

A bare simple Moose tutorial

Nicolas Anquetil

RMod team
INRIA Lille Nord-Europe
nicolas.Anquetil@INRIA.fr

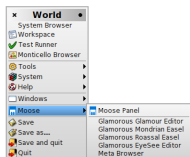
September 18, 2012

Warnings

- ▶ This very simple tutorial should help you understand some ideas behind Moose and what it can do
- ▶ You are supposed to have some basic knowledge of Pharo (e.g. what is the “World Menu”) to follow this
- ▶ We will assume you have a working Moose image. This tutorial uses Moose 4.6, but it should work for many other versions
- ▶ This tutorial is not a step by step recipe, it is expected that you explore a bit around the few simple tasks illustrated here

The Moose Panel

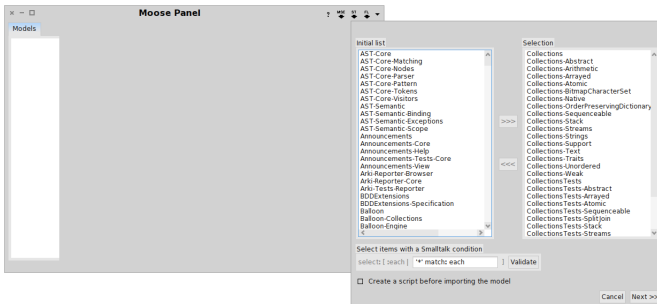
- ▶ First you should bring on the “Moose Panel”, accessible from the World Menu



- ▶ The Moose Panel allows to manipulate models (create, delete, import, export), navigate them and call other tools on them

Creating a Smalltalk model

- ▶ We will start by creating a model, which means analyzing some piece of code and represent its entities and their dependencies into a Moose model
- ▶ We will create a model of some Smalltalk code, for this use the “ST” button at the top right corner of the Moose Panel

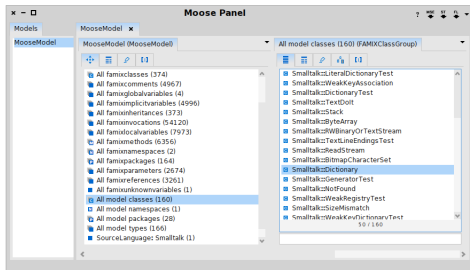


Creating a Smalltalk model

- ▶ The “ST” button opens a wizard where we can select some Pharo packages (in our example, Collections) and send them to the “Selection” area
- ▶ When this is done, finish the wizard simply accepting the default options
- ▶ Models can also be loaded from .mse files (“MSE” button next to the “ST” one) or from “Fuel” files
- ▶ You can generate .mse files of programs in other languages using external tools. For example, verveineJ generates a .mse file from a Java project

Exploring a model

- ▶ After finishing the wizard, you should be back to the Moose Panel, with a model in it (left most pane)
- ▶ Click on the model name and on any element of the list to bring something like this view



Exploring a model

- ▶ You can already navigate the model for example:
 - ▶ Select “All model classes” in the second pane (next to the list of models)
 - ▶ Select a class in the list that appears
 - ▶ Select all methods of that class (if it has more than one)
 - ▶ Select a method of that list
 - ▶ Select all the methods it invokes
 - ▶ Select one of these methods
 - ▶ Select the class owning it
 - ▶ ...

Exploring a model

- ▶ Moose models contains several kind of Entities¹:
 - ▶ Packages (or Namespaces in Java)
 - ▶ Classes, Interface, Parameterized classes
 - ▶ Methods, Functions
 - ▶ Attributes, Variables, Method parameters
- ▶ These entities are related by four possible Associations:
 - ▶ Invocations (between two methods or functions)
 - ▶ Reference of a Class in some code (typically a method)
 - ▶ Access of an Attribute in some code
 - ▶ Inheritance between two classes
- ▶ There is also belongs to relationships as between a method and its class

¹class FAMIXSourcedEntity

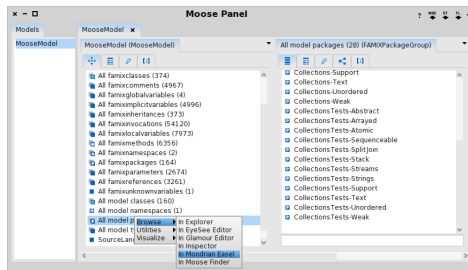
Exploring a model

- ▶ You will notice that there are tabs in the Moose Panel's panes
- ▶ You are encouraged to try up these tabs and see what they offer.
- ▶ Some interesting ones at this stage are:
 - ▶ Displays various metrics and properties of an entity
 - ▶ Displays the source code of an entity
 - ▶ Displays a graphical visualization using polymetric view (look for this in the Moose Book² on chapter "6.1 – System Complexity")
 - ▶ Pharo interpret on the entity (self represents the entity (or group of entities), you can explore it with the Pharo inspector)

²<http://www.themoosebook.org/>

A tool: Mondrian

- ▶ Mondrian is a scripting tool to create visualizations on the models. For example the graphical tab in the Moose Panel for some entities (or group of entities) displays a Mondrian visualization of that entity (or group of)
- ▶ You can call the Mondrian tool by right clicking on a model's name, or on an entry in the Moose Panel's lists.



A tool: Mondrian

- ▶ We will not teach here how to use Mondrian (see the Moose Book) for this
- ▶ You can try out the very script we entered in the example and click on the black triangle to have the visualization appear. It will only work if you called Mondrian on the entry “All model packages”

The screenshot shows the Mondrian Easel tool interface. On the left, a tree view displays the MooseModel structure, with "All model packages (29)" selected. The main window shows a visualization of these packages as a complex network of nodes and edges. Below the visualization, a script editor contains the following code:

```
view shape rectangle
width: #numberOfClasses
height: #numberOfMethods
linearFillColor: #numberOfClientPackages within packageGroup,
view nodes: packageGroup,
view edges ToAll: #clientPackages,
view treeLayout
```

To the right of the script editor, a table displays the variable and value for the selected script:

Variable	Value
packageGroup	All model...ackages)

A tool: Mondrian

- ▶ In the preceding view, nodes are packages and there are links when one package uses another (e.g. a method in a package needs a method in another one)
- ▶ The parameters of the polymetric view are:
 - ▶ Width of the package indicates how many classes it has
 - ▶ Height indicates how many methods it contains
 - ▶ Color indicates how many clients it has, darker means more clients

Closing up

- ▶ You can explore other functionalities of the Moose Panel, for example by exporting and importing models
- ▶ You can look at the Meta-Model behind Moose (Famix), using the “Meta Browser” available from the World Menu. For example look below Famix.Entity / Famix.SourcedEntity
- ▶ A great power of Moose also comes from the possibility to manipulate and query the models programatically and to extends the exsiting tools

Have Fun !