Grant ANR-12-INSE-0011

GEMOC

ANR Project, Program INS

D4.1.2 - Eclipse-based tool to model heterogeneous model execution, and GEMOC studio (SOFTWARE) Task 4.1

Version 0.1

©Consortium GEMOC, 2012- 2016

Document Control

	- :	A : 07/05/2014	B : 13/05/2014	C :	D :
Written by Signature	Mélanie Bats - Obeo	Didier Vojtisek - Inria	Didier Vojtisek - Inria		
Approved by Signature					

Revision index	Modifications
A	Typo, links to web and other deliverables, template, section numbering
B	Clarification of the workbench features, added screenshots, added MoCML
C	
D	

©Consortium GEMOC, 2012- 2016

Authors

Author	Partner	Role
Mélanie Bats	Obeo	Lead author
Didier Vojtisek	Inria	Contributor

©Consortium GEMOC, 2012- 2016

Table of Content

1. Introduction 5
   1.1 Purpose 5
   1.2 Definitions & acronyms 5

2. GEMOC Studio Overview 6

3. Features 8
   3.1 GEMOC Workbench Features 8
      3.1.1 GEMOC Language Workbench 8
      3.1.2 GEMOC Modeling Workbench 8
   3.2 GEMOC Main Features 8
      3.2.1 Ecore Tools 8
      3.2.2 OCL 9
      3.2.3 XTend 9
      3.2.4 Kermeta 3 9
      3.2.5 TimeSquare 9
      3.2.6 ECL 9
      3.2.7 MoCCML 9
      3.2.8 XText 9
      3.2.9 Sirius 10
      3.2.10 GEMOC Execution Engine 10
      3.2.11 GEMOC Animation 10
   3.3 Additional Features 10
      3.3.1 Acceleo 10
      3.3.2 ATL 10
      3.3.3 QVTo 11
      3.3.4 EEF 11
      3.3.5 UML Designer 11
      3.3.6 EMF Compare 11
      3.3.7 TESL 11
      3.3.8 Modhel'X 11
      3.3.9 Groovy 11
      3.3.10 m2e 12
      3.3.11 Mylyn 12
      3.3.12 EGit/JGit 12
      3.3.13 Subversive SVN 12

 ©Consortium GEMOC, 2012- 2016

Eclipse-based tool to model heterogeneous model execution, and GEMOC studio

1. Introduction

1.1 Purpose

This document aims to describe the different components integrated in the GEMOC Studio.

The Gemoc Studio is available on: http://gemoc.org/studio/studio-download/

This software is provided for all major OS: Windows 32 and 64 bit, Linux 32 and 64 bit and Mac OS X.
The stable version 0.1.0 is distributed as deliverable D4.1.2 for task T4.1.

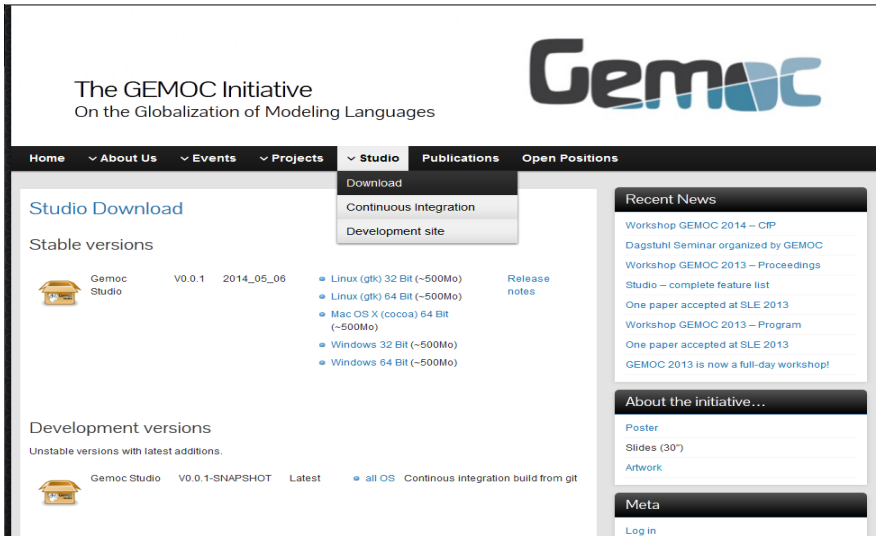

Figure 1. Screenshot of the download page

1.2 Definitions & acronyms

- **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.

