CR-20 Proposal

Specification and Meta-Model Modifications

*In table 5.1 - **CMMNElement Attributes** the following lines need to be added*

extensionDefinitions: ExtensionDefinition [0..*]	This attribute is used to attach additional attributes and associations to any CMMN Element. This association is not applicable when the XML schemainterchange is used, since the XSD mechanisms for supporting anyAttribute and any element already satisfy this requirement. See section 5.1.5 for additional information on extensibility.
extensionValues: ExtensionAttributeValue [0..*]	This attribute is used to provide values for extended attributes and model associations. This association is not applicable when the XML schema interchange is used, since the XSD mechanisms for supporting anyAttribute and any element already satisfy this requirement. See section 5.1.5 for addition information on extensibility.

A new figure of the meta model showing the relation between CMMNElement and extension need to be added in section 5.1.1

In table 5.2 – Definitions attributes the following lines need to be added after the imports line

extensionDefinitions: ExtensionDefinition [0..*]	This attribute is used to attach additional attributes and associations to CMMN Definitions. This association is not applicable when the XML schemainterchange is used, since the XSD mechanisms for supporting anyAttribute and any element already satisfy this requirement. See section 5.1.5 for additional information on extensibility.
extensionValues: ExtensionAttributeValue [0..*]	This attribute is used to provide values for extended attributes and model associations. This association is not applicable when the XML schema interchange is used, since the XSD mechanisms for supporting anyAttribute and any element already satisfy this requirement. See section 5.1.5 for addition information on extensibility.

In table 5.2 – Definitions attributes the following lines need to be added at the end

relationships: Relationship [0..*]	This attribute enables the extension and integration of CMMN models into larger system/development Case Management.

Figure 5.1 – Definitions class diagram need to be updated with Extensions depiction.

The following sections need to be added to 5.1

5.1.5 Extensibility

The **CMMN** metamodel is aimed to be extensible. This allows **CMMN** adopters to extend the specified metamodel in a way that allows them to be still **CMMN**-compliant. It provides a set of extension elements, which allows **CMMN** adopters to attach additional attributes and elements to standard and existing **CMMN** elements. This approach results in more interchangeable models, because the standard elements are still intact and can still be understood by other **CMMN** adopters. It's only the additional attributes and elements that MAY be lost during interchange.

[Figure of the meta model should be inserted here, in the meta model we need to add the four classes described]

A **CMMN** Extension basically consists of four different elements:

1. Extension
2. ExtensionDefinition
3. ExtensionAttributeDefinition
4. ExtensionAttributeValue

The core elements of an Extension are the `ExtensionDefinition` and `ExtensionAttributeDefinition`. The latter defines a list of attributes that can be attached to any **CMMN** element. The attribute list defines the name and type of the new attribute. This allows **CMMN** adopters to integrate any meta model into the **CMMN** meta model and reuse already existing model elements. The `ExtensionDefinition` itself can be created independent of any **CMMN** element or any **CMMN** definition. In order to use an `ExtensionDefinition` within a **CMMN** model definition (`Definitions` element), the `ExtensionDefinition` MUST be associated with an `Extension` element that binds the `ExtensionDefinition` to a specific **CMMN** model definition. The `Extension` element itself is contained within the **CMMN** element `Definitions` and therefore available to be associated with any **CMMN** element making use of the `ExtensionDefinition`. Every **CMMN** element can be extended by additional attributes. This works by associating a **CMMN** element with an `ExtensionDefinition`, which was defined at the **CMMN** model definitions level (element `Definitions`). Additionally, every "extended" **CMMN** element contains the actual extension attribute value. The attribute value, defined by the element `ExtensionAttributeValue` contains the value of type `Element`. It also has an association to the corresponding attribute definition.

Extension

The `Extension` element binds/imports an `ExtensionDefinition` and its attributes to a **CMMN** model definition. Table 5.8 presents the attributes and model associations for the `Extension` element.

Table 5.8 - Extension attributes and model association

Attribute Name	Description/Usage
mustUnderstand: boolean [0..1] = False	This flag defines if the semantics defined by the extension definition and its attribute definition MUST be understood by the CMMN adopter in order to process the CMMN model correctly. Defaults to False.

definition: ExtensionDefinition	Defines the content of the extension. Note that in the XML schema, this definition is provided by an external XML schema file and is simply referenced by QName.

