**7. CMMN Diagram interchange (CMMN DI)**

**7.1 Scope**

This chapter specifies the meta-model and schema for **CMMN 1.1** Diagram Interchange (**CMMN DI**). The **CMMN DI** is meant to facilitate interchange of **CMMN** diagrams between tools rather than being used for internal diagram representation by the tools. The simplest interchange approach to ensure the unambiguous rendering of a **CMMN** diagram was chosen for **CMMN DI**. As such, **CMMN DI** does not aim to preserve or interchange any “tool smarts” between the source and target tools (e.g., layout smarts, efficient styling, etc.).

**CMMN DI** does not ascertain that the **CMMN** diagram is syntactically or semantically correct.

**7.2 Diagram Definition and Interchange**

The **CMMN DI** meta-model, similar to the CMMN abstract syntax meta-model, is defined as a MOF-based meta-model. As such, its instances can be serialized and interchanged using XMI. CMMN DI is also defined by an XML schema. Thus its instances can also be serialized and interchanged using XML.

Both, CMMN DI meta-model and schema are harmonized with the OMG Diagram Definition (DD) standard version 1.1. The referenced DD contains two main parts: the Diagram Commons (DC) and the Diagram Interchange (DI). The DC defines common types like bounds and points, while the DI provides a framework for defining domain specific diagram models. As a domain specific DI, CMMN DI defines a few new meta-model classes that derive from the abstract classes from DI.

The focus of CMMN DI is the interchange of laid out shapes and edges that constitute a CMMN diagram. Each shape and edge references a particular CMMN model element. The referenced CMMN model elements are all part of the actual CMMN model. As such, CMMN DI is meant to only contain information that is neither present, nor derivable, from the CMMN model whenever possible. Simply put, to render a CMMN diagram both the CMMN DI instance(s) and the referenced CMMN model are REQUIRED.

From the CMMN DI perspective, a CMMN diagram is a particular snapshot of a CMMN model at a certain point in time. Multiple CMMN diagrams can be exchanged referencing model elements from the same CMMN model. Each diagram may provide an incomplete or partial depiction of the content of the CMMN model. As described in Chapter 9, a CMMN model package consists of one or more files. Each file may contain any number of CMMN diagrams. The exporting tool is free to decide how many diagrams are exported and the importing tool is free to decide if and how to present the contained diagrams to the user.

**7.3 How to read this chapter**

Section 7.4 describes in details the meta-model used to keep the layout and the look of CMMN Diagrams. Section 7.5 presents in tables a library of the CMMN element depictions and an unambiguous resolution between a referenced CMMN model element and its depiction.

**7.4 CMMN Diagram Interchange Meta-Model**

**7.4.1 Overview**

The CMMN DI is an instance of the OMG DI meta-model. The basic concept of CMMN DI, as with DI in general, is that serializing a diagram [CMMNDiagram] for interchange requires the specification of a collection of shapes [CMMNShape] and edges [CMMNEdge].

The CMMN DI classes only define the visual properties used for depiction. All other properties that are REQUIRED for the unambiguous depiction of the CMMN element are derived from the referenced CMMN element [cmmnElementRef].

CMMN diagrams may be an incomplete or partial depiction of the content of the CMMN model. Some CMMN elements from a CMMN model may not be present in any of the diagram instances being interchanged.

Multiple depictions of a specific CMMN element in a single diagram is NOT allowed (except for OnPart relations that can be displayed once for each entry/exit criterion using their sentry.) Thus, it is not allowed to depict a PlanItem twice in the same diagram, but it is allowed to depict the same PlanItem in two different diagrams.

CMMN DI does not directly provide for any containment concept. The CMMNDiagram is an ordered collection of mixed CMMNShape(s) and CMMNEdge(s). The order of the CMMNShape(s) and CMMNEdge(s) inside a CMMNDiagram determines their Z-order (i.e., what is in front of what). CMMNShape(s) and CMMNEdge(s) that are meant to be depicted “on top” of other CMMNShape(s) and CMMNEdge(s) MUST appear after them in the CMMNDiagram. Thus, the exporting tool MUST order all CMMNShape(s) and CMMNEdge(s) such that the desired depiction can be rendered.