©Consortium GEMOC, 2012- 2016

- **GUI**: Graphical User Interface.
- **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.
- **TESL**: Tagged Events Specification Language.
- **xDSML**: Executable Domain-Specific Modeling Language.

2. GEMOC Studio Overview

The GEMOC Studio is an Eclipse package that contains components supporting the GEMOC methodology for building and composing executable DSMLs.
It includes the two workbenches:

- The GEMOC Language Workbench, intended to be used by language designers: it allows building and composing new executable DSMLs.
- The GEMOC Modeling Workbench, intended to be used by domain designers: it allows creating and executing models conformant to executable DSMLs.

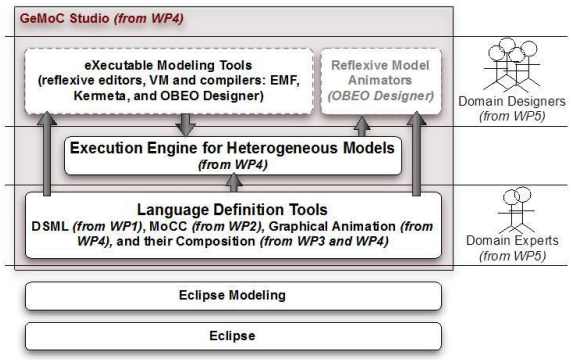

Figure 2. Studio general view

Each workbench is delivered with some basic illustrating examples that can be installed from Eclipse (File>New>Example…)

