Katie Nguyen

Sofia Ruiz

Moriah Tolliver

HW Assignment #2

Problem 5.1, Stephens page 116

In a component-based architecture, pieces of the system are loosely coupled components that provide services for one another. On the other hand, a service-oriented architecture implements the pieces as services that are more separate than in a component-based architecture. Here, the services run on separate computers, which communicate via a network.

Problem 5.2, Stephens page 116

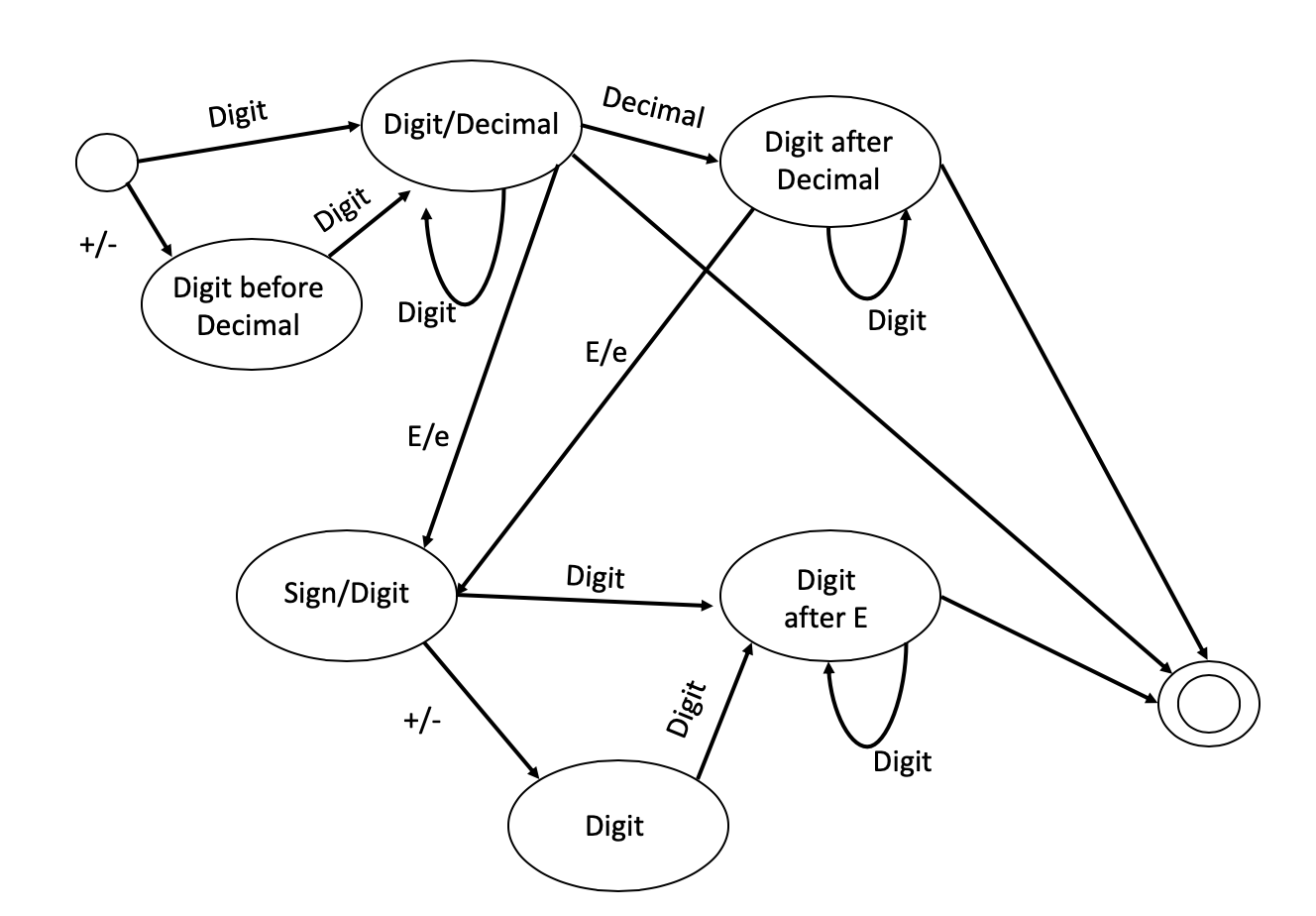
A monolithic architecture would probably work well because it’s a relatively small, self-contained application. A data-centric approach would also be acceptable. For tic-tac-toe, it is straightforward to create tables of user moves and the best way to respond to them, so it will probably use some data-centric or rule-based approaches. The user interface will be event-driven.

