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: 1min37 sec
(a) In number of packages
11
(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: 6 min 55

(a) In terms of outgoing dependencies
  Glamour-Core
  Glamour-Browsers

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

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

3) Focus on package Glamour-Morphic:
Time taken: 7min 50
(a) list all package dependencies which are external to Glamour.
  Announcements-Core
  Morphic-MorphTreeWidget
  Morphic-Base (Morphic in general is a dependency)
  Shout-Windows
  Polymorph-Widgets-Windows
  Polymorph-Widgets-Themes
  Morphic-Worlds
  Collections
  Polymorph-Widgets
  Mondrian

(b) in this list, please signal any external package which is not part of Pharo base (i.e., package must be loaded with Glamour).
As far as I know, the only package in the above list that is not a part of the Pharo base is the MorphTreeWidget.

(c) are there other unexpected/unwanted package dependencies?
It seems a bit strange that Glamour-Morphic depends on the Polymorph-Widgets (especially as it directly subclasses from one particular UI theme).
Mondrian is a visualization framework: it is clear to me that there exist some integration between Glamour and Mondrian, but it would have been nice to load Glamour without having to load Mondrian.

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: 5min9
- a provider package for external clients (package with which external clients interact)
  Glamour-Core

Glamour-Browsers
Glamour-Scripting
Glamour-Examples
Glamour-Tools

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

5) Are some Glamour packages optional/modular (package can be unloaded without impacting application core)?
   Time taken: 2min24
   Glamour-Examples
   Glamour-Scripting
   Glamour-Tools

6) What are the important classes (consider incoming, outgoing, inheritance dependencies) in Glamour-Core? If possible, explain their roles.
   Time taken: 3min38
   GLMLoggedObject (root of all classes that somewhere send notifications using Announcements).
   GLMPane (representation of a pane within a Glamour browser; linked to the presentation and the ports)
   GLMPort
   GLMPresentation (root of all possible presentations)

7) Are there direct cyclic dependencies from Glamour-Core to another package?
   Time taken: 2min43
   Yes, e.g. GLMPane uses the announcements in Glamour-Announcements.
   Within this package GLMMatchingPresentationsChanged requires code that is present in GLMPane.

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: DNA
   With the tool that has to be used during this experiment, it is very hard (if not impossible) to assess cohesion of the packages in the application. It would take more than 20 minutes to accomplish this task.

8) 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: 5min36
GLMMorphicRenderer>>treeMorphFor:and: uses the Preferences class to get the standard menu font.

9) Can you explain the organization of Glamour-Morphic and its relationship with other packages?
Time taken: 1min50
The package consists of a number of classes that implement announcements (and depend on the Announcements package) and a number of special morphs for Glamour (that depend on Morphic). Furthermore, there appear to be a couple of classes that glue the morphs into the Glamour framework (such as GLMMorphicRenderer).

10) 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: 5min25
GLMPresentations contains the class GLMMondrianPresentation that uses a Mondrian View Renderer. Glamour-Scripting contains the methods GLMBrowser>>mondrian, GLMCompositePresentation>>mondrian and GLMTPresentationBuilder>>mondrian that create such presentations. Making a separate package that contains this code seems feasible on first sight.

D. Personal remarks

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

The problem with the System Browser is that it is not easy to get an overview of the interactions between packages. While most of the tasks above can be done using this tool, this process is tedious (lots of different classes/methods need to be manually browsed) and error-prone (it is very easy to miss a dependency).