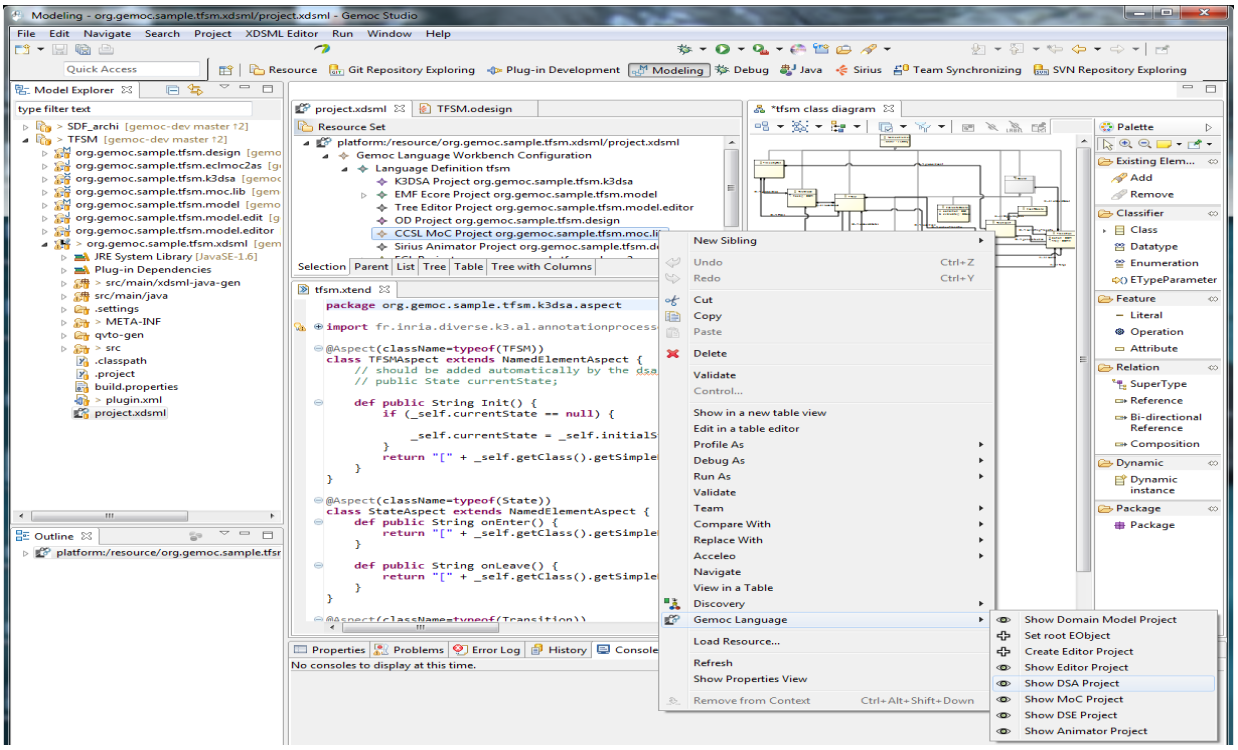

Figure 3. Screenshot of GEMOC Studio Language Workbench on the TFSM (Timed Finite State Machine) example

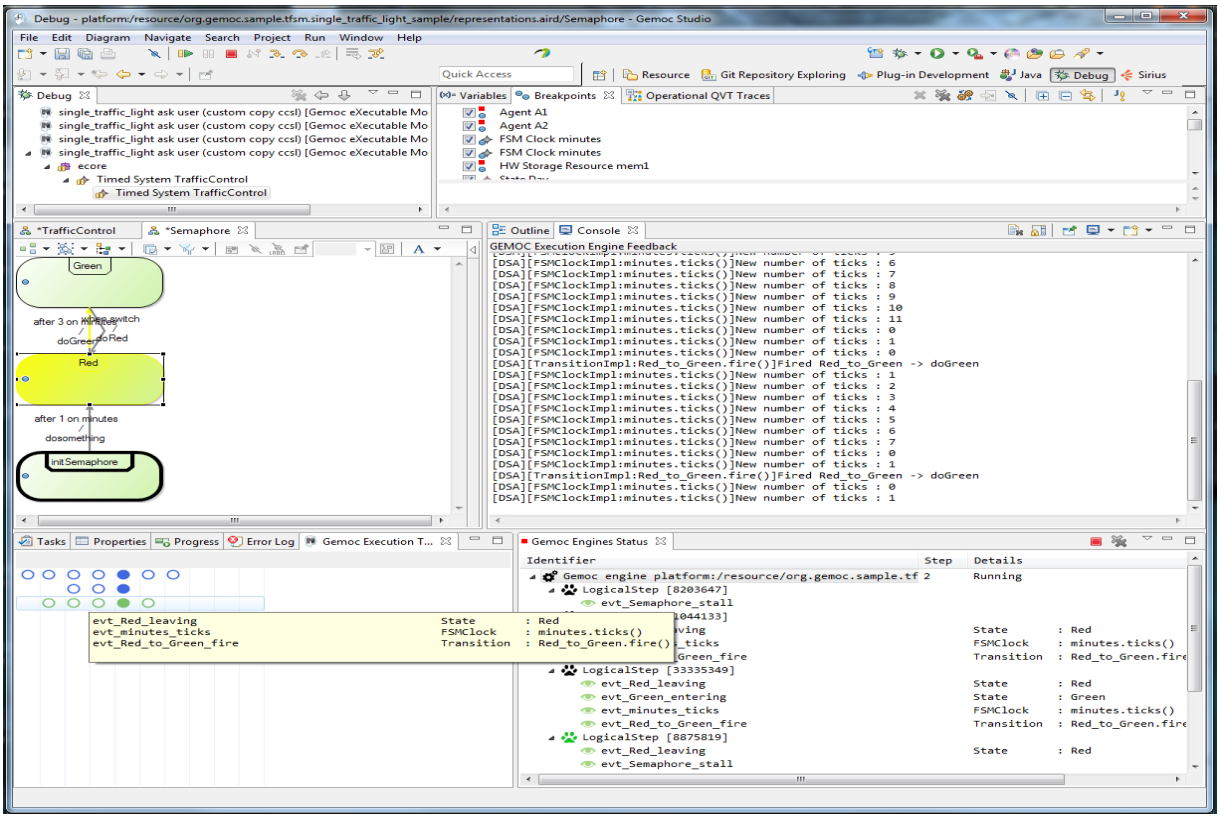

Figure 4. Screenshot of GEMOC Studio Modeling Workbench on the TFSM example (execution and animation)

