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: 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: 2 minutes
(a) In number of packages
11
(b) In number of classes (one of the following ranges):
[X] <100; [ ] 100-200; [ ] 200-300; [ ] > 300

2) What are the most important packages?
Time taken: 15 minutes

PS: it was not clear to me if you wanted the dependencies _internal_ to Glamour (between Glamour packages) or only the _external_ dependencies. I have therefore analyzed each package seperately, not making a distinction between these two kinds of dependencies.

(a) In terms of outgoing dependencies
    Glamour-Morphic
    Glamour-Announcements

(b) In terms of incoming dependencies
    Glamour-Core
    Glamour-Presentations
    Glamour-Announcements

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

3) Focus on package Glamour-Morphic:
Time taken: 20 minutes
(a) list all package dependencies which are external to Glamour.
    Announcements
    Morphic-MorphTreeWidget
    Morphic-Basic
    Morphic-Windows
    Shout-Windows
    Polymorph-Widgets
    Polymorph-Widgets-Themes
    Polymorph-Widgets-Windows
    Mondrian-Morphic
    Morphic-Menus
    Morphic-Pluggable Widgets
    Polymorph-Tools-Diff
    Morphic-Layouts

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

If I can only use the browser, how can I know what packages are in Pharo base? I cannot... So I take a look at the metacello conf and I guess it is Morphic-MorphTreeWidget and I would suspect Mondrian-Morphic as well.

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

    Mondrian-Morphic

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-Presentations
    Glamour-Browsers
    Glamour-Tools
    Glamour-Examples

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

5) Are some Glamour packages optional/modular (package can be unloaded without impacting application core)?
Time taken: 3 minutes

    Glamour-Tests
    Glamour-Tests-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: too tedious to achieve

7) Are there direct cyclic dependencies from Glamour-Core to another package?
Time taken: too tedious to achieve

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: too tedious to analyze with the browser

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: 5 minutes

GLMMorphicRenderer references the class Preferences. It can be solved by replacing the reference to the method 'menuFont' by the code of that method: StandardFonts menuFont.

8) Can you explain the organization of Glamour-Morphic and its relationship with other packages?
Time taken: 10 minutes

It is a specialization package of Morphic elements. Therefore, it depends on the Morphic packages. The dependency on Mondrian seems to be only when you use the GLMMondrianPresentation. I also notice models, views (the morphic subclasses) and announcements.

9) 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: 8 minutes

Glamour-Morphic
Glamour-Presentation

The methods #mondrian on different classes, the GMLMondrianPresentation and the mondrianCanvasFor: method should be separated to a separate package. I cannot see internal senders of the #mondrian methods, which leads me to believe it can be extracted.

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

I would never use the browser to analyze package dependencies ;-)