Problem 5.4, Stephens page 116

In the case that users could play against each other over an Internet connection, the computer player is removed, meaning that the program now needs to transmit data to another running instance of the program via the Internet. Since the computer will no longer be acting as a player, the program will not need distributed pieces. In this sense, the program then becomes based on a monolithic rule-based, service-oriented architecture.

Problem 5.6, Stephens page 116

The application does not have a real need for a database because it can store each drawing in a stand-alone file, which the user can manage on their own outside of the application. In terms of maintenance, the app could store the current drawing in a temp file so that this could be restored in the event of a possible crash.

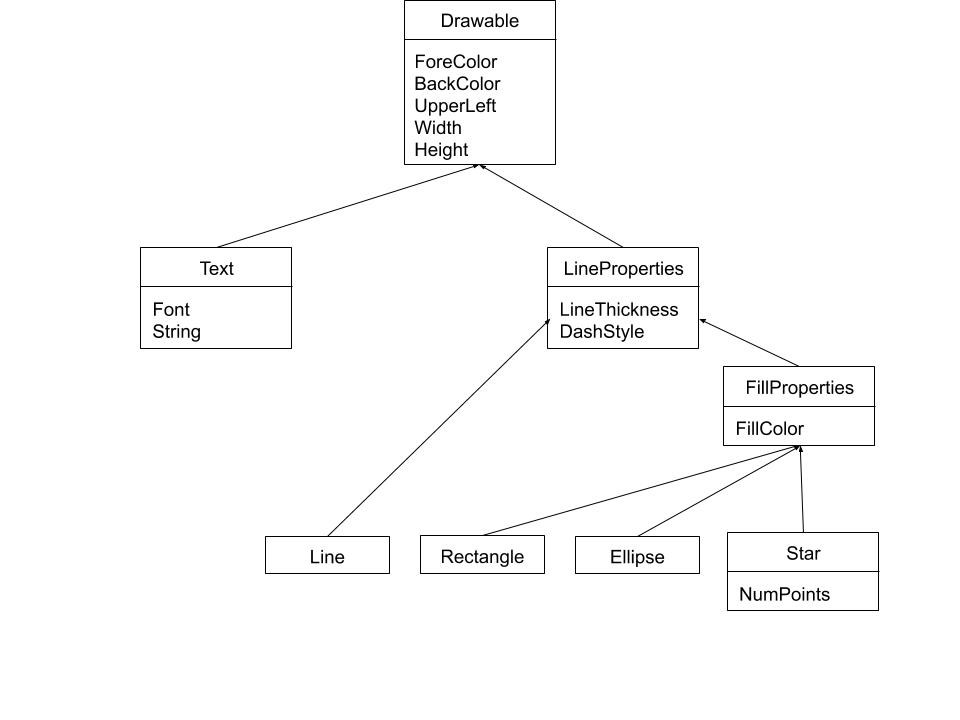