**7.4.2 Measurement Unit**

As per OMG DD, all coordinates and lengths defined by CMMN DI are assumed to be in user units, except when specified otherwise. A user unit is a value in the user coordinate system, which initially (before any transformation is applied) aligns with the device’s coordinate system (for example, a pixel grid of a display). A user unit, therefore, represents a logical rather than physical measurement unit. Since some applications might specify a physical dimension for a diagram as well (mainly for printing purposes), a mapping from a user unit to a physical unit can be specified as a diagram’s resolution. Inch is chosen in this specification to avoid variability but tools can easily convert from/to other preferred physical units. Resolution specifies how many user units fit within one physical unit (for example, a resolution of 300 specifies that 300 user units fit within 1 inch on the device).

**7.4.3 CMMNDI [Class]**

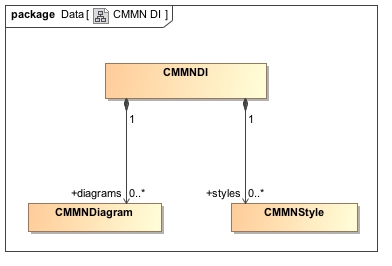


Figure 7. CMMNDI

The class CMMNDI is a container for the shared CMMNStyle and all the CMMNDiagram defined in a Definitions.

Table 7. CMMNDI attributes

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| styles:CMMNStyle [0..\*] | A list of shared CMMNStyle that can be referred by all CMMNDiagram and CMMNDiagramElement. |
| diagrams:CMMNDiagram [0..\*] | A list of CMMNDiagram. |

**7.4.4 CMMNDiagram [Class]**

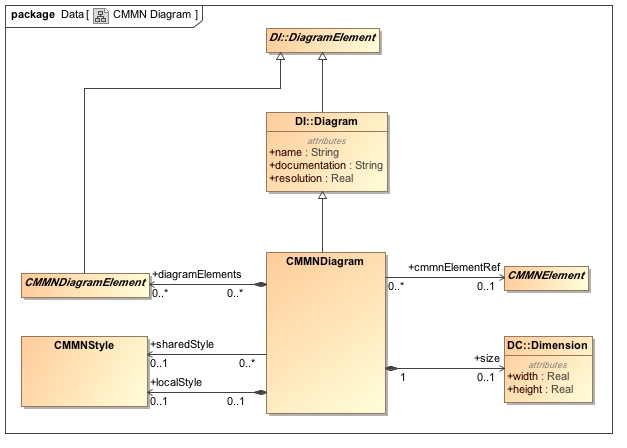


Figure 7. CMMNDiagram

The class CMMNDiagram specializes DI::Diagram. It is a kind of Diagram that represents a depiction of all or part of a CMMN model.

CMMNDiagram is the container of CMMNDiagramElement (CMMNShape(s) and CMMNEdge(s)). CMMNDiagram cannot include other CMMNDiagram.

A CMMNDiagram can define a CMMNStyle locally and/or it can refer to a shared one defined in the CMMNDI. Properties defined in the local style overrides the one in the referred shared style. That combined style (shared and local) is the default style for all the CMMNDiagramElement contained in this CMMNDiagram.

The CMMNDiagram class represents a two dimensional surface with an origin of (0, 0) at the top left corner. This means that the x and y axes have increasing coordinates to the right and bottom. Only positive coordinates are allowed for diagram elements that are nested in a CMMNDiagram.

The CMMNDiagram has the following attributes.

