GEMOC Studio Guide

Francois Tanguy <francois.tanguy@inria.fr>,
Didier Vojtisek <didier.vojtisek@inria.fr>

GEMOC Studio Guide

Francois Tanguy <francois.tanguy@inria.fr>, Didier Vojtisek <didier.vojtisek@inria.fr>

Table of Contents

List of Figures

Introduction

The GEMOC Studio offers 2 main usages:

- Building and composing new executable DSML. This mode is intended to be used by language designers (aka domain experts).
- Creating and executing models conformant to executable DSMLs. This mode is intended to be used by domain designers.

Each of these usage has it own set of tools that are referenced as **Gemoc Language workbench** for the tools for *language designers* and **Gemoc Modeling Workbench** for the tools for *domain designers*.

Chapter 1. Gemoc Language workbench

1.1. Language Workbench overview

The GEMOC Language Workbench, intended to be used by language designers: it allows building and composing new executable DSMLs.

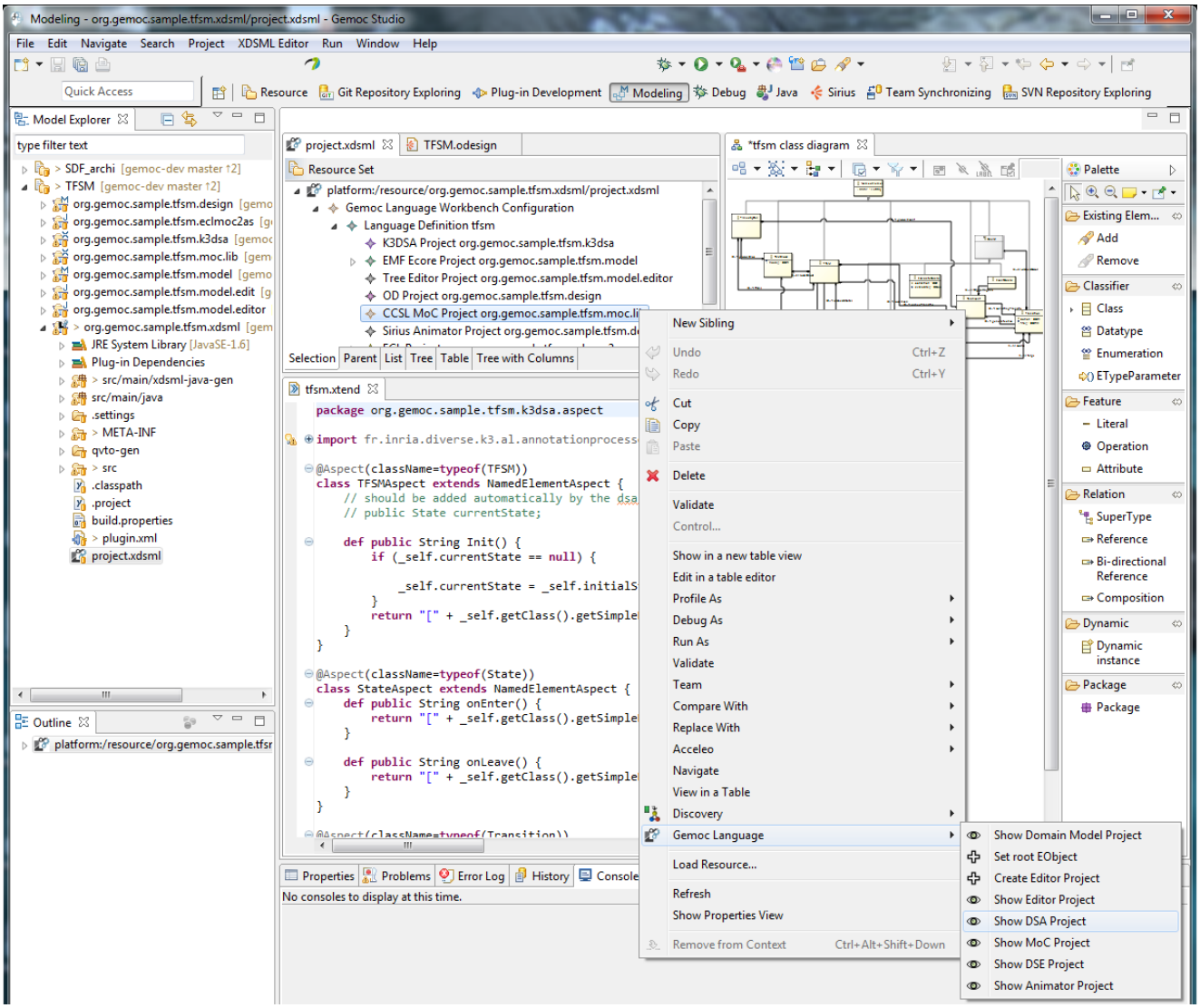

Figure 1.1. Screenshot of GEMOC Studio Language Workbench on the TFSM (Timed Finite State Machine) example.

