



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 2.0

ANR INS GEMOC / Task 4.1	Version: 2.0
Eclipse-based tool to model heterogeneous model execution, and GEMOC studio (SOFTWARE)	Date: 12/9/15
D4.1.2	

Document Control

	- :	A :	B :	C :	D :
Written by Signature	Mélanie Bats - Obeo				
Approved by Signature					

Revision index	Modifications
A	
B	
C	
D	

Authors

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

ANR INS GEMOC / Task 4.1	Version: 2.0
Eclipse-based tool to model heterogeneous model execution, and GEMOC studio (SOFTWARE)	Date: 12/9/15
D4.1.2	

Table of Content

1.Introduction	4
1.1Purpose.....	4
1.2Definitions & acronyms	5
2.GEMOC Studio Overview	6
3.GEMOC Studio v2.0 Release notes	8
3.1Migration & Refactoring	8
3.1.1Model debugging	9
3.1.2Concurrency modeling and analysis	9
3.1.3Behavioral coordination.....	9
3.2GEMOC Discovery catalog	9
3.3Documentation update	10
3.4Tutorials update	10
3.5Sample update.....	10

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 2.0.0 is distributed as deliverable D4.1.2 for task T4.1.

ANR INS GEMOC / Task 4.1	Version: 2.0
Eclipse-based tool to model heterogeneous model execution, and GEMOC studio (SOFTWARE)	Date: 12/9/15
D4.1.2	








The GEMOC Initiative

On the Globalization of Modeling Languages

[Home](#)
[About Us](#)
[Events](#)
[Projects](#)
[Studio](#)
[Publications](#)

Studio Download

Stable versions

	Gemoc Studio	V2.0.0	2015_12_05	<ul style="list-style-type: none"> Linux (gtk) 32 Bit (~570Mo) Linux (gtk) 64 Bit (~570Mo) Mac OS X (cocoa) 64 Bit (~570Mo) Windows 32 Bit (~570Mo) Windows 64 Bit (~570Mo) 	Release notes
	Gemoc Studio	V1.0.0	2015_06_03	<ul style="list-style-type: none"> All OS 	Release notes
	Gemoc Studio	V0.1.3	2014_12_19	<ul style="list-style-type: none"> all OS 	Intermediate version
	Gemoc Studio	V0.1.2	2014_11_14	<ul style="list-style-type: none"> all OS 	Intermediate version
	Gemoc Studio	V0.1.1	2014_09_01	<ul style="list-style-type: none"> all OS 	Intermediate version

Latest tweets

RT @bruncedric: Want to know how this guy relates to EMF Core ? Vote for my talk ! #eclipsecon <https://t.co/G9lFc2ZOZE> <https://t.co/lg9QSZE...> 15 days ago

RT @eclipsecon: Session Proposal: Modeling Avengers : Open Source Technology Mix for Saving the World <https://t.co/n07OD2LhKc> #EclipseCon 16 days ago

RT @bruncedric: Just proposed a talk mixing Eclipse Modeling, @optaplanner, @EclipseSirius, @text & @gemocinitiative among others <https://t.co/...> 17 days ago

[Follow @gemocinitiative](#)

ANR INS GEMOC / Task 4.1	Version: 2.0
Eclipse-based tool to model heterogeneous model execution, and GEMOC studio (SOFTWARE)	Date: 12/9/15
D4.1.2	

