

Questions about Object Structure during Coding Activities

Marwan Abi-Antoun Nariman Ammar
Wayne State University

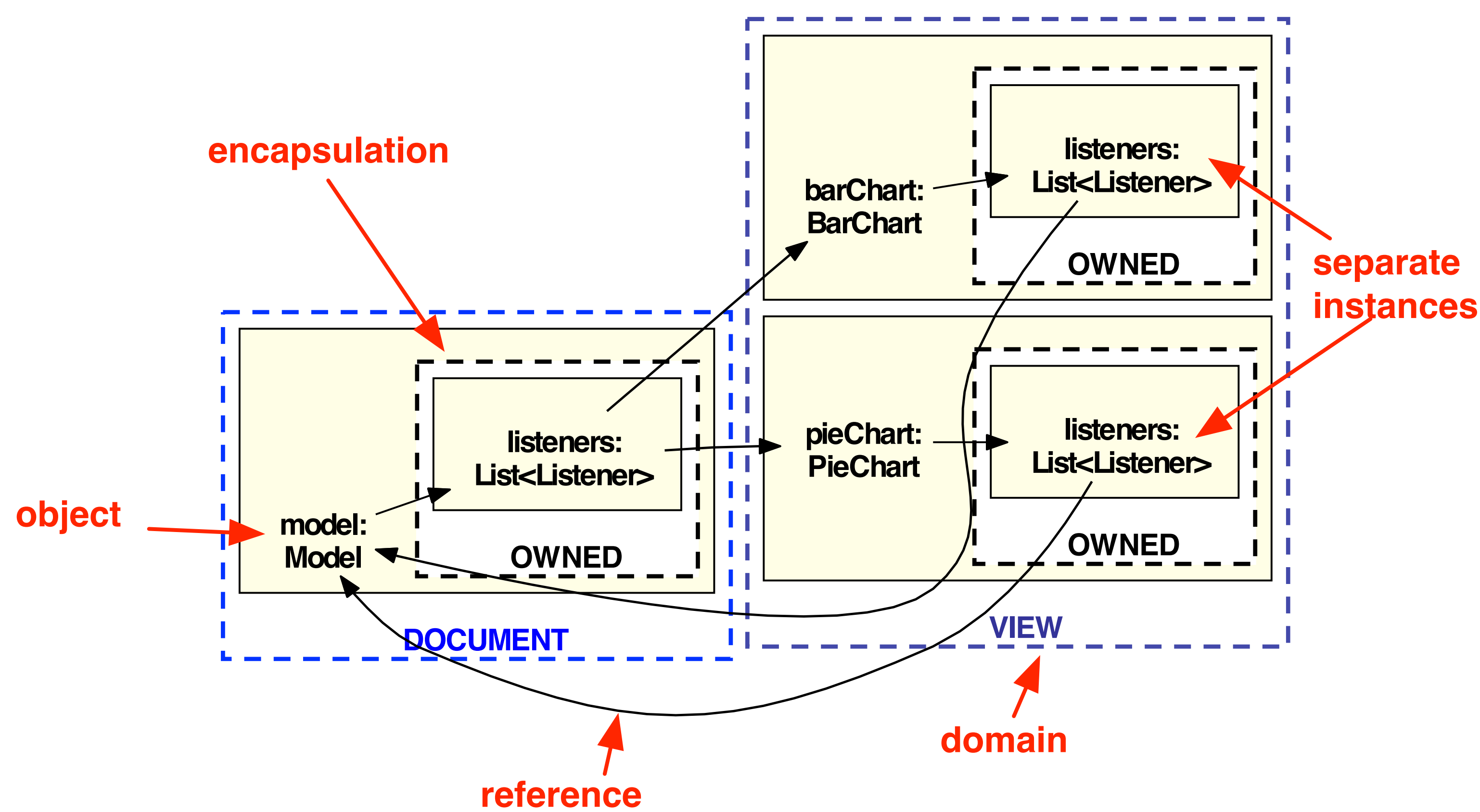
Thomas D. LaToza
Carnegie Mellon University

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

Ownership Object Graph (OOG)



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

Method

We performed an **exploratory** observational study of developers performing coding tasks using a prototype OOG tool.

Participants 2 grad students and 1 current software developer
Tools Eclipse IDE, OOG tool, UML class diagram tool
Codebase JHotDraw 5.3 (200 classes, 15 KLOC), MVC design
Tasks bug fixes, feature requests. Varied across participants
Setup thinkaloud, audio recording, screen recording

Procedure

Overview Demo JHotDraw and Eclipse navigation
10 - 15 mins
Do tasks ~ 85 mins
Interview 10 mins
Discuss how participants worked and perceived challenges

Analysis

Time into video	Code	Navigation event	Navigation target	Think aloud of participant
0:15	Fact Is-In-Tier			so you want me to impl guess this is related to I I can get to know how y the list since you want r the list or all the figures so when I open a new v should be 0?
	Fact Part-Of			
	Question			

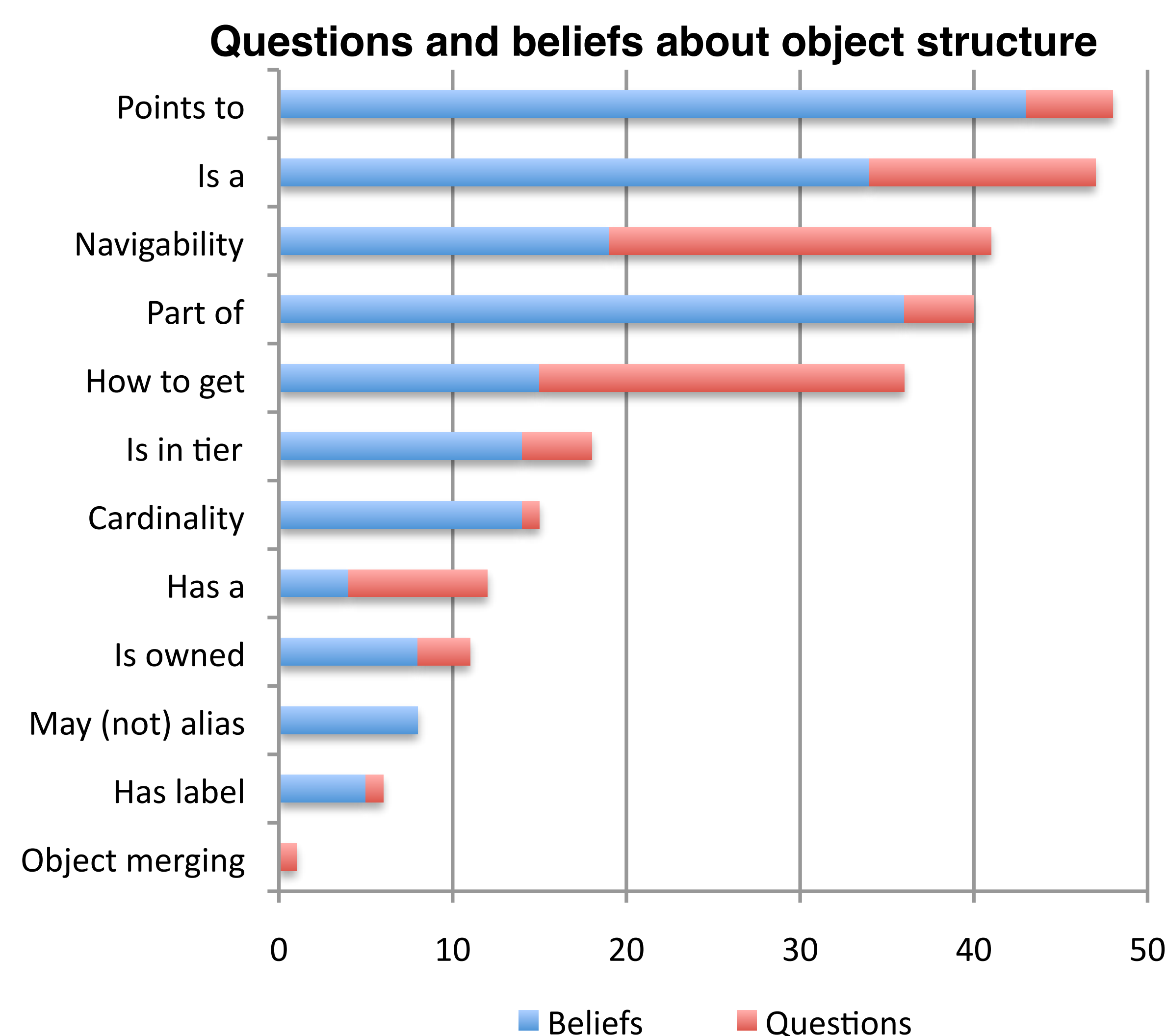
Transcripts

Coding

Questions and beliefs about object structure

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.



Is a	Who implements type X? [who can be an object or a type]
Navigability	Let's say I am in the StandardDrawing class and I want the JavaDrawApp object which is a DrawingEditor [...]. What would save me a lot of time is to say now I am at the Drawing and I want to go to the DrawingEditor, show me my options.
Part of	Maybe I would start with the Drawing object and that should have a list of listeners?
How to get	How I will get hold of the DrawingEditor object? [...] Basically I need to know the instance of the current window. I know I need to get the view from here; so how do I do that?
Is in tier	What I would be interested in is looking in the code to try to understand where are the view and model
Cardinality	The class diagram says that the DrawingEditor has one DrawingView and the StandardDrawingView may or may not have a Drawing. I would like to know the cardinality: so Window has one or more StandardDrawingViews?
Has a	Maybe I would start with the Drawing object and that should have a list of listeners
Is owned	[...] the window itself has a reference to the UndoManager but you can't tell from this diagram whether each window has its own UndoManager, or whether it is just one global manager.
May alias	So I have different selections in the different views. Both of them are two views on the same Drawing, but if there are two windows...

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.