The Observer Pattern through Athlete Observation

Dave Browning | A01256705

October 14, 2017

1 Introduction

The purpose of this report is to demonstrate familiarity with UML Class and Interaction diagrams, the Observer pattern, and the Decorator pattern.

2 Diagrams

2.1 Visual Paradigm Class and Interaction Diagrams

Figure 1 is the App Layer of the final project. The key pieces are the Singleton Pattern as seen in the ObservedAthletes class, and the Athlete, Subject, and AthleteObserver relationships.

Figure 2 outlines the Gui Layer. Key points include the strategy pattern being employed in the Observers, and the relationship between the control form and the observers.

Figure 3 is my attempt at an interaction diagram. I've outlined the interaction between receiving an update, processing it, and sending the relevant information on to the athletes.

Figure 4 is a sequence diagram outlining the same relationship as figure 3, but adds to the model by showing how the process flows.

3 Insights

3.1 Design

I struggled with this assignment more than I have with all other programming assignments up to this point in my academic career. I've spent several hours collaborating with classmates in an attempt to understand the code base and be able to interact with it intelligently, however the practical implementation is beyond my current skill level given the time allotted.

I believe this failure on my part is largely due to the fact that I could not internalize the code and follow it from end to end until the day the assignment was due. I believe I have spent too little time on the design and too much time trying to hack my way through the implementation.

3.2 Implementation

Due largely to the fact that I dislike merely taking ideas from a sample project and adapting them to the assignment, I spent a great deal of time trying to build pieces of the project in order to better understand it as a whole.

As a consequence, I think that given another week or two I would be able to come up with my own implementation of the control form and the corresponding GUI's.

I have not been able to successfully implement the decorator pattern; however, if I could spend more time getting a graphical observer working, I think the most obvious place to put decorators is in objects moving across a gui, for example color, size of object etc.

In a way the updateMessage gets decorated by the various updateMessages.

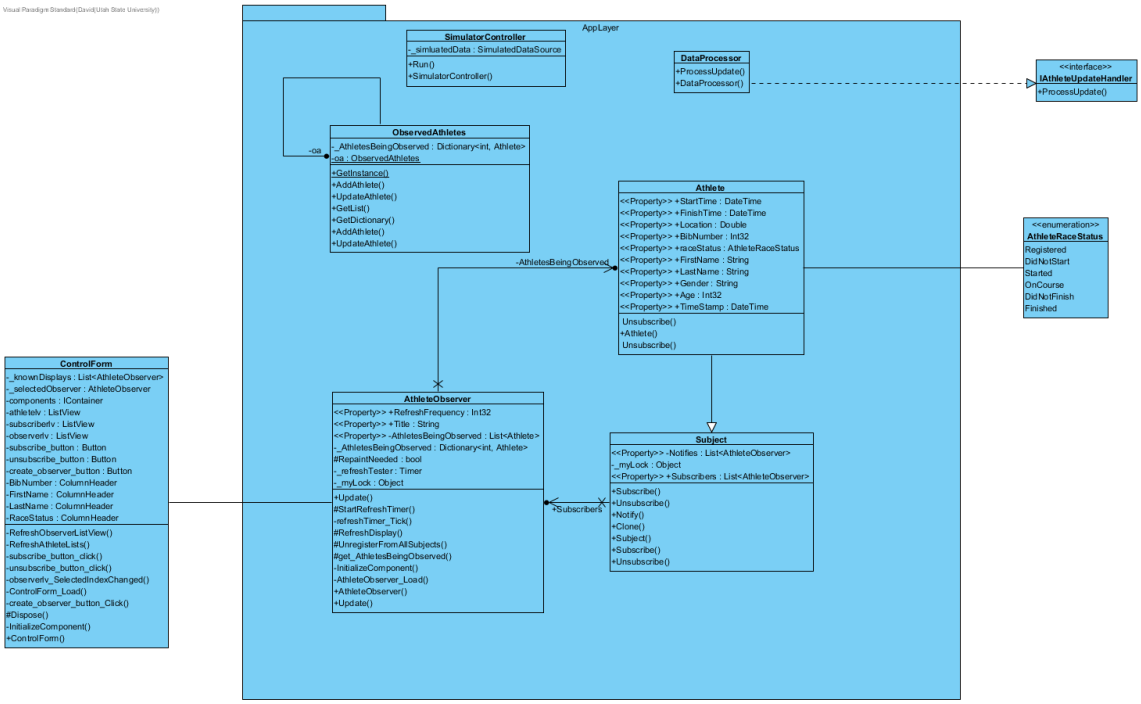

Figure 1: App Layer

3.3 Testing

As anyone familiar with the development process knows, the ideal situation is to be able to test your code as your write it. The trouble is the insecurity that comes with not adequately understanding what you're doing well enough to test it in any kind of intelligent way.