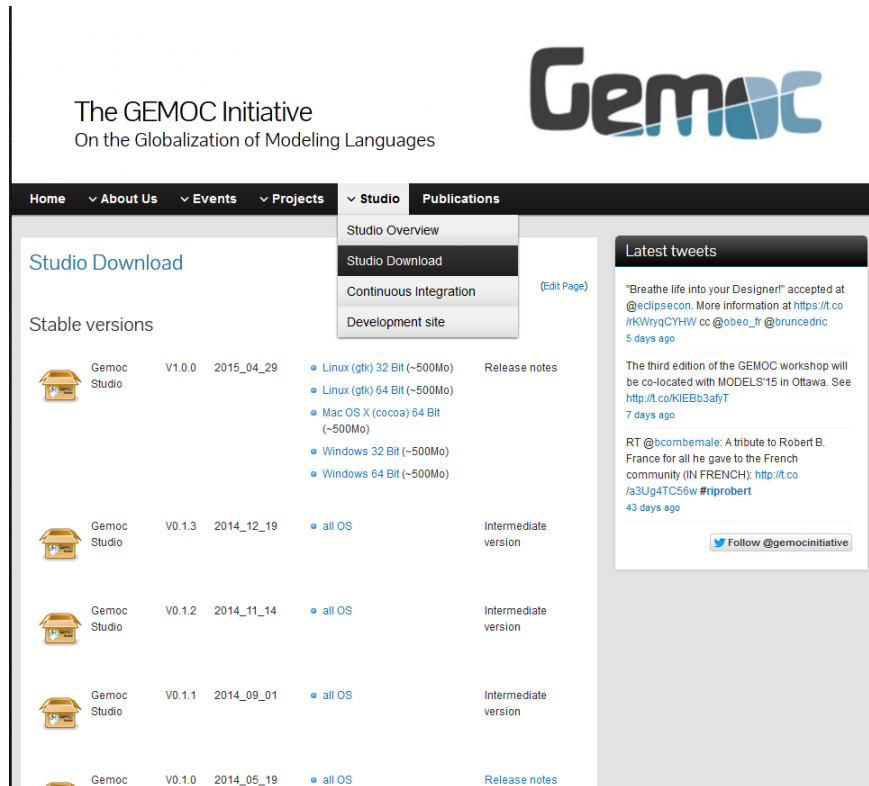


Figure 1. Screenshot of the download page

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.
- **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

ANR INS GEMOC / Task 4.1	Version: 2.0
Eclipse-based tool to model heterogeneous model execution, and GEMOC studio (SOFTWARE)	Date: 12/9/15
D4.1.2	

- **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 domain experts: 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.

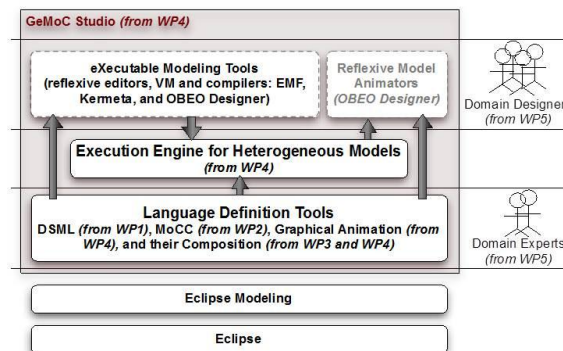


Figure 2. Studio general view

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

Figure 3 shows the GEMOC Language Workbench. It illustrates the GEMOC Studio on TFSM (Timed Finite State Machine) language which is one of the official samples deployed with the Studio.

Figure 4 shows the GEMOC Modeling Workbench. It illustrates the GEMOC Studio on the execution of a TFSM model.

ANR INS GEMOC / Task 4.1	Version: 2.0
Eclipse-based tool to model heterogeneous model execution, and GEMOC studio (SOFTWARE)	Date: 12/9/15
D4.1.2	

