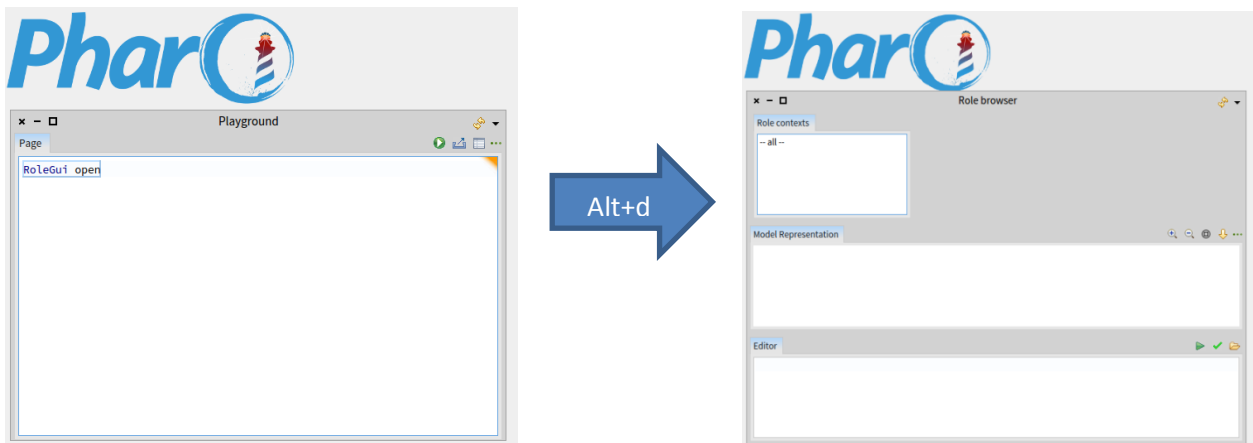


Tutorial on Role4All

This is a tutorial of Role4All. You need to install Pharo 4 and download Role4All first.

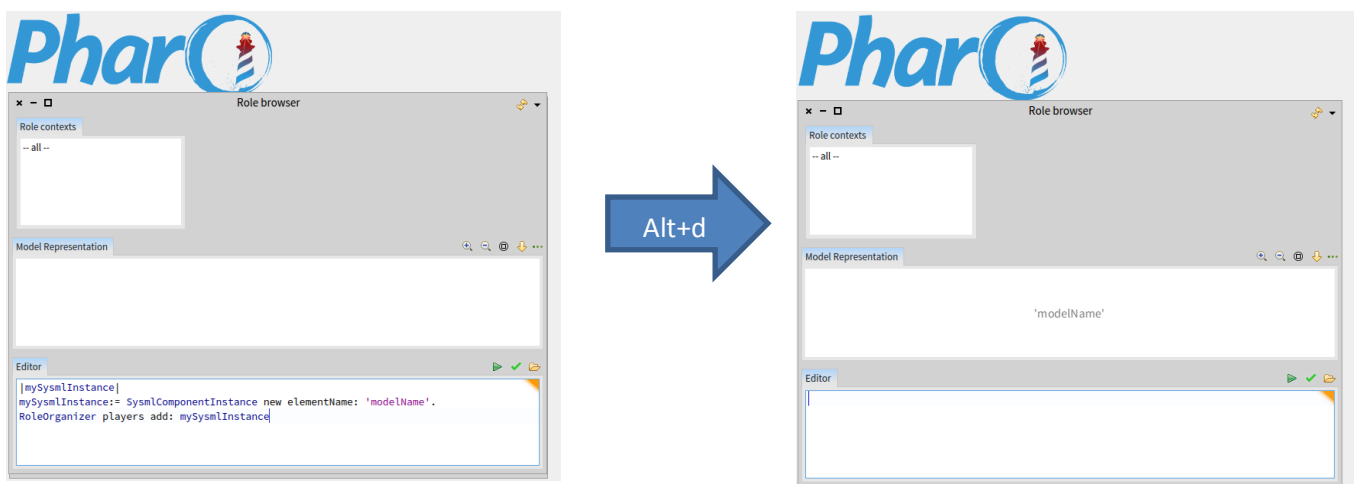
Open Role4All browser

Open a playground and run (alt+d) “RoleGui open”



