# Add boxed conditional expression

*In section 10.2.1, add the following to the dotted list:*

* A boxed conditional

*Add a sub section to section 10.2.1 named* “Boxed conditional” *with the following content:*

Boxed conditional offers a visual representation of an if statement using three rows. The first one is labelled “if”; the second one is labelled “then” and the last one is labelled “else”. In the right part, another FEEL expression is expected. The expression in the “if” part MUST resolve to a boolean.



Figure Conditional expression

Color is suggested.



Figure Use of conditional expression with decision table and invocation

*In* “Figure 10.17 Expression class diagram”:

*Add the following class extending Expression*

|  |
| --- |
| Conditional |

*Add the following class*

|  |
| --- |
| ChildExpression |
| +id: ID [0..1] |

*Create one aggregation between* ChildExpression *and* Expression *named* value *with a cardinality of 1.*

*Create three aggregations between* Conditional and *the new* ChildExpression *class. One named* “if”*, one named* “then” *and the last one named* “else” *all with a cardinality of 1.*

*Add a sub-section in section 10.5 Metamodel named* “Conditional metamodel” *with the following content*

A Conditional is a visual way to express an if statement.

Conditional inherits all the attributes and model associations from Expression. Table XX presents the additional attributes and model associations of the Conditional element.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| if: ChildExpression | This attribute holds the expression that is evaluate by the conditional expression. |
| then: ChildExpression | This attribute holds the expression that will be evaluated when the condition in the if statement evaluates to true. |
| else: ChildExpression | This attribute holds the expression that will be evaluated when the condition in the if statement evaluates to false. |

*Add another subsection following named* “ChildExpression metamodel” *with the following content.*

A ChildExpression is used to hold an expression inside a node. Table XX presents the attributes of a ChildExpression.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| id: ID[0..1] | Optional identifier for this element. SHALL be unique within its containing Definitions element. |
| value: Expression | The instance of Expression that is the expression in this ChildExpression |

*Add the following elements to the semantic XSD*

<xsd:complexType name="tChildExpression">

<xsd:sequence>

<xsd:element type=”tExpression”/>

</xsd:sequence>

<xsd:attribute name="id" type="xsd:ID" use="optional"/>

</xsd:complexType>

<xsd:element name="conditional" type="tConditional" substitutionGroup="expression"/>

<xsd:complexType name="tConditional">

<xsd:complexContent>

 <xsd:extension base="tExpression">

 <xsd:sequence>

 <xsd:element name="if" type="tChildExpression"/>

 <xsd:element name="then" type="tChildExpression"/>

 <xsd:element name="else" type="tChildExpression"/>

 </xsd:sequence>

 </xsd:extension>

 </xsd:complexContent>

</xsd:complexType>