Decorator Subsystem

This subsystem describes the various adornments that can be placed on or in the vicinity of a connector stem. These include text labels and graphic symbol geometry.

Relationship numbering range: R100-R149

(Conflict with Binary Connector! Renumber as R200-R249 in next version)

Class Descriptions

Arrow Symbol

Describes a triangular geometry that can be used to define an arrow head.

Attributes

Name

Same as **Shape Element.Name**

Base

The width of the triangle base

Type: Distance

Height

The height of the triangle

Type: Distance

Stroke

The width and pattern of the border around the triangle

Type: Stroke Style

Fill

Defines the overall look of the Arrowhead as either a hollow arrow (border as a closed triangle), solid arrow (solid fill triangle) or open (v-shape with no base line drawn)

Type: Hollow_Solid_Open `:: [ hollow | solid | open ]`

Identifiers

Name

Unique across all Shape ELements

Circle Symbol

Describes a circular geometry. These appear on the initial and final transitions on state diagrams, for example.

Attributes

Name

Same as **Simple Symbol.Name**

Radius

The radius of the circle

Type: Distance

Solid

Whether or not the circle is filled

Type: Boolean

Identifiers

1. **Name**

Compound Symbol

As the name suggests, a Compound Symbol is built up from multiple Simple Symbols stacked together in some arrangement.

Attributes

Name

Same as **Symbol.Name**

Stroke

The border line width, pattern and color

Type: Stroke style

Identifiers

1. **Name**

Cross Symbol

Describes a line drawn at an angle to the Stem. These appear on Shlaer-Mellor superclass stems, for example.

Attributes

Name

Same as **Simple Symbol.Name**

Width

The length of the crossing line segment

Type: Distance

Angle

Angle relative to the Stem axis. 90 degrees yields a cross normal to the Stem.

Type: Degrees

Identifiers

1. **Name**

Decoration

Any notational element, graphical or textual that adorns the vicinity of a Stem is a Decoration.

Attributes

Name

In the case of Annotation the name is the same text that is drawn for the notation. So the name could be: `0..1` or `{ disjoint, complete }`

For a Symbol, the name is purely descriptive such as double solid arrow or small solid circle.

It is important not to use a model semantic name such as initial psuedo state since the symbol might be used with other notations with other meanings. So it is safest to stick to a description of appearance.

Type: Name

Size

When drawn, this is the total rectangular area consumed. This may be useful for detecting and avoiding overlapping drawn elements.

Type: Rect Size

Identifiers

1. **Name**

Decorations are named uniquely by policy.

Label

A fixed text annotation drawn next to a Stem. On a class diagram, these could be standard UML labels such as `1..*` or a tag like`{ disjoint, complete }`. These are not to be confused with variable text such as the name of an event on a state diagram or a relationship such as R33 on a class diagram.

Attributes

Name

Same as **Decoration.Name**. Since Labels are text, it is convenient to simply make the label content the name of the Label. For example, `0..1` serves as both the name and rendered content of a UML class diagram multiplicity label.

Identifiers

1. **Name**

Simple Symbol

A Simple Symbol is a graphical element that may form all or part of an entire Symbol. It is "simple" in the sense that it is an atomic geometric element.

Attributes

Name

Same as **Symbol.Name**

Stroke

The border line width, pattern and color

Type: Stroke style

Terminal offset

Distance from either end (root/vine) of a Stem. An arrow symbol can be drawn so that it touches the Stem end (0 distance) from a Node face in the case of a root Stem end. Or a cross used in a Shlaer-Mellor super-class might be drawn at some distance from the vine end.

This value is distinct from the **Stack Placement.Offset** which defines an offset between Simple Symbols that are stacked.

Identifiers

1. **Name**

Symbol

A geometric shape such as an arrow drawn at either end of a Stem is a Symbol.

Attributes

Name

Same as **Decoration.Name**. The name can refer to the visual appearance of the Symbol (wide hollow arrow) or to its general usage (gen arrow). Care should be taken to avoid model semantic names such as '1 multiplicity' since the same symbol might be useful for a variety of meanings in different contexts or Diagram Types. A solid arrow, for example, could be used to indicate a method invocation, a state transition or a unit of multiplicity.

