

Moose for Java Enterprise Application

Perin Fabrizio

SCG - Software Composition Group

Institut für Informatik und Angewandte Mathematik
University of Bern, Switzerland

18/03/10

Revision History

Date	Ver.	Main Changes	Responsible	Authorized
23/02/10	0.1	Initial Release	Perin Fabrizio	
18/03/10	0.2	Import GUI	Perin Fabrizio	

CONTENTS

Contents

List of figures	3
1 How to start	4
2 How to use	11

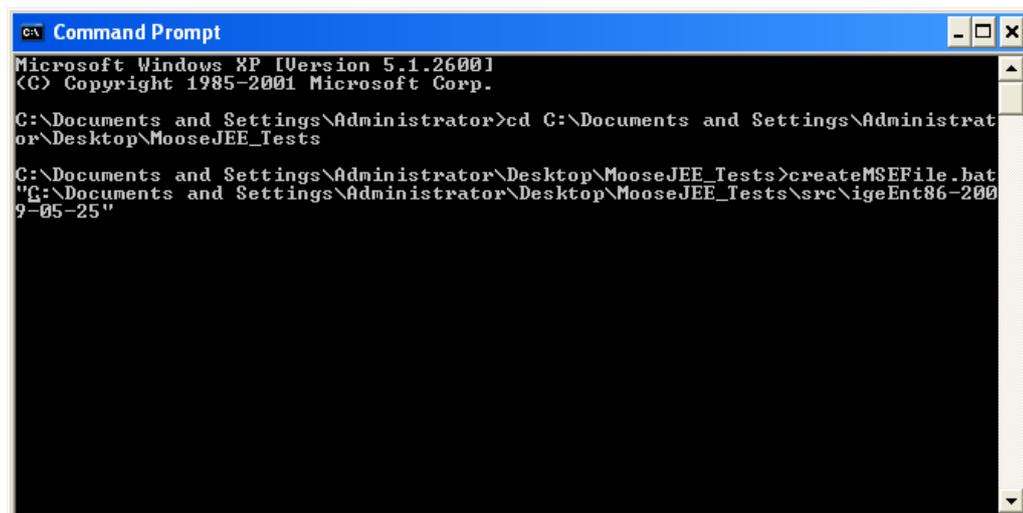
List of Figures

1	Command line to create the mse file	4
2	Import your project from MSE file	5
3	Information panel	6
4	MSE file panel	7
5	Root directory panel	8
6	Post import operations panel	9
7	The image is working	10
8	System ready	11
9	Transaction flow visualization	12
10	Transaction flow visualization particular	13
11	Unsafe Query visualization	14

1 How to start

To start to work with Moose completed with the add-on to analyze enterprise applications follow the instruction below:

1. Unzip the Moose image.
2. Create the mse file using the command createMSEFile.bat:
 - Open a terminal (cmd.exe)
 - go into the directory where you unzip Moose
 - use the command createMSEFile.bat passing as parameter the absolute path of the directory containing your Java source code to analyse. "createMSEFile.bat absolute_path_of_project_source_code". See Fig. 1
 - E.g.: createMSEFile.bat C:\development\projects\ProjectA
 - **IMPORTANT:** In order to load properly your application remember to put into the folder of your project all the external JAR used from your application.



```
Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>cd C:\Documents and Settings\Administrator\Desktop\MooseJEE_Tests

C:\Documents and Settings\Administrator\Desktop\MooseJEE_Tests>createMSEFile.bat
"C:\Documents and Settings\Administrator\Desktop\MooseJEE_Tests\src\IgeEnt86-2009-05-25"
```

Figure 1: Command line to create the mse file

3. Run the Moose image double clicking on RunMooseJEE.bat.
4. In the Moose image that appear there will be the Moose Panel open as well as another window. Click on the import menu' of the Moose Panel and select *Import Java EJB 2.1 from mse file* as shown in figure 2.

