Problem 5.1, Stephens page 116

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

Component-based architecture regards the system as a collection of loosely coupled components that provide services for each other. It decouples the pieces of code much as a multitier architecture does, but the pieces are all contained within the same executable program so they communicate directly instead of across a network. Service-oriented architecture is similar to component-based architecture except the pieces are implemented as services. A service is a self-contained program that runs on its own and provides some kind of service for its client (ex: web service). As a condensed summary, component-based architecture treats a system of different parts as a whole while SOA treats each component separately.

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?

A rule-based structure would work best for a tic-tac-toe game due to its simplicity, clear objectives, and limited unexpected scenarios. A monolithic system is also ideal as the game is self-contained, not relying on external databases and operates on a single device.

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.

Chess pieces would return to being rule-based as they all adhere to a set of rules that govern their interactions and specific movements. A client/server structure would be utilized to facilitate online chess games.

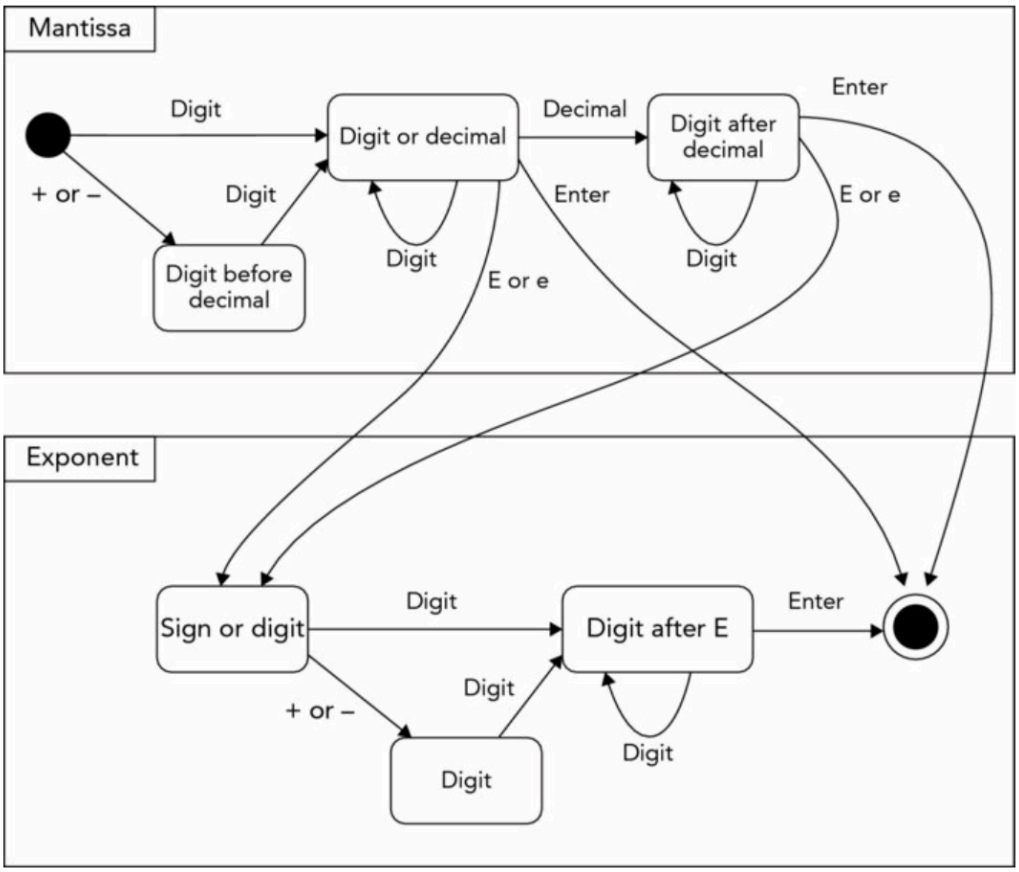
Problem 5.6, Stephens page 116

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

The application doesn't really need a database since any file created could be stored already as an individual file, and for maintenance, the app could create temporary files if any changes are being made and store those until edits are done being made.

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 10^{17}). 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?

The properties they all share are for drawing position, width, height, background and foreground coloring. The properties they don't share are for the Text class, string and font. The properties shared by some but not others are LineThickness, FillColor, and DashStyle. For Rectangle, Ellipse and Star, the property needed is FillColor, and for Rectangle, Ellipse, Star, and Line, the properties needed are LineThickness and DashStyle.

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.)