

Team: W9-5

sd.mycontroller.adapters.AdapterFactory.getAdapter(MapTile) - Communications

: TrapTileAdapter

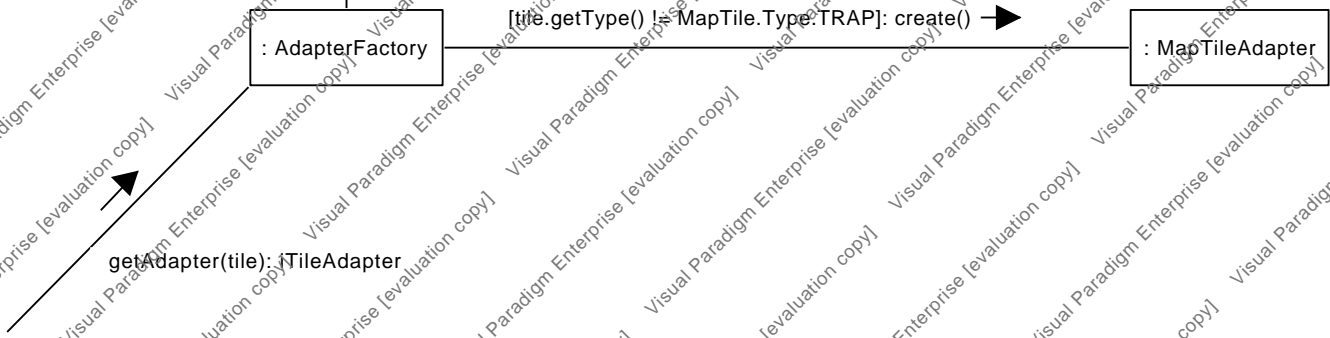
[tile.getType() == MapTile.Type.TRAP]: create()

: AdapterFactory

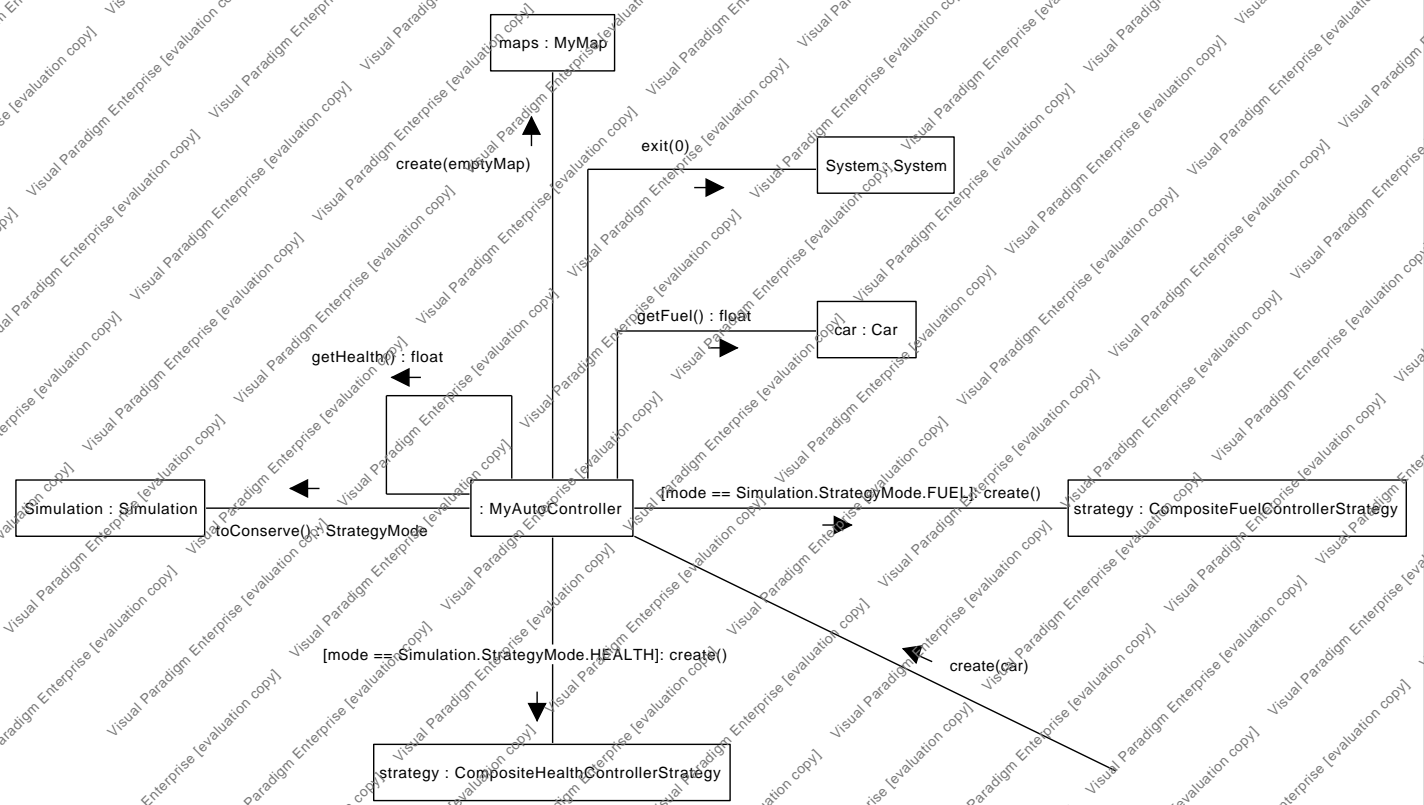
[tile.getType() != MapTile.Type.TRAP]: create() →