3. Features

This section describes all the features available in the studio. Those features fall into three categories:

- GEMOC Workbench features: the two main workbench features provide integration support for the methodology.
- GEMOC main features : the set of components used by the GEMOC methodology (defining a new language or using a workbench language)
- Additional features: other components that are integrated to the studio to ease the development of modelling frameworks.

For sake of simplification, features that are technically deployed as several Eclipse features are grouped for presentation. The precise top level Eclipse features are listed in Annex 1.

3.1 GEMOC Workbench Features

This section presents the two main features of the Studio. These features plays a specific role in the studio, they organize the use of the other provided components.

3.1.1 GEMOC Language Workbench

The language workbench provides means to create new executable languages.
It gathers services from other features to provide the language definition tools
This component is developed in the context of the GEMOC ANR project.

Website: Not yet available
Documentation: Not yet available / partly documented in D1.3.1 and D4.1.1
Source code: Available on the gemoc-dev git repository : gemoc-dev/org/gemoc/gemoc_language_workbench/

3.1.2 GEMOC Modeling Workbench

The Modeling Workbench provides the means to use and run languages (as defined in the language workbench aforementioned) in an integrated way. It gathers services from other features to provide to the user:

- Model edition
- GEMOC Execution Engine
- GEMOC Animation

This component is developed in the context of the GEMOC ANR project.

Website: Not yet available
Documentation: Not yet available / partly documented in D1.3.1 and D4.1.1
Source code: Available on the gemoc-dev git repository: gemoc-dev/org/gemoc/gemoc_modeling_workbench/

3.2 GEMOC Main Features

This section lists the features that are explicitly used by the current GEMOC methodology in the two workbenches.

3.2.1 Ecore Tools

EcoreTools provides a complete and state of the art graphical modeler to create, edit and analyze Ecore models. The new ecore tools editors are based on Sirius.

Website: http://www.eclipse.org/ecoretools/
Documentation : http://www.eclipse.org/ecoretools/doc/

 ©Consortium GEMOC, 2012- 2016

3.2.2 OCL

The Eclipse OCL Project provides an implementation of the Object Constraint Language (OCL) OMG standard for EMF-based models. In GEMOC methodology, OCL is used to add constraints on AS models.

Website: http://www.eclipse.org/modeling/mdt/?project=ocl
Documentation: https://wiki.eclipse.org/OCL

3.2.3 XTend

Xtend is a statically-typed programming language which translates to comprehensible Java source code. It allows to use advanced language constructs with a seamless integration in the Java programming language.

Website: https://www.eclipse.org/xtend
Documentation: https://www.eclipse.org/xtend/documentation.html

3.2.4 Kermeta 3

Kermeta 3 is a metaprogramming environment based on an object-oriented DSL optimized for metamodel engineering. It works as an extension of the xTend language to provide aspect oriented and model typing facilities. It is used to define the Domain Specific Action (DSA) in the GEMOC process.

Website: https://github.com/diverse-project/k3/wiki
Documentation: Not yet available / partly documented in D1.1.1

3.2.5 TimeSquare

TimeSquare is an MDK (Model Development Kit) provided as a set of Eclipse plugins that can be downloaded or installed over an existing Eclipse. TimeSquare provides an engine, a solver and editors for CCSL specifications (Clock Constraint Specification Language).
CCSL specifications are used to define MoCC.

Website: http://timesquare.inria.fr/
Documentation: Not yet available

3.2.6 ECL

ECL is an Event Constraint Language which allows the manipulation of logical time. An ECL model defines the link between the couple AS/DSA and the MoCC.

