

Gemoc Studio Guide

Francois Tanguy

Gemoc Studio Guide

Francois Tanguy

Preface	v
1. Gemoc Language workbench	1
1.1. Introduction	1
2. Gemoc Modeling workbench	3
2.1. Introduction	3
3. Example Chapter	5
3.1. Introduction	5
4. Second Example Chapter	7
4.1. Introduction	7
Bibliography	9
Glossary	11
Index	13

Preface

The Gemoc Studio offers 2 main usages:

- Building and composing new executable DSML. This mode is intended to be used by language designers (aka domain experts).
- Creating and executing models conformant to executable DSMLs. This mode is intended to be used by domain designers.

Each of these usage has it own set of tools that are referenced as **Gemoc Language workbench** for the tools for *language designers* and **Gemoc Modeling Workbench** for the tools for *domain designers*.

Chapter 1. Gemoc Language workbench

1.1. Introduction

Intro description for Language workbench.

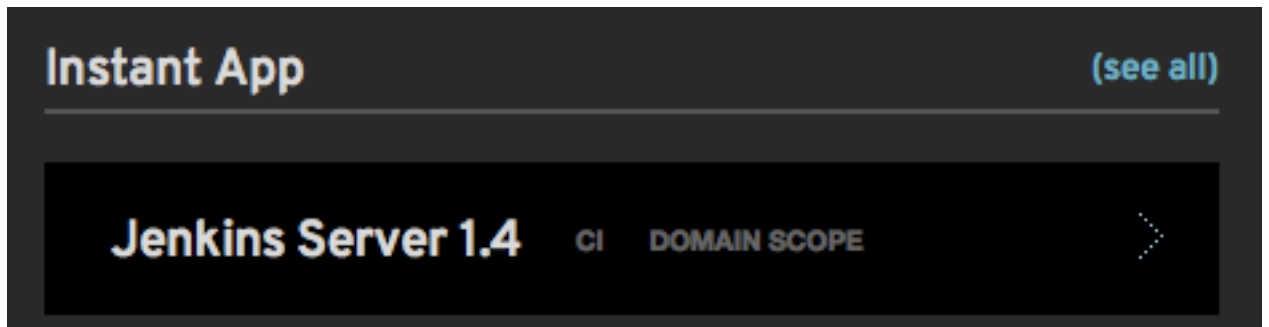


Figure 1.1. Screenshot of the Openshift.

Some more text.

Text on first line Text on second line.

Chapter 2. Gemoc Modeling workbench

2.1. Introduction

Intro description for Modeling workbench.

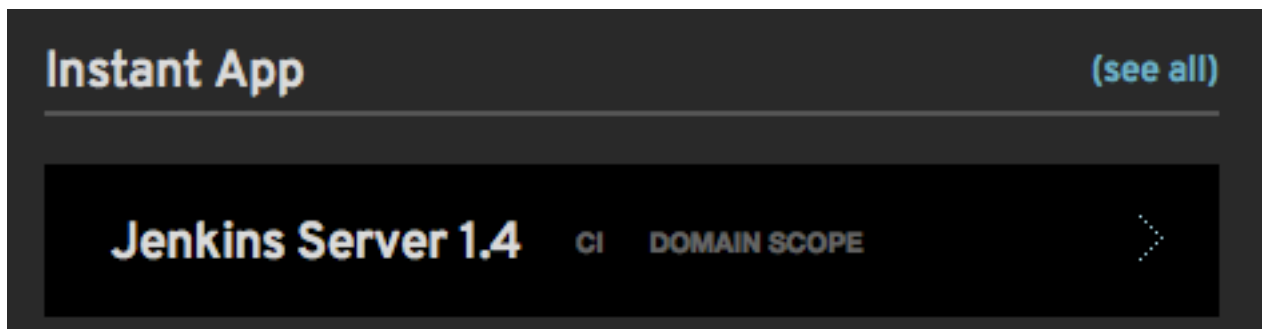


Figure 2.1. Screenshot of the Openshift.