: MapTileAdapter

getAdapter(tile): TileAdapter



sd.mycontroller.MyAutoController.Car - Communications



sd.mycontroller.MyAutoController.update() - Communications

getView() : HashMap<Coordinate, MapTile>
[maps.getExit(),contains(currPos)]: applyBrake() : void
updateState(parcelPos) : void
checkWallAhead(currPos, map) : boolean
makeAction(start, des) : void
changeState(newState) : void

maps : MyMap

updateMap(view) : void
getParcel() : ArrayList<Coordinate>

update()

MyAutoController

create(position)

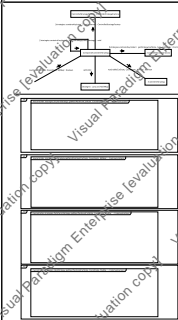
currPos : Coordinate

[currState != CarState.COLLECTING]: getNextPosition(state, curr, map) : Coordinate
[currState == CarState.COLLECTING && strategy.reachable(currState, currPos, maps)]: getNextPosition(state, curr, map) : Coordinate
[currState == CarState.COLLECTING && not strategy.reachable(currState, currPos, maps)]: getNextPosition(state, curr, map) : Coordinate

strategy : ControllerStrategy

ref

mycontroller.strategies.CompositeControllerStrategy.getNextPosition(CarState, Coordinate, MyMap) - Communications



sd.mycontroller.MyMap.MyMap(HashMap) - Communications

myMap : java.util.HashMap

```
for next in emptyMap.keySet()[adapter.getType(tile) == TileType.WALL]: put(next, new MapTile(MapTile.Type.WALL))
for next in emptyMap.keySet()[adapter.getType(tile) == TileType.ROAD]: put(next, new MapTile(MapTile.Type.ROAD))
for next in emptyMap.keySet()[adapter.getType(tile) == TileType.FINISH]: put(next, new MapTile(MapTile.Type.FINISH))
for next in emptyMap.keySet()[adapter.getType(tile) == TileType.START]: put(next, new MapTile(MapTile.Type.START))
```

factory : AdapterFactory

getAdapter(tile) iTileAdapter

: MyMap

get(next)

emptyMap : HashMap<Coordinate, MapTile>

adapter.getType(tile) == TileType.FINISH: add(next)

