

# Sheet Subsystem

This subsystem considers the placement of graphics, title blocks and text fields on the sheet underlying a model diagram. This makes it possible to display metadata such as title, copyright, date, version etc. All of this is in the Sheet Subsystem since the placement and sizing of these elements typically varies by sheet size.

Relationship numbering range: R300-R350

# Class Descriptions

# Box

A bounded rectangle forming part of a Title Block Pattern.

## Attributes

### ID

Arbitrary identifying value, local to a Title Block Pattern.

Type: Based on Nominal

## Identifiers

### 1. ID + Pattern

ID value is unique within each Pattern

# Box Placement

When a Title Block Pattern is scaled and then positioned within a Frame it becomes possible to compute the placement of each of its Boxes.

## Attributes

### Placement

The location of the lower left corner of the Box in Canvas coordinates.

Type: Position

### Box size

The height and width of the Box

Type: Rect Size

## Identifiers

1. **Frame + Sheet + Box + Pattern**

Many to many identifier

# Boxed Field

A Boxed Field is placed inside a Data Box within a Title Block Pattern.

## Attributes

### Order

When there is more than one unit of Metadata present in the same Data Box, the items are stacked vertically. If there is only one Boxed Field in a Data Box, that field has an order value of 1. If there is more than one item, the bottommost item is 1 with ascending values going vertically.

## Identifiers

### 1. Metadata + Frame + Sheet

Super class reference

## Boxed Text Line

Each unit of Metadata is displayed as a single line of text in a Data Box. These lines are ordered vertically if there is more than one Metadata item sharing the same Data Box.

So a Boxed Text Line is the entry of Metadata content in some line number of a Data Box.

### Identifiers

1. **Metadata + Box + Title block pattern**

A Metadata item can appear at most once in a Data Box

2. **Box + Title block pattern + Order**

The numbering of text lines is local to each Data Box

### Attributes

#### Order

When there is more than one unit of Metadata present in the same Data Box, the items are stacked vertically. If there is only one Boxed Field in a Data Box, that field has an order value of **1**. If there is more than one item, the bottommost item is **1** with ascending values going vertically.

# Compartment Box

A Compartment Box is split in two with a Partition yielding two Partitioned Boxes. So a Compartment Box is a rectangular border wrapping two internal Boxes.

## Attributes

(No non-referential attributes)

## Identifiers

1. **ID + Pattern**

Forms the union of the subclass identifier values

# Data Box

A Box that is not further partitioned, but is instead intended to wrap presented Meta Data is a Data Box.

## Attributes

### Alignment

The vertical and horizontal alignment of text content displayed in this Data Box.

Type: VH Align :: ( [ T | C | B ], [ L | C | R ] )

## Identifiers

### 1. ID + Pattern

Refers to each superclass identifier



# Envelope Box

A Title Block Pattern is wrapped by a single Envelope Box. So it constitutes the outer boundary of the entire Title Block Pattern.

## Attributes

(No non-referential attributes)

## Identifiers

1. **ID + Pattern**

Refers to each superclass identifier

## Field

A location within a Frame where an item of Metadata is placed.

## Attributes

### Type

The location in Canvas coordinates of the lower left corner where the Meta Data will be rendered.

Type: Open Box :: [ Open | Block ]

### Max area

The maximum rectangular dimension to be used for rendering the Metadata.

Type: Rect Size

## Identifiers

### 1. Metadata + Frame + Sheet + Type

Since the same unit of Metadata could be displayed both in a Boxed and an Open Field, we need the **Type** attribute to distinguish the two cases.

# Field Content

This is data supplied with a Canvas that is entered into a Field.

## Attributes

### Value

This is the data supplied with the Canvas that goes into the Field.

