

Ecore MDK for Kermeta

Ecore MDK manual

Didier Vojtisek

Abstract

This document presents the Ecore Model Development Kit (MDK) for Kermeta.

Published Build date: 3-November-2010
Published \$Date: 2009-02-05 16:22:33 \$



Preface	v
Chapter 1. Introduction to the Ecore MDK	1
1.1. Short description of the available require	1
1.2. Using Ecore behavior aspect	2
Chapter 2. Presentation of the Ecore Metamodel	3
2.1. Global view	3
2.2. Generics	4

List of Figures

2.1. Ecore metamodel main class diagram.	3
2.2. Representation of Generics in the Ecore metamodel.	4

Preface

In this document we will focus on the Ecore MDK. This MDK applies to Ecore metamodel.

Important

Kermeta and its MDKs are evolving softwares and despite that we put a lot of attention to this document, it may contain errors (more likely in the code samples). If you find any error or have some information that improves this document, please send it to us using the bug tracker in the forge: **http://gforge.inria.fr/tracker/?group_id=32** or using the developer mailing list (kermeta-developers@lists.gforge.inria.fr) Last check: v1.2.1

Tip

The most update version of this document is available on line from <http://www.kermeta.org> .

Introduction to the Ecore MDK

The Model Development Kit for Kermeta contains various kind of codes that work with Ecore models: some helper classes and some transformations.

Tip

To use them you simply have to add the corresponding require statement to your kermeta code. Then, the outline and the KermetaDoc views should help you to use its content.

1.1. Short description of the available *require*

Ecore MDK available files

Currently, the MDK provides the following files :

<code>require "http://www.eclipse.org/emf/2002/Ecore"</code>	This is the Ecore metamodel itself, without behavior. All other files have a direct or indirect dependency to that one.
<code>require "platform:/plugin/fr.irisatechlab/kermeta.ecore/src/main/fr.irisatechlab/kermeta.ecore/src/ker-</code>	This file adds the behavior on all metaclasses of Ecore metamodel.
<code>require "platform:/plugin/fr.irisatechlab/kermeta.ecore/src/main/fr.irisatechlab/kermeta.ecore/src/ker-</code>	This file contains some helper classes for loading Ecore models, retrieving elements in the Ecore model (by name for example), adding EAnnotation to the model, ...
<code>require "platform:/plugin/fr.irisatechlab/kermeta.ecore/src/main/fr.irisatechlab/kermeta.ecore/src/ker-</code>	This file contains some helper operations for manipulating EPackage. Ex: obtaining the root package, obtaining the qualified name of the package, ...
<code>require "platform:/plugin/fr.irisatechlab/kermeta.ecore/src/main/fr.irisatechlab/kermeta.ecore/src/ker-</code>	This file contains a transformation class that allows to transform your Ecore model and apply Design Patterns to it. Provided Design Pattern (and variants) are : Visitor, Command, Builder and Observer.
<code>require "platform:/plugin/fr.irisatechlab/kermeta.ecore/src/main/fr.irisatechlab/kermeta.ecore/src/ker-</code>	This file contains a transformation class that allows to generate a visitor to traverse the EReferences that are isComposite by using the Aspect way. The result is a kmt file containing the architecture to perform actions (function type passed in parameter) during the traverse of a model.
<code>meta/transformations/ContainmentBasedAction-PerformerGenerator.kmt"</code>	

<code>/kernel/meta/transformations/CleanEcoreKMAnotation.kmt</code>	This file contains a transformation class that allows to remove all EAnnotation in the ecore model that may have been introduced by kernelmeta2ecore transformation.
<code>require "platform:/plugin/fr.irisatriskell.kermeta.ecore/src/kernelmeta.ecore/require "platform:/plugin/fr.irisatriskell.kermeta.ecore/src/kernelmeta.ecore/require "platform:/plugin/fr.irisatriskell.kermeta.ecore/src/kernelmeta/persistence/EcoreResource.kmt"</code>	This file contains an experimental transformation class that allows to produce the text of a dot file for producing a graphical representation of the ecore model.
<code>require "platform:/plugin/fr.irisatriskell.kermeta.ecore/src/kernelmeta/persistence/EcoreResource.kmt"</code>	This file adds some capabilities to the EMFRepository in order to support RegisteredEcoreResource. This kind of Resource allows to load an ecore model from Eclipse registry. Note that this resource is read-only, changes in this won't be saved.