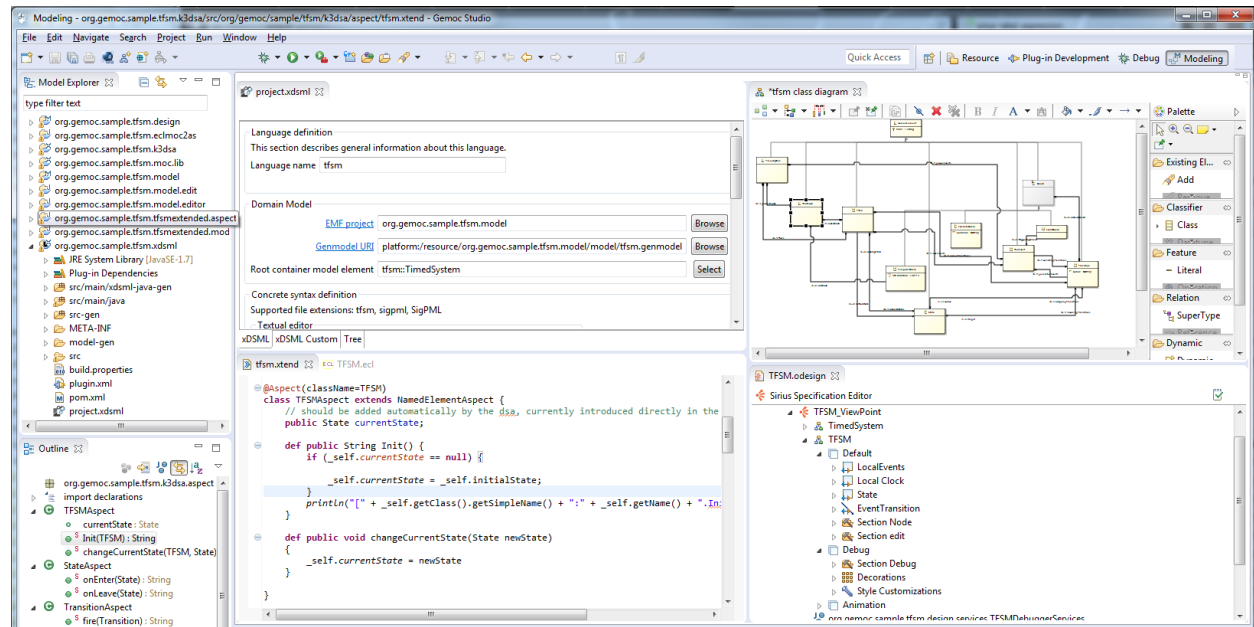


Figure 3. Screenshot of GEMOC Studio Language Workbench on the TFSM (Timed Finite State Machine) example

ANR INS GEMOC / Task 4.1	Version: 2.0
Eclipse-based tool to model heterogeneous model execution, and GEMOC studio (SOFTWARE)	Date: 12/9/15
D4.1.2	

