Justin Sanny

2/13/2017

Assignment #2

Problem 5.1, Stephens page 116

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

Component-based architecture is architecture that regards the system as a collection of loosely coupled components that provide services for each other. Service-oriented architecture have pieces that are implemented as services.

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?

Simple monolithic architecture rule-based architecture would be most appropriate. Since there is no need for a external database and everything is within the phone, simple monolithic architecture cuts down on work and need for communication. It is a standalone game so monolithic makes the most sense.

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.

Since Internet connection will be used, a monolithic rule-based (data-centric) service-orientated application seems best. This is due to the fact that there needs to be multiple pieces that are being used by the application.

Problem 5.6, Stephens page 116

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

ClassyDraw does not seem to need a database since drawings are separate files. ClassyDraw can have a separate or temporary memory file that stores the last drawing the player uses. In addition a save/load function or operating system can allow players to store copies of drawings and use them again.

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*???]

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?

Properties that all of these share are forecolor, back color, upper left, width, and height.

Properties that are specific to text is font and string, specific to star is number of points.

Fill color is shared specifically by Rectangle, Ellipse, and Star.

Lastly, line thickness and dash-style are shared by Line, Rectangle, Ellipse, and Star.

Yes there are properties that are specific to some classes.

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.)

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.)

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.