Grant ANR-12-INSE-0011

GEMOC
ANR Project, Program INS

WP5 – GEMOC EXPERIMENTATIONS

D5.3.1 Use cases models and simulation (SOFTWARE)
Task 5.3

V0

DOCUMENT CONTROL

	- :	A :	B :	C :	D :
Written by Signature	Le Noir (TRT)				
Approved by Signature					

Revision index	Modifications
A	
B	
C	
D	

Authors

Author	Partner	Role
Jérôme Le Noir	TRT	Lead author
Xavier Crégut	IRIT	Contributor
Ali Koudri	TRT	Contributor
Julien DeAntoni	I3S	Contributor
Joël Champeau	ENSTA-B	Contributor
Marc Pantel	IRIT	Contributor

Table of content

D5.3.1 Use cases models and simulation

1. Introduction

The GEMOC project targets a language design studio providing methods and tools to ease the design executable DSMLs (xDSMLs) relying on the use of proven technologies developed in previous research projects such as Cometa, CCSL, Kermeta. In Deliverable D1.1.1, the architecture of a Concurrency-Aware xDSML has been defined. Such an xDSML is defined by its Abstract Syntax (AS), its Domain-Specific Actions (DSAs) (Execution Functions, Execution Data also called Execution State), its Model of Concurrency and Communication (MoCC) (events and constraints on these events) and its Domain-Specific Events (DSEs). Essentially, the AS specifies the concepts of the domain and their relations, the DSAs specify the dynamic information of the xDSML and how they evolve during the execution, and the MoCC specifies the concurrency aspects of the xDSML (synchronization, causalities, etc...). The D1.1.1 also describes a methodology as well as guidelines (good and bad practices) to define an xDSML from its abstract syntax.

1.1 Purpose

This document describes the experiments conducted with the language workbench from the toolset developed in GEMOC for two of the three use cases provided by WP5.

Two case studies are implemented for the validation of the solutions established in GEMOC. The industrial case studies rely on several heterogeneous cooperative xDSMLs relying on various MoCCs that are implemented using the GEMOC Studio provided by WP4, using the processes, methods and tools defined in the deliverable D1.1.1. The experiments from the task T5.2 have defined the languages and then models for validating the proposed technologies through simulation are illustrated in this deliverable.

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.
- **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 GEMOC in WP1, WP2, WP3 and WP4 partners.

1.4 References

- D5.1.1: Technical Requirements, uses-cases specification and metrics for experimentations
- [UML2012] The Unified Modeling Language release 2.4.1, Infrastructure & Superstructure, The Object Management Group, 2012 (http://www.omg.org/spec/UML)
- [fUML2013] The Semantics Of A Foundational Subset For Executable UML Models, release 1.1, The Object Management Group, 2013 (http://www.omg.org/spec/FUML)
- [SysML2013] The System Modeling Language, release 1.3, The Object Management Group, 2013 (http://www.omg.org/spec/SysML)
- [OCL2012] The Object Constraint Language, release 2.3.1, The Object Management Group, 2012 (http://www.omg.org/spec/OCL)
- [Simulink2014] Simulink Reference Documentation, release 2014a, The MathWorks, 2014 (http://www.mathworks.fr/fr/help/simulink/index.html)
- [Stateflow2014] Stateflow Reference Documentation, release 2014a, The MathWorks, 2014 (http://www.mathworks.fr/fr/help/stateflow/index.html)
- [AADL2012] Architecture Analysis and Design Language, release 2.0, Society of Automotive Engineers, 2012 (http://standards.sae.org/as5506b/)
- [D1.13:GA2011] D1.13: Functional Modeling Guidelines, 2011, GeneAuto consortium (https://gforge.enseeiht.fr/docman/view.php/25/4247/D1.13+Functional+Modelling+Guidelines_1.3_110126_AnTo.pdf)
- [D1.14:GA2009] D1.14: Modeling Guidelines for State Diagrams, 2009, GeneAuto consortium (https://gforge.enseeiht.fr/docman/view.php/25/3399/D1.14_Modelling_guidelines_for_state_diagrams_v1.1_090130_AnTo.pdf)
- [BL2014] GeneAuto/HiMoCo/P Block Library Specification Language, work in progress, 2014, P/HiMoCo consortium (http://block-library.enseeiht.fr/bl/)
- [Kahn74] G. Kahn, "The semantics of a simple language for parallel programming," in Proc. IFIP Cong. '74, Amsterdam: NorthHolland, 1974.
- [Petri66] Carl Adam Petri. Communication with Automata. PhD Thesis, Universitat Hamburg, 1966.
- [Lee87] Edward A. Lee, David G. Messerschmitt. Proceedings of the IEEE, vol. 75, no. 9, p 1235-1245, September, 1987.
- [Harel84] David Harel. StateCharts: A Visual Formalism for Complex Systems. 1984.
- [Murthy02] Praven K. Murthy, Edward A. Lee: Multidimensional Synchronous Dataflow. IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 50, NO. 7, JULY 2002