1 How to start

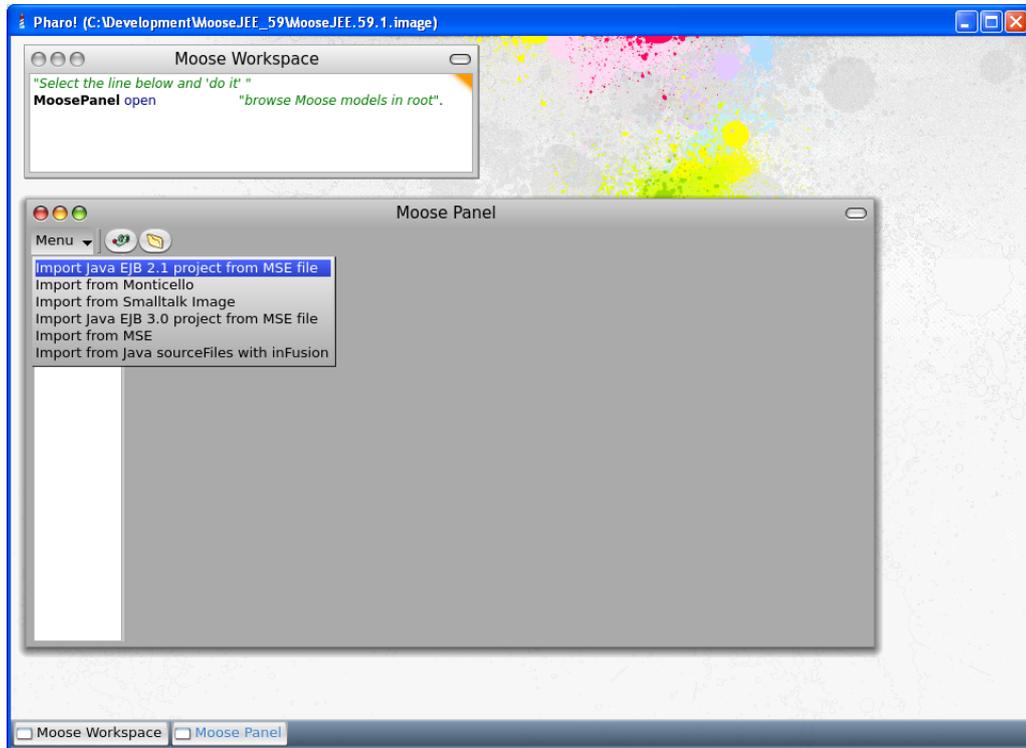


Figure 2: Import your project from MSE file

5. A sequence of panels will appear asking for different information useful for the analyses of the Java application.
6. A simple panel appear with some information regarding the Import process as shown in figure 3. Just click Next.