Ultimately, I decided to focus my time on the subject and observer tests as those were central to the assignment. I did include a Singleton test to be sure I had implemented it correctly.

Figure 2: Gui Layer

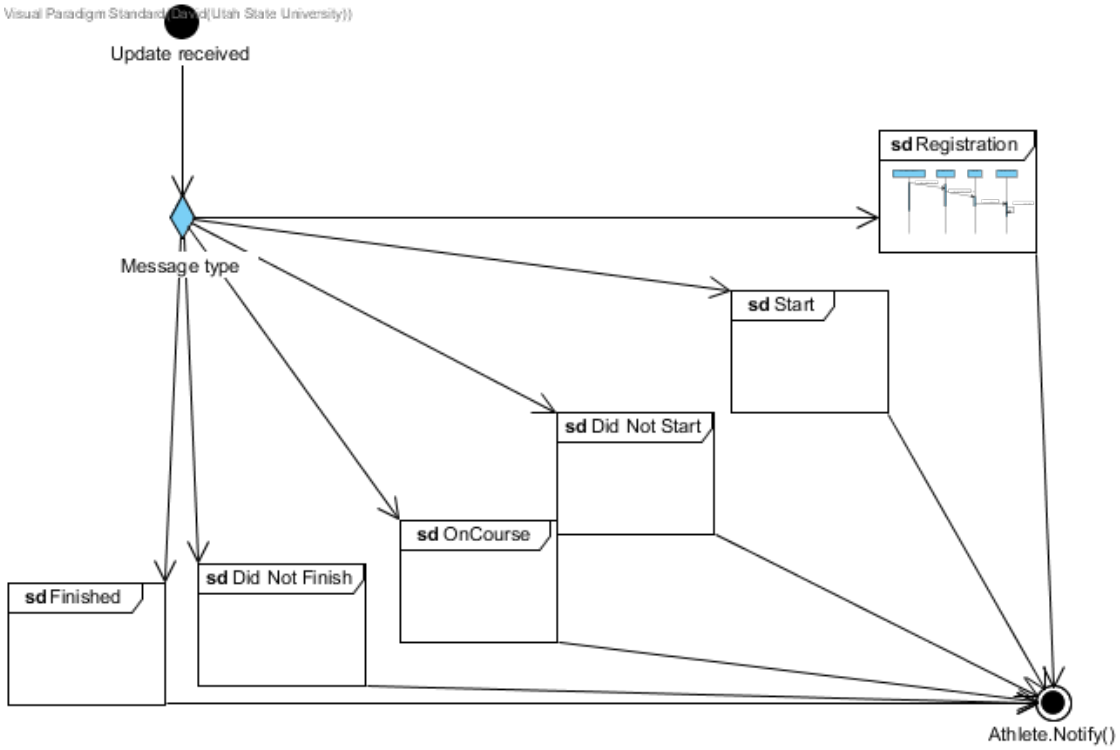

Figure 3: Interaction Diagram

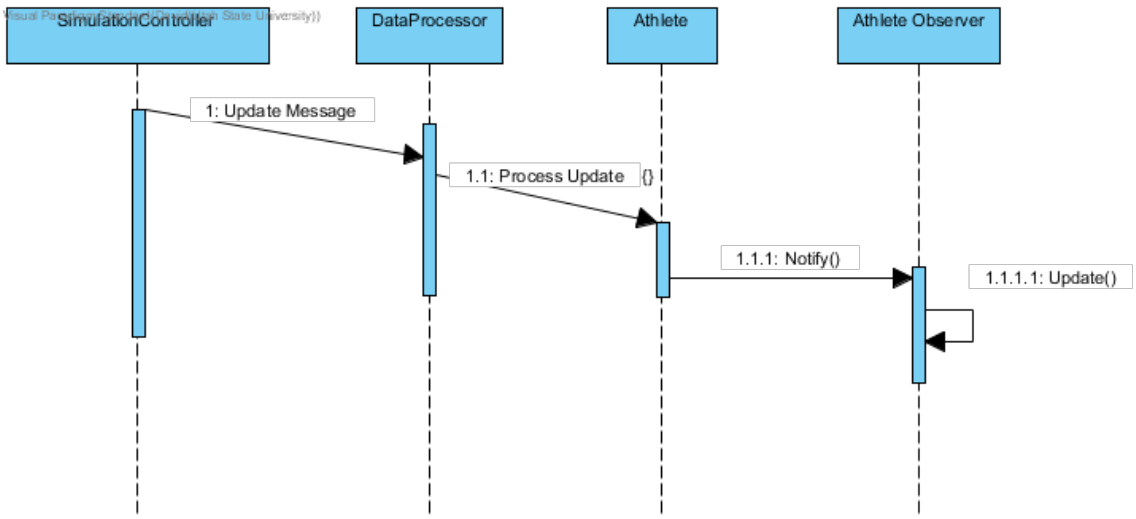

Figure 4: Sequence Diagram