Identifiers

1. **Name**

Symbol Stack Placement

This class represents both the inclusion and arrangement of a Simple Symbol in a Compound Symbol.

In the case of a double headed arrow, for example, one Arrow Symbol is drawn at the head of a Stem end and the other behind it. Ordering progresses either along the stem (adjacency), or upward toward the viewer on the z axis (layering).

In the case of an xUML final psuedo state, a solid arrow is drawn at the tip of the Stem end with a large circle after the tip. Then a small solid circle is drawn on top of a large unfilled circle.

Attributes

Position

The order proceeding from the Stem end outward and upward by layer.

Type: Ordinal

Compound symbol

Same as **Compound symbol.Name**

Simple symbol

Same as **Simple Symbol.Name**

Arrange

Whether the next Simple Symbol in the Position sequence will be layered or placed adjacent to this Simple Symbol. If this is the final Simple Symbol in the position sequence, it is simply marked as the last one.

Type: `[ adjacent | layer | last ]`

Offset

This is the distance in the opposite direction of the Stem end relative to the next Simple Symbol, if any, in the position order. For example, one Arrow might be spaced at a certain distance from the adjacent Arrow. Or a circle might be off center when layered on top of another circle. A double hatched cross would have a certain amount of space between the two cross line segments along the Stem axis.

Type: Distance

Identifiers

1. **Position** + **Compound symbol**

Each position within a Compound Symbol is unique. We define positions 1 and 2 within the 'double solid arrow' Compound Symbol, for example, where each simple 'solid arrow' is rendered.

Relationship Descriptions

R100 / Generalization

Simple Symbol is an **Arrow**, **Circle** or **Cross Symbol**

All of the Stem symbols in supported notations can be created by an arrangement of these Simple Symbol subclasses. Any notation that requires a stem end shape that cannot be created with these elements, will need to extend this generalization (or some subclass) to include the desired element geometry.

Formalization

Name referential attribute in each subclass

R101 / M:Mc-M

Compound Symbol stacks *one or many* **Simple Symbol**

Simple Symbol is stacked in *zero, one or many* **Compound Symbol**

A Simple Symbol is drawn relative to another (or the same) Simple Symbol to form all or part of a Compound Symbol. For example, in a double arrowhead configuration, one arrow is drawn adjacent to the other on a Decorated Stem End. Or, in the case of an xUML final pseudo state, a small solid circle is drawn on top of a larger hollow circle.

The same Simple Symbol can be useful in multiple Compound Symbols. A solid arrow, for example, is useful in both a single and double arrow configuration.

The same Simple Symbol may be used more than once in a Compound Symbol where each usage corresponds to a unique Stack Placement. For example, a solid arrow appears in both the first and second Stack Placement Positions of a double solid arrowhead.

Without any Simple Symbol elements, there would be nothing to draw, so a Compound Symbol requires at least one with at least two Stack Placement positions.

Formalization

Stack Placement association class

R102 / Ordinal

Stack Placement draw order

In a Compound Symbol, each constituent Simple Symbol is drawn in a specific relative location. This order corresponds to a progression along the stem axis or a z-axis toward the viewer.

See the **Stack Placement.Arrange** attribute description for an explanation of which axis applies for a given stacking direction.

Formalization

Position is ordered sequentially using an Ordinal value within each **Compound Shape**

R103 / Generalization

Symbol is a **Compound Symbol** or **Simple Symbol**

A Symbol is either made up of an arrangement of one or more Simple Symbols in various positions or it is just a single Simple Symbol in one position.

Formalization

Name referential attribute in each subclass

R104 / Generalization

Decoration is a **Label** or **Symbol**

In all cases, a Symbol is just a name associated with an icon that can be drawn on the end of a Stem.

There are only two ways to designate the meaning of a Stem end. In some notations, such as Starr class diagramming, hollow and solid arrows indicate multiplicity and conditionality of class model associations. In xUML notation, text labels are used instead. With Shlaer-Mellor notation, arrows are used to indicate multiplicity while a text `c` symbol (conditional) indicates when zero is an option.

Formalization

Name referential attribute in each subclass