Type: Text

## Identifiers

1. **Metadata + Location + Frame + Sheet**

Identifier is taken from the many side of the association

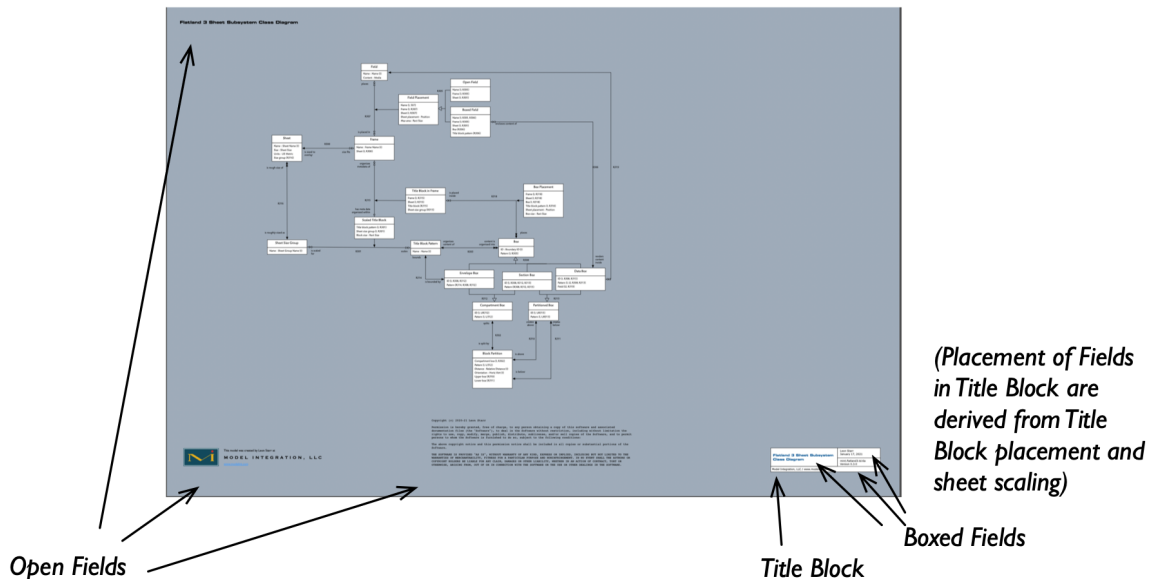
# Frame

A Frame makes it possible to ‘frame’ a diagram with important metadata such as authors, dates, revision numbers, copyright notices and organization logos. This data is displayed in one or more fields positioned on a given sheet (size) and orientation as shown:

*A Frame specifies the inclusion and placement of an optional Title Block and Fields*

Name: Model Integration Open Source

Sheet: D



A Frame represents a pattern of Fields and/or a Title Block Pattern on the surface area defined by a Sheet. The lower left corner placements of each Frame element (Field or Scaled Title Block) are customized to fit the dimensions of a given Sheet and orientation.

## Attributes

### Name

Describes the usage or organization affiliation, Model Integration Open Source or Model Integration Proprietary, as examples.

Type: Frame Name based on system type Name

### Orientation

A Sheet positioned such that it is wider than it is tall (landscape) or taller than it is wide (portrait)

Type: Portrait Landscape

## Identifiers

### 1. Name + Sheet + Orientation

Since a Frame is custom fit to a Sheet and its orientation we will refer to a Frame using the Sheet Name, D Landscape–Model Integration Open Source, for example.

# Frame Metadata

Metadata that is displayed in either an Open Field or a Block Field or both.

## Identifier

1. **Metadata + Frame + Sheet**

## Attributes

(No non-referential attributes)

# Metacontent

The user supplies a value for each relevant Metadata item as Metacontent. If you think of Metadata as a variable, then Metacontent is the value assigned to that variable.

## Identifiers

1. Metadata

## Attributes

No non-referential attributes

# Metadata

Information that can be displayed in the Frame of a model diagram other than the model itself.

## Attributes

### Name

A name that reflects the content to be inserted in a Frame such as `Modification date`, `Author`, `Organization Logo`.

Type: Field Name based on Name system type.

### Media

They kind of data to be inserted in the Field. Since the underlying graphics library will handle various image formats, we need not specify them here. We really just need to know what rendering facilities to use.

Type: [ `text` | `image` ]

## Identifiers

1. Name



# Open Field

An Open Field is not part of a Title Block and can be placed anywhere in a Frame.

## Identifiers

1. **Metadata + Frame + Sheet**

Refers to Superclass identifier

## Attributes

### Placement

The location, in Canvas Coordinates, within the Frame of the lower left corner of the Open Field.

Type: Position

# Partition

A Compartment Box is divided horizontally or vertically to yield two Partitioned Boxes.

## Attributes

### Distance

A fraction of the distance moving upward or rightward. For example, a Compartment Box split at **0.33** yields two Partitioned Boxes with the lower being one third the area of the upper Box. So the distance is always measured away from the lowest boundary coordinate value either vertically or horizontally.