Table 7. CMMNDiagram attributes

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| name: String | The name of the diagram. Default is empty String. |
| documentation:String | The documentation of the diagram. Default is empty String. |
| resolution:Real | The resolution of the diagram expressed in user units per inch. Default is 300 |
| cmmnElementRef:CMMNElement [0..1] | A reference to either a Definitions, a CasePlanModel, a Stage, or a PlanFragment. |
| diagramElements:CMMNDiagramElement [0..\*] | A list of CMMNDiagramElement (CMMNShape and CMMNEdge) that are depicted in this diagram. |
| sharedStyle:CMMNStyle[0..1] | A reference to a CMMNStyle defined in the CMMNDI that serves as the default styling of the CMMNDiagramElement in this CMMNDiagram. |
| localStyle:CMMNStyle [0..1] | A CMMNStyle that defines the default styling for this diagram. Properties defined in that style overrides the one in the sharedStyle. |
| size:DC::Dimension [0..1] | The size of this diagram. If not specified, the CMMNDiagram is unbounded. |

**7.4.5 CMMNDiagramElement [Class]**

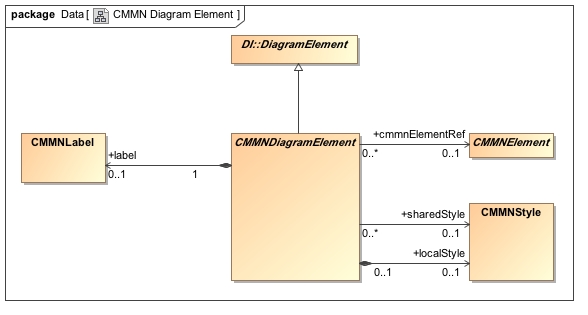


Figure 7. CMMNDiagramElement

The CMMNDiagramElement class is contained by the CMMNDiagram and is the base class for CMMNShape and CMMNEdge.

CMMNDiagramElement inherits its styling from its parent CMMNDiagram. In addition, it can refer one of the shared CMMNStyle defined in the CMMNDI and/or it can define a local style. See section 7.4.9 for more details on styling.

CMMNDiagramElement MAY also contain a CMMNLabel when it has a visible text label. If no CMMNLabel is defined, the CMMNDiagramElement should be depicted without a label.

CMMNDiagramElement has the following attributes:

Table 7. CMMNDiagramElement

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| cmmnElementRef:CMMNElement [0..1] | A reference to the CMMNElement that is being depicted. |
| sharedStyle:CMMNStyle[0..1] | A reference to a CMMNStyle defined in the CMMNDI. |
| localStyle:CMMNStyle [0..1] | A CMMNStyle that defines the styling for this element. |
| label:CMMNLabel [0..1] | An optional label when this CMMNElement has visible text label. |

**7.4.6 CMMNShape [Class]**

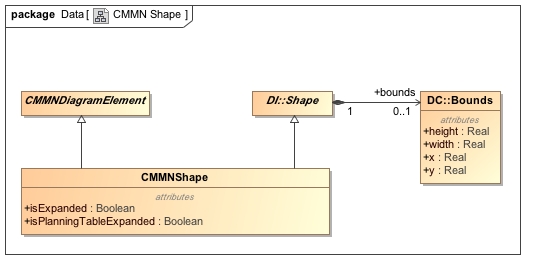


Figure 7.2 CMMN Shape

The CMMNShape class specializes DI::Shape and CMMNDiagramElement. It is a kind of Shape that depicts a CMMNElement from the CMMN model.

CMMNShape represents a CasePlanModel, a PlanItem, a DiscretionaryItem, an EntryCriterion, an ExitCriterion or a CaseFileItem that is depicted on the diagram.

CMMNShape has two additional properties (isCollapsed and isPlanningTableCollapsed) that are used to further specify the appearance of some shapes that cannot be deduced from the CMMN model.

