Class.inverseOfCoreClass

ComplexDataType

ComplexDataTypes
RALComplexDataTypes

(AggregatingOperator value O_O)

(List of DataTypes value O_O)

AggregatingOperator

AggregatingOperators
RALAggregatingOperators

(Code value O_O)

```
SetOf
ListOf
Mapping
```

(Integer value O_O)

(Template value O_O)

Emoji

Emojis
RALEmojis

String

Strings
RALStrings

CamelName

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

CamelNames
RALCamelNames
String
UpperCamel , LowerCamel

(String value O_O)

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

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

UpperCamel a CamelName that begins with a capital letter

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

RALCamelName

LowerCamel a CamelName that begins with a lower case letter

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

QualifiedCamel
an expression consisting of Camel Names separated by periods

QualifiedCamels
RALQualifiedCamels
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
RALValueTypeRichTexts
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
RALOneLiners
RichText

the string content *(String value O_O)*

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

PrimitiveType A basic, built-in data type

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

String

Strings
RALStrings
PrimitiveType
CamelName , QualifiedCamel , ValueTypeRichText

Integer

Integers
RALIntegers
PrimitiveType

Decimal

Decimals
RALDecimals
PrimitiveType

Boolean

Booleans
RALBooleans
PrimitiveType

Date

Dates
RALDates
PrimitiveType

Time

Times
RALTimes
PrimitiveType

DateTime

DateTimes
RALDateTimes
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)

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

(Annotation value M_1)

Annotation.annotationType

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

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

Annotation.annotationType

Annotation
A note or comment associated with a model element

Annotations
RALAnnotations
Component

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

An Annotation is considered to *recognized* if the label is associated with an Annotation Type. otherwise it is *ad hoc* .
Should be a Value Type
AnnotationType.inverseOfAnnotationType

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

But any short label is valid.

from annotationType

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

from annotation type

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

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

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

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)

tge recommended lanquage for expressing derivation, defaults, and constraints

(CodingLanguage value O_O)

OCL

ges *(Optional List of CodingLanguages value O_O)*

tnge recommended lanquage for expressing derivation, defaults, and constraints

(TemplateLanguage value O_O)

Handlebars

uages *(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 synonymousle 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 it's 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
RALSubtypings
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
RALReferenceTypes
Class

CodeType
A data type or enumeration used in the model

CodeTypes
RALCodeTypes
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
RALCodeValues
CodeType

A short code or abbreviationi 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
RALKeys
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 synonymousle 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 it's 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
RALSubtypings
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
RALReferenceTypes
Class

CodeType
A data type or enumeration used in the model

CodeTypes
RALCodeTypes
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
RALCodeValues
CodeType

A short code or abbreviationi 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
RALKeys
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

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 synonymousle 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 it’s 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
RALSubtypings
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
RALReferenceTypes
Class

CodeType
A data type or enumeration used in the model

CodeTypes
RALCodeTypes
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
RALCodeValues
CodeType

A short code or abbreviationi 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
RALKeys
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
RAL
Key

Attributes

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

AttributeSections
RALAttributeSections
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 sectional is optional ot required, depending on how it is marked.

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

bnverse 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)

bnverse attribute for Class.attributeSections from which this was implied.
(Class value M_1)

Class.attributeSections

bverse attribute for Class.attributeSections from which this was implied. ***(Class value M_1)***

Class.attributeSections

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

Attribute
A property or characteristic of a class

Attributes
AttributeSection
AttributeConstraint
Component

(LowerCamel value O_O)

Component.name

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

But,

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

the section below on Data Type Specifiers.

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

*** False

The cardinality of the relationship represented by the attribute _
(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 arrribute 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 parenthentical 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
RALMessages

ClassConstraint

ClassConstraints
RALClassConstraints
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
RALAttributeConstraints
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

RAL CodeExpressions

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 _
(List of Parameters value O_O)

ParameterAnInputToAMethod.inverseOfParameters

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

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

Class.methods

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

Class.methods

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

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
RALDataTypes

SimpleDataTypeSubtpeOfDataType

SimpleDataTypeSubtpeOfDataTypes
RALSimpleDataTypeSubtpeOfDataTypes

(Class value O_O)

Class.inverseOfCoreClass

ComplexDataType

ComplexDataTypes
RALComplexDataTypes

(AggregatingOperator value O_O)

(List of DataTypes value O_O)

AggregatingOperator

AggregatingOperators
RALAggregatingOperators

(Code value O_O)

```
SetOf
ListOf
Mapping
```

(Integer value O_O)

(Template value O_O)

Low level Data Types

insert Camel Case.md

Emoji

Emojis
RALEmojis

String

Strings
RALStrings

CamelName

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

CamelNames
RALCamelNames
String
UpperCamel , LowerCamel

(String value O_O)

