

Example: Robot System

Mikael Svahnberg*

May 3, 2019

Contents

1	Introduktion	1
2	Package Diagram	1
3	Use Case: Navigate to Point	2
4	Systemsekvensdiagram: Navigate to Point	2
5	Sekvensdiagram - selectNavigationMethod	3
6	Klassdiagram	4

1 Introduktion

Det här är ett exempel som vi började arbeta med på lektionen den *<2019-05-03 Fri>*, 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 Package Diagram

```
package UI
```

```
package Robot {  
package ControlInterface  
package Navigation  
package Steering  
package ArmControl
```

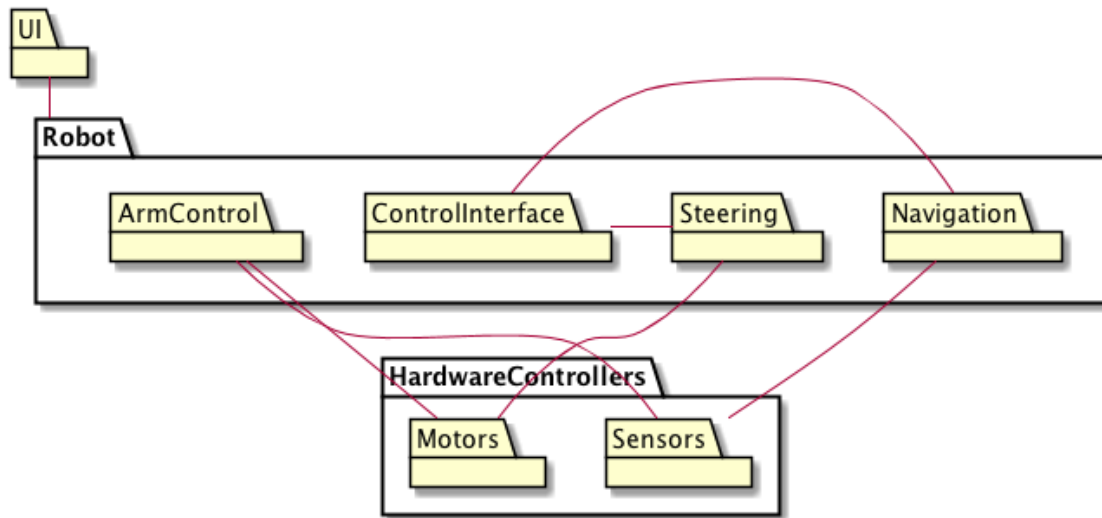
```
ControlInterface - Navigation  
ControlInterface - Steering
```

```
}
```

```
package HardwareControllers {  
package Motors  
package Sensors  
}
```

```
UI - Robot  
Steering - Motors  
Navigation - Sensors  
ArmControl - Motors  
ArmControl - Sensors
```

*Mikael.Svahnberg@bth.se



3 Use Case: Navigate to Point

Use Case: Navigate to Point

Actors: User

Description: User selects a coordinate and asks for routes to this point. System displays possible routes. User selects one route.

Requirements: FR120, QA2, FR111

Main Course of Events

Actor	System
1. User selects "Navigate to Point"	
	2. System displays a map and asks user to select a point
3. User selects a point	
	4. System calculates routes and displays them.
5. User selects one route.	