CMMNShape extends DI::Shape and CMMNDiagramElement and has the following attributes:

Table 7. CMMNShape attributes

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| bounds:DC::Bounds [1] | The Bounds of the shape relative to the origin of its parent CMNNDiagram. The Bounds MUST be specified. |
| cmmnElementRef:CMMNElement[1] | A reference to a CasePlanModel, a PlanItem, a DiscretionaryItem, an EntryCriterion, an ExitCriterion or a CaseFileItem MUST be specified. |
| isCollapsed:Boolean[0..1] | If the CMMNShape refers to a PlanItem or DiscretionnaryItem that refers to a Stage or to a DiscretionaryItem that refers to a PlanFragment, then this attribute is used to determine if the Stage or PlanFragment is depicted collapsed (true) or expanded (false). Default, when applicable, is false. |
| isPlanningTableCollapsed:Boolean[0..1] | When a CMMNShape depicts a CasePlanModel that has a Planning Table or when it depicts a PlanItem or a DiscretionaryItem referring to a Stage containing a PlanningTable or when it depicts a PlanItem or a DiscretionaryItem referring to a HumanTask containing a PlanningTable, then this attribute is used to determine if the PlanningTable is depicted collapsed (true) or expanded (false). Default, when applicable, is false. |

**7.4.7 CMMNEdge [Class]**

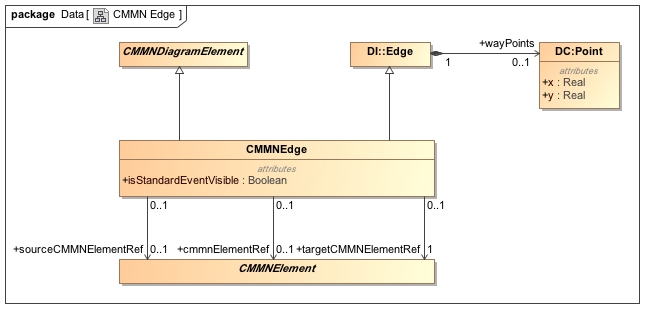


Figure 7.3 CMMN Edge

The CMMNEdge class specializes DI::Edge and CMMNDiagramElement. It is a kind of Edge that can depict a relationship between two CMMN model elements.

CMMNEdge are used to depict links in the CMMN model. As specified in chapter 6, a link is used to illustrate two things: a relation from a DiscretionaryItem to its HumanTask (Discretionary Association) or an OnPart relation (Connector).

When the CMMNEdge is used to depict a Discretionary Association, no element should be specified in the cmmnElementRef attribute of the CMMNEdge. In that particular case, the targetCMMNElementRef MUST be a DiscretionaryItem. The sourceCMMNElementRef MUST be a PlanItem or a Discretionary Item representing the HumanTask that holds the DiscretionaryItem referred by the targetCMMNElementRef.

When the CMMNEdge is used to depict an OnPart, the cmmnElementRef attribute of the CMMNEdge MUST be the id of that OnPart. In that particular case the sourceCMMNElementRef MUST NOT be specified since it can be obtained by the sourceRef of the OnPart or by the exitCriterionRef when specified in a PlanItemOnPart. The targetCMMNElementRef MUST be the id of one of the criterion (either an EntryCriterion or an ExitCriterion) that is linked to the Sentry holding the OnPart. An additional property (isStandardEventVisible) is used to determine if the StandardEvent should be depicted.

CMMNEdge extends DI::Edge and adds the following properties:

Table 7. CMMNEdge attributes

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| wayPoints:DC::Point[2..\*] | A list of points relative to the origin of its parent CMNNDiagram that specifies the connected line segments of the edge. At least two (2) waypoints MUST be specified. |
| cmmnElementRef:CMMNElement [0..1] | A reference to an OnPart when representing a connector.  MUST NOT be specified when representing a Discretionary Association. |
| sourceCMMNElementRef:CMMNElement [0..1] | MUST NOT be specified when cmmnElementRef is an OnPart  When used to depict a Discretionary Association, a reference to a PlanItem or to a DiscretionaryItem that represents a HumanTask. That HumanTask MUST have a planning table where the DiscretionaryItem used in the targetCMMNElementRef is defined. |
| targetCMMNElementRef[1] | MUST be the id of an EntryCriterion or an ExitCriterion when cmmnElementRef is an OnPart. That Criterion MUST refer to the Sentry holding the OnPart.  MUST be the id of a DiscretionaryItem when representing a Discretionary Association. |
| isStandardEventVisible:Boolean [0..1] | When cmmnElementRef is an OnPart, then this attribute is used to determine if the StandardEvent should be visible (true) or hidden (false) in the displayed label. Default, when applicable, is false. |

