

Nolan Anderson

CS 307

Jacob Hauenstein

28 March 2021

Programming Assignment 2: Preliminary class Diagram

Section 1 – General Overview

The functionality and basic overview of assignment 2 will be similar to assignment 1, but the implementation will be very different due to the fact that we now have to deal with aircraft types.

Section 2.1 describes the main executer of the program, and sections 2.2-2.4 will describe the classes that will hold data and do calculations for section 2.1's output. Sections 2.5 and 2.6 describe the parsers, but I did not go into much detail as they are simply data parsers. Lastly, an additional class described in section 2.7 will be important for dealing with the new aircraft types.

Section 2 – Class Overview

2.1 Class Simulation

This class will be the main rapper for initializing the data files / classes, sending out and collecting data on time intervals, maintaining time, and ending the program. It will have a Aggregation relationship, one to many, for the 3 classes described in sections 2.2, 2.3, and 2.4. It will call functions defined in these classes to output data. Continually, no calculations will be done here other than time. The data functions will take care of this as well.

2.2 Class CityData

Here, we will be gathering and compiling data from the CityData.xml file provided in the zip file. It will contain distances, names and other relevant information about all of the different cities. Continually, it will hold the calculations for distances between cities. Lastly, it will have a dependency relationship to CityDataParser described in section 2.5. CityData will gather the data it needs and store in the form of a list of structs. It will only need one instance of the CityData parser.

2.3 Class FlightData

Holding all of the flight data is very important in doing our calculations, specifically concerning where the flights are going, where they're leaving from, the type of aircraft etc. It will do the calculations for how long a flight will take, the current location among other calculations. Continually, it will have a dependency relationship with CityData and AircraftData to do its calculations. This will be the main class that does the brunt of the data collection from the other two classes. Additionally, the simulation class will call the functions defined in FlightData.

2.4 Class AircraftData

This class will hold all of the data of the different types of aircraft including cruise speed, altitude, type of aircraft etc. It will have dependencies of section 2.6 and 2.7. Here, we will be using the Flyweight Design pattern to hold the data to avoid redundancy. Additionally, this class will do all of the calculations for time to cruise speed, current speed etc. This class is fundamentally important for holding aircraft data and doing important calculations for FlightData.

2.5 Class CityDataParser

This class is provided for us and will extract the data from the xml files. The sections described previously need this class to get the data. I will not go into detail for this since it is simply a parser.

2.6 Class FlightDataParser

This class is provided for us and will extract the data from the xml files. The sections described previously need this class to get the data. I will not go into detail for this since it is simply a parser.

2.7 Class AircraftFactory

Aircraft factory will be a subclass of section 2.4, AircraftData. It will be implemented using the Singleton design pattern. It will return pointers to aircraft objects. Overall, this class will be added in from the previous assignment due to the fact that we need to know where the aircraft are produced. The different types are defined in the AircraftTypes header file and the FlightDataParser will give us an aircraft type this time instead of just the data of a specific aircraft. To avoid redundancy, we will need to use the flyweight design patter. This class has no dependencies, but the AircraftData class depends on its data.

Section 3 – Class UML Diagram

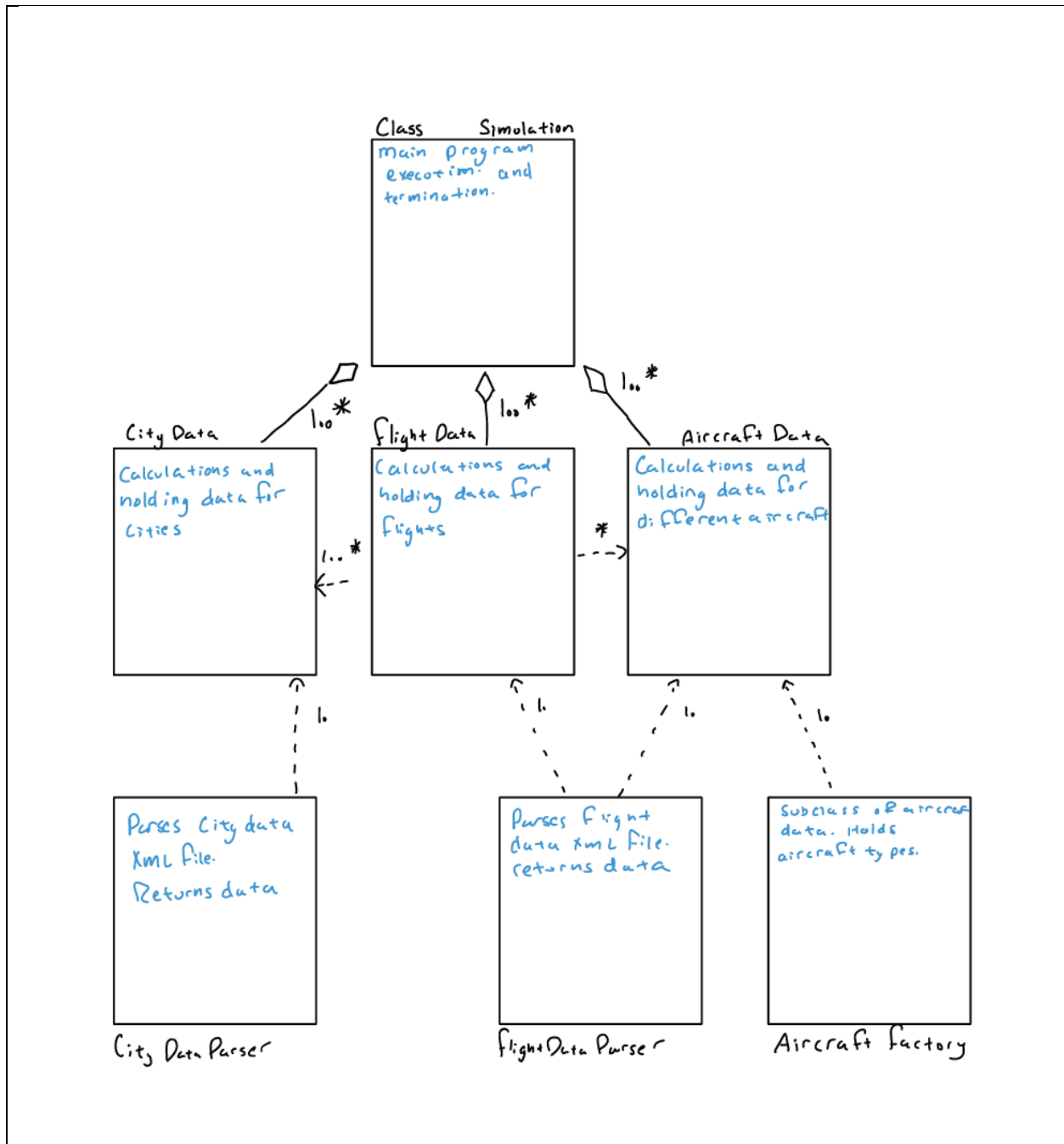


Figure 1: Preliminary Class Diagram