Website: http://timesquare.inria.fr/
Documentation: Not yet available

3.2.7 MoCCML

MoCML is modeling language for defining MoCC. The component comes with a textual and a graphical editor. This component is developed in the context of the GEMOC ANR project.

Website: Not yet available
Documentation: Not yet available
Source code: Available on the gemoc-dev git repository: gemoc-dev/org/gemoc/MoCC/

3.2.8 XText

Xtext offers to create textual syntaxes and associated editor on top of grammars and/or metamodels.

Website: http://www.eclipse.org/Xtext/
Documentation: http://www.eclipse.org/Xtext/documentation.html

 ©Consortium GEMOC, 2012- 2016

3.2.9 Sirius

Sirius is an Eclipse project which allows to easily creating graphical modelling workbenches easily by leveraging the Eclipse Modeling technologies, It is built on top of EMF and GMF.

Website: http://eclipse.org/sirius/
Documentation: http://www.eclipse.org/sirius/doc/

3.2.10 GEMOC Execution Engine

This feature allows to run any model according to its xDSML definition. It uses the solver to compute the steps of the simulation. The interface provides launcher facilities to run a new simulation. This component is developed in the context of the GEMOC ANR project.

Website: Not yet available
Documentation: Deliverable notes D4.2.1
Source code: Available on the gemoc-dev git repository: gemoc-dev/org/gemoc/execution/engine

3.2.11 GEMOC Animation

This feature provides animation of graphical models according to a running execution. It allows define specific animation views or animate existing ones. It is integrated to the Eclipse Debugger UI, and it highlights running model components on graphical views and editors. It is based on Sirius. This component is developed in the context of the GEMOC ANR project.

Website: Not yet available
Documentation: Not yet available
Source code: Available on the gemoc-dev git repository: gemoc-dev/fr/obeo/dsl/sirius/animation

3.3 Additional Features

This section lists the features that have been added to the studio in order to ease the development or that are used internally by the main features.

3.3.1 Acceleo

Acceleo is a pragmatic implementation of the Object Management Group (OMG) MOF Model to Text Language (MTL) standard.

Website: http://www.eclipse.org/acceleo/
Documentation: http://www.eclipse.org/acceleo/support/

3.3.2 ATL

ATL (ATL Transformation Language) is a model transformation language and toolkit. In the field of Model-Driven Engineering (MDE), ATL provides ways to produce a set of target models from a set of source models.

Website : https://www.eclipse.org/atl/
Documentation : https://www.eclipse.org/atl/documentation/

 ©Consortium GEMOC, 2012- 2016

3.3.3 QVTo

QVT Operational component is a partial implementation of the Operational Mappings Language defined by the OMG standard specification (MOF) 2.0 Query/View/Transformation. In long term, it aims to provide a complete implementation of the operational part of the standard.

Website: http://www.eclipse.org/mmt/?project=qvto
Documentation: https://wiki.eclipse.org/QVTo

3.3.4 EEF

The Extended Editing Framework aims at enhancing EMF model creation phase by providing services dedicated to advanced model editing.

Website: http://www.eclipse.org/modeling/emft/?project=eef
Documentation: https://wiki.eclipse.org/EEF

3.3.5 UML Designer

UML Designer is a graphical tool to edit and visualize UML 2 models.

Website: http://marketplace.obeonetwork.com/module/uml
Documentation: http://www.obeonetwork.com/group/uml-designer/page/reference-documentation

3.3.6 EMF Compare

EMF Compare provides comparison and merge facility for any kind of EMF Model.
In a nutshell this project provides:

- a framework you can easily reuse and extend to compare instances of your models
- a tool integrated in the Eclipse IDE to see the differences and merge them.

It includes a generic comparison engine and the ability to export differences in a model patch. It is integrated with the Eclipse Team API, meaning that it enables collaborative work on models using CVS, SVN and GIT.

Website: http://www.eclipse.org/emf/compare/
Documentation: http://www.eclipse.org/emf/compare/documentation/latest/

3.3.7 TESL

TESL is a tagged event solver used by Modhel'X. It comes with a language dedicated to clock constraints specification. It is studied in GEMOC as a possible solver.

