|  |  |  |
| --- | --- | --- |
| Design Patterns | February 29  2016 | |
| Monica Stoica  Rosen Danev  Public Transportation Application | | Observer pattern |

Table of Contents

[Introduction 2](#_Toc444544259)

[UML Diagram 2](#_Toc444544260)

[User interface 3](#_Toc444544261)

[Reusability 4](#_Toc444544262)

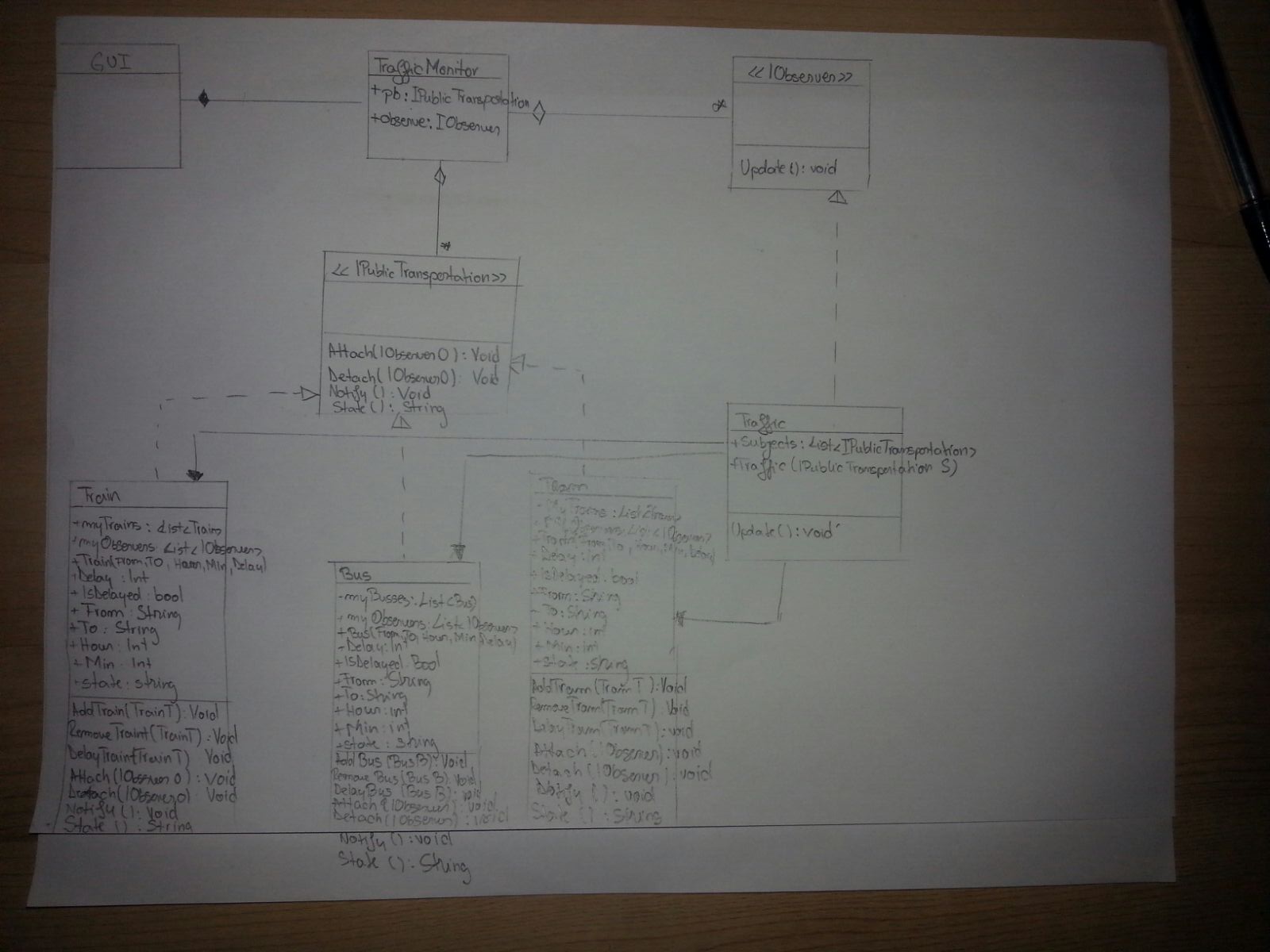
[Maintainability 4](#_Toc444544263)

[Extensibility 4](#_Toc444544264)

