1. Introduction
2. Identify the issues of design patterns and code quality

The design pattern of this metro simulation project is good enough for only current needs. However, some parts of designs may be not upgradable for Metro Melbourne’s future satisfactions. The following paragraphs will identify this type of issues and treatments of associated issue will be provided.

The first issue of this design pattern is that there are only two types of trains currently, which are the train that can take 10 or 80 passengers. As one of the aims of this project is that simulating the network of trains to help the Melbourne Metro making the decisions of where new stations should be built, the simulation should allow the client to vary the size of trains to analysis the behaviour of passengers when the train size varies.

The modification that should be made for the first issue is that creating new classes replacing the original type classes. Specifically, the two classes – SmallPassengerTrain and BigPassengerTrain should be replaced by a single class called PassengerTrain. The constructor of the new class should include a new variable, which controls the size of train, and update this variable as the size of the train in its methods. The other contents in the PassengerTrain should keep the same as either SmallPassengerTrain or BigPassengerTrain. Because the size of trains is not fixed anymore, few more changes should be made in the classes and the map file. The type of each train under <trains> section in the map file should not be strict as BigPassenger or SmallPassenger, but also (train.java, mapreader.java, map file)

No controller of track, train and station service

Passenger connects to passenger generator

Code quality, magic numbers, commands