Traceability MDK for Kermeta

MDK manual

Didier Vojtisek

Abstract

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

Published Build date: 3-November-2010
Published $Date: 2009-01-30 12:51:01 $

List of Figures

CHAPTER

Preface

In this document we will focus on the Traceabilty MDK. This MDK applies to Traceabilty metamodel. Ie. it provides functionalities to manipulate traces.

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 .

CHAPTER 1

Introduction to the Traceability MDK

The Model Development Kit for Kermeta contains various kind of codes that work with Tracability 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*

Traceability MDK available files

Currently, the MDK provides the following files :

`require "http://www.kermeta.org/Traceability"`

This is the Traceability metamodel itself, without behavior.

`require "platform:/plugin/fr.irisa.triskell.traceability.model/model/kmt/Traceability_Behavior.kmt"`

This file adds various helper operations to the Metaclasses of the Traceability. In particular, it provides some support for optimizing the trace model and ensuring that there is one and only one refeence to a given Object.

CHAPTER 2

Presentation of the Traceability Metamodel

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

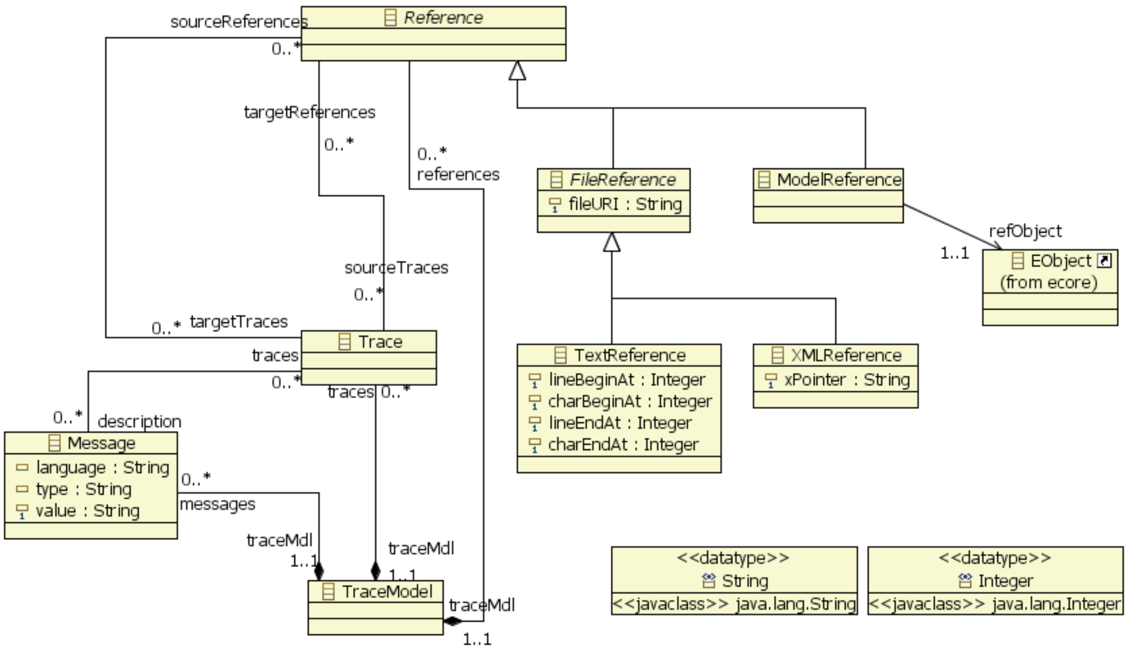

Figure 2.1. Traceability metamodel class diagram

Basically, a tracemodel is a set of Reference (textual or model) and a set of Trace. A Trace can be decorated by a Message that gives information about who and why the trace was generated.

> **Note**
>
> You choose the level of granurarilty of your traces and if they are optimized or not by optimized we mean garanting that there is one and only one ModelReference for a given Object. This optimization can be done while building the trace or in a post processing.