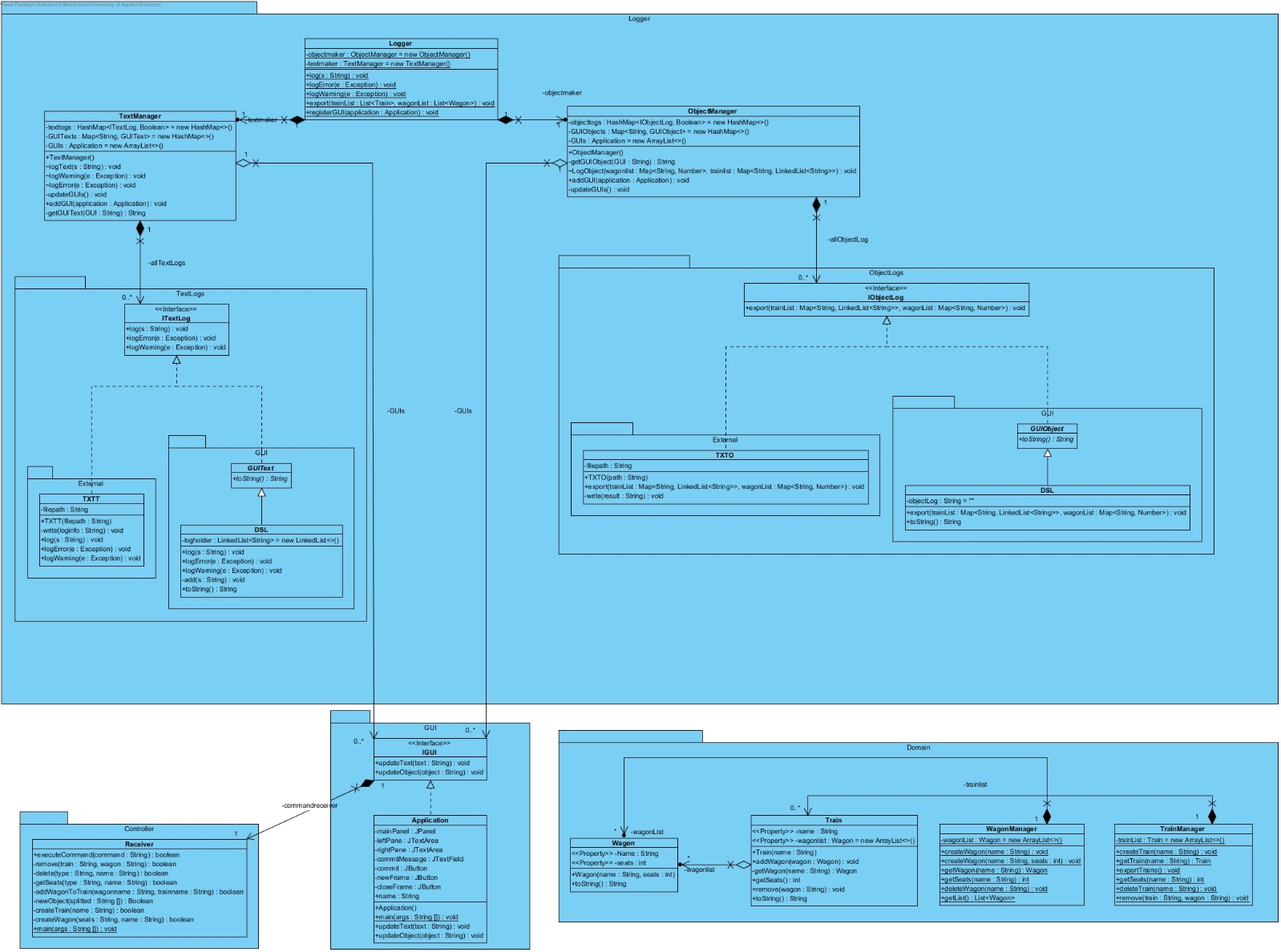
Motivatie Code



(Klassendiagram: model.jpg) (**note:** receiver geeft static commando’s aan wagonmanager en trainmanager, verder is er geen kennis over het domeinmodel bekend bij de receiver)

**Voor een beter kwaliteit en inzoomen, zie model.jpg.**

# GUI

IGUI vormt een laag tussen de GUI’s en de back-end applicatie. Hierdoor kunnen er verschillende GUI’s simpel op de applicatie gebouwd worden door de interface te implementeren, er zijn dan geen aanpassingen in de applicatie meer nodig.

Door middel van een observer pattern vanuit ‘Textmanager’ en ‘Objectmanager’ wordt de tekst in de gemaakte GUI actueel gehouden. Als de informatie anders weergegeven moet worden kan dit aan GUItext of GUIobject gehangen worden. (Strategy pattern & open-close principle)

# Controller

Als een gebruiker in de GUI een command geeft (new/add/delete etc.) dan krijgt de ‘Receiver’ klasse deze binnen. De commando’s worden dan door deze klasse gedecodeerd en doorgestuurd naar de verantwoordelijke domeinklasse.

# Domain

Het factory-pattern wordt o.a. gebruikt door ‘TrainManager’ en ‘WagonManager’, zodat het beheren van de treinen en wagons gescheiden is van de objecten. Hierdoor hebben de klassen maar één taak (Single-responsibility Principle)

De overzichten worden ge-update door middel van commando’s naar de logger die de overzichten bijwerkt.

# Logger

De logger krijgt Textoutputs binnen, deze worden doorgestuurd naar de ‘TextManager’. Objectoutputs worden doorgestuurd naar de ‘ObjectManager’. Ook de geregistreerde observer GUI’s worden naar beide managers meegestuurd. Alles samen zorgt ervoor dat ‘Logger’ het enige aanspreekpunt is voor buitenaf en de aansturing van deze layer verzorgt.

Doordat TextGUI’s en externe Textbestanden dezelfde kennis van informatie mogen gebruiken, erven ze dezelfde interface en kunnen dus hierdoor gezamenlijk door de Textmanager beheerd worden. Ditzelfde geldt voor ObjectGUI’s en externe Objectbestanden met de Objectmanager. Ook deze erven de interface die beheerd wordt door Objectmanager.

omdat GUI-logs intern opslaan om hun informatie door te sturen naar de GUI’s en externe log-schrijvers hun informatie direct wegschrijven, onderscheiden de GUI’s zich van de externe log’s d.m.v. een extra interface (Strategy Pattern & Liskov substitution principle &Dependency Inversion principle & Interface segregation principle).

De manager stuurt nieuwe outputs ook gelijk door naar de subscribed GUI’s. Dit gebeurt met het in “GUI” uitgelegde observer pattern.

IObjectLog en ITextLog volgen een adapter pattern om de ontvangen informatie op de juiste manier weg te schrijven of door te geven aan de GUI. (Open-close principle)

**De SOLID Principles zijn vaak meerdere keren op dezelfde plekken terug te vinden.**