Problem 5.8, Stephens page 116

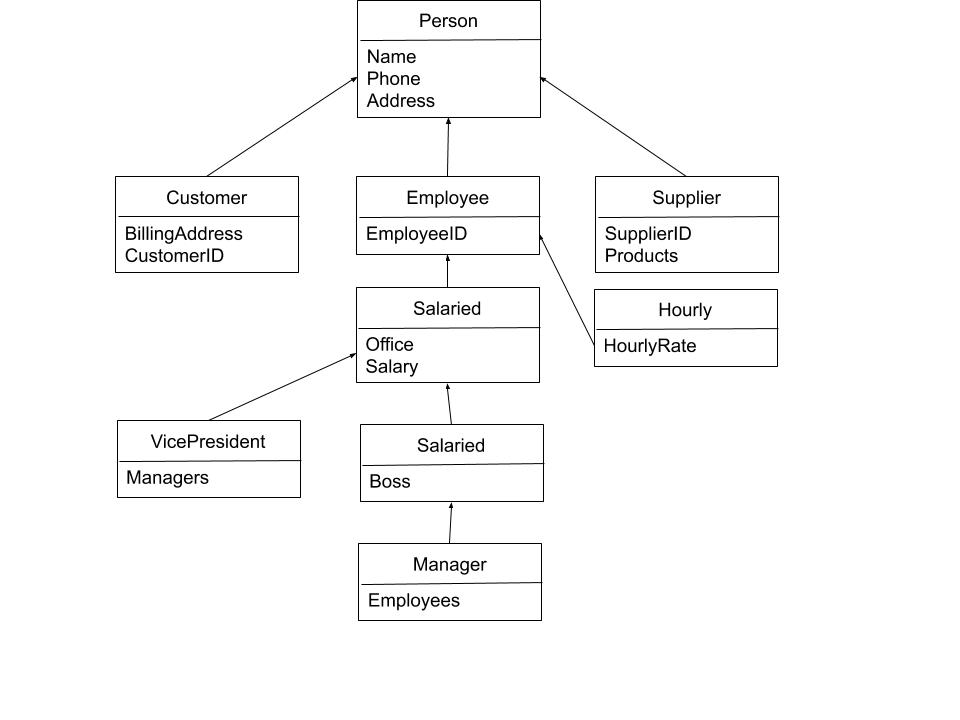
Problem 6.1, Stephens page 138

Below is a table of shared and not-shared properties:

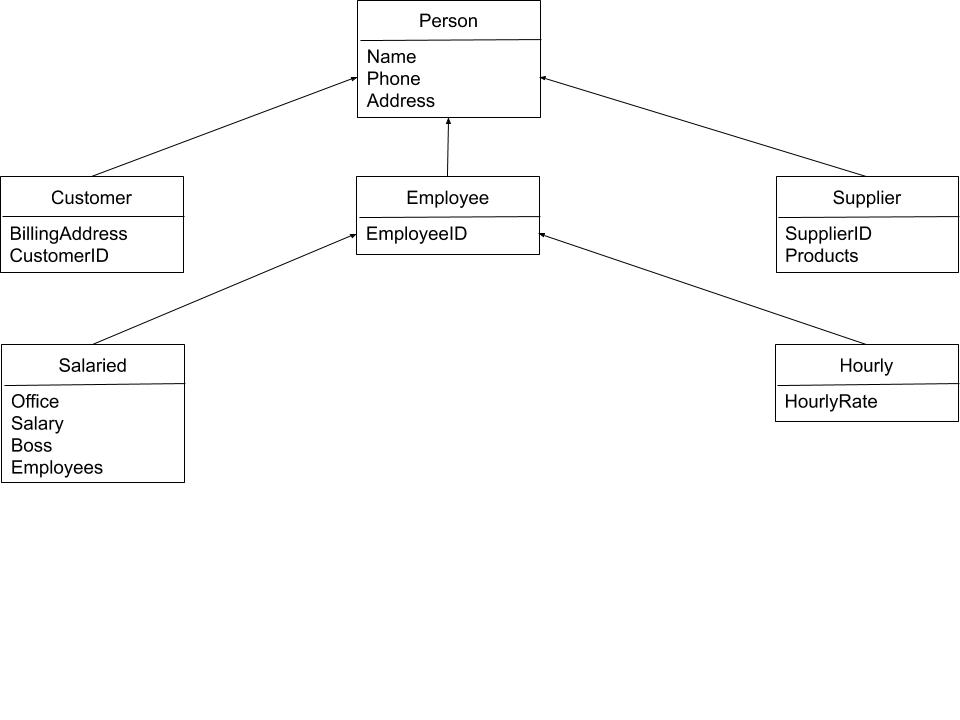
|  |  |
| --- | --- |
| Property | Used by: |
| ForeColor | All |
| BackColor | All |
| UpperLeft | All |
| Width | All |
| Height | All |
| Font | Text |
| String | Text |
| NumPoints | Star |
| FillColor | Rectangle, Ellipse, Star |
| LineThickness | Rectangle, Ellipse, Star, Line |
| DashStyle | Rectangle, Ellipse, Star, Line |

Problem 6.2, Stephens page 138



Problem 6.3, Stephens page 139

Problem 6.6, Stephens page 139

You could combine all the salaried employees, then the “employees” field would be empty for non-managers and the “boss” field would be empty for anyone without a boss. The restructured inheritance hierarchy is below: