Questions about Object Structure during Coding Activities

Marwan Abi-Antoun
Wayne State University

Nariman Ammar
Carnegie Mellon University

Thomas D. LaToza

Introduction

Recent tools have been designed to help developers understand the potential runtime structure of objects at compile time. But do developers ask questions about object structure? If so, why?

Runtime object structure

The Ownership Object Graph (OOG) depicts:
- instances at runtime
- objects grouped into conceptual groups (domains)
- potential points-to-relationships
- encapsulation and logical containment

Questions about object structure

Developers both asked questions about object structure and stated beliefs about object structure that they held. To analyze this data, we grouped each question instance into question types and counted the number of each type.

Benefits of a diagram

In many cases, the diagram can answer these questions directly. Overall, a diagram may have important benefits.

Obtaining a high level overview
I am not familiar with JHotdraw or any other similar application and I really don’t know where to start, and I think this will be a pain.

Helping me figure out where to insert a given piece of code
Until now, I was unable to find a place where the figure is being added [...] and I am stuck. Eclipse does not help any more.

Questions about object identity

Most diagrams of structure depict class structure rather than object structure. In a UML class diagram, there is only a single box for all instances of a class. In contrast, the OOG can show different instances of a class that have unique identity and different relationships.

Listener interfaces
In JHotDraw, there are several different listener interfaces. Developers asked questions about how objects were sending or receiving notifications.

So something had to fire off the view update, so what I am thinking about is that all registered observers are notified if the drawing view has been changed.

Developers found it difficult to answer these questions using a class diagram.

I referred to [the class diagram] because I am used to it, but if I use this [object] diagram more and more, I will get used to it.