Mitchell Simon

Prof. B. J. Johnson

Assignment #2

5.1: What's the difference between a component-based architecture and a service-oriented architecture?

A component-based architecture is a a set of loosely coupled components that provide parts of the application to each other. Where as a service-based architecture is similar in that the different components provide things to each other, but it involves more separation, typically where each service is standalone and possibly even on other machines across a network.

5.2: Suppose you're building a phone application that lets you play tic-tac-toe against a simple computer opponent. It will display high scores stored on the phone, not in an external database. Which architectures would be most appropriate and why?

A small application without a need for a database or any outside contact would make a good monolithic application. A data driven architecture would work well with the data for moves in a game.

5.4: Repeat question 3 [after thinking about it; it repeats question 2 for a chess game] assuming the chess program lets two users play against each other over an internet connection.

It would like be best to use a web service as a middle man between client applications. So it would likely be monolithic client applications and a data driven,

monolithic web server to act as the middle man, but because it's broken up this way it’s also a service based architecture.

5.6: What kind of database structure and maintenance should the ClassyDraw application use?

It doesn’t really need a database, perhaps for settings or user preferences. Otherwise the images should be saved to either project files or image files.

5.8: Draw a state machine diagram to let a program read floating point numbers in scientific notation as in +37 or -12.3e+17 (which means -12.3x10^17). Allow both E and e for the exponent symbol. [Jeez, is this like Dr. Dorin’s DFAs, or what???]

Check picture included.

6.1: Consider the ClassyDraw classes Line, Rectangle, Ellipse, Star, and Text. What properties do these classes all share? What properties do they not share? Are there any properties shared by some classes and not others? Where should the shared and nonshared properties be implemented?

They are all things you can draw, so they share properties associated with that like colors. They also have some independent properties, so they should use inheritance to share properties in common and save time coding.

6.2: Draw an inheritance diagram showing the properties you identified for Exercise 1. (Create parent classes as need, and don’t forget the Drawable class at the top.)

Check picture included.