Website: http://wwwdi.supelec.fr/software/ModHelX/
Documentation: Not yet available

3.3.8 Modhel'X

Modhel'X is an heterogeneous execution engine based on TESL. This engine is distributed with a few models of computation.

Website: http://wwwdi.supelec.fr/software/ModHelX/
Documentation: Not yet available

3.3.9 Groovy

Groovy is an agile and dynamic language for the Java Virtual Machine.

Website: http://groovy.codehaus.org/
Documentation: http://groovy.codehaus.org/Documentation

 ©Consortium GEMOC, 2012- 2016

3.3.10 m2e

The goal of the m2e project is to provide a first-class Apache Maven support in the Eclipse IDE, making it easier to edit Maven's pom.xml, run a build from the IDE and much more. For Java developers, the very tight integration with JDT greatly simplifies the consumption of Java artifacts either being hosted on open source repositories such as Maven Central, or in your in-house Maven repository.

Website: https://www.eclipse.org/m2e/
Documentation : https://www.eclipse.org/m2e/documentation/

3.3.11 Mylyn

Mylyn is the task and application lifecycle management (ALM) framework for Eclipse. It provides:

- A task-focused interface : Realigns the IDE around tasks so that you see only the code that's relevant
- A task management tool for developers
- A broad ecosystem of Agile and ALM integrations : Dozens of extensions integrate Mylyn with ALM and developer collaboration tools

Website: https://www.eclipse.org/mylyn/
Documentation: https://www.eclipse.org/mylyn/start/

3.3.12 EGit/JGit

EGit is an Eclipse Team provider for the Git version control system. Git is a distributed SCM (Source Control Management) which means every developer has a full copy of all history of every revision of the code, making queries against the history very fast and versatile.

Website: https://www.eclipse.org/egit/
Documentation: https://www.eclipse.org/egit/documentation/

3.3.13 Subversive SVN

The Subversive project is aimed to integrate the Subversion (SVN) version control system with the Eclipse platform.

Website: https://www.eclipse.org/subversive/
Documentation: https://www.eclipse.org/subversive/documentation.php

©Consortium GEMOC, 2012- 2016

Annex 1 - Internal Eclipse feature list

This annex presents screenshots of the Eclipse features installed in the Studio version 0.0.1 – 2014-05-12.