**7.4.8 CMMNLabel [Class]**

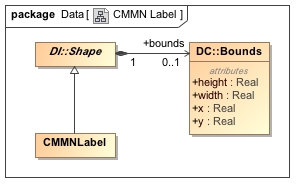


Figure 7.4 CMMN Label

CMMNLabel represents the depiction of some textual information about a CMMN element.

A CMMN label is not a top-level element but is always nested inside either a CMMNShape or a CMMNEdge. It does not have its own reference to a CMMN element but rather inherits that reference from its parent CMMNShape or CMMNEdge. The textual information depicted by the label is derived from the name attribute of the referenced CMMNElement.

CMMNLabel extends DI::Shape and has the following properties:

Table 7. CMMNLabel attribute

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| Bounds:Bounds[0..1] | The bounds of the CMMNLabel. When not specified, the label is positioned at its default position as determined in section 7.5 |

**7.4.9 CMMNStyle [Class]**

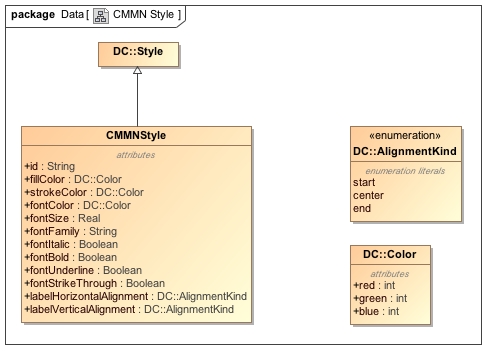


Figure 7. CMMNStyle

CMMNStyle specializes DC::Style. It is a kind of Style that provides appearance options for a CMMNDiagramElement.

CMMNStyle is used to keep some non-normative visual attributes such as colors and font. CMMN doesn’t give any semantic to color and font styling, but tools can decide to use them and interchange them.

CMMNDiagramElement style is calculated by percolating up CMMNStyle attributes defined at different level of the hierarchy. Each attribute is considered independently (meaning that a CMMNStyle attribute can be individually overloaded). The precedence rules are as follow:

* The CMMNStyle defined by the localStyle attribute of the CMMNDiagramElement
* The CMMNStyle referenced by the sharedStyle attribute of the CMMNDiagramElement
* The CMMNStyle defined by the localStyle attribute of the parent CMMNDiagram
* The CMMNStyle referenced by the sharedStyle attribute of the parent CMMNDiagram
* The default attribute value defined in Table 7.7 (CMMNStyle attributes).

For example, let’s say we have the following:

* CMMNDiagramElement has a local CMMNStyle that specifies the fillColor and strokeColor
* Its parent CMMNDiagram defines a local CMMNStyle that specifies the fillColor and fontColor

Then the resulting CMMNDiagramElement should use:

* The fillColor and strokeColor defined at the CMMNDiagramElement level (as they are defined locally).
* The fontColor defined at the CMMNDiagram level (as the fillColor was overloaded locally).
* All other CMMNStyle attributes would have their default values.

