

Glamour packages - User study

Presentation

The goal of this study is to assess packages forming an application. You will assess packages through their classes and class references, analyzing what they use and how they are used, both internally and externally to the application.

This study is organized in three sections which require more and more in-depth look at packages and their classes. Each section implies that you play a different role when assessing the application, first as newcomer and potential client who wants to use the application, second as an architect who needs to assess the organization, third as a developer who performs maintenance. There are 11 questions in this study, each question relating to a task.

You will perform the study on Glamour packages. Glamour is an engine for scripting browsers for any kind of models. If you are unfamiliar with Glamour, do not hesitate to test it before the study to get a basic understanding of Glamour capabilities. Information on usage and samples are available on: <http://www.moosetechnology.org/tools/glamour>

Tool used: Package blueprint browser

Instructions

- Use only the tool indicated for the study. Do not use another browser/tool.
- Browse the documentation before performing the study:
 - **PackageBlueprintsPrinciples.pdf** explains the basics of package blueprints
 - **packageblueprints.mov** shows interactions with the package blueprint browser
 - a draft version of the journal paper describing package blueprints is provided
- Process questions in the given order (do not read questions in advance!)
- Please provide only accurate answers like the name of a class or a package, the association between two classes, the method with references.
- **Time yourself** for each question.
- Do not spend more than **20 minutes** on a question. If you reach this limit, write it down, stop the task, and proceed to the next question.
- You also have a time limit of **1h30** to answer the 11 questions so take care of your time.

A. Application assessment

As a potential client, you are assessing the package dependencies of the application. You want an idea about the size of the application and the kind of dependencies needed, especially if it involves new dependencies to be loaded with the application.

1) How big is the application?

Time taken: **2 Minutes**

(a) In number of packages **11 Packages**

(b) In number of classes (one of the following ranges):

[] <100; [] 100-200; [] 200-300; [☒] > 300

2) What are the most important packages?

Time taken: **15 minutes**

(a) In terms of outgoing dependencies

Glamour-Core, Glamour-Examples, Glamour-Morphic, Glamour-Tests

(b) In terms of incoming dependencies

Glamour-Core, Glamour-Browser, Glamour-Examples

(c) Overall, considering both outgoing and incoming dependencies
Glamour-Core, Glamour-Examples,

3) Focus on package Glamour-Morphic:

Time taken: 12 minutes

(a) list all package dependencies which are external to Glamour.

Morphic, Polymorph-Widgets, Mondrian, Morphic-MorphTreeWidget, Balloon, Collections-Unordered, Polymorph-Tools-Diff, FreeType, Collections-Arrayed, Collections-Sequenceable, Collections-Strings, Graphics, Magritte-Morph, Shout

(b) in this list, please signal any external package which is not part of Pharo base (i.e., package must be loaded with Glamour).
Don't know.

(c) are there other unexpected/unwanted package dependencies?
Don't know.

B. Application architecture assessment

As an architect, you now want to check the organization of your packages. You want your packages to have a good rationale for existence in the application. You want some parts of the application to be modular.

4) Please characterize **each** Glamour package as either:

Time taken: 8 minutes

- a provider package for external clients (package with which external clients interact)

Glamour-Core, Glamour-Browsers, Glamour-Helpers, Glamour-Morphic, Glamour-Tests, Glamour-Announcements, Glamour-Presentations, Glamour-Tools

- an internal package (package which should not be accessed by external clients)

Glamour-Examples, Glamour-Test-Morphic,

5) Are some Glamour packages optional/modular (package can be unloaded without impacting application core)?

Time taken: 4 minutes

What are the packages that make up application core? If that is Glamour-Core then yes because Glamour-Core only references Glamour-Helpers and Glamour-Announcements packages.

6) What are the important classes (consider incoming, outgoing, inheritance dependencies) in Glamour-Core? If possible, explain their roles.

Time taken: 11 minutes

GLMPresentation, GLMPane, GLMLoggedObject, GLMBrowsers, GLMTransmission

7) Are there direct cyclic dependencies from Glamour-Core to another package?

Time taken: 3 minutes

NO

C. Detailed assessment

As a developer, you want a detailed comprehension and assessment of dependencies between classes and packages and optionally to refactor such dependencies, assessing impact of change.

First give a precise answer then provide your explanation.

8) What are the most cohesive packages of the application?

Time taken: **5 minutes**
Glamour-Core, Glamour-Browsers

- 9) There is a dependency to DeprecatedPreferences in Glamour-Morphic. Can you detect the faulty class? Explain the dependency: do you see an easy way to solve it?

Time taken: **2 minutes**
GLMMorphicRenderer

I don't see a way to solve it as I don't know the code of the class.

- 10) Can you explain the organization of Glamour-Morphic and its relationship with other packages?

Time taken: **7 minutes**
The package Glamour-Morphic is largely organized around the class GLMMorphicRenderer as it is suggested by the huge number of dependencies of the class with other classes in the same package. The package is not cohesive: only one class has references to other classes, other classes in the package refer to each other very rarely. The package interacts with almost 19 external packages, another symptom that the package is not self-containing and it has to depend a lot on other classes from different packages. GLMMorphicRenderer is probably a god class or a blob.

- 11) Multiple packages of Glamour have dependencies to external library Mondrian. List such packages. Could you extract this dependency and make it optional (you can propose a solution)?

Time taken: **7 minutes**

Announcements, Collections-Arrayed, Collections-Strings, Collections-Sequenceable, Collections-Unordered, Graphics, Compiler, Kernel, Morpich, Polymorph-Widgets, etc.
Dependencies can be made optional. Use Aspect-oriented Refactoring

D. Personal remarks

You can provide any additional remarks about the study itself, the tasks, the tool used.

E. Personal evaluation of Package blueprints

1	2	3	4	5
Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree

Question	1	2	3	4	5
Does package blueprint help you to understand dependencies between packages?					<input checked="" type="checkbox"/>
Would you use package blueprint when you need to understand packages?					<input checked="" type="checkbox"/>
Did the outgoing view help you?				<input checked="" type="checkbox"/>	
Did the incoming view help you?				<input checked="" type="checkbox"/>	
Did the inheritance view help you?			<input checked="" type="checkbox"/>		
Was package blueprint useful to get an impression of the most used classes in a package?				<input checked="" type="checkbox"/>	

Was package blueprint useful to get an impression of the most referencing classes in a package?				<input checked="" type="checkbox"/>	
Was package blueprint useful to get an impression of package cohesion?					<input checked="" type="checkbox"/>