1.5 Summary

This document defines the content of the software delivered in Task 5.3 and the location of the associated elements (source code, binary code, plugins, features, documentation, tests...).

Section 1 defines the scope of the document and presents the document structure.
Section 2 describes the content of the Thales use case.
Section 3 describes the content of the Airbus and IRIT use case.

2. THALES: Radar Simulation use case

This use case is relying on internal THALES software parts that will be publically available on https://www.polarsys.org/projects/polarsys.capella.

The Arcadia engineering method mainly focuses on functional analysis, complex architecture definition and early validation. It is highly extensible and customizable through viewpoints providing integrated specialty engineering support. Both the method (which is going to be published and standardized) and the Capella ecosystem are already operationally deployed within Thales in defence, aerospace, space, transportation and security business domains, across several countries, thanks to a large-scale rollout of model-based approaches, with hundreds of daily users worldwide, on critical operational projects.

2.1 Simulation

The xDSML components (Figure 1: DSML and MoC for Thales radar use case) for Capella have been developed according to the GEMOC methodology which has been defined in the WP1. In the context of the task 5.3 the simulation Capella animation and debug viewpoints (Figure 2: New Viewpoints (Capella debug & animation) for simulation purpose) have been developed. The xDSML components are located on the git repository at: **org/gemoc/usecases/thales**. For simulation purpose, you have to deploy the "D5.2.1 DSML and MoC for Use Cases" components within the Gemoc language workbench and to use the radar model (Figure 3: Radar model within Capella Gemoc modelling workbench) in the Gemoc modelling workbench.