1.2. Using Ecore behavior aspect

Kermeta Ecore MDK main part consist of the implementation of Ecore behavior in kermeta. Almost all operations and derived properties are available.

To use it, you simply have to add this require statement to your kermeta code :

```
require "platform:/plugin/fr.irisatriskell.kermeta.ecore/src/kermeta/ecore_behavior.kmt"
```

Note

In the current implementation of the derived properties with a multiplicity greater than 1, you'll get a new collection with the desired content. However, even if you can modify the elements themselves, these collections aren't the real container of the elements, if you modify it (adding or removing elements) the real attributes (from which the derived properties are calculated) won't be changed.

Presentation of the Ecore Metamodel

This chapter presents the Metamodel and some considerations about how to use it.

2.1. Global view

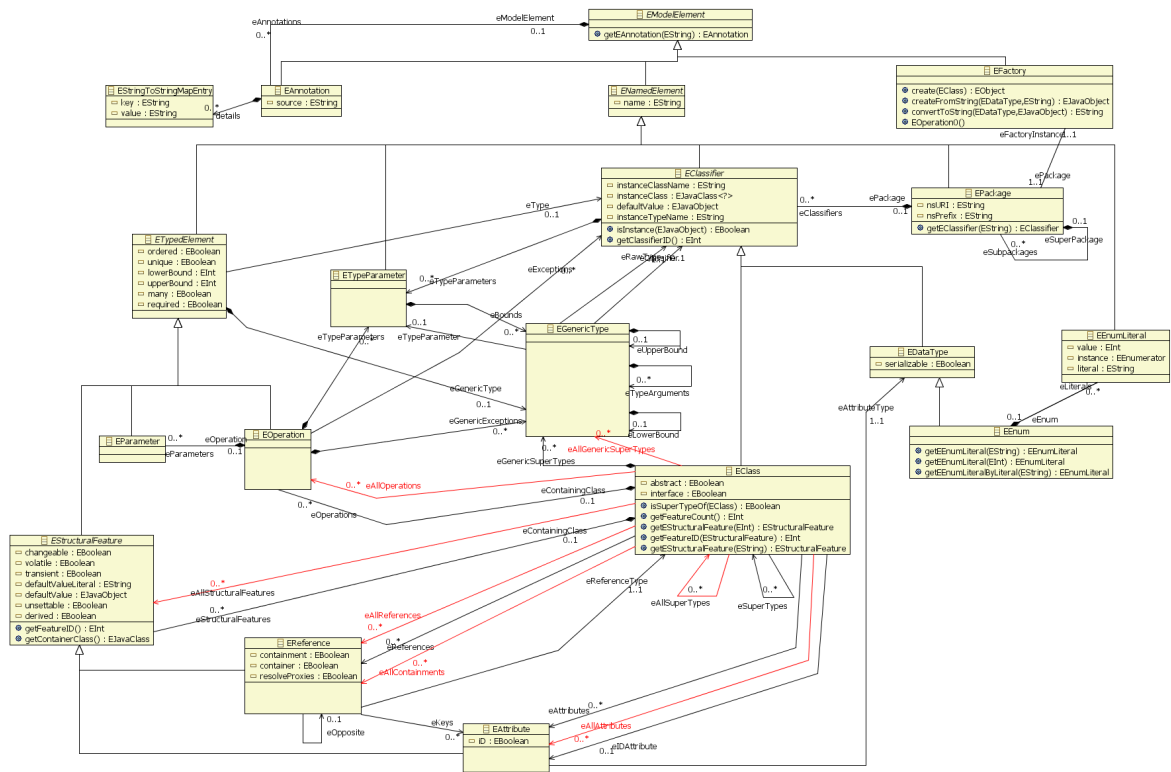


Figure 2.1. Ecore metamodel main class diagram.