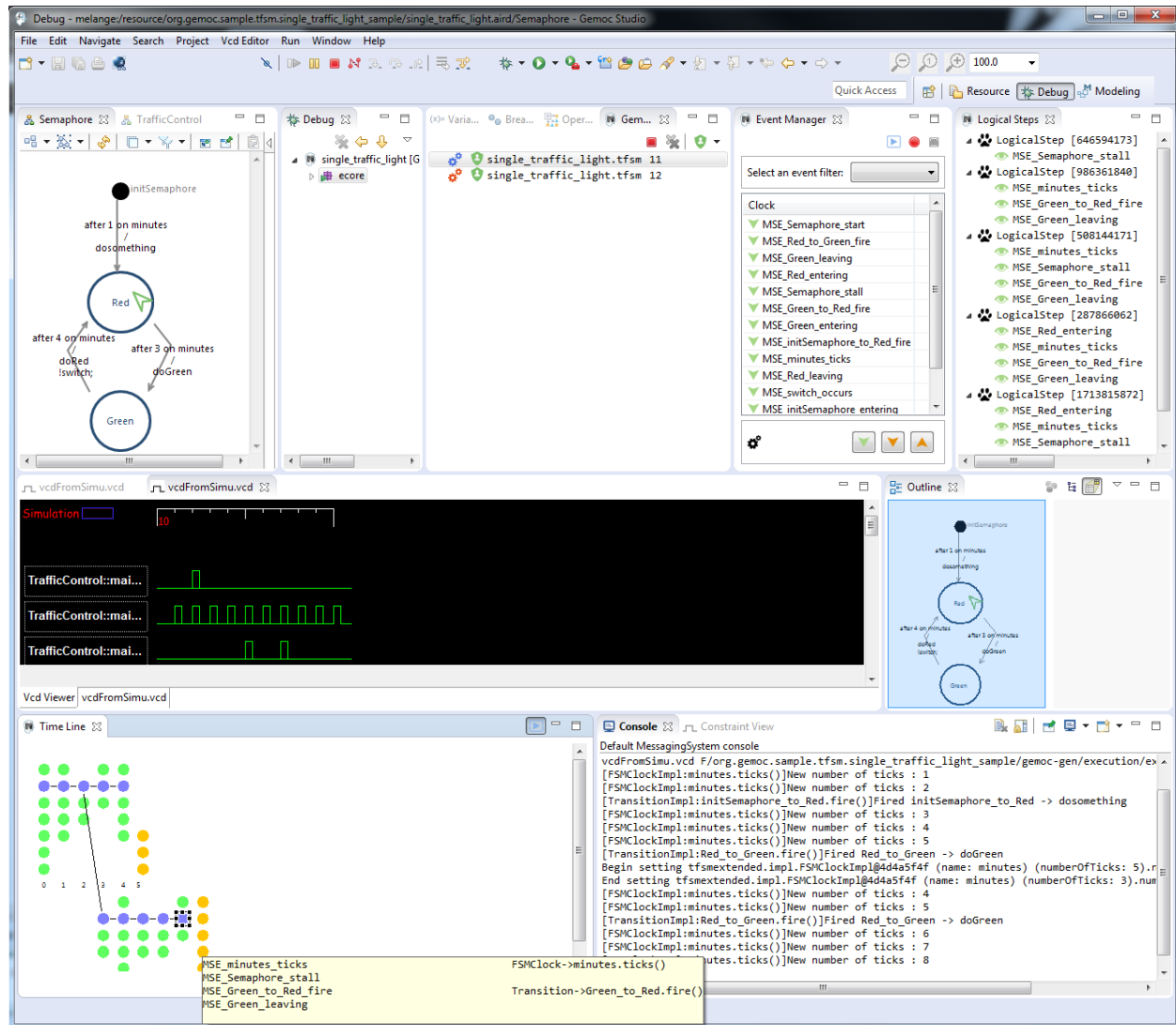


Figure 4. Screenshot of GEMOC Studio Modeling Workbench on the TFSM example (execution and animation)

3. GEMOC Studio v2.0 Release notes

The purpose of this section is to present the work done on the GEMOC Studio between the v1.0 and the v2.0.

3.1 Migration & Refactoring

Since the 1.0 version, the GEMOC studio source code was migrated from the INRIA forge to two public organization on

Github : <https://github.com/gemoc> and <https://github.com/SiriusLab/>

A refactoring was done to separate the multiple existing components to address the following complementary and incremental scenarios:

- Model debugging
- Concurrency modeling & analysis
- Behavioral coordination

The architecture has been updated to enable these scenarios. On Figure 5. GEMOC Studio architecture for incremental scenarios, components in green are intended to Model Debugging, components in grey are intended to Concurrency modelling & analysis, and components in blue are intended to Behavioral coordination.

ANR INS GEMOC / Task 4.1	Version: 2.0
Eclipse-based tool to model heterogeneous model execution, and GEMOC studio (SOFTWARE)	Date: 12/9/15
D4.1.2	