1 How to start

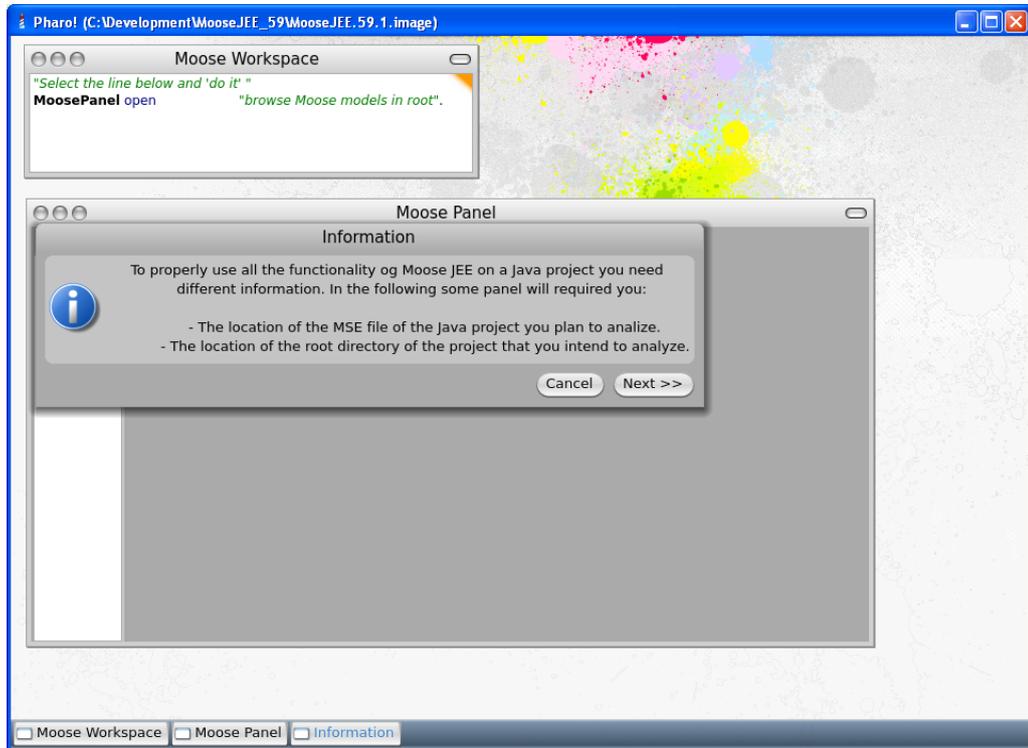


Figure 3: Information panel

7. A panel will ask the user to select the MSE file generated by inFusion. If you follow the creation of the MSE of this document the file is in the current directory and it is called *project.mse*. Select the file and click Next as shown in figure 4.

1 How to start

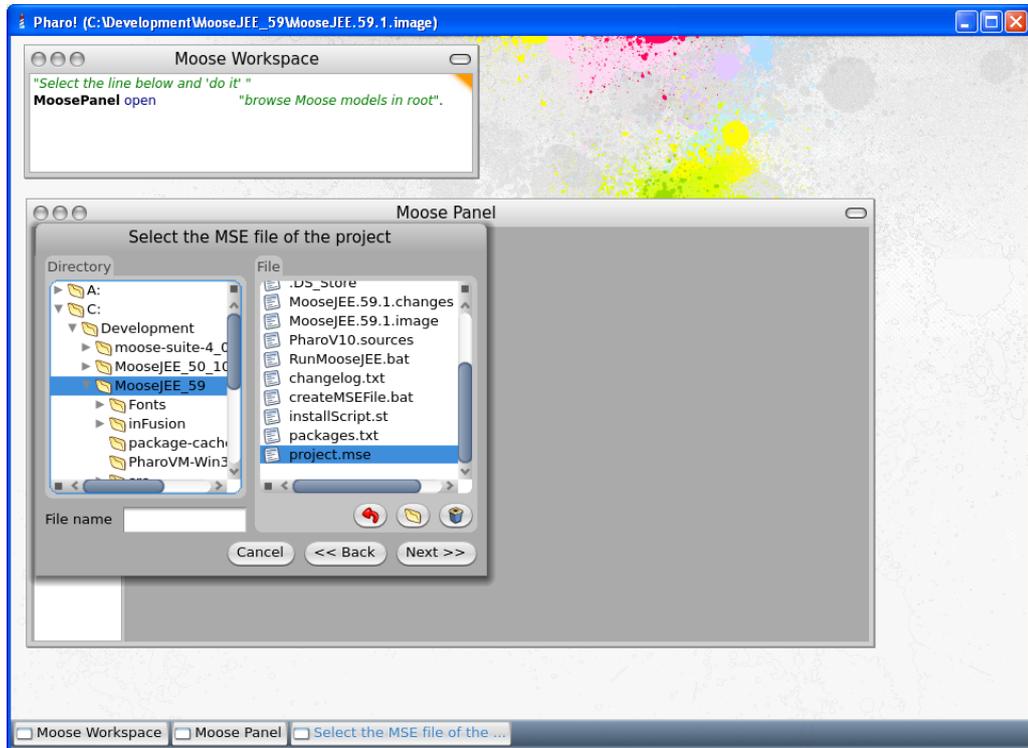


Figure 4: MSE file panel

8. A panel will ask the user to select the root directory containing the project. Select the directory and click Next as shown in figure 5.

