Title

An Executable Formal Framework for Inter-DSL Collaboration

Abstract

Our proposed framework provides a model-based approach for formally coordinating the execution semantics of independent Domain-Specific Languages (DSLs). The framework consists of two main parts: modeling and formal specification. The modeling part of our framework involves several key assets, including DSL Metamodels which specify the semantics of the DSLs being collaborated, Composition Metamodels which relate the semantic domains of independent DSLs, DSL Models that define instances of the metamodels, and BPMN Diagrams which describe the collaboration between the DSLs. To build these models and metamodels, we use the Eclipse Modeling Framework (EMF). In the formal specification part, the DSLs and Composition B Machines are generated from the DSLs models and metamodels to instrument DSLs with formal semantics using the B method, the Communication Sequential Process (CSP) Model is specified from the BPMN diagrams to enable inter-DSL animation and verification. The Meeduse Framework is used for generating the B specifications and the ProB tool for their animation.

Keywords

DSL · BPMN · Model Composition · Models Collaboration· Formal Methods · B Method · CSP · Animation · Verification.

**ID of accepted COORDINATION'23 paper.\*** Submission number of the accepted COORDINATION 2023 paper for which you are submitting this artefact.

2089

**Artefact format and requirements.\*** Briefly describe the format of the artefact (e.g. ready-to-use x86 VM, Docker container, source code, ...) and any relevant software or hardware requirement for compiling or using it.  
  
NOTE: the reviewers will use this information only during bidding. The full instructions for building or running the artefact should be provided in the submitted "paper" for this artefact.

The format of the artefact is source code listed in tow folders: DSLs\_Modeling, and DSLs\_Formal\_Specification.

To view the DSLs metamodels and the models, you should download and run *Eclipse Modeling Framework* (<https://www.eclipse.org/downloads/packages/release/2022-12/r/eclipse-modeling-tools>), and install BPMN2 Modeler (https://www.eclipse.org/bpmn2-modeler/downloads.php).

To animate the DSLS B specifications, you should download and run the the *ProB Animator and Model Checker* (https://prob.hhu.de/w/index.php?title=Download).

**URL for artefact download.** Provide a URL for retrieving the artefact.

<https://github.com/SalimChehida/Inter-DSL-Collaboration.git>

**Artefact validation instructions.** If relevant, explain how to verify whether the artefact download was successful (e.g. computing its sha256sum).

Not relevant

**Copy of the accepted COORDINATION'23 paper.** Upload a copy of the paper accepted at COORDINATION 2023 for which you are submitting this artefact. The file must be in PDF format. No file chosen

OK

**Artefact instructions.** Upload the detailed instructions for artefact reviewers. Supported file extensions: .pdf, .html, .txt. No file chosen

To be completed

**# An Executable Formal Framework for Inter-DSL Collaboration**

A Domain-Specific Language (DSL) is used for building models appropriate to specific application domain or specific aspect of the system. In many cases, a set of models from heterogeneous and independent DSLs are collaboratively combined to specify the same system. This need to define explicit links between the various models and also to manage the collaboration between them.

This directory provides the several assets of our approach for inter-DSL collaboration presented in the following paper accepted in the conference COORDINATION 2023:

- Salim Chehida, Akram Idani, Mario Cortes-Cornax and German Vega " An Executable Formal Framework for Inter-DSL Collaboration", accepted in the conference COORDINATION 2023.

The proposed approach, supported with a formal framework, allows engineers to define how DSLs collaborate with each others. In our approach, the Model-driven engineer specifies the DSLs metamodels and the BPMN models of their collaboration. Then the metamodels and BPMN diagrams are transformed into B and CSP respectively, while integrating the system properties. Afterwards, the operator can animate the formal specifications while observing the respect of the properties.

We applied the proposed approach to a smart grid case study provided by RTE, the energy transmission company in France. The case study involves two DSLs: the first one, named CM-DSL (Configuration Management DSL), focuses on the management of system configurations assigning to a set of applications various infrastructures. The second one named SRA-DSL (Security Risk Assessment DSL), is dedicated to security risk assessment. The composition and the collaboration of these DSLs allow to manage configurations while dealing with security concerns.

**## 1. Modeling DSLs and their Collaboration**

In this step, the engineer builds:

- The abstract syntax of each DSL and the metamodel of their composition using EMF-based modelling tool (Ecore, Xtext, Sirius, GMF, etc.).

- The BPMN diagrams expressing the collaboration between the DSLs using the BPMN2 Modeler.

**### Requirements**

- Download and install the *Eclipse Modeling Framework* <https://www.eclipse.org/downloads/packages/release/2022-12/r/eclipse-modeling-tools>

- Launch your eclipse and install BPMN2 Modeler from the following update sites:

<https://www.eclipse.org/bpmn2-modeler/downloads.php>

The following Update Sites is used: <http://download.eclipse.org/bpmn2-modeler/updates/2020-06/1.5.2>

**### Use case Models**

- The EMF metamodel of CM-DSL (Part A of Figure 4 in the paper) is available at:

<https://github.com/SalimChehida/Inter-DSL-Collaboration/blob/artefacts-coordination/DSLs_Modeling/Metamodels/CM_DSL/model/cM_DSL.ecore.uml>

- The EMF metamodel of SRA-DSL (Part B of Figure 4) is available at:

<https://github.com/SalimChehida/Inter-DSL-Collaboration/blob/artefacts-coordination/DSLs_Modeling/Metamodels/SRA_DSL/model/sRA_DSL.ecore>

- The composition metamodel of SRA-DSL and CM-DSL (Part C of Figure 4) is available at:

<https://github.com/SalimChehida/Inter-DSL-Collaboration/blob/artefacts-coordination/DSLs_Modeling/Metamodels/Inter_DSL_Collaboration_CM_SRA/model/inter_DSL_Collaboration_CM_SRA.ecore>

- The BPMN model of inter-DSL collaboration (Figure 5 in the paper) is available at:

<https://github.com/SalimChehida/Inter-DSL-Collaboration/blob/artefacts-coordination/DSLs_Modeling/Metamodels/Inter_DSL_Collaboration_CM_SRA/model/collaboration_CM_SRA.bpmn>

**## 2. Formal Specification in B language**

In this step, the engineer can:

- Generate automatically a formal B specification from each DSL and the metamodel of their composition using the Meeduse framework.

- Complete manually the execution semantics of the generated machine by specifying the B operations defining actions involved in the collaboration process.

- Specify a CSP model from the BPMN diagram built in the previous section (this mapping is done manually, work in progress intends to automate this transformation)

**### Requirements**

- Launch your eclipse and install B4MSecure and Meeduse from the following update sites:

B4MSecure: http://vasco.imag.fr/tools/b4msecure/updates/build

Meeduse: http://vasco.imag.fr/tools/meeduse/updates/build

**### Use case B specifications**

- The CM-DSL B machine (discussed in Section 4.2 in the paper) is available at:

- The SRA-DSL B machine (discussed in Section 4.2 in the paper) is available at:

- DSL\_Composition B machine (Figure 6 in the paper) is available at:

- The CSP model (Figure 7 in the paper) is available at:

https://github.com/SalimChehida/Inter-DSL-Collaboration/blob/artefacts-coordination/DSLs\_Formal\_Specification/cM\_DSL\_main.mch

**## 3. Animation and Verification**

**### Requirements**

**### Use case animation scenario**

**## 4. Contact**

"SALIM CHEHIDA" salim.chehida@univ-grenoble-alpes.fr