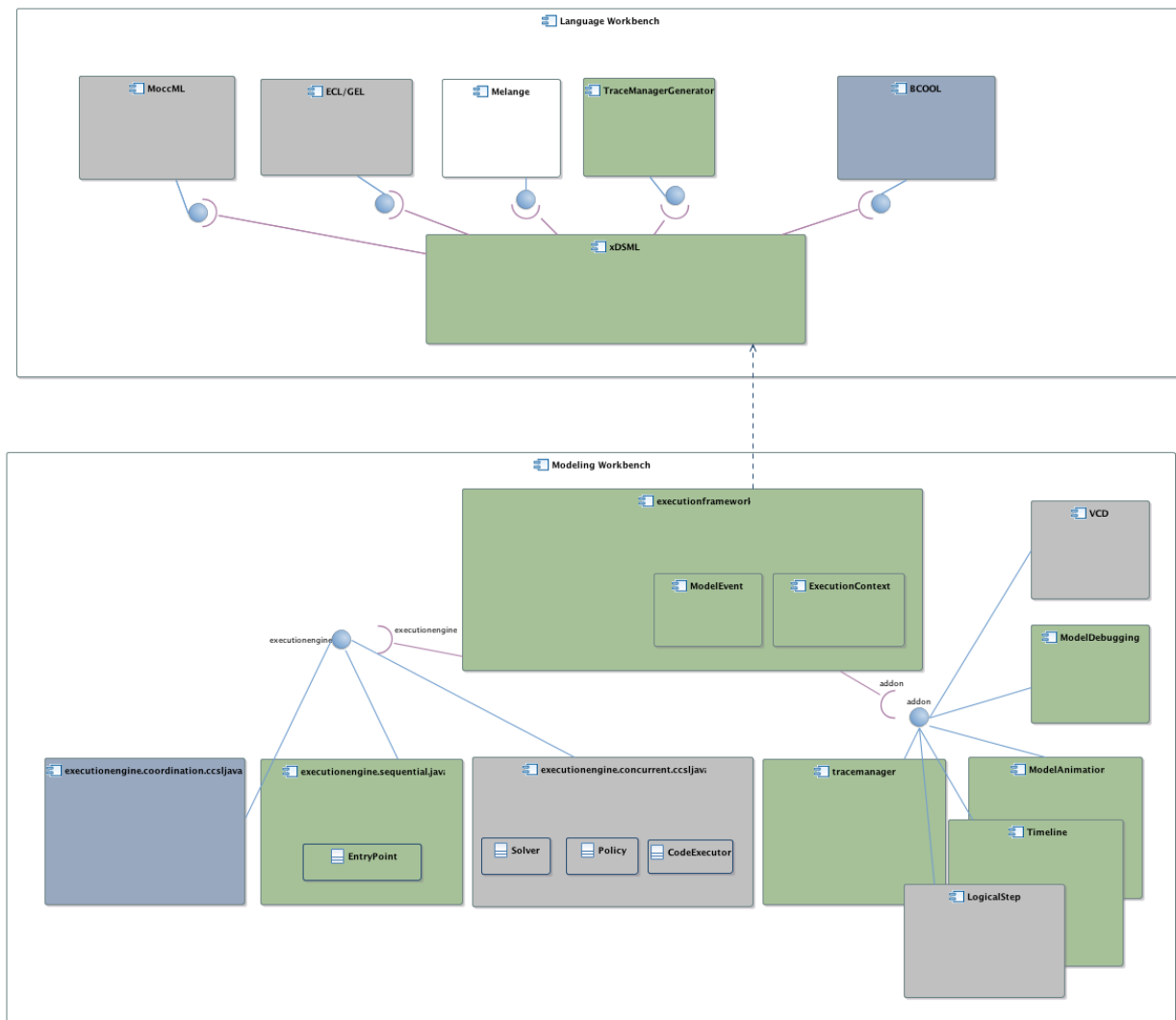


Figure 5. GEMOC Studio architecture for incremental scenarios

3.1.1 Model debugging

Model debugging regroups the GEMOC studio components used to provide the sequential execution, the trace management capability, the debugging facilities and the animation. These components help to bring execution and simulation in domain specific tooling built with Sirius and Eclipse Modeling Framework.

- Generic execution engine
- Generic animation framework : In this new version, it is now possible to contribute the variable changes to the variable view in debug mode. There is no more need to defined by hand the mapping between the execution engine and the dynamic data, it is now automatically provided by the animation framework.
- Trace management : To provide efficient and adapted execution trace management facilities, it is now possible to generated a multidimensional and domain-specific trace metamodel for the considered sequential xDSML defined using Kermeta. This option is available in the language workbench by right-clicking on the xdsml file of a sequential xDSML, and by selection "Generate Trace Addon Project". A new Eclipse plugin is then created containing the trace metamodel as an ecore file, along with corresponding generated Java code, and with a dedicated trace manager that provides a selection of services to construct a trace, query a trace, and restore the executed model in a previous state. This generated eclipse

ANR INS GEMOC / Task 4.1	Version: 2.0
Eclipse-based tool to model heterogeneous model execution, and GEMOC studio (SOFTWARE)	Date: 12/9/15
D4.1.2	

plugin is also a GEMOC Engine Addon that can be used in the modeling workbench to capture an execution trace, and to enable Omniscient Debugging features (see dedicated section). It can be enabled as any other addon in the Run Configuration of a model. (see Figure 6. Screenshot of Trace Management generator)

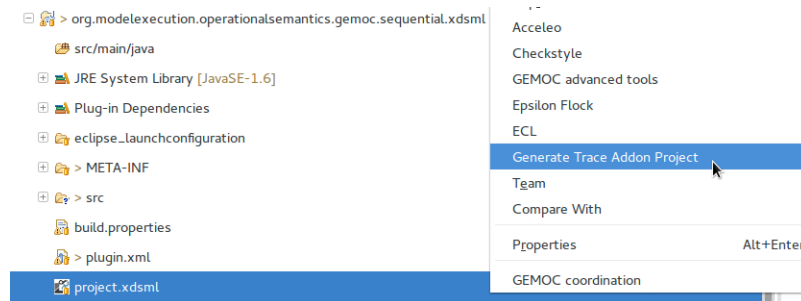


Figure 6. Screenshot of Trace Management generator

- Omniscient debugging : If a trace addon was generated for a sequential xDSML, and is made available in the modeling workbench, then Omniscient Debugging facilities are automatically enabled when a model is executed with the trace addon. These new facilities allow to explore the previous states of a model during its execution. There are two ways to use these backward facilities. First, the Multidimensional Timeline view provides an interactive representation of the execution trace being captured. When double-clicking on a previous state represented in the timeline, the model is brought back into this state. Moreover, the timeline represents all the different dimensions captured in a trace, each being the sequence of values taken by one specific element of the model. When double-clicking on a value that was reached by an element, the complete model is brought back in the state corresponding to this value. Second, the Eclipse debugging view of the GEMOC Studio possesses additional buttons in the toolbar to go backward in an execution. They provide the following actions: backInto, backOver and backOut. All these actions behave similarly to their forward counterparts, but follow execution steps in the opposite direction. (See Figure 7. Screenshot the Omniscient Debugger)

