# **Proposal:**

**Summary**

* Change ItemDefinition metamodel to support function definition
* XSD schema changes at <https://github.com/opatrascoiu/omg-dmn-spec/blob/master/xsd/dmn.xsd>

**Details**

The changes are baselined on **formal-19-01-05.pdf**.

**Position:**

**Page 132 Section 7.3.2**

**Replace**

The inputs and output of decisions are data items whose value, at the decision logic level, is assigned to variables or represented by value expressions.

**with**

The inputs and output of decisions, business knowledge models, and decision services, and the output of input data (all DRGElements)are data items whose value, at the logic level, is assigned to variables or represented by value expressions.

**Position:**

**Page 72 Section 7.3.2 ItemDefinition metamodel**

**Replace**

An alternative way to define an instance of ItemDefinition is as a composition of ItemDefinition elements. An instance of ItemDefinition may contain zero or more itemComponent, which are themselves ItemDefinitions. Each itemComponent in turn may be defined by either a typeRef and allowedValues or a nested itemComponent. In this way, complex types may be defined within DMN. The name of an itemComponent (nested ItemDefinition) must be unique within its containing ItemDefinition or itemComponent. An ItemDefinition element SHALL be defined using only one of the alternative ways:

* reference to a built-in or imported typeRef, possibly restricted with allowedValues;
* composition of ItemDefinition elements

**with (remove last sentence)**

An alternative way to define an instance of ItemDefinition is as a composition of ItemDefinition elements. An instance of ItemDefinition may contain zero or more itemComponent, which are themselves ItemDefinitions. Each itemComponent in turn may be defined by either a typeRef and allowedValues or a nested itemComponent. In this way, complex types may be defined within DMN. The name of an itemComponent (nested ItemDefinition) must be unique within its containing ItemDefinition or itemComponent.

**Position:**

**Page 72 Section 7.3.2**

**Insert**

An alternative way to define an instance of ItemDefinition is by specifying a FunctionItem element, which defines the signature of a function: the parameters and the output of the function. An instance of ItemDefinition may contain one FunctionItem. A FunctionItem may contain zero or more parameters defined as InformationItems and one output type defined as a typeRef. The names of the parameters of a FunctionItem are unique.

An ItemDefinition element SHALL be defined using only one of the alternative ways:

* reference to a built-in or imported typeRef, possibly restricted with allowedValues;
* composition of ItemDefinition elements
* function signature element

**before**

The ItemDefinition element specializes NamedElement and it inherits its attributes and model associations. Table 24 presents the additional attributes and model associations of the ItemDefinition element.

**Position:**

**Page 72 Table 24: ItemDefinition attributes and model associations**

**Append following row**

|  |  |
| --- | --- |
| **functionItem:** FunctionItem[0..1] | This attribute describes an optional FunctionItem that compose this ItemDefinition. |

**Insert new Table XX: FunctionItem attributes and model associations**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| **outputTypeRef**: String[0..1] | Reference to output type of function |
| **parameters**: InformationItem[0..\*] | Function parameters as InformationItems |

**and add Figure below after table.**

****

**Figure 7.7: ItemDefinition class diagram**

**Notes:**

Parameters should be 0..1 on FI side

FunctionItem.outputTypeRef should be optional, for consistency w/ parameter type

**XSD Schema changes**

Change file <https://github.com/omg-dmn-taskforce/omg-dmn-spec/blob/master/xsd/DMN12.xsd>

**Replace**

<**xsd:complexType name="tItemDefinition"**>
 <**xsd:complexContent**>
 <**xsd:extension base="tNamedElement"**>
 <**xsd:choice**>
 <**xsd:sequence**>
 <**xsd:element name="typeRef" type="xsd:string"**/>
 <**xsd:element name="allowedValues" type="tUnaryTests" minOccurs="0"**/>
 </**xsd:sequence**>
 <**xsd:element name="itemComponent" type="tItemDefinition" minOccurs="0" maxOccurs="unbounded"**/>
 </**xsd:choice**>
 <**xsd:attribute name="typeLanguage" type="xsd:anyURI" use="optional"**/>
 <**xsd:attribute name="isCollection" type="xsd:boolean" use="optional" default="false"**/>
 </**xsd:extension**>
 </**xsd:complexContent**>
</**xsd:complexType**>

**with**

<**xsd:complexType name="tItemDefinition"**>
 <**xsd:complexContent**>
 <**xsd:extension base="tNamedElement"**>
 <**xsd:choice**>
 <**xsd:sequence**>
 <**xsd:element name="typeRef" type="xsd:string"**/>
 <**xsd:element name="allowedValues" type="tUnaryTests" minOccurs="0"**/>
 </**xsd:sequence**>
 <**xsd:element name="itemComponent" type="tItemDefinition" minOccurs="0" maxOccurs="unbounded"**/>
 <**xsd:element name="functionItem" type="tFunctionItem" minOccurs="0" maxOccurs="1"**/>
 </**xsd:choice**>
 <**xsd:attribute name="typeLanguage" type="xsd:anyURI" use="optional"**/>
 <**xsd:attribute name="isCollection" type="xsd:boolean" use="optional" default="false"**/>
 </**xsd:extension**>
 </**xsd:complexContent**>
</**xsd:complexType**>
<**xsd:element name="functionItem" type="tFunctionItem" substitutionGroup="namedElement"**/>
<**xsd:complexType name="tFunctionItem"**>
 <**xsd:complexContent**>
 <**xsd:extension base="tNamedElement"**>
 <**xsd:sequence**>
 <**xsd:element name="parameters" type="tInformationItem" minOccurs="0" maxOccurs="unbounded"**/>
 </**xsd:sequence**>
 <**xsd:attribute name="outputTypeRef" type="xsd:string"**/>
 </**xsd:extension**>
 </**xsd:complexContent**>
</**xsd:complexType**>