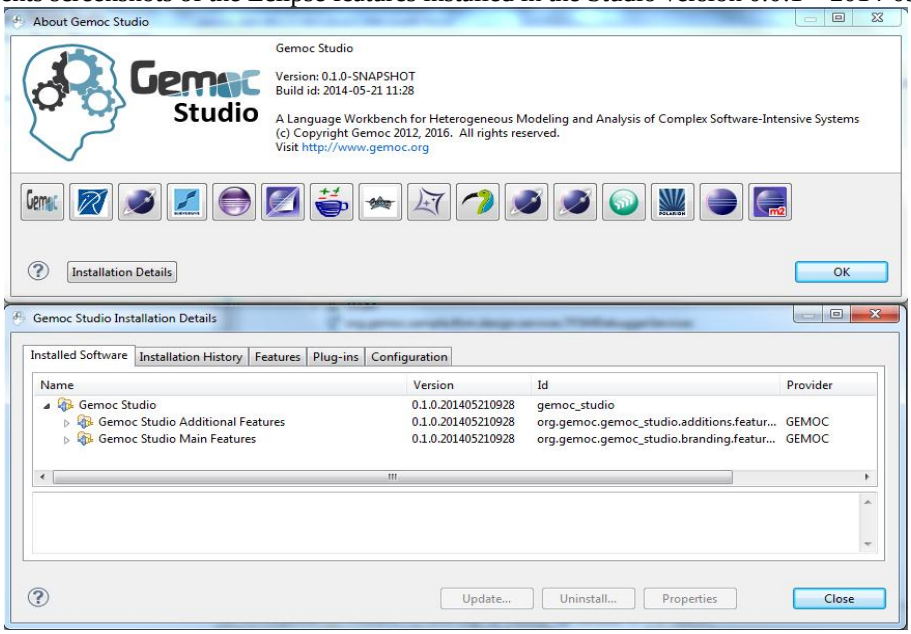

Figure 5. Installation detail page

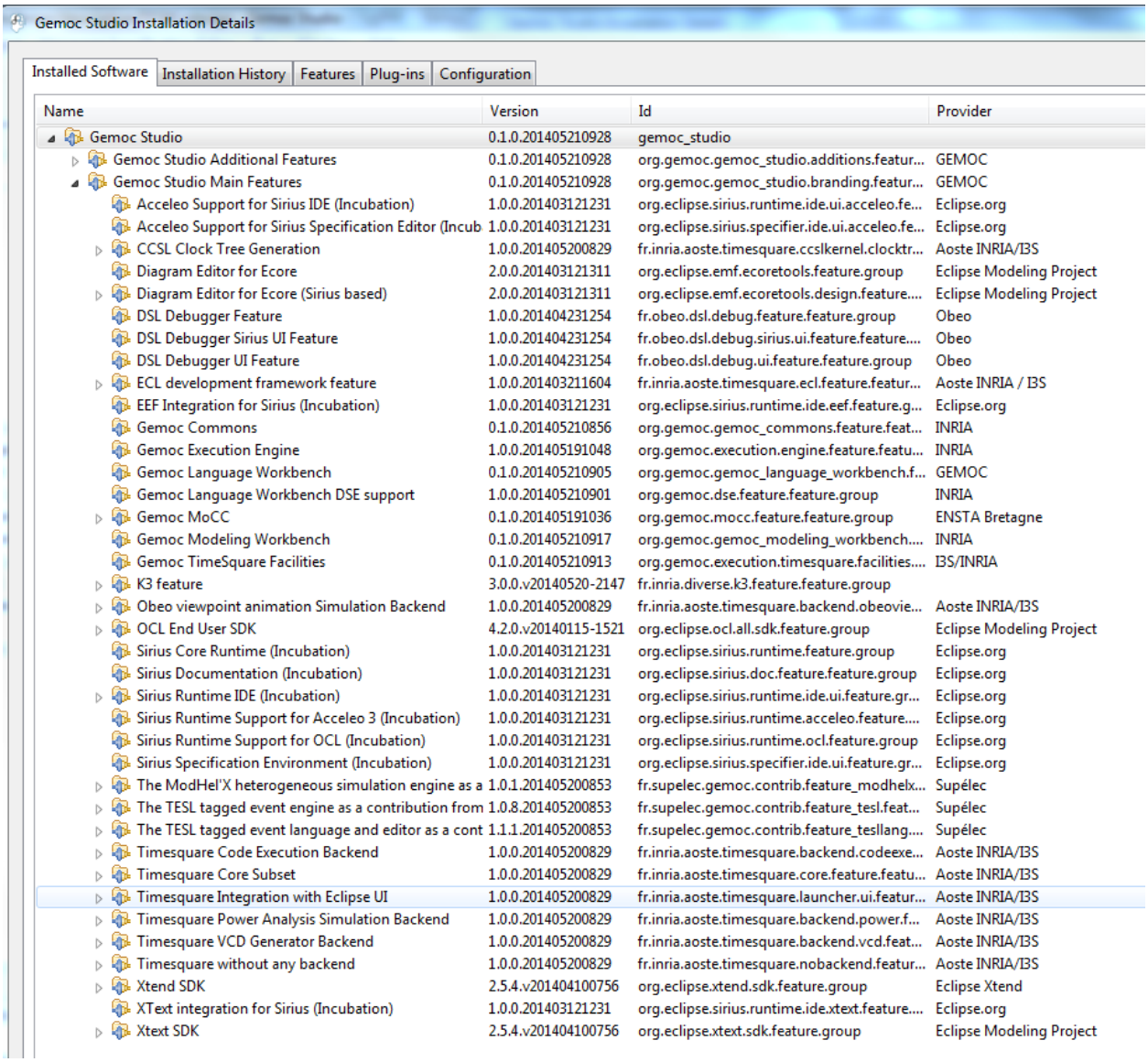

