Metamodel Extensibility

Kitalpha
Metamodel Extensibility

Declarations

- Extensible metamodel
- Ability to extend metamodels

Existing metamodel

Contributive metamodel

Those abilities are expressed by properties in EMF (genmodel: Extensible Provider Factory, Child Creation Extenders)
EMF enables extensibility only by subclass
Pros and Cons

😊 This mechanism enables inheritance of all properties, constraints, and representations (e.g., user interfaces, diagrams)

😢 The type of a model element cannot be changed during its lifecycle
Kitalpha introduces the mechanism of extension by aggregation

New: eMDE (essential MDE metamodel), a top-level metamodel
### eMDE metamodel

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExtensibleElement</td>
<td>Metaclass providing the ability to dynamically aggregate new properties, named ElementExtension</td>
</tr>
<tr>
<td>ExtensionElement</td>
<td>Abstract metaclass which is adapted as new properties in contributive metamodels</td>
</tr>
</tbody>
</table>
Conceptually, ClassA aggregates ClassB and ClassC
• Declaration of the extension by aggregation with inheritance of ExtensibleElement
• In this case, all the subclasses of ComponentElement become extensible

Data ComponentSample.data {
   Class ComponentElement {
      superClass external emde.ExtensibleElement
      abstract: true
            ...
   }
   Class AbstractComponent {
            ...
   }
   Class SoftwareComponent {
      superClass AbstractComponent
   }
   Class HardwareComponent {
      superClass AbstractComponent
            ...
   }
   ...
}

Data ComponentSampleQualityAssessment.data {
   Class QualityAssessment {
      extends ComponentSample.AbstractComponent

   Attributes:
            ...
   Associations:
            ...
      measures contains [0,*] QualityMeasure
   }
   Class QualityMeasure {
            ...
   }

• Declaration of the extension • QualityAssessment extends the abstract class AbstractComponent
The same model element can be decorated by data from different concerns (i.e. viewpoints) during its lifecycle.

All the properties, constraints, and representations defined in the ExtensibleElement are not inherited by the ExtensionElement and must be redefined when needed.
Thank You!

https://polarsys.org/kitalpha/

benoit.langlois@thalesgroup.com

#LangloisBenoit