Example of Design Process

Mikael Svahnberg[*]

2020-03-04

1 Introduktion

Det här är ett exempel som vi började arbeta med på lektionen den *<2020-03-03 Tue>*, med tanke att fortsätta nästa lektion. Exemplet är inte komplett och förmodligen svårt att hänga med på om man inte var med på föreläsningen; om du är osäker så fråga dina studentkollegor.

2 System

☐ Självkörande Robot

☒ Timmer-Skördare

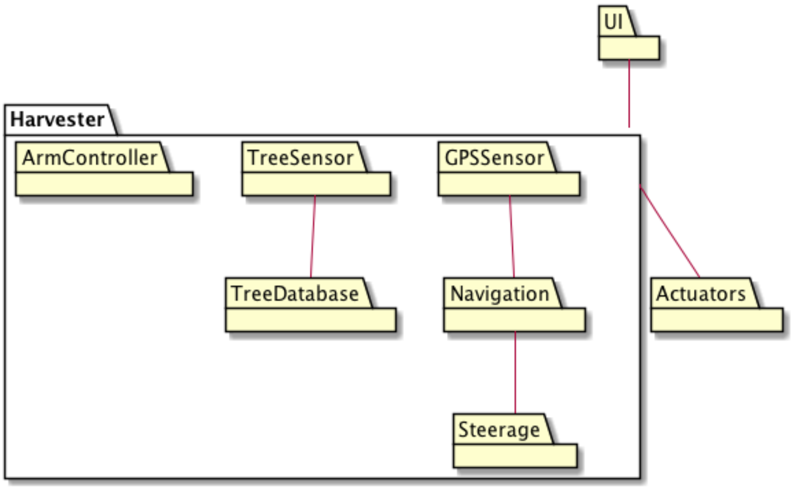
☐ Websida för Blekinge Rovdjursimport

3 Arkitektur – Paketdiagram

```
package Harvester {

package Navigation
package Steerage
package GPSSensor

Navigation -- Steerage
GPSSensor -- Navigation


package TreeSensor
package TreeDatabase

TreeSensor -- TreeDatabase

package ArmController
}
```

[*] Mikael.Svahnberg@bth.se

```
package UI
package Actuators

UI -- Harvester
Harvester -- Actuators
```



4 Use Case

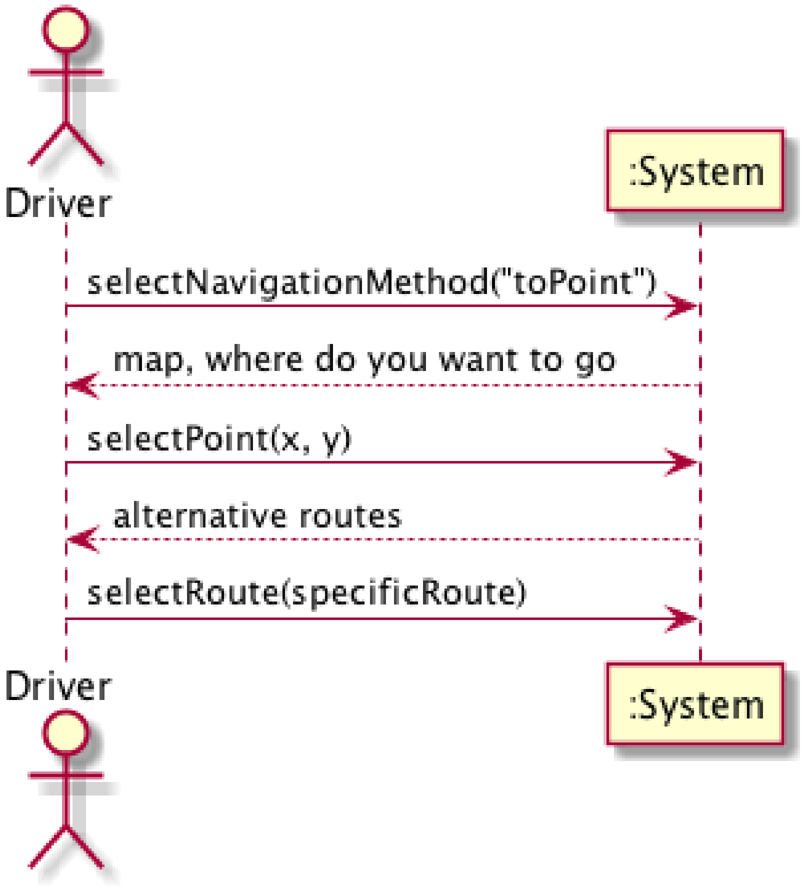
Use Case: Navigate to Point Actors: Driver Description: Driver selects that they want to navigate to a point. The system displays a map, the driver selects a point.