CMMNStyle extends DC::Style and has the following properties:

Table 7. CMMNStyle attributes

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| id:String [0..1] | A unique id for this style so it can be referenced. Only style defined in the CMMNDI can be referred by CMMNDiagramElement and CMMNDiagram. |
| fillColor:DC::Color [0..1] | The color use to fill the shape. Doesn’t apply to CMMNEdge. Default is white. |
| strokeColor: DC::Color [0..1] | The color use to draw the shape borders. Default is black. |
| fontColor:DC::Color [0..1] | The color use to write the label. Default is black. |
| fontFamily:String [0..1] | A comma separated list of Font Name that can be used to display the text. Default is Arial. |
| fontSize:Real [0..1] | The size in points of the font to use to display the text. Default is 8. |
| fontItalic:Boolean[0..1] | If the text should be displayed in Italic. Default is false. |
| fontBold:Boolean[0..1] | If the text should be displayed in Bold. Default is false. |
| fontUnderline:Boolean[0..1] | If the text should be underlined. Default is false. |
| fontStrikeThrough:Boolean[0..1] | If the text should be stroke through. Default is false. |
| labelHorizontalAlignement:AlignmentKind[0..1] | How text should be positioned horizontally within the Label bounds. Default depends of the CMMNDiagramElement the label is attached to (see section 7.5). |
| labelVerticalAlignment: AlignmentKind[0..1] | How the text should be positioned vertically inside the Label bounds. Default depends of the CMMNDiagramElement the label is attached to (see section 7.5). Start means “top” and end means “bottom”. |

**7.5 Notational Depiction Library and Abstract Element Resolutions**

As a notation, CMMN specifies the depiction for each of the CMMN elements.

Serializing a CMMN diagram for interchange requires the specification of a collection of CMMNShape(s) (see section 7.4.6) and CMMNEdge(s) (see section 7.4.7) in the CMMNDiagram (see section 7.4.4). The CMMNShape(s) and CMMNEdge(s) attributes must be populated in such a way as to allow the unambiguous rendering of the CMMN diagram by the receiving party. More specifically, the CMMNShape(s) and CMMNEdge(s) MUST reference CMMN model elements. If no CMMNElement is referenced or if the reference is invalid, it is expected that this shape or edge should not be depicted. The only exception is Discretionary Association. For this kind of link no CMMN model element exists since the CMMNEdge does not depict any model element but rather a containment relation (i.e. the Discretionary Item is linked to the HumanTask having a Planning Table that contains this Discretionary Item). See table 7.18 for the depiction.

When rendering a CMMN diagram, the correct depiction of a CMMNShape or CMMNEdge depends mainly on the referenced CMMN model element and its particular attributes and/or references. The purpose of this section is to: provide a library of the CMMN element depictions, and to provide an unambiguous resolution between the referenced CMMN model element [CMMNElement] and their depiction. Depiction resolution tables are provided below for both CMMNShape (Section 7.5.2) and CMMNEdge (Section 7.5.3).

**7.5.1 Labels**

Both CMMNShape and CMMNEdge may have labels (its name attribute) placed on the shape/edge, or above or below the shape/edge, in any direction or location, depending on the preference of the modeler or modeling tool vendor.

Labels are optional for CMMNShape and CMMNEdge. When there is a label, the position of the label is specified by the bounds of the CMMNLabel of the CMMNShape or CMMNEdge. Simply put, label visibility is defined by the presence of the CMMNLabel element.

The bounds of the CMMNLabel are optional and always relative to the containing CMMNDiagram's origin point. The depiction resolution tables provided below exemplify default label positions if no bounds are provided for the CMMNLabel (for CMMNShape kinds (Section 7.5.2) and CMMNEdge kinds (Section 7.5.3)).

When the CMMNLabel is contained in a CMMNShape, the text to display is the name of the CMMNElement.

When the CMMNLabel is contained by a CMMNEdge referencing a CMMNElement (representing an OnPart) the name of that OnPart is used as the label with optionally the StandardEvent name in brackets.

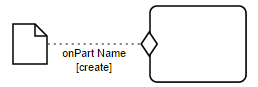


Figure 7.6 OnPart connector displaying the OnPart name and the Standard Event

When the CMMNLabel is contained by a CMMNEdge not referencing a CMMNElement (i.e. representing a Discretionary Association), no label should be displayed. Simply put, no label should be displayed for Discretionary Associations.

**7.5.2 CMMNShape Resolution**

CMMNShape can be used to represent CasePlanModel, PlanItem, DiscretionaryItem, EntryCriterion, ExitCriterion and CaseFileItem.