Figure 6. GEMOC main Eclipse features

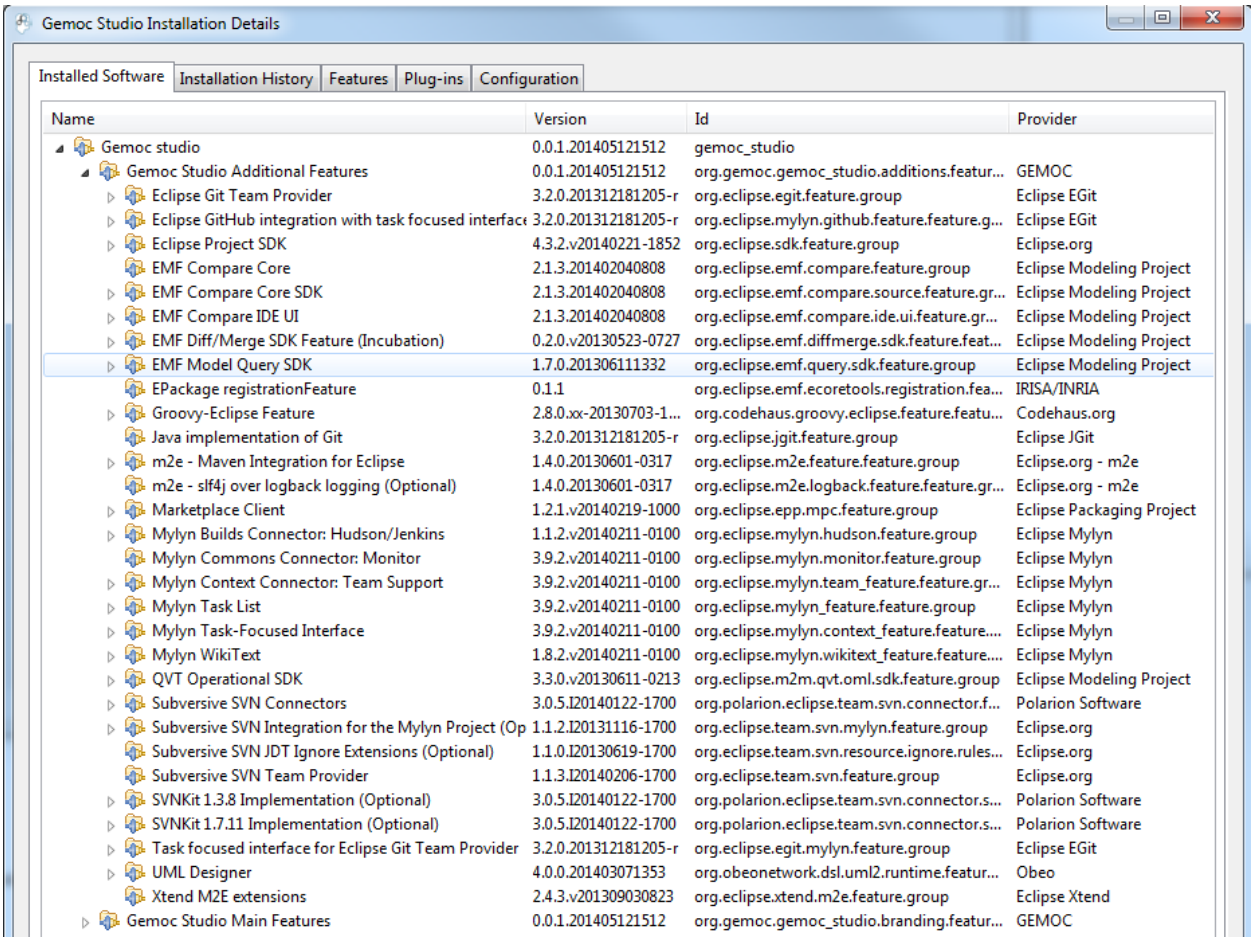

Figure 7. GEMOC additional Eclipse features