Some more text.

Text on first line Text on second line.

Chapter 3. Example Chapter

3.1. Introduction

Intro description for Adam.

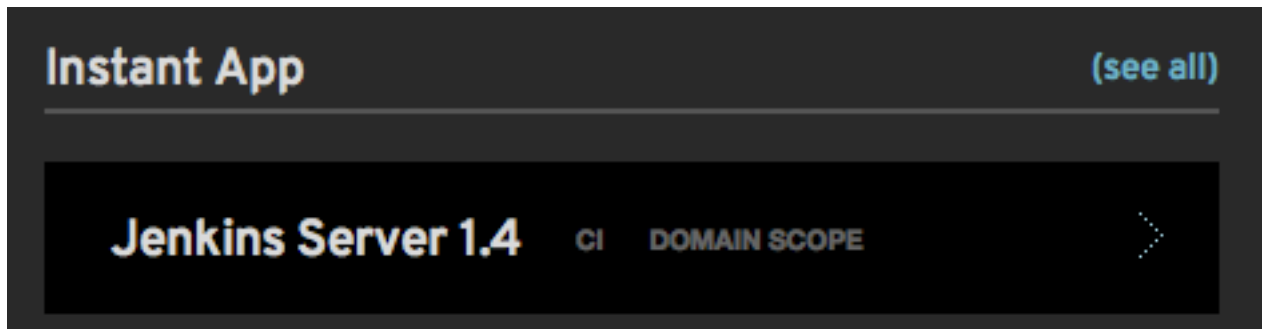


Figure 3.1. Screenshot of the Openshift.

Some more text.

Text on first line Text on second line.

Chapter 4. Second Example

Chapter

4.1. Introduction

Intro blah blah blah

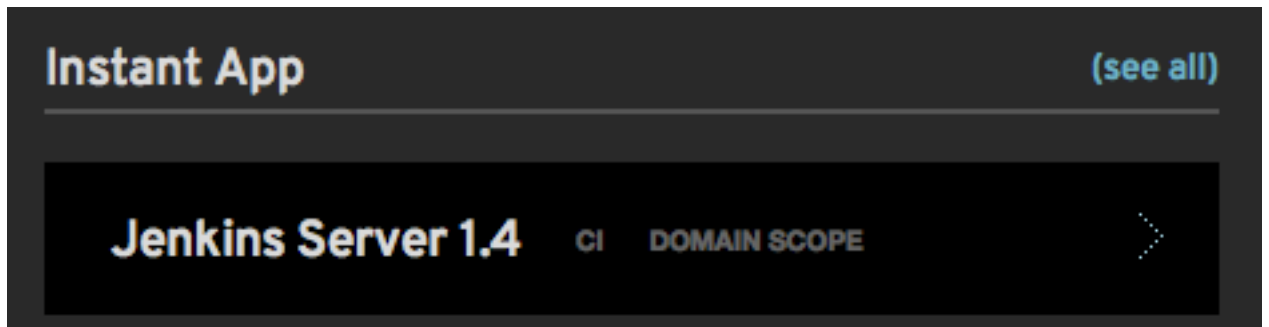


Figure 4.1. Another Screenshot of the Openshift.

Bibliography

The bibliography lists some useful external documents. For a more complete list, please refer to the publications section on <http://gemoc.org> site.

Articles

[globalizing-modeling-languages] Globalizing Modeling Languages [<http://hal.inria.fr/hal-00994551>] (Benoit Combemale, Julien Deantoni, Benoit Baudry, Robert France, Jean-Marc Jezequel, Jeff Gray), In Computer, IEEE, 2014.

Glossary

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

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 content 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.

xDSML

Executable Domain-Specific Modeling Language.

Index

D

description, 1, 3, 5

J

Jenkins, 1, 3, 5

O

OpenShift, 1, 3, 5

Jenkins, 1, 3, 5