### Orientation

This is the direction of the partition axis.

Type: [ `horizontal` | `vertical` ]

## Identifiers

### 1. ID + Pattern

Forms the union of the subclass identifier values

# Partitioned Box

A Partitioned Box results from partitioning a Compartment Box as either the upper (rightmost or uppermost) box or lower (leftmost or lowermost) Box.

## Attributes

(No non-referential attributes)

## Identifiers

1. **ID + Pattern**

Forms the union of the subclass identifier values

# Resource

The name of an image that can be located somewhere.

## Identifiers

1. Metadata

## Attributes

### Name

A descriptive name to be supplied with the diagram semantics that can be mapped to a file somewhere.

Type: Text

### Location

A resource locator such as a file path or URL. (Currently only file name's are supported)

Type: Resource Locator (file path, for now)

# Scaled Title Block

The dimensions of a Title Block and all of the Boxes it contains can be determined for a given Sheet Size Group. A Title Block Pattern is defined as a set of relative Box sizes. The actual Box sizes are determined by ratios relative to the Envelope Box. The size of the Envelope Box is determined by choosing a size that works for a given Sheet Size Group and saving it as a Scaled Title Block.

## Attributes

### Size

The height and width of the Envelope Box for the given Sheet Size Group.

Type: Rect Size

### Margin

The distance from the horizontal and vertical box boundary to any text content that must remain clear for all boxes in the Title Block Pattern (for the current scaling). This spacing is necessary for aesthetics and readability of the text content.

Type: HV Spacing :: (H::distance, V::distance)

## Identifiers

1. Title block pattern + Sheet size group

## Section Box

A rectangle inside the Envelope that is further partitioned is a Section Box.

### **Attributes**

(No non-referential attributes)

### **Identifiers**

#### 1. ID + Pattern

Refers to each superclass identifier

# Sheet

A pre-defined size representing the total rectangular surface area available for drawing is considered a Sheet. In the engineering world standard sheet sizes are named such as A, B, C D and E in the US or A0, A1, etc internationally. Each size constitutes a possible instance of Sheet. Many drawing programs provide the feature of an arbitrary sized sheet that expands as you draw. Flatland, however, encourages, but does not enforce, adherence to a standard sheet size so that you have the option of creating a properly scaled drawing or plot on actual paper if you have the printing equipment. Flatland also encourages you to divide complex domains into appropriately sized subsystems rather than just creating one monolithic tangle of model elements. That said, any size of Sheet can be pre-defined and made available for drawing in Flatland.

## Attributes

### Name

A US name like letter, tabloid, B, C, D or international name like A4, A3, etc. Or even a user defined name like 'extra wide'

Type: Sheet Name based on the Name system type

### Size

Type: Sheet Size, a height and width value as unit-less positive rational numbers

### Units

The units are specified so that the Size values can be converted into the correct internal drawing units (probably points).

Type: [ US | Metric ]

## Identifiers

1. Name

## Sheet Size Group

It is not really necessary to customize a title block for each sheet size. A title block sized to fit in a Tabloid (11inx17in) Sheet is equally adequate to an A3 Sheet. So it is convenient to group these two sizes for the purpose of determining which Scaled Title Block Pattern to use. It might be possible to work out a scaling formula that would perfectly size a Title Block Pattern given Sheet dimensions, but this formula is not obvious. It certainly is not a linear scale. (If you take the Title Block Pattern to Sheet size ration for Letter, and scale that up for an A1 Sheet, it doesn't look right). Rather than try to devise a formula, we can just scale each Title Block Pattern to fit a given Sheet Size Group which will be less work than doing the same for every Sheet size, US and International.

### Attributes

#### Name

A descriptive name like 'Dish' or 'LetterA4'. Or maybe use a 'Tshirt size' system like 'small', 'medium', etc.

Type: Sheet Group Name based on the system Name data type

### Identifiers

1. Name



# Text content

Content such as an author name, date, document title and so forth is defined as text and rendered into any Field that designates the associated Metadata item.

## Identifiers

1. Metadata

## Attributes

### Value

The actual text to be displayed on the diagram wherever the Metadata is referenced.

Type: Text

# Title Block Pattern

This is typically a rectangle partitioned into multiple internal rectangles where fields like “Title”, “Approved by:”, “Version” and so forth appear as in a standard engineering diagram.

