

# <2023-02-07 tis> Example: Robot System

Mikael Svahnberg\*

2023-02-01

## Contents

<b>1 Use Case: Navigate to Point</b>	<b>2</b>
<b>2 Interaction Diagrams</b>	<b>3</b>
<b>3 Class Diagram</b>	<b>4</b>

package UI

package Robot {

package GPSSensor

package MapManager

package Navigator

package RouteExecutor

package DriveControl

GPSSensor -- Navigator

MapManager -- Navigator

Navigator -- RouteExecutor

RouteExecutor -- DriveControl

package ArmControl

}

package Sensors

package Actuators

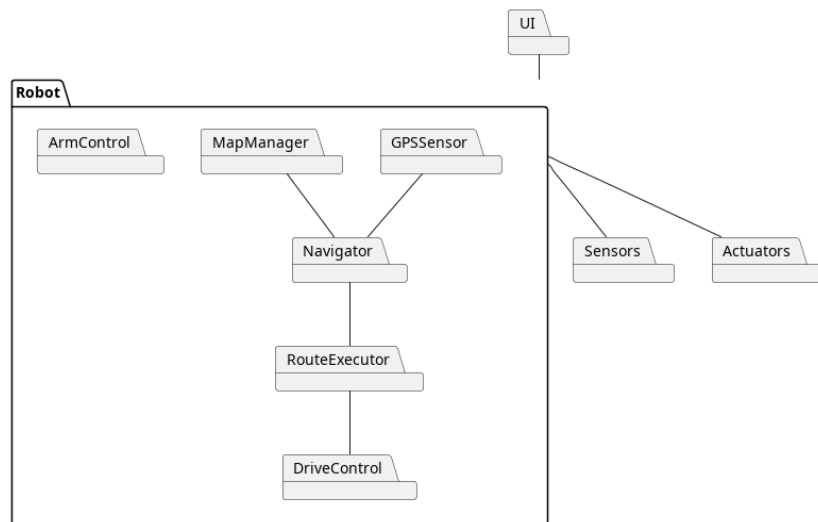
UI -- Robot

Robot -- Sensors

Robot -- Actuators

---

\*Mikael.Svahnberg@bth.se



## 1 Use Case: Navigate to Point

**Use Case** Navigate to Point

**Actors** Driver

**Description** Driver selects that they want to navigate to a point.  
System displays a map. Driver selects a point. System plots a route to point.

### Main Course of Events

Driver	System
1. Driver selects that they want to navigate to a point.	
	2. The system selects "Navigate to point" and pr
3. Driver selects a point on the map.	
	4. System plots a route to the point.
5. Driver confirms route	
	6. System lurches away with a smile on its face.

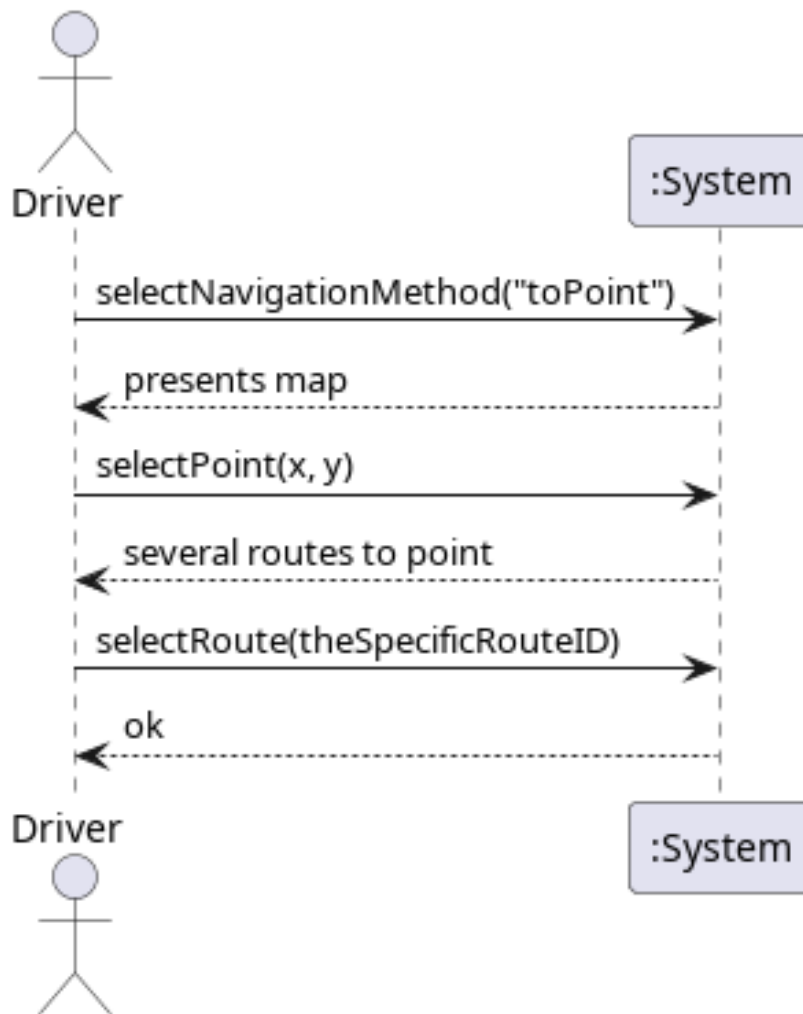
```

actor Driver as d
participant ":System" as sys

d -> sys : selectNavigationMethod("toPoint")
sys --> d : presents map

d -> sys : selectPoint(x, y)
sys --> d : several routes to point

d -> sys : selectRoute(theSpecificRouteID)
sys --> d : ok
  