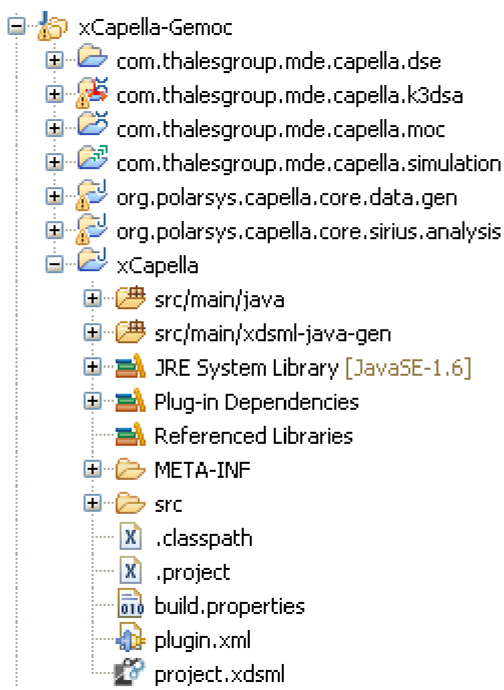

Figure 1: DSML and MoC for Thales radar use case

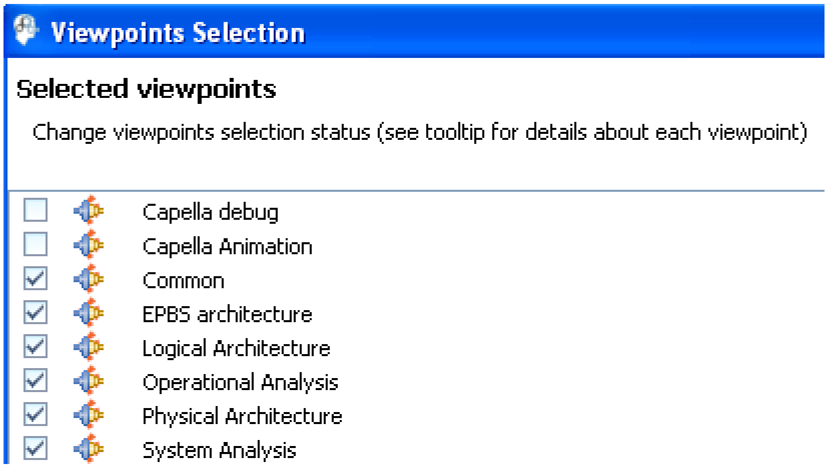

Figure 2: New Viewpoints (Capella debug & animation) for simulation purpose

2.2 Radar model

The figure below represents the physical architecture of a simplified radar model realised with the Capella Gemoc modelling workbench.

