Execution Framework of the GEMOC Studio (Tool demo)

Erwan Bousse ¹ Thomas Degueule ² Didier Vojtisek ² Tanja Mayerhofer ¹ Julien Deantoni³ Benoit Combemale²

¹TU Wien (Austria)

²Université Rennes 1 (France)

³Université Côte d'Azur, I3S (France)



October 30. 2016

emo

Conclusion O

Generic syntactic services for DSMLs

- Many language workbenches to define
 Domain-specific Modeling Languages (DSMLs)
- Generic advanced syntactical services provided by various frameworks (Xtext, Sirius, etc.):
 - editors, parsers, syntax highlighting, auto completion, "quick fix" suggestion systems, etc.
 - no implementation from scratch
- On-going work on a Language Server Protocol supported by big actors (Red Hat, Microsoft) for a standard unified interface for syntactic services





emo

Conclusion O

Generic syntactic services for DSMLs

- Many language workbenches to define
 Domain-specific Modeling Languages (DSMLs)
- Generic advanced syntactical services provided by various frameworks (Xtext, Sirius, etc.):
 - editors, parsers, syntax highlighting, auto completion, "quick fix" suggestion systems, etc.
 - no implementation from scratch
- On-going work on a Language Server Protocol supported by big actors (Red Hat, Microsoft) for a standard unified interface for syntactic services





Demo Conclusion o o

Towards generic runtime services?

- Problem: services at runtime remain mostly handcrafted for each DSML, eg.:
 - interactive debugging
 - trace management
 - runtime monitoring

- How to capitalize logic of runtime services...
 - RQ1: among different DSMLs?
 - *RQ2:* among different metaprogramming approaches used for defining semantics?



Execution Framework of the GEMOC Studio

The GEMOC Studio execution framework •••

Demo O

Overview of the GEMOC execution framework



Core classes of the framework:

- IExecutionEngine
- IExecutionAddon
- AbstractExecutionEngine

The GEMOC Studio execution framework

Demo O Conclusion O

Overview of the GEMOC execution framework



Core classes of the framework:

- IExecutionEngine
- IExecutionAddon
- AbstractExecutionEngine

The GEMOC Studio execution framework •••

Demo O Conclusion O

Overview of the GEMOC execution framework



Core classes of the framework:

- IExecutionEngine
- IExecutionAddon
- AbstractExecutionEngine

The GEMOC Studio execution framework

Demo O Conclusion O

Overview of the GEMOC execution framework



Core classes of the framework:

- IExecutionEngine
- IExecutionAddon
- AbstractExecutionEngine

Context	The GEMOC Studio execution framework	Demo
00	0	0

Metaprogramming approaches

(and associated engines):

- Java (or Java-based)
- ×MOF
- Java+MoCCML (concurrency)
- BCOol (coordination of models)

- Graphical animator (using Sirius)
- Execution trace management
- Sequential omniscient debugger
- Concurrent omniscient debugger
- VCD, Stimuli manager, step decider

Context	The GEMOC Studio execution framework	Demo
00	0	0

Metaprogramming approaches

(and associated engines):

- Java (or Java-based)
- ×MOF
- Java+MoCCML (concurrency)
- BCOol (coordination of models)

- Graphical animator (using Sirius)
- Execution trace management
- Sequential omniscient debugger
- Concurrent omniscient debugger
- VCD, Stimuli manager, step decider

Context	The GEMOC Studio execution framework	Demo
00	0	0

Metaprogramming approaches

(and associated engines):

- Java (or Java-based)
- ×MOF
- Java+MoCCML (concurrency)
- BCOol (coordination of models)

- Graphical animator (using Sirius)
- Execution trace management
- Sequential omniscient debugger
- Concurrent omniscient debugger
- VCD, Stimuli manager, step decider

Context	The GEMOC Studio execution framework	Demo
00	0	0

Metaprogramming approaches

(and associated engines):

- Java (or Java-based)
- ×MOF
- Java+MoCCML (concurrency)
- BCOol (coordination of models)

- Graphical animator (using Sirius)
- Execution trace management
- Sequential omniscient debugger
- Concurrent omniscient debugger
- VCD, Stimuli manager, step decider

Context	The GEMOC Studio execution framework	Demo
00	0	0

Metaprogramming approaches

(and associated engines):

- Java (or Java-based)
- ×MOF
- Java+MoCCML (concurrency)
- BCOol (coordination of models)

- Graphical animator (using Sirius)
- Execution trace management
- Sequential omniscient debugger
- Concurrent omniscient debugger
- VCD, Stimuli manager, step decider

The GEMOC Studio execution framewor

Demo

Con

Demo

"Talk is cheap. Show me the code."

Linus Torvalds

Bousse, Degueule, Vojtisek, Mayerhofer, Deantoni, Combemale Execution Framework of the GEMOC Studio

Context	The GEMOC Studio execution framework	Demo	Conclusion
00		O	•
Conclusion			

- Sharing of syntactic services in good shape (Xtext, Language Protocol, etc.)
- But poor sharing of runtime services among metaprog methods and DSMLs
- Our proposal: the GEMOC execution framework, to integrate generically runtime services independently from metaprogramming approaches and DSMLs

- Eclipse Research Consortium GEMOC: sustains the GEMOC studio as a research platform to experiment on the globalization of, possibly executable and heterogeneous, modeling languages
- Contributors are welcome! (eg. new metaprog. approaches)

http://gemoc.org/ https://github.com/gemoc/



