Problem 5.1, Stephens page 116

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

A component-based architecture thinks of pieces of the system as loosely coupled components that provide serves for each other, while a service-oriented architecture is kind of the same but the pieces are implemented as services, usually running on separate pc’s on a network. They’re pretty similar but the pieces are more separate in the service-oriented architecture.

Problem 5.2, Stephens page 116

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?

This is a really simple self-contained app, so no database is really required. A monolithic architecture makes the most sense because it’s data-centric and it’ll be easy to build tables of moves and the best responses. The UI would be event driven and so would the computers moves.

Problem 5.4, Stephens page 116

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.

The main changes for an online version would be that the program needs to exchange info with another instance of the program across the Internet and there’s no computer opponent. It’d still be a monolithic rule-based service oriented application.

Problem 5.6, Stephens page 116

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

The ClassyDraw app can store each drawing in a separate file, so it doesn’t really need a database. OS’s have tools that let the user manage files, like moving and deleting.

Problem 5.8, Stephens page 116

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 -12.3 x 1017). Allow both E and e for the exponent symbol. [Jeez, is this like Dr. Dorin's DFAs, or *what*???]

IMG ATTACHED

Problem 6.1, Stephens page 138

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?

All the classes represent stuff that’s drawn so they share properties related to being displayed. Star needs to be given how many points on the star and text requires an input string. Those are the main differences.

Problem 6.2, Stephens page 138

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

PUT LATER ALSO

Problem 6.3, Stephens page 139

The following list gives the properties of several business-oriented classes.

* Customer — Name, Phone, Address, BillingAddress, CustomerID
* Hourly — Name, Phone, Address, EmployeeID, HourlyRate
* Manager — Name, Phone, Address, EmployeeID, Office, Salary, Boss, Employees
* Salaried — Name, Phone, Address, EmployeeID, Office, Salary, Boss
* Supplier — Name, Phone, Address, Products, SupplierID
* VicePresident — Name, Phone, Address, EmployeeID, Office, Salary, Managers

Assuming a Supplier is someone who supplies products for your business, draw an inheritance diagram showing the relationships among these classes. (Hint: Add extra classes if necessary.)

PUT LATER ALSO

Problem 6.6, Stephens page 139

Suppose your company has many managerial types such as department namager, project manager, and division manager. You also have multiple levels of vice president, some of whom reprt to other manager types. How could you combine the Salaried, Manager, and VicePresident types you used in Exercise 3? Draw the new inheritance hierarchy.

IMG ATTACHED