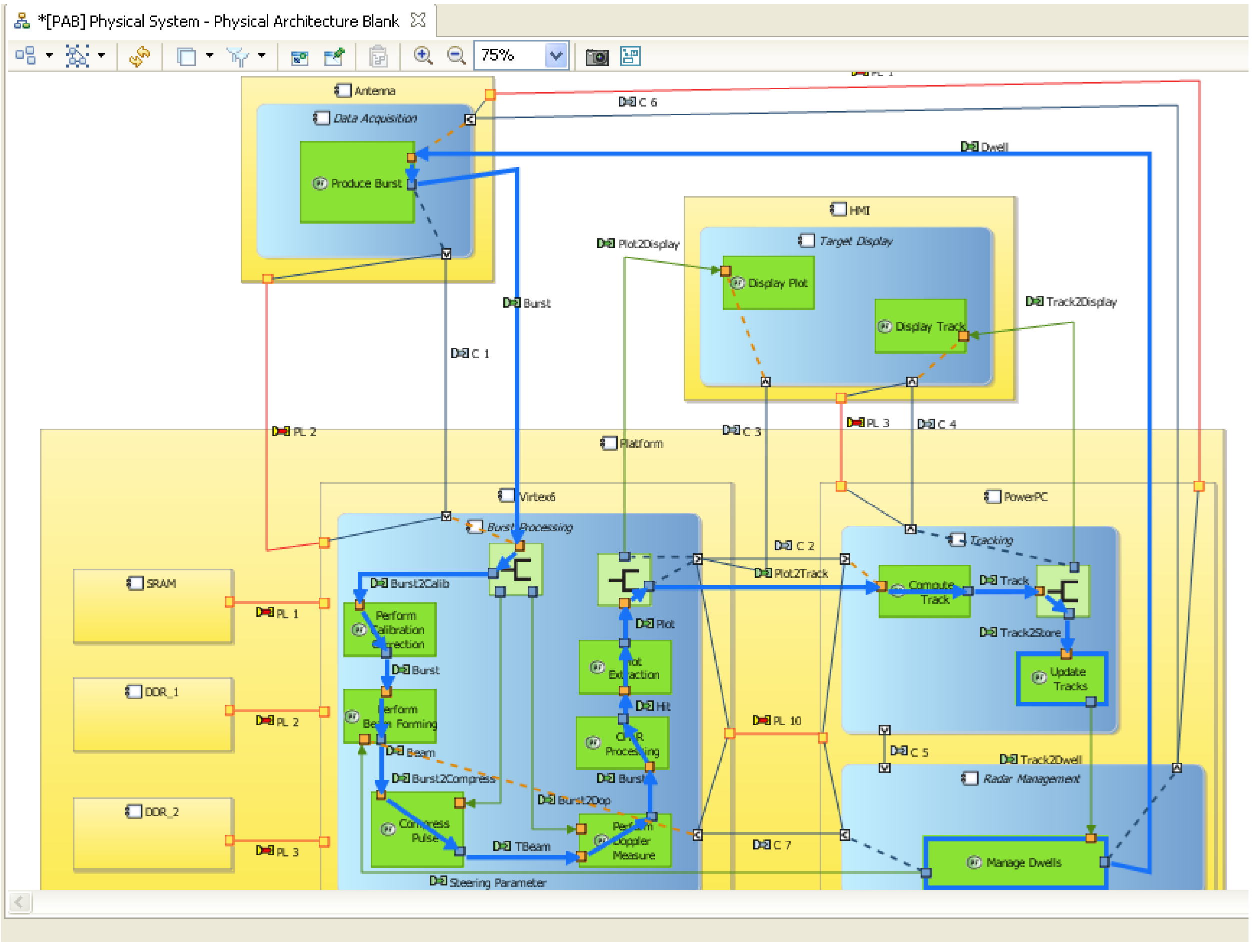

Figure 3: Radar model within Capella Gemoc modelling workbench

3. Airbus and IRIT: Maintenance training use case

In this first version of the report, the designed models are mainly used to validate the language specifications. The purpose is to provide coverage of the semantics of all model elements.

The following figures illustrate the kind of models currently handled. These models are stored in the git repository at org/gemoc/usecases/geneauto/samples.

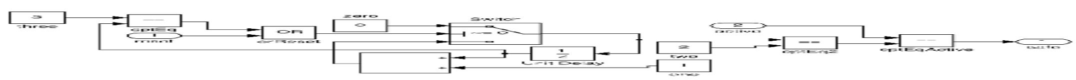

Figure 4 Observer to assess the correctness of the counter

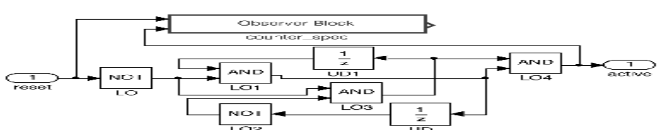

Figure 5 Implementation of a modulo 3 counter

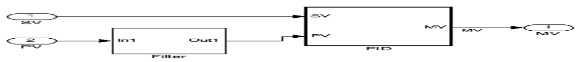

Figure 6 Interface of a PID controler

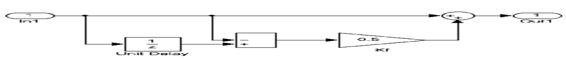

Figure 7 First order filter

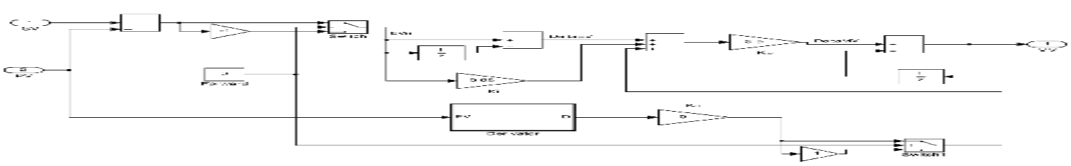

Figure 8 PID controler

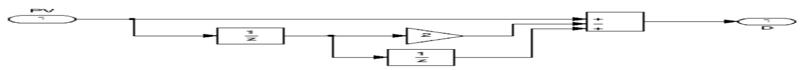

Figure 9 Derivation function