

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: 1 minute

(a) In number of packages

(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

(b) In terms of incoming dependencies

Glamour-Core

(c) Overall, considering both outgoing and incoming dependencies

Glamour-Core

3) Focus on package Glamour-Morphic:

Time taken: 8 minutes

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

Morphic, Polymorph-Widgets, Mondrian, Morphic-MorphTreeWidget, Balloon, Shout, Magritte-Morph, Graphics, Collections-Strings, Collections-Sequenciable, DeprecatedPreferences, Collections-Arrayed, Freetype, Polymorph-ToolsDiff, Collection-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, Magritte-Morph,

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

I am a bit surprised of the dependency with Magritte

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: 5 minutes

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

Glamour-Core

Glamour-Browsers

Glamour-Tools

Glamour-Presentations

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

Glamour-Helpers

Glamour-Tests

Glamour-Announcements

Glamour-Test-Morphic

Glamour-Examples

Glamour-Morphic

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

Time taken: 1 minutes

Yes, Glamour-Test-Morphic, Glamour-Examples, Glamour-Tools

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

Time taken: 8 minutes

GLMPresentation appears to be central in Glamour. This class is not used externally of Glamour, but it is intensively used internally (essentially by the packages Glamour-Presentations and Glamour-Browsers). This class heavily use events. It also has something to do with panes.

GLMBrowser is also central since it is referenced in many packages that do not belong to Glamour. This class seems an heavy user of the transmission mechanism and the panes.

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

Time taken: 9 minutes

With Glamour-Tools

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: 2 minutes

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

The faulty class is GLMMorphicRenderer, its method treeMorphFor:and: refers to Preferences

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

Time taken: 8 minutes

Among dependencies of Glamour-Morphic, this packages depends on Polymorph, Morphic and Mondrian, each of these 3 package is related to a distinct graphic framework. Glamour-Morphic provides a number of classes that seems to act as wrappers. It provides a layer between Glamour and the operating system. Something strange however, is that apparently, Glamour-Test-Morphic seems to be the unique package to use GlamourMorphic.

- 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: 4 minutes

Glamour-Tests, Glamour-Morphic, Glamour0Examples, Glamour-Presentations, Glamour-Test-Morphic refer to Mondrian. One way to reduce the dependency on Mondrian is to provide a new package , let's call it Glamour-Mondrian that would wrap the

essential classes of Mondrian. All these 5 classes will then depends on this Glamour-Mondrian.

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?					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		