## Attributes

### Name

A descriptive name.

Type: Name

## Identifiers

1. Name

# Title Block Placement

Whereas the sizes of each Box can be worked out knowing the Sheet Size Group, the actual coordinates of each Box lower left corner is unique to each Frame. That's because the Frame is custom fit to its Sheet.

## Attributes

(No non-referential attributes)

## Identifiers

1. **Frame + Sheet**

Identifier formed from the association class many-side (Frame)

# Relationship Descriptions

## **OR304 / Ordinal**

### stacking order

Metadata items are stacked vertically within the Data Box with the lowest ordinal value in the highest vertical position.

### Formalization

Boxed Field.Order

## **R300 / 1:Mc**

**Frame** is sized to overlay *one Sheet*

**Sheet** size fits *zero or many Frame*

The dimensions and Fields of a Frame are adjusted to perfectly fit a particular Sheet (sheet size). A D-Model Architect frame, for example will only fit on a D Sheet.

A variety of Frames can be designed for any given Sheet. If no Frames are defined for a given Sheet, no title blocks or Fields will be drawn when a diagram is generated when that Sheet is selected by the user.

### Formalization

Frame.Sheet -> Sheet.Name

## R301 / Mc:Mc

**Title Block Pattern** is scaled to look good in *zero, one or many* **Sheet Size Group**

**Sheet Size Group** scales for appearance *zero, one or many* **Title Block Pattern**

A given Title Block Pattern can be scaled to look nicely (not take up too much space and still display contents legibly) in the Sheet sizes defined by a Sheet Size Group. For each Sheet Size Group where the Title Block Pattern might be used, it would have a different scale and all of its internal Boxes would size differently.

It is possible to define a Title Block Pattern without necessarily scaling it, though this would be unusual as it would get used. But there may be some benefit in defining a Title Block Pattern in advance of employing it.

Similarly a Sheet Size Group may exist that doesn't support any Title Block Pattern. This could be the case in a project where title blocks are not used at all.

### Formalization

Scaled Title Block.Title block pattern -> Title Block Pattern.Name, Scaled Title Block.Sheet size group -> Sheet Size Group.Name

## R302 / 1:1

**Partition** splits *one* **Compartment Box**

**Compartment Box** is split by *one* **Partition**

By definition, a Compartment Box is split by a single Partition to yield an upper and a lower Partitioned Box. By 'upper' and 'lower' we refer to ascending values along the cartesian axes. Upper refers to 'further right' if the Partition is vertical or 'further up' if the Partition is horizontal. In other words, 'up' means a higher coordinate value along the x or y axis.

### Formalization

Partition.Compartment box -> Compartment Box.ID

Partition.Pattern -> Compartment Box.Pattern

## R303 / 1:M

**Title Block Pattern** defines a nested rectangular hierarchy of *one or many* **Box**

**Box** is rectangle in nested hierarchy defined by *one* **Title Block Pattern**

A Title Block Pattern is divides an Envelope Box into multiple sub-rectangles which may be further partitioned to form a number of Data Boxes where Meta Data can be presented. At a minimum, a Title Block consists of an Envelope Box split in two to form two Data Boxes. Anything less would constitute a simple Open Field.

## Formalization

Box.Pattern -> Title Block Pattern.Name

## R305 / Generalization

**Field** is an **Openor Boxed Field**

Each Field is either embedded within a Title Block Pattern or scattered by itself somewhere out on the Canvas.

## Formalization

<subclass>.Metadata -> Field.Metadata

<subclass>.Frame -> Field.Frame

<subclass>.Sheet -> Field.Sheet

<subclass>.Orientation -> Field.Orientation

## R306 / 1:Mc

**Boxed Field** is enclosed as text block within *one* **Data Box**

**Data Box** encloses in text block *zero or many* **Boxed Field**

Each Box Field within a Data Box corresponds to distinct line in a multiline block of text. The lower left corner of this text box is determined by the Scaled Title Block spacing applied to the Data Box's Box Placement.

Since a Data Box is just a position within a Title Block Pattern, it can be referenced each time its Title Block Pattern is applied to a Frame.

## Formalization

Boxed Field.Box -> Data Box.ID

Boxed Field.Title block pattern -> Data Box.Title box pattern

## R307 / Mc:Mc-1

**Metadata** is displayed in *zero, one or many* **Frame**

**Frame** displays *zero, one or many* **Metadata**

When a Frame is specified Fields may be defined at various locations in the Frame, in Canvas Coordinates, where the associated Meta Data will be rendered.