ANR INS GEMOC / Task 4.1	Version: 2.0
Eclipse-based tool to model heterogeneous model execution, and GEMOC studio (SOFTWARE)	Date: 12/9/15
D4.1.2	

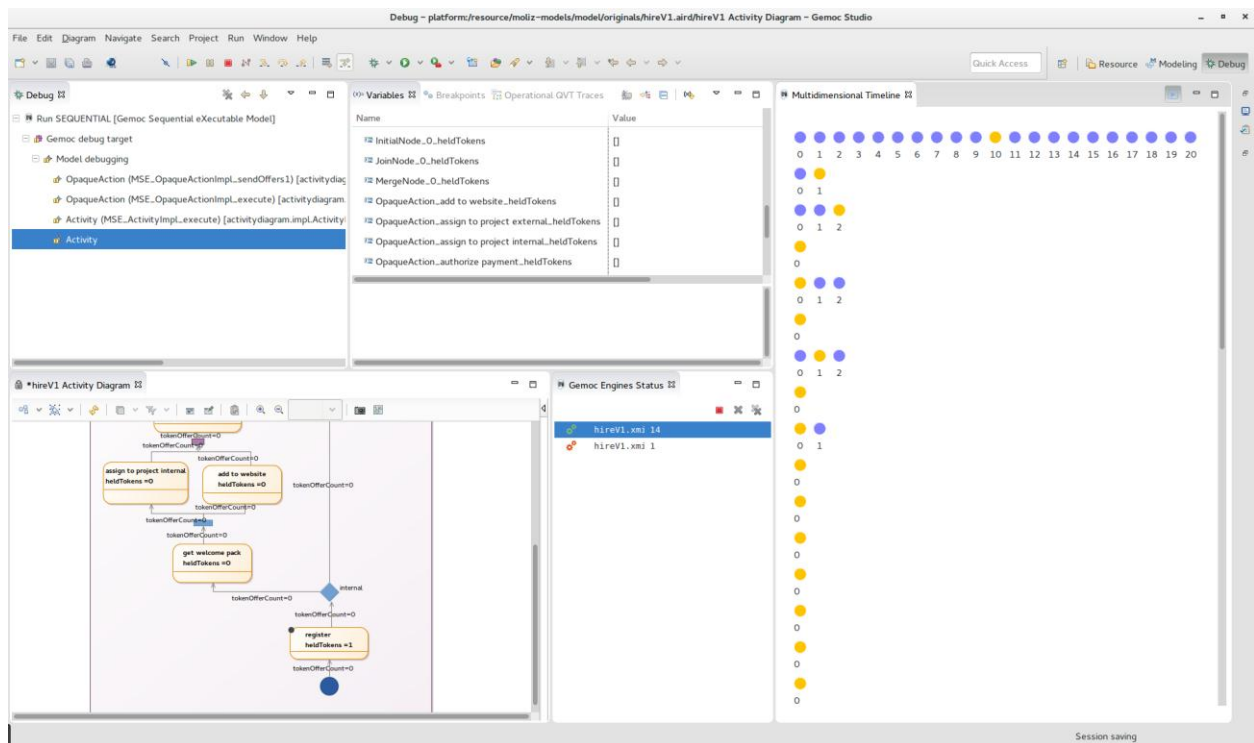


Figure 7. Screenshot the Omniscient Debugger and its multidimensional execution trace

3.1.2 Concurrency modeling and analysis

The concurrency modeling and analysis features are based on the model debugging components and on other specific components :

- MoCCML: the Model of Concurrency and Communication Modeling Language is a metalanguage dedicated to the specification of constraints between some events. These constraints reflect the causal, concurrent and temporal aspect of the operational semantics of a language.
- ECL: the Event Constraint Language is a lightweight extension of OCL (Object Constraint Language) with the notion of event and with behavioral invariant. The Event are used to define the DSE and the behavioral invariants are used to constraint the partial ordering of the DSE. These constraints are defined in MoCCML.
- Gemoc Events Language (GEL): metalanguage used to specify the Communication Protocol between the MoCC and the DSA of a Concurrency-aware xDSML. The GEMOC Studio integrates a textual editor for GEL (based on Xtext), its model-level target language (microGEL) and the compiler to unfold a GEL specification for a specific model.
- 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 is based on the formal Clock Constraint Specification Language (CCSL), which allows the manipulation of logical time. Logical time is a relaxed form of time where any events can be taken as a reference for counting (e.g. do something every 30 openings of the door). It can be used for [specifying classical and multiform real-time requirements](#) as well as formally specifying constraints on the behavior of a model (either a UML-based or a DSL model). TimeSquare can solve these constraints to conduct a simulation or to construct the state space representing all possible partial orders of the clocks (if finite).
- Concurrent execution engine: The concurrent execution engine allows to simulate conjointly

ANR INS GEMOC / Task 4.1	Version: 2.0
Eclipse-based tool to model heterogeneous model execution, and GEMOC studio (SOFTWARE)	Date: 12/9/15
D4.1.2	

heterogeneous models. More precisely, based on a coordination model (usually generated from a BCOoL specification), it starts and coordinates the execution engine of each models that are part of the coordination. It results an execution where the different model interacts to provide an emerging behavior according to the coordination.

- Analysis tool : The Clocksystem exhaustive exploration tool was contributed to the GEMOC Studio. The code generator to create the exploration file was updated. Finally, some plugins dedicated to the Clocksystem service calls were integrated to the studio.

3.1.3 Behavioral coordination

The last scenario supported by the GEMOC Studio is the ability to specify behavioral coordination of possibly heterogeneous models. This is done thanks to a coordination framework based on:

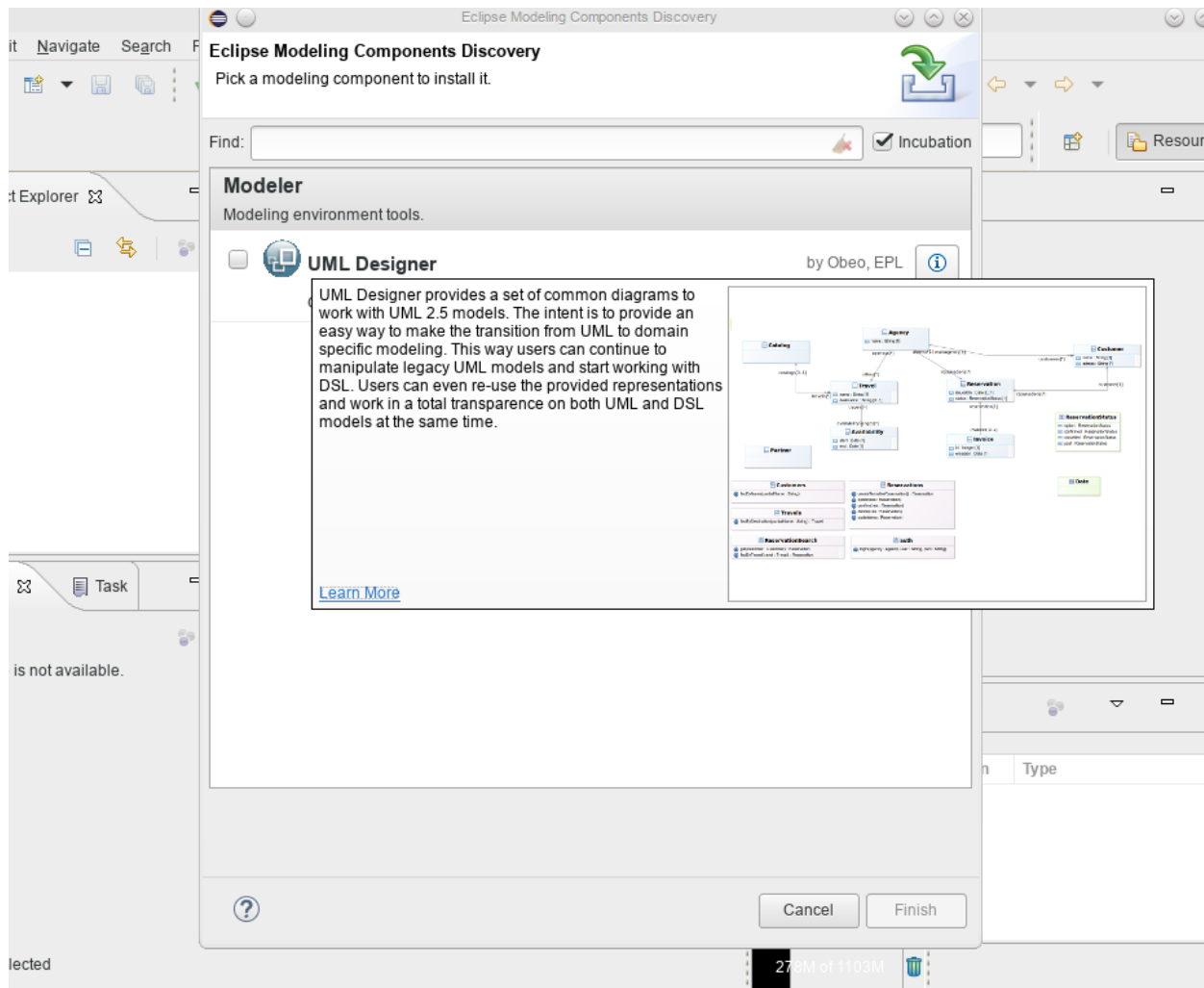
- BCOoL : a (meta)language to capture behavioral coordination patterns between DSMLs. It is a dedicated language to system designer that eases the understanding and adaptation of coordination patterns. By instantiating a coordination pattern, models can be automatically coordinated. The generated coordination conforms to a formal language, i.e., MOCCML, thus enabling the verification and validation of the coordinated system. While the GEMOC studio supports several facilities, e.g., editing, graphical representation, animation and execution of domain-specific tools, BCOoL adds coordination facilities.
- Coordination Engine : a dedicated launcher allows to start heterogeneous models and coordinate their execution according to BCOoL patterns