Important

Ecore metamodel use a lot of calculated properties. (In red in our figures) If you need to use them, make sure to require their kermeta implementation (require "platform:/plugin/fr.irisa.triskell.kermeta.ecore/src/kermeta/ecore_behavior.kmt")

2.2. Generics

Since EMF 2.4, Eclipse has introduced a support for generics. Here is a quick overview of the introduces elements.

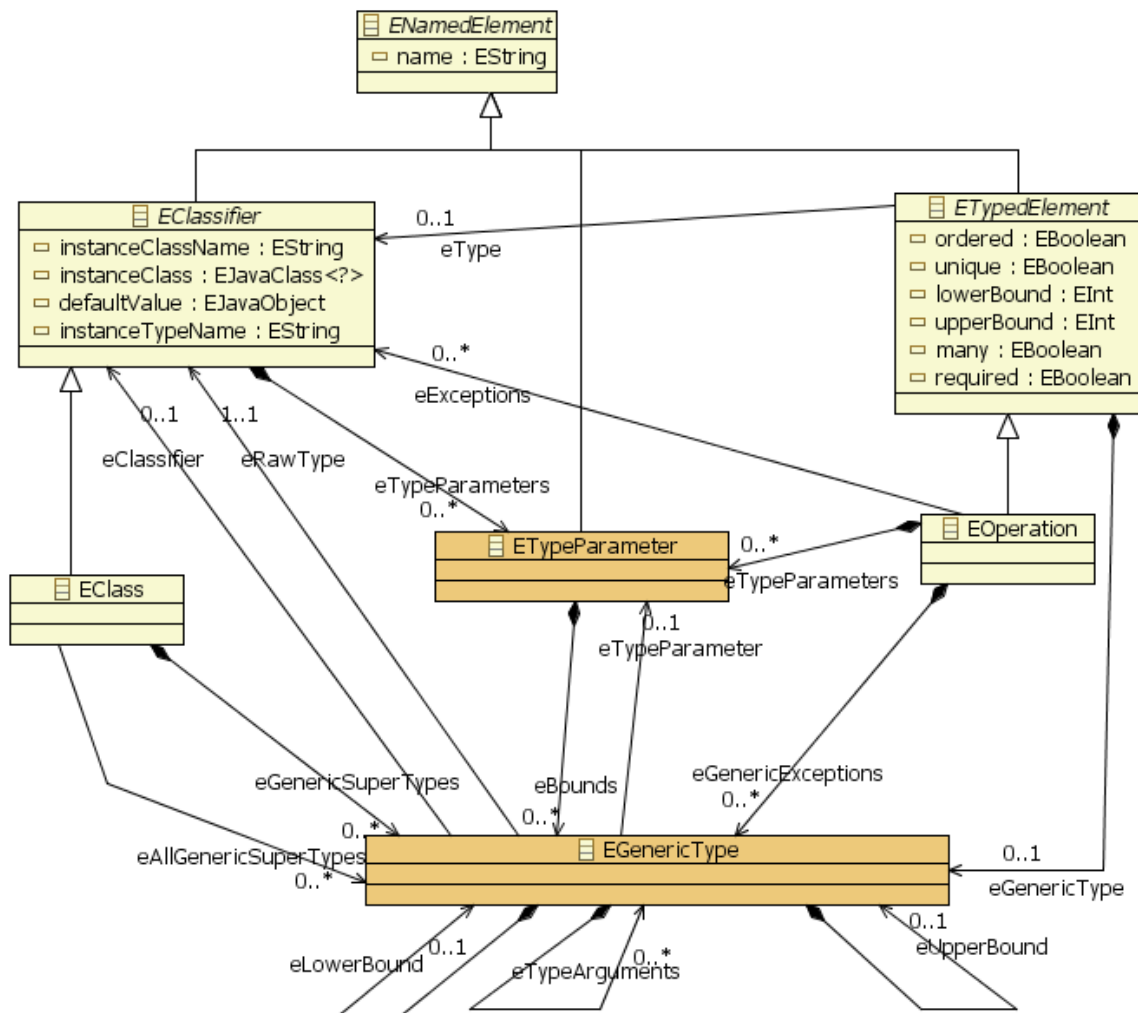


Figure 2.2. Representation of Generics in the Ecore metamodel.