In some cases the same piece of Metadata, Title, let's say, might be presented in more than one location for easy reference. Consequently, we cannot use the Meta Data name exclusively as the name of the field.

Fields are not required since a Frame may be empty or specify just a single Scaled Title Block.

Metadata may be defined which is never used, though this should be a rare occurrence.

### Formalization

Frame Metadata.Name -> Metadata.Name

Frame Metadata.Frame -> Frame.Name

Frame Metadata.Sheet -> Frame.Sheet

Frame Metadata.Orientation -> Frame.Orientation

## R308 / Generalization

**Box** is an **Envelope**, **Section** or **Data Box**

A Title Block Pattern is divided up into a number of sub-rectangles. The outermost rectangle is the Envelope. A partition of this Box results in two internal Boxes. If neither is partitioned further, each becomes a Data Box where Meta Data can be presented. But if one of the boxes is further subdivided via a Partition, that enclosing Box is a Section Box which simply encloses two other Boxes either or both of which may be a Section Box or a Data Box. Sooner or later we are left with nothing but Data Boxes and the Title Block Pattern is complete. Envisioned as a binary tree, we can think of the Envelope Box as the root, intermediate nodes as Section Boxes and the leaves as Data Boxes.

### Formalization

<subclass>.ID -> Box.ID

<subclass>.Pattern -> Box.Pattern

## R309 / 1:1c

**Metacontent** is provided for *one* **Metadata**

**Metadata** has provided *zero or one* **Metacontent**

The user supplies content to be filled in for any reference to a given Metadata item.

Only one value may be supplied for each item of Metadata. If no value is supplied, any associated Metadata Field will be rendered blank.

### Formalization

Metacontent.Metadata -> Metadata.Name

## R310 / 1:1

**Partition** creates above *one* **Partitioned Box**

**Partitioned Box** is above *one* **Partition**

A Partition will always yield a rightmost or uppermost Partitioned Box.

## Formalization

Partition.Upper box -> Partitioned Box.ID

Partition.Pattern -> Partitioned Box.Pattern

## R311 / 1:1

**Partition** creates below *one* **Partitioned Box**

**Partitioned Box** is below *one* **Partition**

A Partition will always yield a leftmost or lowermost Partitioned Box.

## Formalization

Partition.Lower box -> Partitioned Box.ID

Partition.Pattern -> Partitioned Box.Pattern

## R312 / Generalization

**Compartment Box** is an **Envelope** or **Section Box**

Both an Envelope and a Section Box are, by definition, partitioned.

## Formalization

<subclass>.ID -> Compartment Box.ID

<subclass>.Pattern -> Compartment Box.Pattern

The union of <subclass> ID + <subclass> Pattern values forms the domain of the Compartment Box.ID + Compartment Box.Pattern values.

## R313 / Generalization

**Partitioned Box** is a **Section** or **Data Box**

A Box resulting from a Partition is either another Compartment Box, which is a Section Box or it is not further partitioned in which case it is a Data Box.

## Formalization

<subclass>.ID -> Partitioned Box.ID

<subclass>.Pattern -> Partitioned Box.Pattern

The union of <subclass> ID + <subclass> Pattern values forms the domain of the Partitioned Box.ID + Partitioned Box.Pattern values.



## R314 / 1:1

**Title Block Pattern** is bounded by *one* **Envelope Box**

**Envelope Box** bounds *one* **Title Block Pattern**

The rectangle that completely surrounds a Title Block is the Envelope Box.

### Formalization

Envelope Box.Pattern -> Title Block Pattern.Name

## R315 / 1c:Mc

**Frame** places *zero or one* **Scaled Title Block**

**Scaled Title Block** is placed in *zero, one or many* **Frame**

A Scaled Title Block is positioned within a Frame by creating a Title Block Placement where a lower left corner is specified.

A Frame can position at most one title block, typically in the lower right corner of the Canvas.

A Title Block Pattern, if it is used, will likely have a different position in each Frame.