1 How to start

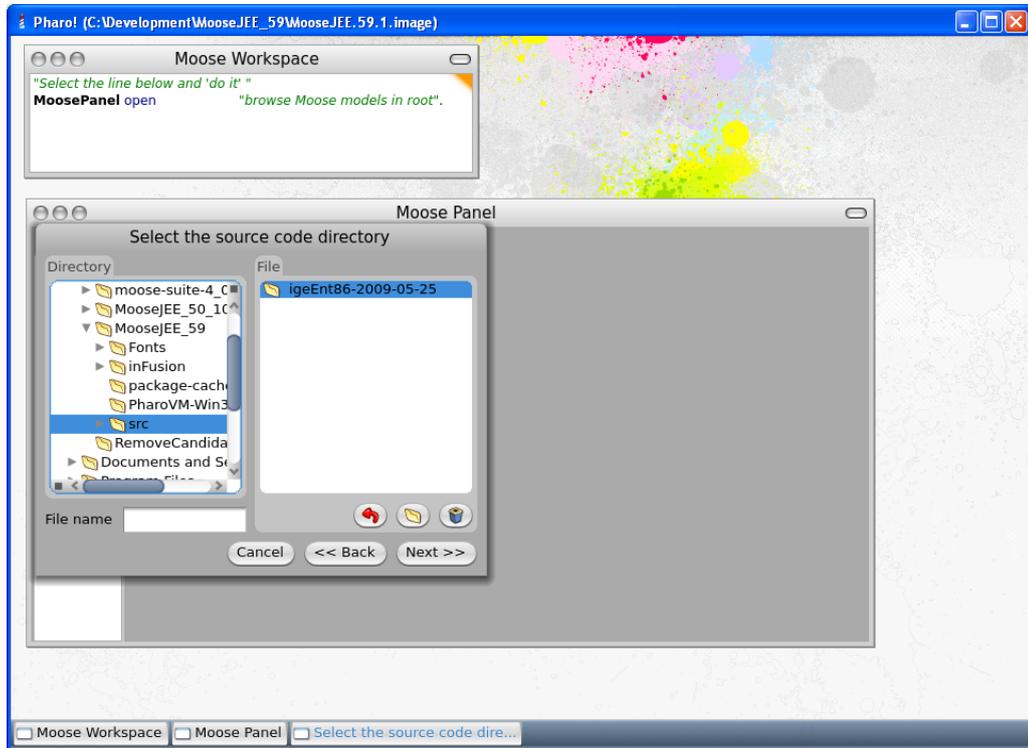


Figure 5: Root directory panel

9. A final panel will ask the user to select the post operation to perform. By default at the end of the import operations the system will open two visualizations and a code browser. Don't modify anything and click Terminate as shown in figure 6.

1 How to start

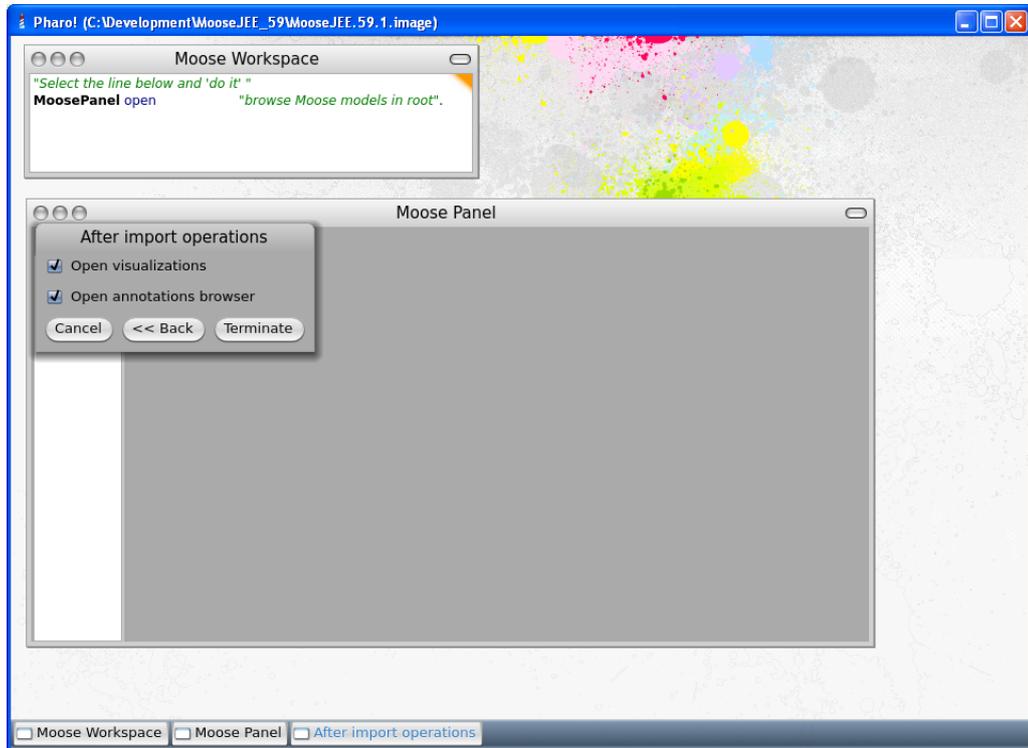


Figure 6: Post import operations panel

10. A progress bar will appear keeping the user up to date on the operations the system is performing as shown in Fig. 8.