1.2. Gemoc Language project

In the menu go to: File> New> Projects…> Gemoc Language project

This project will be the aggregator of the other Eclipse projects that will constitute the components of a Language Unit of the xDSML.

It mainly defines a project.xdsml file that references these other components.

The xDSML file isn't supposed to be edited manually. The pop up menu on the project (available on right click) or the Section 1.11, "Process support view" provides a set of wizard to fill this file.

From these information, the project will generate additional code that will be used by the execution engine when the language will be deployed and used by the Chapter 2, *Gemoc Modeling workbench*.

1.3. Defining Abstract Syntax (or domain concepts)

Defining the abstract syntax (AS) in the Language Workbench is done thanks to an EMF Ecore project.

All ecore editors can be used to edit the concepts in the ecore file.

TODO new project wizard (via popup, via process) TODO select existing project TODO Ecore editor

1.4. Defining RunTime Data

1.5. Defining Domain-Specific Actions (DSAs)

1.6. Defining Domain-Specific Constraints

1.7. Defining a concrete syntax

The xDSML can support different concrete syntaxes. Most EMF based editors should work however Gemoc provides additional support for some editors. Editors explictly supported are: EMF tree editor, xText editor, Sirius editor.

1.7.1. Defining a concrete syntax with xText

1.7.2. Defining a concrete syntax with Sirius

1.8. Defing Model of Concurrency (MoC)

1.9. Defining Domain Specific Events (DSE)

1.10. Defining an animation view

The animation layer is an extension on top of a graphical editor defined with Sirius.

TODO Debug layer, Animation layer

1.11. Process support view

TODO present process view

Chapter 2. Gemoc Modeling workbench

2.1. Modeling workbench overview

The GEMOC Modeling Workbench, intended to be used by domain designers: it allows creating and executing models conformant to executable DSMLs.