4 Systemsekvensdiagram: Navigate to Point

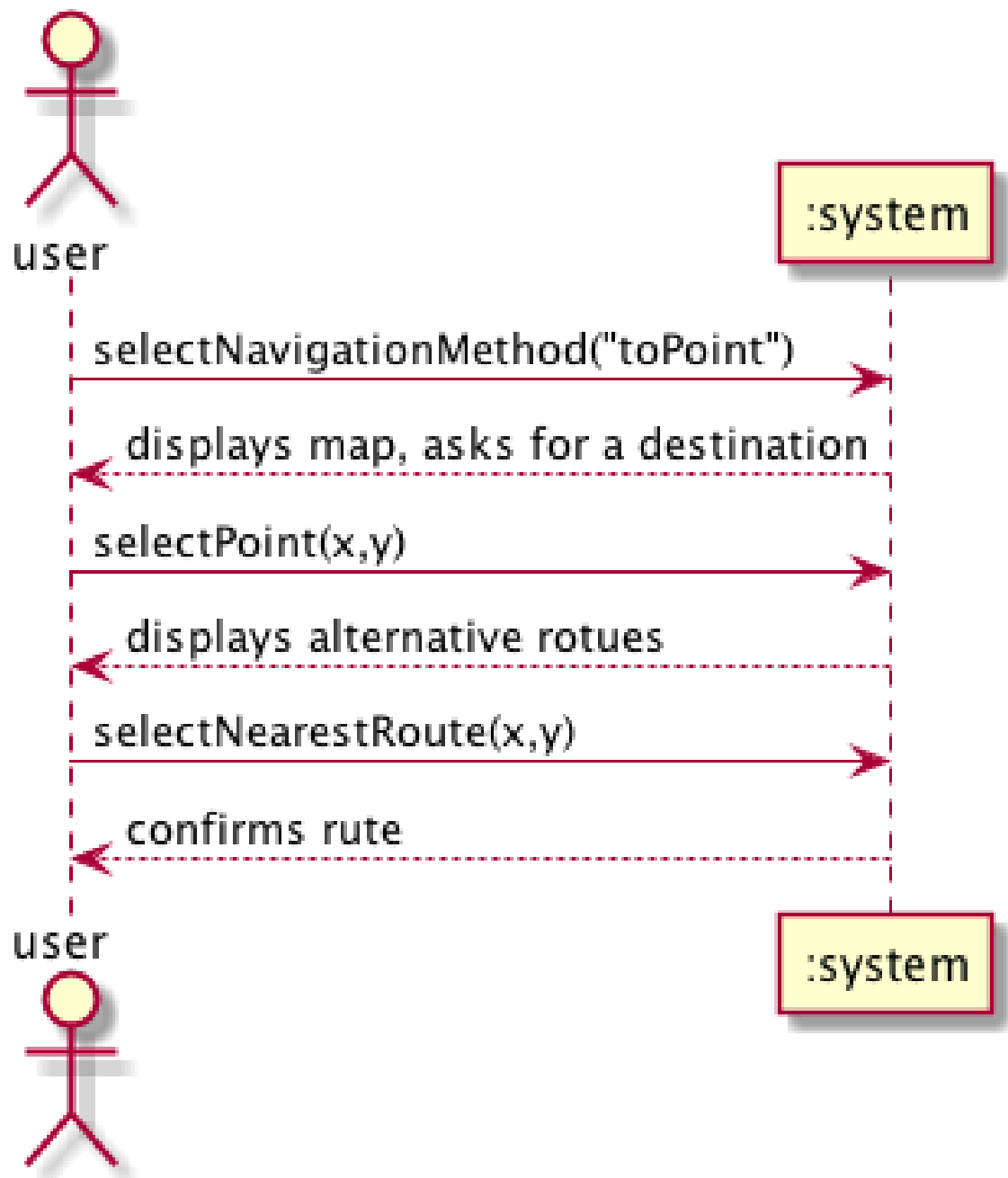
```

actor user
participant ":system" as sys

user -> sys : selectNavigationMethod("toPoint")
sys --> user : displays map, asks for a destination

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

user -> sys : selectNearestRoute(x,y)
sys --> user : confirms route
  
```



5 Sekvensdiagram - selectNavigationMethod

```

[-> ":system" : selectNavigationMethod("toPoint")
activate ":system"

create "currentNavigationStrategy:PointNavigation" as cns
":system" -> cns : create()
activate cns
cns -> ":GPS" : currentCoordinates = getCurrentCoordinates()
activate ":GPS"
deactivate ":GPS"

cns -> ":MapCollection" : getMap(currentCoordinates, zoomLevel)
activate ":MapCollection"
create "map:Map"
":MapCollection" -> "map:Map" : create()
":MapCollection" --> cns : map:Map
  
