CONVENTION ANR-12-INSE-0011

ANR INS GEMOC

D5.3.1 - Uses-case models and simulation
Task 5.3

Version 1.0

DOCUMENT CONTROL

	–: 31/03/2016	A:	B:	C:	D:
Written by Signature	S. MADELENAT (TRT)				
Approved by Signature	B. COMBEMALE (INRIA)				

Revision index	Modifications
–	Initial version.
A	
B	
C	
D	

Authors

Author	Partner	Role
Sébastien Madelénat	Thales	Lead author
Name	Organisation	Contributor

Contents

Uses-case models and simulation

1. Introduction

The GEMOC project targets a language design studio providing methods and tools to ease the design and integration of new MoCCs and executable DSMLs (xDSMLs) relying on the use of proven technologies developed in previous research projects such as Cometa, CCSL, Kermeta and the meta-modelling pattern to build xDSML in order to define:

- Modelling languages with associated methods and tools for the modelling of both MoCCs and xDSMLs;
- A single cooperative heterogeneous execution framework parameterized by the MoCCs and xDSMLs definitions;
- A global MoCCs and xDSMLs design methodology encompassing these two items;A formal specification of the previous cooperation framework to prove its completeness, consistency and correctness with respect to the cooperative heterogeneous model execution needed.

1.1 Purpose

This document aims at presenting the xCapella demonstrator installation for study and experimentation.

1.2 Definitions, Acronyms and Abbreviations

- 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.
- 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.

1.3 Intended Audience

This document mainly targets the Gemoc Studio end-users.

1.4 Summary

This document defines provides a description of the xCapella demonstrator installation and execution.

2. xCapella

2.1 Set up

2.1.1 Download links

Here is the list of download URL of required software.

- openSUSE Leap 42.1 64 bits (native installer):
  https://software.opensuse.org/421/en
- openSUSE Leap 42.1 64 bits (virtual machine):
  http://www.osboxes.org/opensuse/
- xCapella demonstrator platform:
  http://gemoc.org/pub/xCapella.zip

In case of problem with a listed link, feel free to contact the authors.

2.1.2 Operating System

The xCapella demonstrator has been successfully experimented on a Linux as well as a native platform as a Virtual Machine using VMWare Workstation 12. The Linux distribution is OpenSUSE Leap 42.1 64 bits.

2.1.3 xCapella demonstrator installation

The xCapella demonstrator is a GEMOC Studio modeling platform extended with Capella and xCapella extensions. The demonstrator installation requires to download and unzip the xCapella demonstrator platform in your Linux distribution.

2.1.4 xCapella demonstrator

To run the demonstrator, execute the GEMOC Studio executable from your unzipped xCapella demonstrator. The xCapella demonstrator offers 3 execution demonstration of standard Capella models:

- The mode automata concurrent execution illustrating an operational semantics interpretation of an experimental mode automata. To execute the mode automata demonstration, use the command Debug → ModeAutomata.
- The data flow concurrent execution illustrating an operational semantics interpretation of the standard Capella data flow. To execute the data flow demonstration, use the command Debug → Dataflow.
- The mode automata and the data flow coordinated execution illustrating the coordination of both operational semantics listed above. To execute coordination demonstration, use the command Debug → Coordination. The result shall be similar to the Figure 2.1.

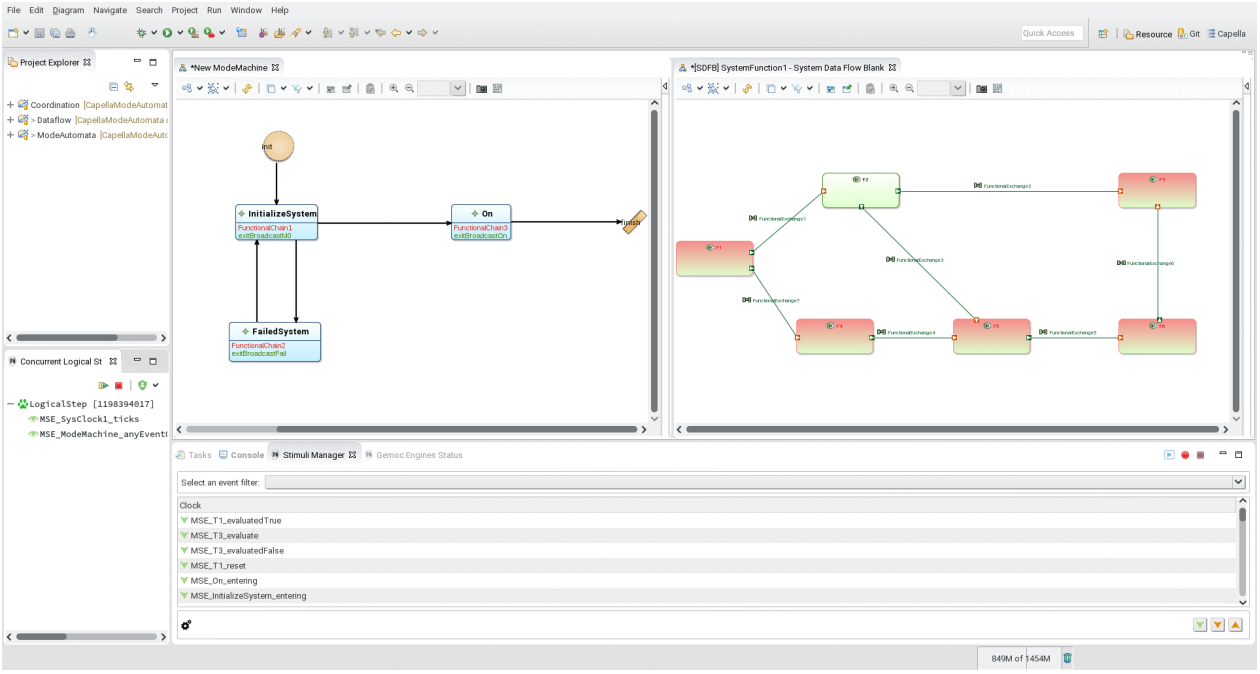

Figure 2.1: xCapella coordination execution example snapshot

3. Conclusion

The xCapella demonstrator shows the GEMOC Studio capability of extending an existing industrial engineering environment.
This demonstration has been made using Capella as is, i.e. without any change in the original code.
More over, the xCapella demonstrator illustrates the robustness of the GEMOC pattern. This demonstrator adds AS information used in DSA definition by the standard Capella extension mechanism (Capella Studio) instead of the extension mechanism provided by GEMOC Studio (Melange).