Main Course of Events:

Actor	System
1. Driver selects that they want to navigate to a point	
	2. The system selects “Navigate to Point” and presents a map. Asks where to go.
3. The Driver selects a point.	
	4. The system calculates a couple of alternative routes and displays them.
5. The Driver selects one route.	
	6. The system lurches away into the forest with a smile on its face.

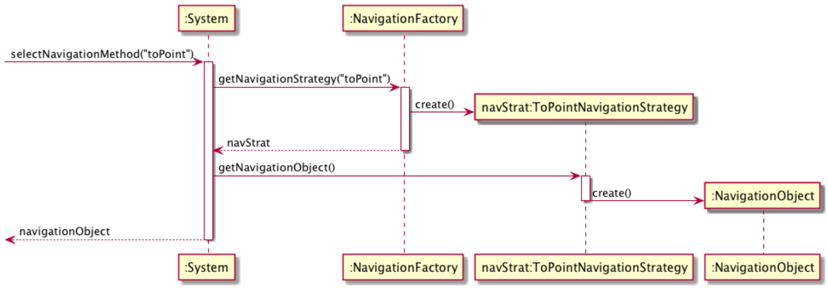
5 Systemsekvensdiagram

```
actor Driver as d
participant ":System" as sys

d -> sys : selectNavigationMethod("toPoint")
sys --> d : map, where do you want to go

d -> sys : selectPoint(x, y)
sys --> d : alternative routes

d -> sys : selectRoute(specificRoute)
```


Driver
:System
selectNavigationMethod("toPoint")
map, where do you want to go
selectPoint(x, y)
alternative routes
selectRoute(specificRoute)
Driver
:System

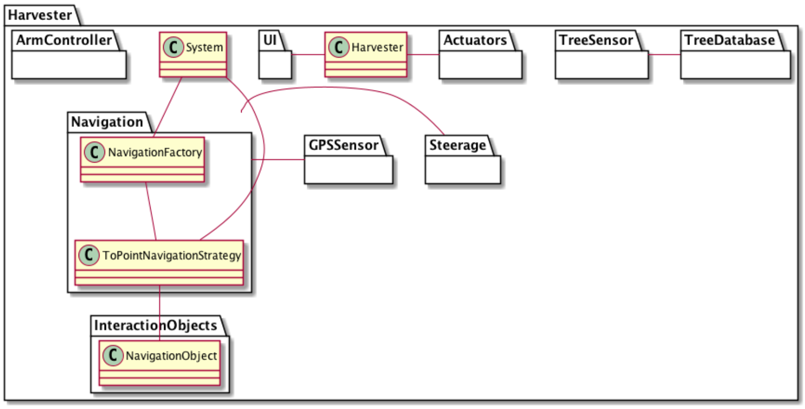
6 Interaktionsdiagram

```
[-> ":System" : selectNavigationMethod("toPoint")
activate ":System"
":System" -> ":NavigationFactory" : getNavigationStrategy("toPoint")
activate ":NavigationFactory"

":NavigationFactory" -> "navStrat:ToPointNavigationStrategy"** : create()
":NavigationFactory" --> ":System" : navStrat

deactivate ":NavigationFactory"

":System" -> "navStrat:ToPointNavigationStrategy" : getNavigationObject()
activate "navStrat:ToPointNavigationStrategy"
"navStrat:ToPointNavigationStrategy" -> ":NavigationObject"** : create()
deactivate "navStrat:ToPointNavigationStrategy"


[<-- ":System" : navigationObject
deactivate ":System"
```



7 Klassdiagram

```
package Harvester {

class System

package Navigation {
class NavigationFactory
class ToPointNavigationStrategy
}

package InteractionObjects {
class NavigationObject
}

System -- NavigationFactory
NavigationFactory -- ToPointNavigationStrategy
System -- ToPointNavigationStrategy
ToPointNavigationStrategy -- NavigationObject


package Steerage {}
package GPSSensor {}

Navigation - Steerage
GPSSensor - Navigation

package TreeSensor {}
package TreeDatabase {}

TreeSensor - TreeDatabase

package ArmController {}

package UI {}
package Actuators {}

UI - Harvester
Harvester - Actuators
```


Harvester
ArmController
System
UI
Harvester
Actuators
TreeSensor
TreeDatabase
Navigation
NavigationFactory
GPSSensor
Steerage
ToPointNavigationStrategy
InteractionObjects
NavigationObject