1 How to start

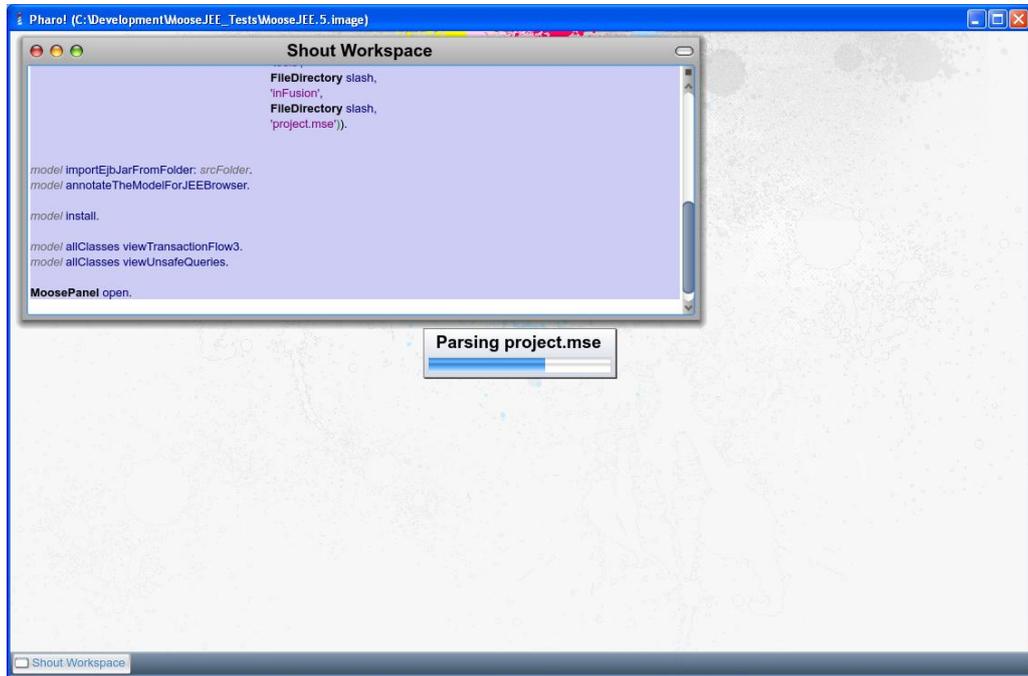


Figure 7: The image is working

11. Now you are ready to inspect the code with Moose and its add-on for enterprise applications.

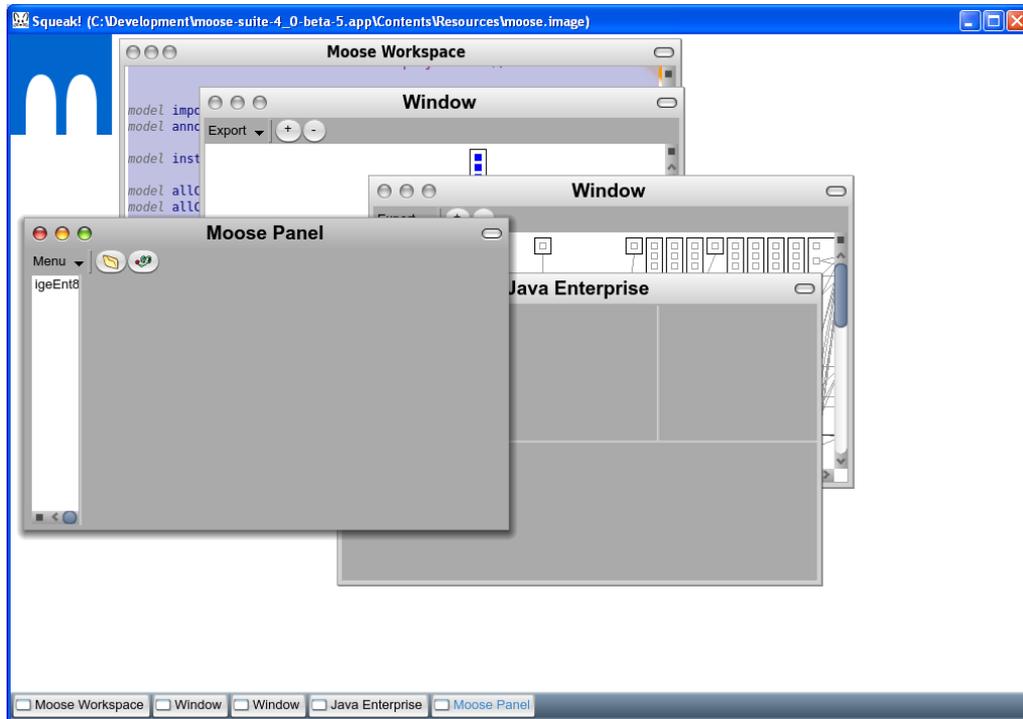


Figure 8: System ready

2 How to use

After the import process 4 windows appear on your screen. One of them is called Transaction Flow visualization Fig. 9. The purpose of that visualization is to show all methods that are involved in an application transaction. The elements in the image have the following meaning:

1. *Blue squares* are methods that start a transaction (i.e. their transaction attribute is 'Required' or 'RequiresNew').