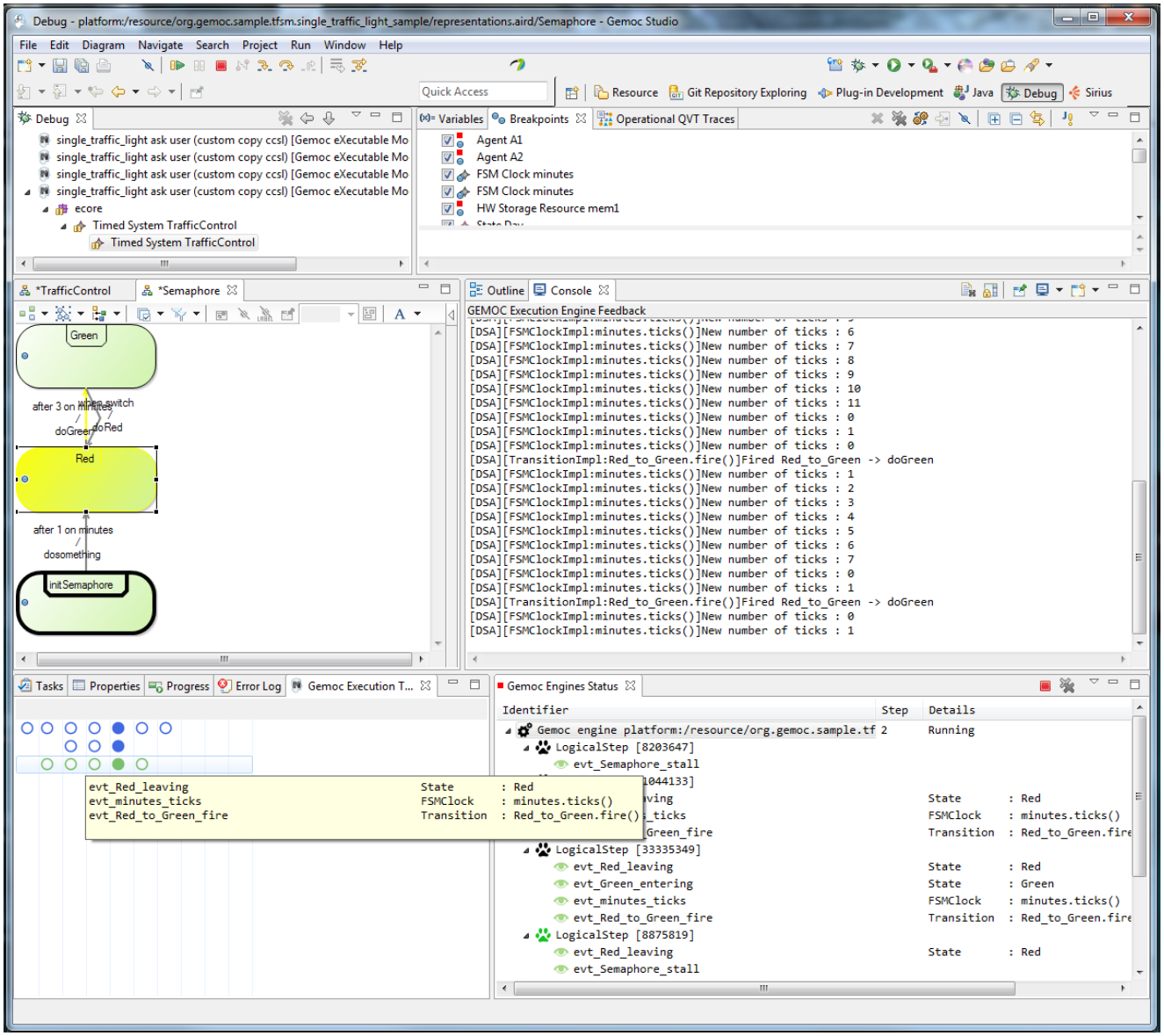

Figure 2.1. Screenshot of GEMOC Studio Modeling Workbench on the TFSM example (execution and animation).

2.2. Editing model

2.3. Executing model

2.3.1. Launch configuration

The Gemoc launch configuration offers both a Run and a Debug mode.

General options

- Model to execute : this is the model that will be run
- xDSML : this field allows to select among the valid variants of the executable language that are available for the model (Ie. the combinaison of DSA, DSE and MoCC that can be used on the given domain model)
- Decider : this field allows to select the solver strategy used by the engine when several Logical Steps can be triggered. Possible choice are :
  - Solver proposition : the solver internal stategy will be used to selecton Logical Step
  - Random : will randomly select one of the available Logical Step (warning: execution cannot be reproduced when using this Decider)
  - Ask user : (available only in Debug mode), this option will use the Logical Step View or the Timeline View to present the available Logical Steps and pause. The user will then need to click on one of the Logical Step to continue.

More Deciders will be developped (for example for playing predefined scenario).

Run mode

In run mode, it offers the faster way to run the model. It cannot be paused. However, you can stop it. It offers a limited set of views: - the Engine view allows to stop a running model.

If more feed back are required, please use one tof the front end back end available for the xDSML.

Debug mode

In debug mode, the engine offers more control on the execution. It allows to pause, add break point, and run in a step by step mode.

If offers several views.

logical steps - event manager - timeline

- activate layer debug
- activate layer animation
- show EMF tree in variables view

Backends and frontends

Back ends and front ends offer additionnal view that can respectivly display informations from the running model or provide event input to the running model.

These backends and front ends usually open dedicated views. These views are always opened in all modes (Run or Debug).

2.3.2. Engine View

2.3.3. Logical Step View

2.3.4. Timeline View

2.3.5. Event Manager View

2.3.6. Animation View

Bibliography

The bibliography lists some useful external documents. For a more complete list, please refer to the publications section on http://gemoc.org site.

Articles

[globalizing-modeling-languages] Globalizing Modeling Languages [http://hal.inria.fr/hal-00994551] (Benoit Combemale, Julien Deantoni, Benoit Baudry, Robert France, Jean-Marc Jezequel, Jeff Gray), In Computer, IEEE, 2014.

Glossary

AS
: Abstract Syntax.

API
: Application Programming Interface.

Behavioral Semantics
: see Execution semantics.

CCSL
: Clock-Constraint Specification Language.

Domain Engineer
: user of the Modeling Workbench.

DSA
: Domain-Specific Action.

DSE
: Domain-Specific Event.

DSML
: Domain-Specific (Modeling) Language.

Dynamic Semantics
: see Execution semantics.

Eclipse Plugin
: an Eclipse plugin is a Java project with associated metadata that can be bundled and deployed as a contribution to an Eclipse-based IDE.

ED
: Execution Data.

Execution Semantics
: Defines when and how elements of a language will produce a model behavior.

GEMOC Studio
: Eclipse-based studio integrating both a language workbench and the corresponding modeling workbenches

Language Workbench
: a language workbench offers the facilities for designing and implementing modeling languages.

Language Designer
: a language designer is the user of the language workbench.

MoCC
: Model of Concurrency and Communication

Model
: model which contributes to the convent of a View

Modeling Workbench
: a modeling workbench offers all the required facilities for editing and animating domain specific models according to a given modeling language.

MSA
: Model-Specific Action.

MSE
: Model-Specific Event.

RTD
: RunTime Data.

Static semantics
: Constraints on a model that cannot be expressed in the metamodel. For example, static semantics can be expressed as OCL invariants.

xDSML
: Executable Domain-Specific Modeling Language.

Index