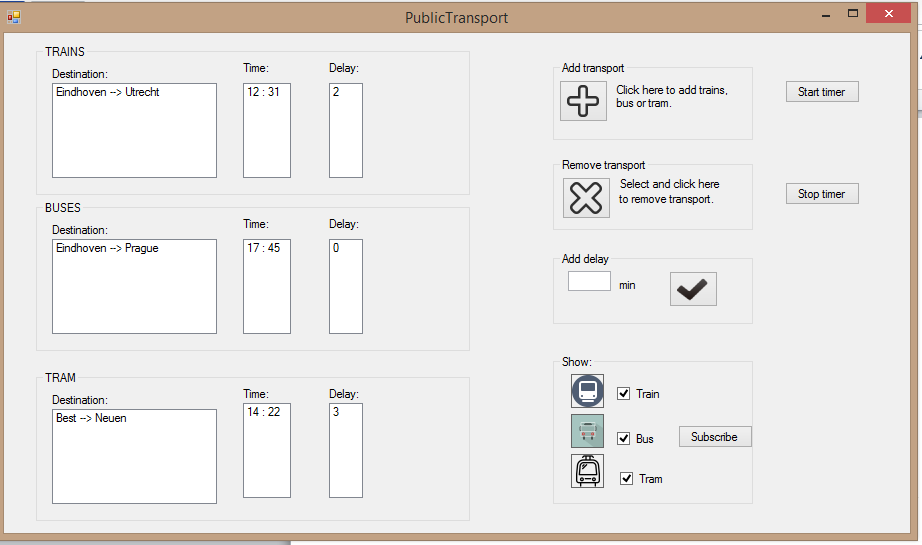
# Introduction

The following document analyses the three main characteristics of the pattern used to develop an application that monitories the public transportation such as: trains, buses and trams. This application allows the user to see the schedule of the above named public transports. The user can also add or remove a vehicle and/or delay it. The observer, the traffic monitor is subscribed to both three types of commuting. Therefore, whenever the state of one of vehicles changes, the observer will be notified. The observer can also choose to subscribe/unsubscribe only to the subjects that interest it (e.g Monitor only the trains and unsubscribe for the busses and trams). A pattern is general reusable solution to a commonly occurring problem within a giving context. The purpose of using patterns is to speed up the developing process and helps preventing issues that can cause major problems.

# UML Diagram



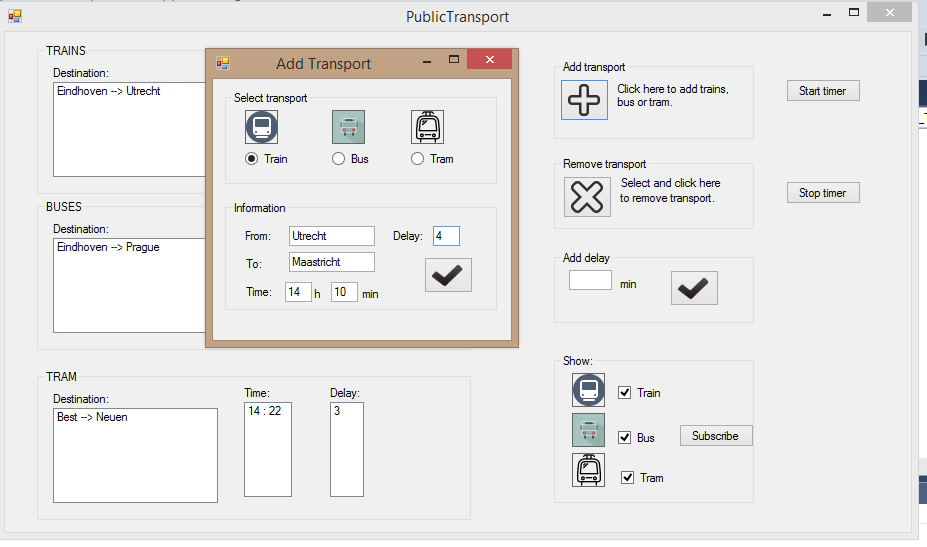
# User interface



The three groupboxes on the right represent the subjects – the public transportation: trains, busses and trams. Every time an object is added, removed (or delayed) from the list, the observer will be informed by its status. An observer has the possibility to ‘subscribe’ to these three subjects. If he/she wants to receive updates only about busses, then the user has to uncheck the train an tram checkboxes and press the button subscribe.

The items can be removed from the listobx automatically by pressing the button Start timer. The first item of each listbox will be removed and the timer will be incremented with 5 seconds.

To add a subject, the user has to press the button add. A new window will be displayed in which the details of the journey have to be inserted.



# Reusability

One of the main reasons why patterns are useful is because they can be easily reused without changing the code. Considering our above described application, we can attest that the pattern is reusable. The observer pattern can be reused and new behaviors can be added without modifying the TrafficMonitor class. Moreover, the behaviors, in this case Bus, Train, Tram, can be changed at run time.

# Maintainability

Because the classes are not tightly coupled and the pattern is quite small, the system is easily maintained. When a system is easy to maintain it means that new features can be added.

# Extensibility

New functionality can be provided by adding new code without changing the initial one. Therefore, there is no need to worry about bugs or causing problems. The current pattern can be easily extended because it does not depend on other classes. There can be added as many subjects or observers as wanted. They can also be connected in any way (e.g one observer subscribed for 4 subjects while the other subscribed for the remaining 5)