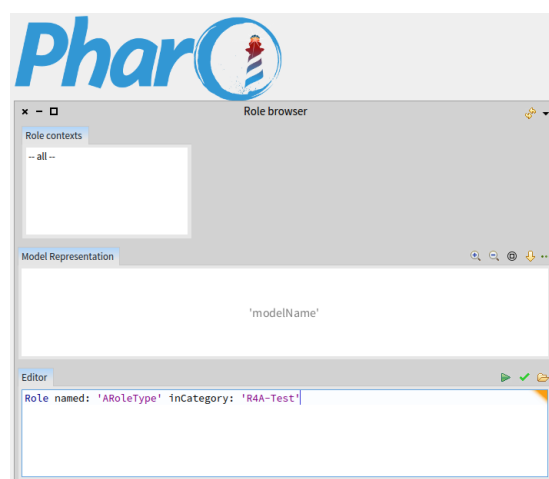
Create a new Player

To create a new Player we create an instance of SysmlComponentInstance here “mySysmlInstance” with an element name here “modelName”. Finally we add this instance in the players list.



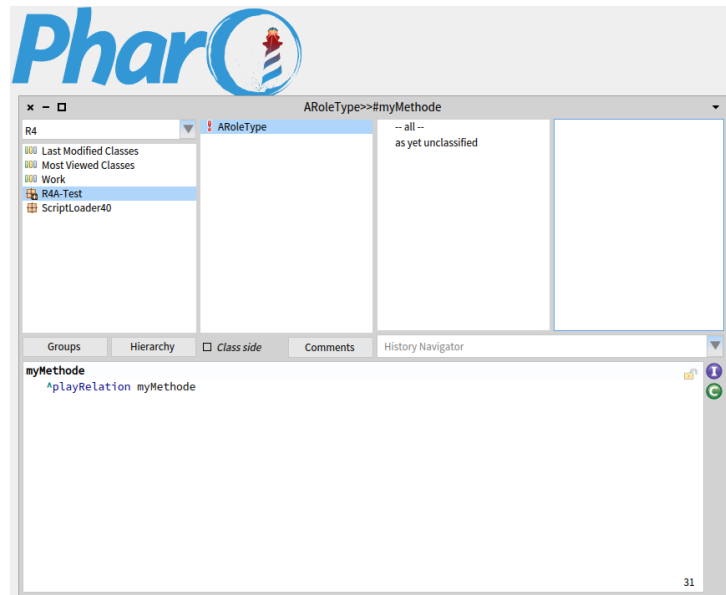
Create a role type

Create a role type with a name here “ARoleType” and a category here “R4A-Test”



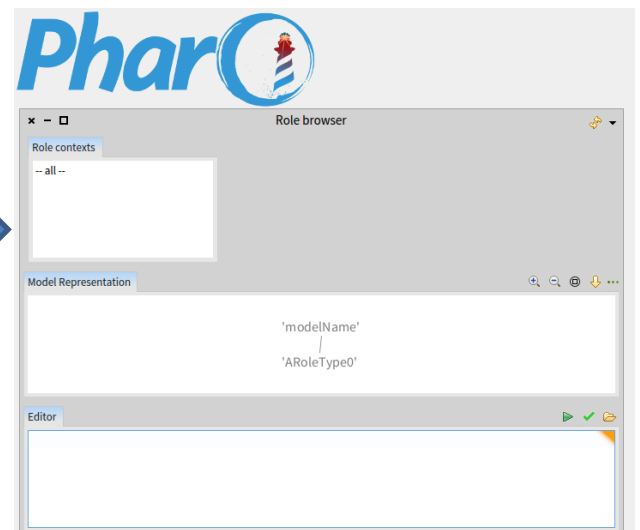
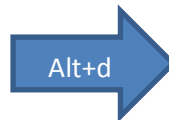
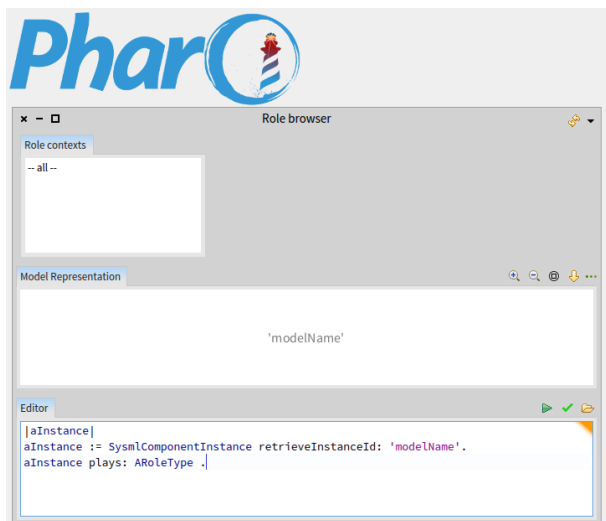
Add a new method in the role type

Add a method in “ARoleType”. This method call another method with the same name (here “myMethode” defined in the class named PlayRelation).



