

Developer documentation of the SOFA 2 UML

Lukas Hermann

July 19, 2011

Contents

1	Introduction	2
2	SOFA 2 UML Mapping	3
2.1	Content of plug-ins	3
2.2	Extension points	4
3	SOFA 2 UML Preparation	5
4	SOFA 2 UML Model	7
5	SOFA 2 UML Entity	8

Chapter 1

Introduction

The SOFA 2 UML is a compound of Eclipse plug-ins which extends the SOFA 2 IDE with an ability of generation SOFA 2 entities [1] from the UML component model [2]. It introduces a mapping model, that connects a source model with a SOFA 2 project, and its editor where a mapping between source model elements and SOFA 2 entities can be easily specified. It also defines a preparation model, which is a platform independent component model staying in the middle of a transformation from a source model to a target component system.

This guide discusses extension points and API provided by the SOFA 2 UML. You should read it when you want to change the transformation of a UML model to SOFA 2 entities in some of its steps or to extend this tool for supporting another source model.

Parts: This bundle is divided into four major components:

SOFA 2 UML Mapping – the mapping model with its editor

SOFA 2 UML Preparation – the preparation model

SOFA 2 UML Model – the infrastructure for working with UML models

SOFA 2 UML Entity – the infrastructure for working with SOFA 2 entities

Chapter 2

SOFA 2 UML Mapping

2.1 Content of plug-ins

The SOFA 2 UML Mapping plug-ins contain the EMF mapping model with generated classes for editing it both programmatically and graphically.

- **org.objectweb.dsrg.sofa.uml.mapping** plug-in contains the mapping model with its diagram and infrastructure for model processing.
 - **org.objectweb.dsrg.sofa.uml.mapping** package contains a mapping model interface.
 - **org.objectweb.dsrg.sofa.uml.mapping.impl** package contains mapping model implementation.
 - **org.objectweb.dsrg.sofa.uml.mapping.util** package provides classes supporting the mapping model and infrastructure for model processing.
- **org.objectweb.dsrg.sofa.uml.mapping.edit** plug-in provides mapping model item providers used to edit a model content.
 - **org.objectweb.dsrg.sofa.uml.mapping.edit.provider** package contains the mapping model item providers.
- **org.objectweb.dsrg.sofa.uml.mapping.editor** plug-in implements the mapping model graphical editor, that manipulates model elements, and defines extension points for customizing a transformation of a source model.
 - **org.objectweb.dsrg.sofa.uml.mapping.editor.extensions** package contains interfaces and a provider of editor extension points.

- **org.objectweb.dsrg.sofa.uml.mapping.presentation** package contains both generated and manually created graphical editors for the mapping model.
- **org.objectweb.dsrg.sofa.uml.mapping.presentation.actions** package contains actions for refreshing the mapping model from a source model and generating all SOFA entities.
- **org.objectweb.dsrg.sofa.uml.mapping.presentation.databinding** package contains helper classes for data binding between a model and editor fields.
- **org.objectweb.dsrg.sofa.uml.mapping.presentation.forms** package contains mapping model form pages for processing model elements.

2.2 Extension points

The **org.objectweb.dsrg.sofa.uml.mapping.editor** plug-in defines several extension points for customizing source model processing.

- **validator** – Clients of this extension point are called when a source model file should be validated. They return whether the model satisfy rules that they define. The extension point has the *obligatory* property that defines whether the whole validation fails if the model breaks concrete client rules, and the *type* property that stores the extension of model files supported by the client plug-in.
- **transform** – Clients of this extension point get a source model file and a mapping model instance and they should fill the mapping model with data according to the source model. They returns whether the transformation was successful. The extension point has the *types* property that stores the extensions of model files supported by client plug-in separated by commas.
- **generator** – A client of this extension point generates or repairs a SOFA 2 entity of a mapping model element according to a source model.
- **searcher** – A client of this extension point returns a list of available SOFA 2 entities of a particular type in a specific SOFA 2 project or a concrete entity according to its name.
- **compatibility** – A client of this extension point returns whether a SOFA 2 entity is compatible with a mapping model element according to a source model.

Chapter 3

SOFA 2 UML Preparation

The SOFA 2 UML Preparation plug-ins contain the EMF preparation model, its transformation to the mapping model and extension points for its generation from source models.

- **org.objectweb.dsrg.sofa.uml.preparation** plug-in contains the preparation model that is generated from a source model and serves as a connection model between the source model and SOFA 2 entities for other plug-ins.
 - **org.objectweb.dsrg.sofa.uml.preparation** package contains a preparation model interface.
 - **org.objectweb.dsrg.sofa.uml.preparation.generation** package contains an extension point for **generation** of the preparation model from a source model.
 - **org.objectweb.dsrg.sofa.uml.preparation.impl** package contains preparation model implementation.
 - **org.objectweb.dsrg.sofa.uml.preparation.util** package provides classes supporting the preparation model and infrastructure for model processing.
- **org.objectweb.dsrg.sofa.uml.preparation.transform** plug-in implements the **transform** extension point with the preparation model usage. In its *types* property, it stores file extensions of source models that use the preparation model for their transformations. Now, it is set to **uml**. Update this property after creating a new plug-in implementing the **generation** extension point.

The **org.objectweb.dsrg.sofa.uml.preparation** plug-in defines a extension point for adding other source models.

- **generation** – A client of this extension point transforms a source model to a preparation model. The extension point has the *type* property that stores the extension of model files supported by the client plug-in.

Chapter 4

SOFA 2 UML Model

The SOFA 2 UML Model plug-ins operate with a UML component model and prepare it for a transformation to SOFA 2 entities.

- **org.objectweb.dsrg.sofa.uml.model.generator** plug-in adds menu items for creating or binding a SOFA 2 project to a new mapping model instance generated from a UML model file.
 - **org.objectweb.dsrg.sofa.uml.model.generator** package contains a plug-in activator.
 - **org.objectweb.dsrg.sofa.uml.model.generator.popup.actions** package contains actions for creating or binding a new mapping model instance.
- **org.objectweb.dsrg.sofa.uml.model.validator** plug-in implements the **validator** extension point. It uses OCL constraints loaded from the `validator.ocl` file for a UML model validation. If a constraint is broken, it marks the file with the UML model with a corresponding message from the `validator.properties` file. Its *obligatory* property is set to `true`, its *type* property is set to `uml`.
- **org.objectweb.dsrg.sofa.uml.model.transform** plug-in implements the **generation** extension point and transform a UML model to a preparation model. Its *type* property is set to `uml`.

Chapter 5

SOFA 2 UML Entity

The SOFA 2 UML Entity plug-ins implement mapping editor extension points and work with SOFA 2 entities.

- **org.objectweb.dsrg.sofa.uml.entity.compatibility** plug-in implements the **compatibility** extension point with the preparation model usage.
- **org.objectweb.dsrg.sofa.uml.entity.generator** plug-in implements the **generator** extension point with the preparation model usage.
- **org.objectweb.dsrg.sofa.uml.entity.searcher** plug-in implements the **searcher** extension point with the SOFA 2 IDE plug-in usage.

Bibliography

- [1] SOFA 2 component system – <http://sofa.ow2.org>
- [2] Unified Modeling Language 2.x – <http://www.eclipse.org/uml2>