### Formalization

Title Block Placement.Frame -> Frame.Name

Title Block Placement.Sheet -> Frame.Sheet

(constrained to be a Sheet belonging to the Title Block Placement.Sheet size group)

Title Block Placement.Orientation -> Frame.Orientation

Title Block Placement.Title block pattern -> Scaled Title Block.Title block pattern

Title Block Placement.Sheet size group -> Scaled Title Block.Sheet size group

## R316 / 1:M

**Sheet** is roughly sized as *one* **Sheet Size Group**

**Sheet Size Group** roughly sizes *many* **Sheet**

Sheet Size Groups are used to determine the scaling for each available Title Block Pattern.

Roughly similar sizes such as Letter, A4 and Legal may be grouped together in the same Sheet Size Group since the same Title Block scale will work for all three sizes.

Since any Sheet must specify a scale to be used for any Title Block Patterns, each Sheet must be categorized in a Sheet Size Group.

## Formalization

Sheet.Size group -> Sheet Size Group.Name

### R317 / 1:1c

**Frame Metadata** appears in *zero or one* **Boxed Field**

**Boxed Field** shows *one* **Frame Metadata**

A unit of Metadata that is designated for display within a Frame might be rendered in a Box Field. (If not, it must be rendered in an Open Field).

A Boxed Field must be assigned a Metadata item, but that item may not have any content supplied, in which case the field will appear blank when rendered.

## Formalization

Open Field.Metadata -> Frame Metadata.Name

Open Field.Frame -> Frame Metadata.Frame

Open Field.Sheet -> Frame Metadata.Sheet

Open Field.Orientation -> Frame Metadata.Orientation

### R318 / M:Mc

**Title Block Placement** determines placement of *one or many* **Box**

**Box** placement is determined by *zero, one or many* **Title Block Placement**

Once a Title Block Pattern is scaled and positioned in a Frame, it is possible to derive the placement and size of each Box in the Title Block Pattern with respect to the drawing Canvas.

## Formalization

Box Placement.Frame -> Title Block Placement.Frame

Box Placement.Sheet -> Title Block Placement.Sheet

Box Placement.Sheet -> Title Block Placement.Orientation

Box Placement.Box -> Box.ID

Box Placement.Title block pattern -> Box.Pattern

### R319 / 1:1c

**Frame Metadata** appears in *zero or one* **Open Field**

**Open Field** shows *one* **Frame Metadata**

A unit of Metadata that is designated for display within a Frame might be rendered in an Open Field. (If not, it must be rendered in a Boxed Field).

An Open Field must be assigned a Metadata item, but that item may not have any content supplied, in which case the field will appear blank when rendered.

### Formalization

Open Field.Metadata -> Frame Metadata.Name

Open Field.Frame -> Frame Metadata.Frame

Open Field.Sheet -> Frame Metadata.Sheet

Open Field.Orientation -> Frame Metadata.Orientation

## R320 / 1:M

**Frame Metadata** must appear in *one or many* **Field**

**Field** must show *one* **Frame Metadata**

If an item of Metadata is included in a Frame, it must appear in at least one Field. At most, it can appear in two, one of each type, Boxed and Open. The reasoning is that there shouldn't be any need to duplicate frame data in multiple locations inside or outside of a title block. But it can be helpful to take some piece of data that appears in a title block and present it outside for emphasis. The specific case motivating this rule is the diagram **Title Metadata** item. Even though the title information could be included in the title block, it is often nice to show it in the upper left corner of the sheet also for quick reference.

### Formalization

Field.Metadata -> Frame Metadata.Name

Field.Frame -> Frame Metadata.Frame

Field.Sheet -> Frame Metadata.Sheet

Field.Orientation -> Frame Metadata.Orientation

## R321 / 1:Mc

**Boxed Field** is placed on *one* **Box Text Line**

**Box Text Line** places *zero, one or many* **Boxed Field**

.

### Formalization

Boxed Field.Metadata -> Boxed Text Line.Metadata

Boxed Field.Box -> Boxed Text Line.Box

(constrained such that Boxed Field.Metadata == /R321/Box Text Line.Metadata)

Boxed Field.Title block pattern -> Boxed Text Line.Title block pattern

## **R322 / Generalization**

**Metacontent** is **Text Content** or **Resource**

The data to be filled in for an item of Metadata can be plain text, such as an author name or a document title, or it can be a resource such as an image in an png file. (In fact, that's the only kind of resource currently supported).

### **Formalization**

<subclass>.Metadata -> Metacontent.Metadata