

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: System browser or OB System browser

Instructions

- Use only the tool indicated for the study. Do not use another browser/tool.
- 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 which makes 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: 5 min

(a) In number of packages: *there are 11 packages, but 2 of them seem to be tests and another seems to contain examples, so from a runtime point of view 8 packages should be enough.*

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

<100; 100-200; 200-300; > 300

Including tests and examples

2) What are the most important packages?

I will not consider Tests and examples

Time taken: 10 min

(a) In terms of outgoing dependencies: **Browsers**

(b) In terms of incoming dependencies: **Core, Presentations**

(c) Overall, considering both outgoing and incoming dependencies: **Browsers, Core, Presentations**

3) Focus on package Glamour-Morphic:

Time taken: 7 min

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

- **Morphic**
- **Collections**

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

- **None**

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

- **I didn't found them**

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 min

- a provider package for external clients (package with which external clients interact)
Browsers, Tools, Scripting

- an internal package (package which should not be accessed by external clients)
Core, Presentations, Helpers, Morphic, Announcements

Test and Examples are not considered

5) Are some Glamour packages optional/modular (package can be unloaded without impacting application core)? Yes, the Scripting, Tools and of course tests and examples

Time taken: 2 min

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

GMLBrowser seems to be the entry point of the "framework"

Time taken: 3 min

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

Time taken: 15 min

I didn't find any direct dependency.

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:

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: 20 min

I couldn't find it, maybe didn't get the meaning of "DeprecatedPreferences", I looked for dependencies to Morphic stuff marked as deprecated.

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

It takes care of the binding to Morphic.

It depends on Glamour-Core, Glamour-Presentation, Morphic and Announcements

Time taken: 20 min

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)?

I only found references in the Presentation package, in order to remove the dependency I would need to study that is it used for, and I have almost used the available time for this question.

Time taken: 15 min

D. Personal remarks

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

In order to do an analysis this, I would only read code as the LAST OPTION. I would start by reading some documentation, UML diagrams and that kind of things And then I could read some integration test code.

I would also use some other tools, not just the browser. To answer the questions I had to read much code and I am almost sure that some answers are uncompleted wrong.