```

```

deactivate ":MapCollection"
deactivate cns

":system" -> cns : getResponseObject()
activate cns
create "response:ResponseObject"
cns -> "response:ResponseObject" : create()

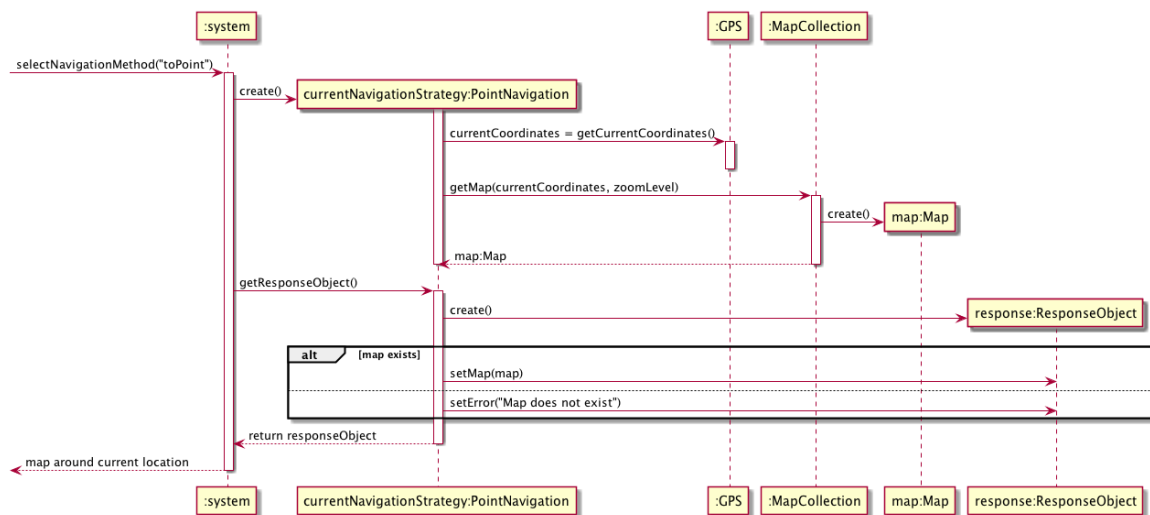
alt map exists
cns -> "response:ResponseObject" : setMap(map)
else
cns -> "response:ResponseObject" : setError("Map does not exist")
end

cns --> ":system" : return responseObject
deactivate cns

":system" -->[ : map around current location

deactivate ":system"

```



6 Klassdiagram

Såhär ser t.ex. metoden selectNavigationMethod i klassen system:

```

system::selectNavigationMethod(String theMethod) {
// ...
NavigationMethod* currentNavigationMethod = new PointNavigation(); // create()
// ...
responseObject* ro = currentNavigationMethod->getResponse();
// ...
return ro;
}

```

Klassdiagram med alla klasser, metoder, och relationer från sekvensdiagrammet för systemhändelsen selectNavigationMethod():

```

class system {
+selectNavigationMethod(theMethod)
}
class PointNavigation {
+create()
+getResponse()
}

```

```

- currentPosition:Position
- map : Map
}
class GPS {
+ getCurrentCoordinates()
}
class MapCollection {
+ getMap(coordinates, zoomLevel)
}
class Map {
+create()
}
class ResponseObject {
+setMap(theMap:Map)
+setError(string theError)
}

```

```

system - NavigationMethod
NavigationMethod <|-- PointNavigation
system -- ResponseObject
PointNavigation -- GPS
PointNavigation -- MapCollection
PointNavigation -- Map
PointNavigation -- ResponseObject
MapCollection - Map
ResponseObject - Map

```