**7.5.2.1 Case Plan Model**

A CMMNShape referring a CasePlanModel doesn’t need any further CMMNShape attributes to specify its depiction.

Table 7. Depiction Resolution for CasePlanModel

|  |  |  |
| --- | --- | --- |
| **CMMNElement** | **CMMNShape attributes** | **Depiction** |
| CasePlanModel | None |  |

**7.5.2.2 Plan Item and Discretionary Items**

When a CMMNShape is used to depict a PlanItem or a DiscretionaryItem, the actual shape is determined by: the referred PlanItemDefinition. PlanItem are displayed with a solid border while DiscretionaryItem are displayed with a dashed border.

Table 7. Depiction for PlanItem and DiscretionaryItem

|  |  |  |  |
| --- | --- | --- | --- |
| **PlanItemDefinition** | **CMMNShape Attributes** | **Depiction** | |
| **PlanItem** | **DiscretionaryItem** |
| Stage | isCollapsed=true |  |  |
| Stage | isCollapsed=false |  |  |
| PlanFragment | isCollapsed=true | N/A |  |
| PlanFragment | isCollapsed=false | N/A |  |
| Task | None |  |  |
| HumanTask  isBlocking=false | None |  |  |
| HumanTask  isBlocking=true | None |  |  |
| ProcessTask | None |  |  |
| CaseTask | None |  |  |
| DecisionTask | None |  |  |
| Milestone | None |  |  |
| EventListener | None |  |  |
| UserEventListener | None |  |  |
| TimerEventListener | None |  |  |

**7.5.2.3 Auto Complete Decorator**

When the CMMNShape depicts a CasePlanModel, a PlanItem referring to a Stage or a DiscretionaryItem referring to a Stage and when the Stage has the attribute “autoComplete” set to true, then the Shape MUST display the autoComplete decorator at the bottom center of the shape.

Table 7. Auto Complete Decorator depiction

|  |  |  |
| --- | --- | --- |
| **Stage Attribute** | **Depiction** | |
| autoComplete=true |  | |
|  |  |

**7.5.2.4 ItemControl Decorators**

When a PlanItem or the DiscretionaryItem has defined ItemControl or when its referred PlanItemDefinition has ItemControl, then the proper ItemControl decorators MUST be depicted in the bottom center of the shape.

Table 7. Item Control Decorators Depiction

|  |  |  |
| --- | --- | --- |
| **ItemControl** | **Depiction** | |
| Manual Activation Rule |  |  |
| Required Rule |  |  |
| Repetition Rule |  |  |

**7.5.2.5 Planning Table Decorator**

When a CMMNShape depicts a CasePlanModel that has a PlanningTable; or when it depicts a PlanItem or a DiscretionaryItem referring to a Stage containing a PlanningTable; or when it depicts a PlanItem or a DiscretionaryItem referring to a HumanTask containing a PlanningTable, then the planning table marker MUST be depicted attached to the top border of the Stage shape, somewhere between the top left corner and the middle point.

CMMNDI doesn’t provide any interchange for the PlanningTable bounds.

Table 7. PlanningTable decorator depiction

|  |  |  |
| --- | --- | --- |
| **CMMNShape Attribute** | **Depiction** | |
| isPlanningTableCollapsed=true |  | |
|  |  |
| isPlanningTableCollapsed=false |  | |
|  |  |
| When the PlanningTable is expanded, DiscretionaryItem(s) MUST be depicted either using Discretionary Association for HumanTask or inside the Stage/CasePlanModel. | |

**7.5.2.6 Entry criterion**

When a CMMNShape refers to an Entry Criterion, it is depicted attached to the border of the PlanItem or the DiscretionaryItem holding that Entry Criterion.

Only PlanItem and DiscretionaryItem referring to a Task, a Stage or a Milestone can have Entry Criterion.