3.2 GEMOC Discovery catalog

In order to reduce the size of the GEMOC Studio, a new discovery catalog feature was developed. This component provides a simplified UI to allow the installation of additional components in the studio. The installable components catalog is hosted on the GEMOC github repository : https://github.com/gemoc/gemoc-studio/blob/master/gemoc_studio/discovery/catalog.xmi

The UI is accessible to the GEMOC Studio end user from the toolbar menu and shows for each component referenced in the catalog its name, a small description, a website URL, a screenshot. To install a new component the end user just need to select the component and click on the *Install* button.

ANR INS GEMOC / Task 4.1	Version: 2.0
Eclipse-based tool to model heterogeneous model execution, and GEMOC studio (SOFTWARE)	Date: 12/9/15
D4.1.2	



3.3 Documentation update

The documentation of the studio is available :

- on Github : <http://gemoc.github.io/gemoc-studio/>
- in the Gemoc Studio bundle : Help > Help Contents > GEMOC Studio

The documentation contains a User Guide and the tutorials.

3.4 Tutorials update

The tutorials were updated and are available on :

http://gemoc.github.io/gemoc-studio/publish/tutorials/html_single/TutorialMain.html

The Marked Graph tutorial was reviewed to fixed some issues :

http://gemoc.github.io/gemoc-studio/publish/tutorial_markedgraph/html_single/GuideTutorialMarkedGraph.html

and a new tutorial presenting the SigPML example was contributed :

http://gemoc.github.io/gemoc-studio/publish/tutorial_sigpml/html_single/Tutorial_SigPML.html

ANR INS GEMOC / Task 4.1	Version: 2.0
Eclipse-based tool to model heterogeneous model execution, and GEMOC studio (SOFTWARE)	Date: 12/9/15
D4.1.2	

3.5 Sample update

The samples integrated to the GEMOC Studio were updated.

Additionally some more advanced examples are publicly available on the organization repository

<https://github.com/gemoc/>:

- Arduino Modeling : <https://github.com/gemoc/arduino modeling> : a sequential language that simulates Arduino components
- DSL for Activity Diagram : <https://github.com/gemoc/activity diagram> : several implementations of the Activity Diagram from FUML. (it features both concurrent and sequential versions)
- Farming Modeling : <https://github.com/gemoc/farming modeling> : a DSL that simulates a farm. It uses a sequential engine.