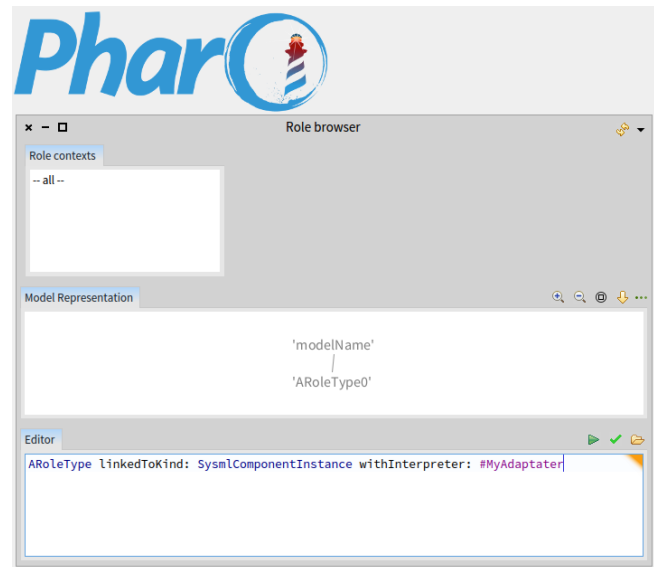
Associate our role type and our instance of Player

Our instance of Player (named “modelName”) play a role (automatically named “ARoleType0”). This role is an instance of “ARoleType” .



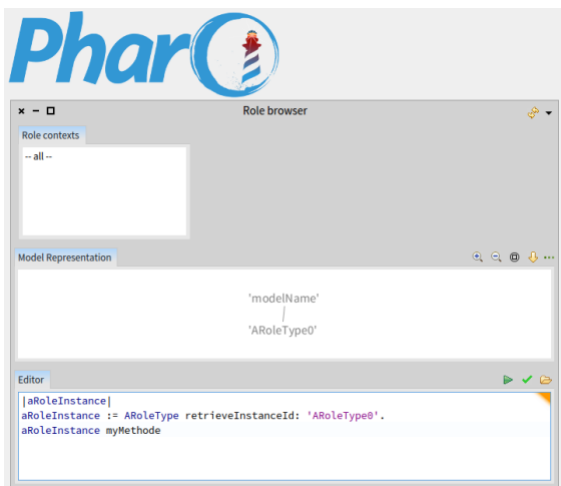
Add an adaptor

Add an adaptor (an instance of “**MyAdaptater**”) on each relation between a role (an instance of “**ARoleType**”) and a player (an instance of “**SysmlComponentInstance**”).



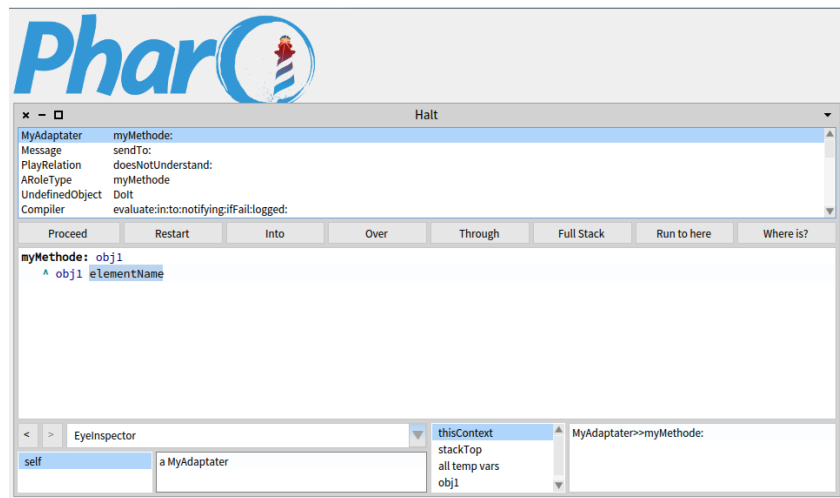
Call our method

Call the method named “**myMethod**” define in the class “**ARoleType**”. This method call the method “**myMethod**” defined in the class “**PlayRelation**” but this method doesn’t exist. However this method will be automatically open a debug window (a **halt**).



Implement the method

We use the halt window to implement the method “**myMethod**” define in the class “**MyAdaptater**”. Here we implemented a method who returns the name of the player associate.



Call our method

We call one more time our method “**myMethod**” define in the class “**MyAdaptater**”. This method return the name of the player who play the role named “**ARoleType0**” .