2. *Red squares* are methods with a transaction attribute equals to 'RequiresNew' and are invoked by methods that are already involved in a transaction.
3. *Cyan squares* are methods with a transaction attribute 'Mandatory', 'Required' or 'Supports' that are invoked by methods already part of a transaction.
4. *Orange squares* are entry point methods that have a transaction attribute equal to 'Supports'.

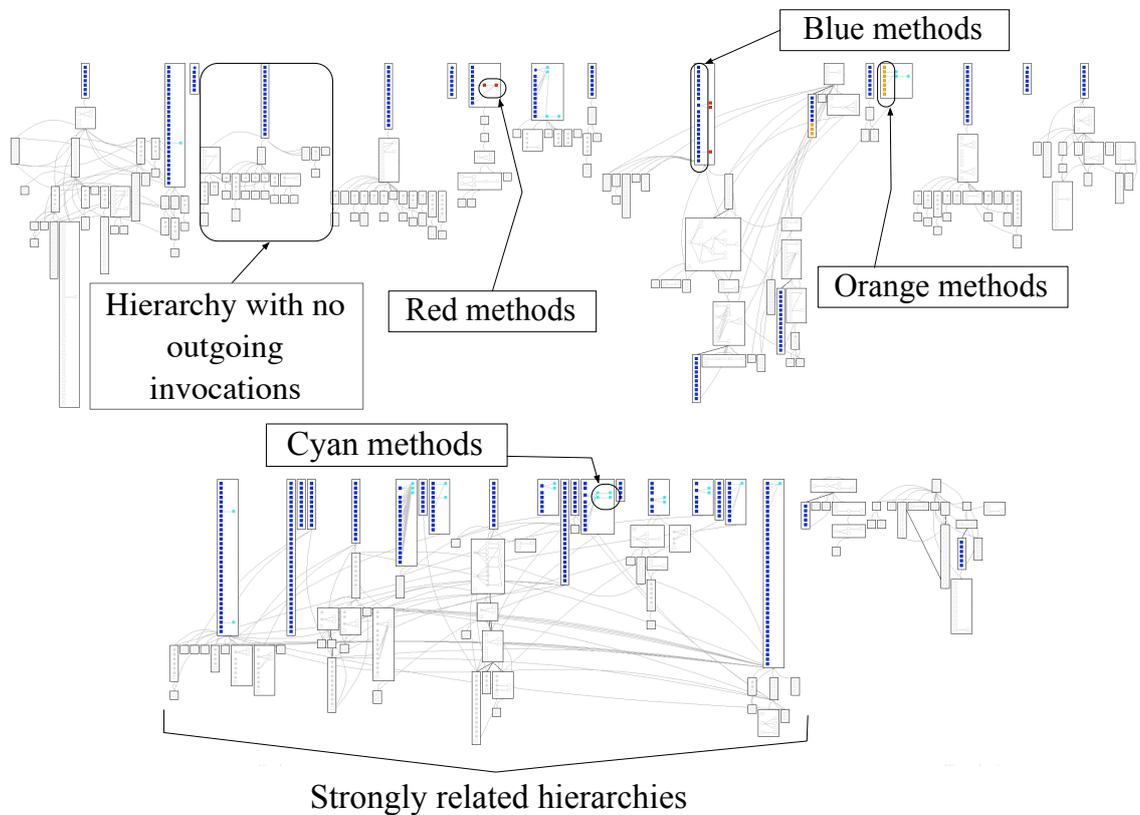


Figure 9: Transaction flow visualization

5. *Grey squares* are methods without a transaction attribute and are invoked by methods already part of a transaction. Looking at the application using the visualization in figure 9 it will be possible identify possible problem like methods that start a useless transaction because they already are into a transaction scope.