```



## 2 Interaction Diagrams

```

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

```

```

":System" -> ":NavigationFactory" : getNavigationMethod("toPoint")
activate ":NavigationFactory"

```

```

":NavigationFactory" -> "currentNavigationStrategy:ToPointNavigationStrategy" : create()
":NavigationFactory" --> ":System" : currentNavigationStrategy
deactivate ":NavigationFactory"

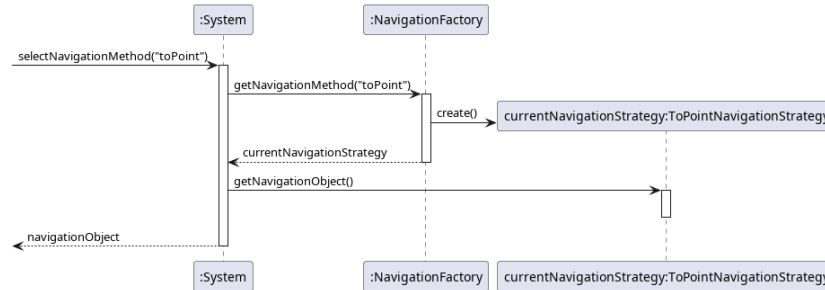
```

```

":System" -> "currentNavigationStrategy:ToPointNavigationStrategy" : getNavigationObject()
activate "currentNavigationStrategy:ToPointNavigationStrategy"
deactivate "currentNavigationStrategy:ToPointNavigationStrategy"

```

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

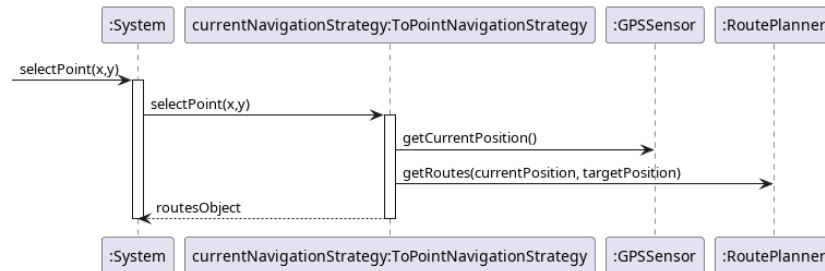


```
[-> ":System" : selectPoint(x,y)
activate ":System"
```

```
"":System" -> "currentNavigationStrategy:ToPointNavigationStrategy" : selectPoint(x,y)
activate "currentNavigationStrategy:ToPointNavigationStrategy"
"currentNavigationStrategy:ToPointNavigationStrategy" -> "GPSensor" : getCurrentPosition()
"currentNavigationStrategy:ToPointNavigationStrategy" -> "RoutePlanner" : getRoutes(currentPosition, targetPosition)
```

```
"currentNavigationStrategy:ToPointNavigationStrategy" --> ":System" : routesObject
deactivate "currentNavigationStrategy:ToPointNavigationStrategy"
```

```
deactivate ":System"
```



### 3 Class Diagram

```
package UI {
}

package Robot {

class System {
+selectNavigationMethod()
+selectPoint(x,y)
}

}
```

```

package NavigationStrategies {
class NavigationFactory {
+getNavigationStrategy()
}

abstract class NavigationStrategy {
+getNavigationObject()
+selectPoint()
}

NavigationStrategy <|-- ToPointNavigationStrategy
}

package GPSSensor {
class GPSSensor {
+getCurrentPosition()
}
}
package MapManager {
}

package Navigator {
class RoutePlanner {
+getRoutes(startPos, targetPos)
}
}

package RouteExecutor {
}
package DriveControl {
}

GPSSensor -- Navigator
MapManager -- Navigator

Navigator -- RouteExecutor
RouteExecutor -- DriveControl

package ArmControl {
}
}

package Sensors {
}
package Actuators {
}

UI -- Robot
Robot -- Sensors

```

Robot -- Actuators

System -- NavigationFactory

System -- NavigationStrategy

NavigationStrategy - GPSSensor

NavigationStrategy - RoutePlanner

