

Glamour packages - User study

Presentation

The goal of this study is to assess packages forming a 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: 1m30

(a) In number of packages

10

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

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

2) What are the most important packages?

Time taken: 3 minutes

(a) In terms of outgoing dependencies

Glamour-Tests, Glamour-Morphic

(b) In terms of incoming dependencies

Glamour-core, Glamour-Browsers

(c) Overall, considering both outgoing and incoming dependencies

Glamour-core

3) Focus on package Glamour-Morphic:

Time taken: 6 minutes

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

Morphic, Polymorph-widgets, Mondrian, Morphtreewidget, Balloon, Shout, Magritte-Morph, Graphics, Collection-Strings, Collections-Sequenceables, DeprecatedPreferences, Collections-Arrayed, Freetype, Polymorph-Tools-Diff, Collections-Unordered

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

Mondrian, Balloon, Magritte-morph, Freetype

(c) are there other unexpected/unwanted package dependencies?

Balloon, Magritte-morph

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: 4 minutes 50 seconds

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

Glamour-core, Glamour-Browsers, Glamour-Presentations, Glamour-Tools

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

Glamour-Helpers, Glamour-Tests, Glamour-Announcements, Glamour-Test-Morphic, Glamour-Examples

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

Time taken: 2 minutes 50 seconds

Yes: Glamour-Tests, Glamour-Test-Morphic, Glamour-Examples,

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

Time taken: 9 minutes, 30 seconds

GLMPresentation: a graphical element, several subclasses.

GLMPane: a graphical panel class

GLMTransmissionContext: unknown, related to transmissions

GLMPort: some kind of port between other objects? several subclasses

GLMLoggedObject: top of a hierarchy including Presentation and Port

GLMTransmissionStrategy: Strategy pattern, for transmissions (a hierarchy)

GLMBrowser: Main browser object.

GLMTransmission: not sure, related to communication.

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

Time taken: 3minutes

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: 8minutes

Glamour-Core: Has a few classes referencing a lot of other classes. The majority of classes however are not referencing a lot of other classes.

Glamour-Morphic: Has one class referencing a large majority of the other classes.

Glamour-Browsers: has several classes referencing several other. Overall quite cohesive.

Glamour-Tools: Each class uses another class. The package is small though.

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: 1minute40seconds

The class is GLMRenderer. It could be fixed by changing the font used in the method involved, so it would be pretty easy.

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

The package seems to have one main interface class, the GLMRenderer. Other important classes are GLMMorphic, and the GLMTreeMorph classes. The main packages it uses are: Morphic, PolyMorph, and Mondrian, all three graphical packages. It doesn't use a lot of functionality from Glamour itself (Glamour-Browsers and Glamour-Announcements). It is not really used by Glamour packages, besides the Glamour-Morphic-Tests package, which is obvious.

Time taken: 7minutes, 40 seconds

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: 5minutes, 10seconds

The packages are Glamour-Morphic, Glamour-Presentations, Glamour-Examples and the Tests packages. To remove the dependency, one would start by removing the optional packages (examples and tests).

In the presentations, there is a class, GLMMondrianPresentation, interfacing with Mondrian that needs to be moved somewhere else (eg a Glamour-Mondrian optional package).

In Glamour-Morphic, the GLMMorphicRenderer also uses Mondrian, in a single method (mondrianCanvasFor:in:). This method could be moved to an extension part of the new Glamour-Mondrian package.

D. Personal remarks

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

- A large screen certainly helps. My window was 4 times larger than the original dimensions.
- There was a large amount of information to process before starting. I under-utilized some features like the color picking (only towards the end).
- Some of the tasks mix closed and open-ended questions that should be separated. I'm not sure about timing for open-ended questions.
- The view gets a bit burdensome when there are a lot of small packages involved in the blueprint (like one band each).
- Having the names of the packages without hovering would be great sometimes.
- This question requires maybe too much knowledge of what's in Pharo:
(a) in this list, please signal any external package which is not part of Pharo base (i.e., package must be loaded with Glamour).

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?					X
Would you use package blueprint when you need to understand packages?				X	
Did the outgoing view help you?				X	
Did the incoming view help you?				X	
Did the inheritance view help you?				X	
Was package blueprint useful to get an impression of the most used classes in a package?				X	
Was package blueprint useful to get an impression of the most referencing classes in a package?				X	
Was package blueprint useful to get an impression of package cohesion?					X