Now that the project as been loaded into the image you have in front of you 3 windows To start to work with Moose completed with the add-on to analyze enterprise applications follow the instruction below:

But also it will be possible identify which Session Bean share its business logic and which one works independently so can be easily extract and reuse. See Fig. 10

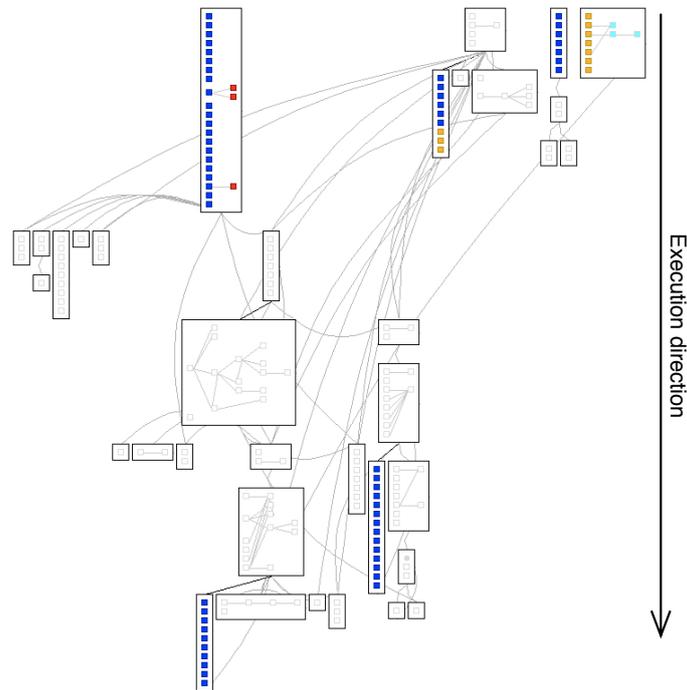


Figure 10: Transaction flow visualization particular

Another visualization is the Unsafe Query Fig. 11. If on one hand it is possible to identify methods involved in a transaction, on the other hand we can show which methods perform queries outside of a transaction.

In Figure 11 in the top part of the figure there are a few classes. Generally they don't access the database and they are the entry points for unsafe paths. Below them there are classes that perform queries on the database. In particular:

1. *yellow dots* are methods that perform a select.
2. *orange dots* are methods that perform an update.
3. *red dots* are methods that perform both.

The last window is the standard Moose browser that provides a basic tool to browse the project that has been loaded and show you all the elements composing the system. It provides also Metrics and two standard visualizations the Class Blueprint and the System Complexity. Refer to this link for a more accurate documentation.

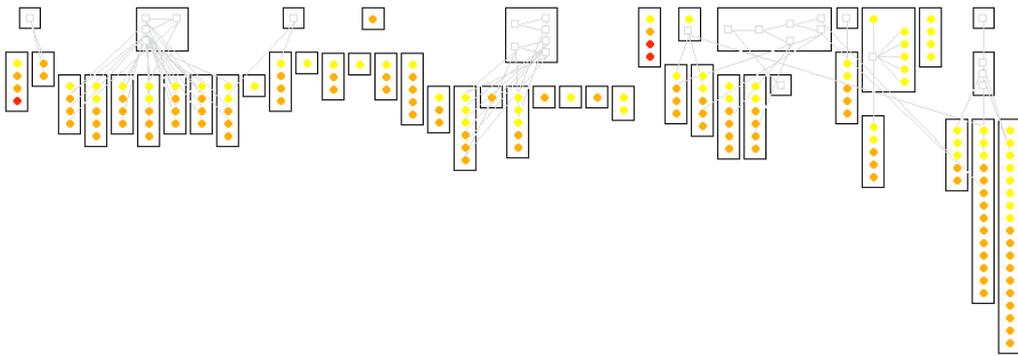


Figure 11: Unsafe Query visualization