Table 7. Depiction Resolution of Entry Criterion

|  |  |  |
| --- | --- | --- |
| **CMMNElement** | **CMMNShape Attributes** | **Depiction** |
| EntryCriterion | None | *MUST be attached to a PlanItem or a DiscretionaryItem* |

**7.5.2.7 Exit criterion**

When a CMMNShape depicts an ExitCriterion, it is depicted attached to the border of the CasePlanModel, the PlanItem or the DiscretionaryItem holding that ExitCriterion.

Only Case Plan Model, Plan Item and Discretionary Item referring to a Task or a Stage can have Exit Criterion.

Table 7. Depiction Resolution of Exit Criterion

|  |  |  |
| --- | --- | --- |
| **CMMNElement** | **CMMNShape Attributes** | **Depiction** |
| ExitCriterion | None | *MUST be attached to a Case Plan Model, or to a Plan Item or a Discretionary Item referring to a Task or a Stage* |

**7.5.2.8 Case File Item**

A CMMNShape referencing a CaseFileItem doesn’t need any further CMMNShape attributes to specify its depiction.

Table 7. Depiction Resolution of Case File Item

|  |  |  |
| --- | --- | --- |
| **CMMNElement** | **CMMNShape Attributes** | **Depiction** |
| Case File Item | None |  |

**7.5.3 CMMNEdge Resolution**

**7.5.3.1 On Part Connector referring to CaseFileItemOnPart**

When the CMMNEdge depicts a CaseFileItemOnPart, its source is the CaseFileItem referred by the sourceRef of the CaseFileItemOnPart. Its target is determined by its targetCMMNElementRef.

Table 7. Depiction Resolution of OnPart connector referring to a CaseFileItemOnPart

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CMMNElement** | **CMMNEdge Attribute** | **CMMNEdge sourceCMMNElementRef** | **CMMNEdge targetCMMNElementRef** | **Depiction** |
| CaseFileItemOnPart | isStandardEventVisible=true | None.  *The source of the connector is the CaseFileItem defined in the sourceRef of the CaseFileItemOnPart* | Entry Criterion or Exit Criterion |  |
| CaseFileItemOnPart | isStandardEventVisible=false | None.  *The source of the connector is the CaseFileItem defined in the sourceRef of the CaseFileItemOnPart* | Entry Criterion or Exit Criterion |  |

**7.5.3.2 On Part Connector referring to PlanItemOnPart**

When the CMMNEdge depicts a PlanItemOnPart, its source is the PlanItem referred by the sourceRef of the PlanItemOnPart or the ExitCriterion if one is defined in the exitCriterionRef of the PlanItemOnPart. Its target is determined by its targetCMMNElementRef.

Table 7. Depiction Resolution of OnPart connector referring to a PlanItemOnPart

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CMMNElement** | **CMMNEdge Attribute** | **CMMNEdge sourceCMMNElementRef** | **CMMNEdge targetCMMNElementRef** | **Depiction** |
| PlanItemOnPart | isStandardEventVisible=true | None.  *The source of the connector is the PlanItem defined in the sourceRef of the PlanItemOnPart or the ExitCriterion defined in the exitCriterionRef when one is specified in the PlanItemOnPart* | Entry Criterion or Exit Criterion |  |
| PlanItemOnPart | isStandardEventVisible=false | None.  *The source of the connector is the PlanItem defined in the sourceRef of the PlanItemOnPart or the ExitCriterion defined in the exitCriterionRef when one is specified in the PlanItemOnPart* | Entry Criterion or Exit Criterion |  |

**7.5.3.3 Discretionary Association**

When the CMMNEdge depicts a Discretionary Association, its source and its target need to be specified.

Table 7. Depiction Resolution of Discretionary Association

|  |  |  |  |
| --- | --- | --- | --- |
| **CMMNElement** | **CMMNEdge sourceCMMNElementRef** | **CMMNEdge targetCMMNElementRef** | **Depiction** |
| None | Plan Item or Discretionary Item referring to a Human Task holding the Planning table defining the depicted Discretionary Item | Discretionary Item |  |