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

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

UpperCamel
a CamelName that begins with a capital letter

_ "Customer", "ProductCategory", "PaymentMethod"
content begins with an upper case letter.
UpperCamels
RALUpperCamels
CamelName

LowerCamel
a CamelName that begins with a lower case letter

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

QualifiedCamel
an expression consisting of Camel Names separated by periods

QualifiedCamels
RALQualifiedCamels
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
RALValueTypeRichTexts
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
RALOneLiners
RichText

the string content
(String value O_O)

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

PrimitiveType
A basic, built-in data type

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

String

Strings
RALStrings
PrimitiveType
CamelName , QualifiedCamel , ValueTypeRichText

Integer

Integers
RALIntegers
PrimitiveType

Decimal

Decimals
RALDecimals
PrimitiveType

Boolean

Booleans
RALBooleans
PrimitiveType

Date

Dates
RALDates
PrimitiveType

Time

Times
RALTimes
PrimitiveType

DateTime

DateTimes
RALDateTimes
PrimitiveType

Annotation Types Used

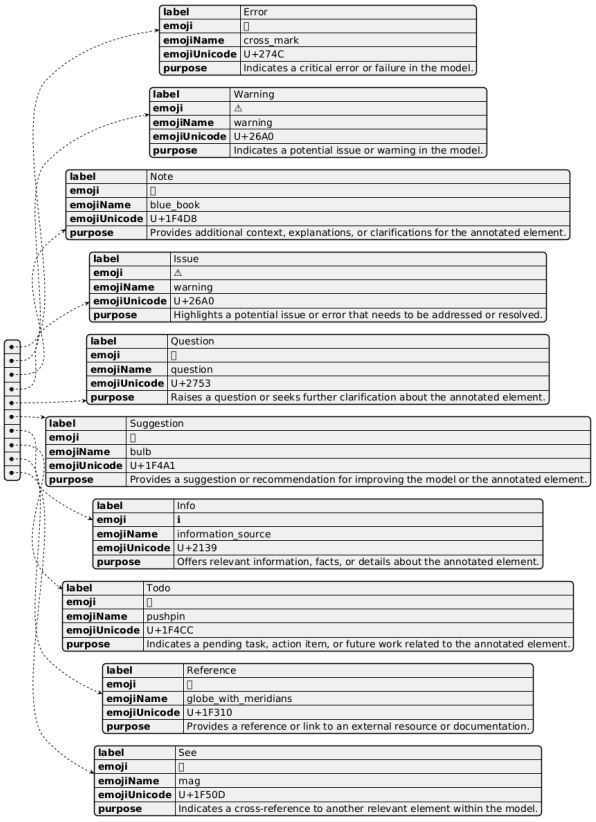
These are the recognized Annotation Types for the LDM model.

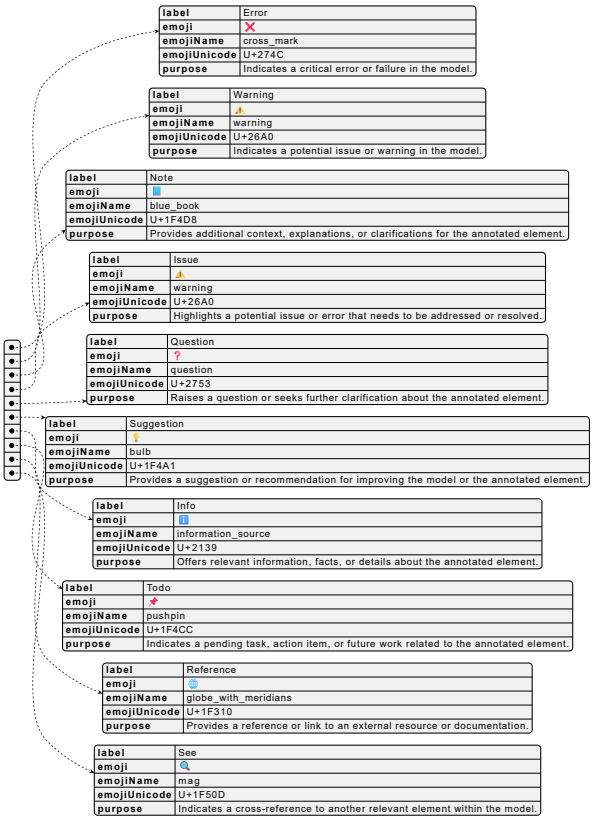
And this is how you register the AnnotationTyped for a model. By including this sort of array in the DSL document for the model.

```
@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": "ℹ️",
"emojiName": "information_source",
"emojiUnicode": "U+2139",
"purpose": "Offers relevant information, facts, or details about the annotated element."
},
{
"label": "Todo",
"emoji": "📌",
"emojiName": "pushpin",
"emojiUnicode": "U+1F4CC",
"purpose": "Indicates a pending task, action item, or future work related to the annotated
```

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


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

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,ℹ️,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	ℹ️	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