ExtensionDefinition

The ExtensionDefinition class defines and groups additional attributes. This type is not applicable when the XML schema interchange is used, since XSD Complex Types already satisfy this requirement. Table 5.9 presents the attributes and model associations for the ExtensionDefinition element.

Table 5.9 - ExtensionDefinition attributes and model association

Attribute Name	Description/Usage
name: string	The name of the extension. This is used as a namespace to uniquely identify the extension content.
extensionAttributeDefinitions: ExtensionAttributeDefinition [0..*]	The specific attributes that make up the extension.

ExtensionAttributeDefinition

The ExtensionAttributeDefinition defines new attributes. This type is not applicable when the XML schema interchange is used; since the XSD mechanisms for supporting “AnyAttribute” and “Any” type already satisfy this requirement.

Table 5.10 presents the attributes for the ExtensionAttributeDefinition element.

Table 5.10 - ExtensionAttributeDefinition attributes and model association

Attribute Name	Description/Usage
name: string	The name of the extension attribute.
type: string	The type that is associated with the attribute.
isReference: boolean [0..1] = False	Indicates if the attribute value will be referenced or contained.

ExtensionAttributeValue

The ExtensionAttributeValue contains the attribute value. This type is not applicable when the XML schema interchange is used; since the XSD mechanisms for supporting “AnyAttribute” and “Any” type already satisfy this requirement.

Table 5.11 presents the model associations for the ExtensionAttributeValue element.

Table 5.11 – ExtensionAttributeValue model association

Attribute Name	Description/Usage
value: Element [0..1]	The contained attribute value, used when the associated ExtensionAttributeDefinition.isReference is false. The type of this Element MUST conform to the

	type specified in the associated `ExtensionAttributeDefinition`.
valueRef: Element [0..1]	The referenced attribute value, used when the associated `ExtensionAttributeDefinition.isReference` is true. The type of this Element MUST conform to the type specified in the associated `ExtensionAttributeDefinition`.
extensionAttributeDefinition: ExtensionAttributeDefinition	Defines the extension attribute for which this value is being provided.

5.1.6 External Relationships

It is the intention of this specification to cover the basic elements necessary for the construction of semantically rich and syntactically valid **Case** models to be used in the description of various ad-hoc situations. As the specification indicates, extension capabilities enable the enrichment of the information described in **CMMN** and supporting models to be augmented to fulfill particularities of a given usage model. These extensions intention is to extend the semantics of a given **CMMN** Artifact to provide specialization of intent or meaning.

Case models do not exist in isolation and generally participate in larger, more complex business and system development. The intention of the following specification element is to enable **CMMN** Artifacts to be integrated in these development **Processes** via the specification of a non-intrusive identity/relationship model between **CMMN** Artifacts and elements expressed in any other addressable domain model.

The 'identity/relationship' model it is reduced to the creation of families of typed relationships that enable **CMMN** and non-**CMMN** Artifacts to be related in non intrusive manner. By simply defining 'relationship types' that can be associated with elements in the **CMMN** Artifacts and arbitrary elements in a given addressable domain model, it enables the extension and integration of **CMMN** models into larger system/development **Processes**.

It is that these extensions will enable, for example, the linkage of 'derivation' or 'definition' relationships between UML artifacts and **CMMN** Artifacts in novel ways. So, a UML use case could be related to a **Case** element in the **CMMN** specification without affecting the nature of the Artifacts themselves, but enabling different integration models that traverse specialized relationships.

Simply, the model enables the external specification of augmentation relationships between **CMMN** `Artifacts` and arbitrary relationship classification models, these external models, via traversing relationships declared in the external definition allow for linkages between **CMMN** elements and other structured or non-structured metadata definitions.

The UML model for this specification follow a simple extensible pattern as shown below; where named relationships can be established by referencing objects that exist in their given namespaces.

[Figure of the meta model should be inserted here. Relationship is a new class that extends CMMNElement and is related to the Definition]

The Relationship element inherits the attributes and model associations of CMMNElement (see Table 5.1).

Table 5.12 presents the additional attributes for the Relationship element.

Table 5.12 – Relationship attributes

Attribute Name	**Description/Usage**
type: string	The descriptive name of the element.
direction: RelationshipDirection {None | Forward | Backward | Both}	This attribute specifies the direction of the relationship.
sources: Element [1..*]	This association defines artifacts that are augmented by the relationship.
targets: Element[1..*]	This association defines artifacts used to extend the semantics of the source element(s).