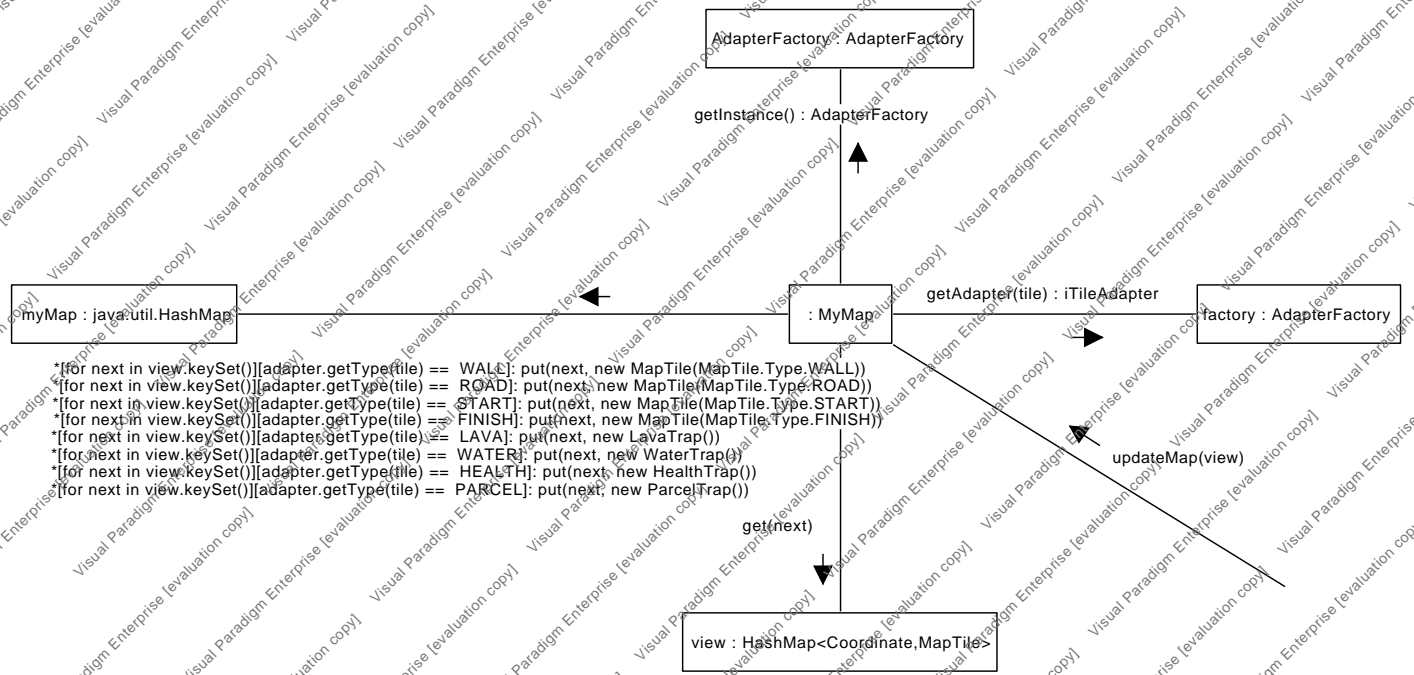
exits : java.util.ArrayList

getInstance() AdapterFactory

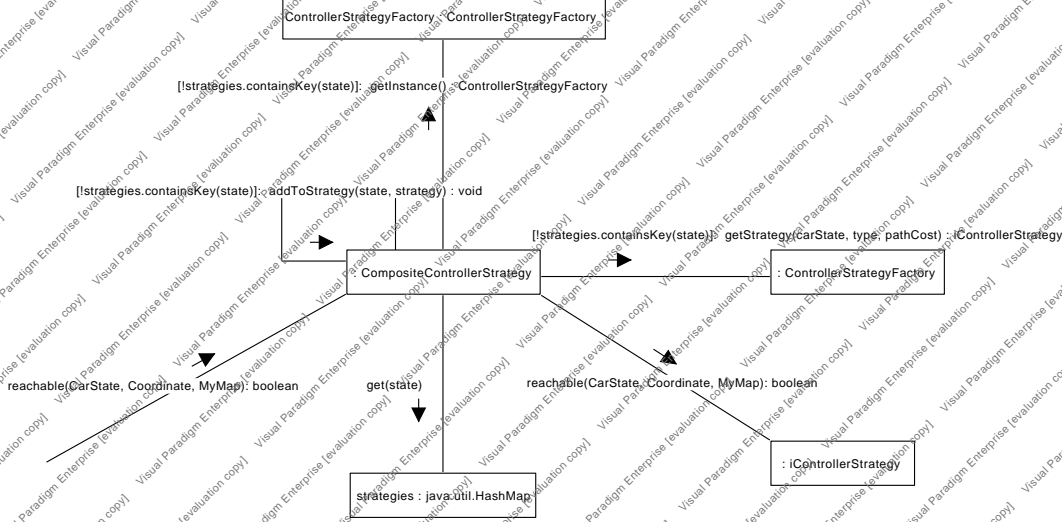
AdapterFactory : AdapterFactory

create(emptyMap)

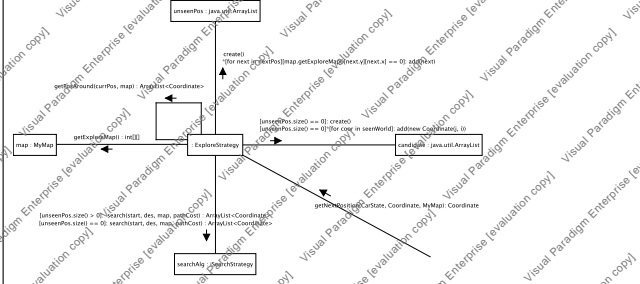
sd.mycontroller.MyMap.updateMap(HashMap) - Communications



sd.mycontroller.strategies.CompositeControllerStrategy.getNextPosition(CarState, Coordinate, MyMap) - Communications



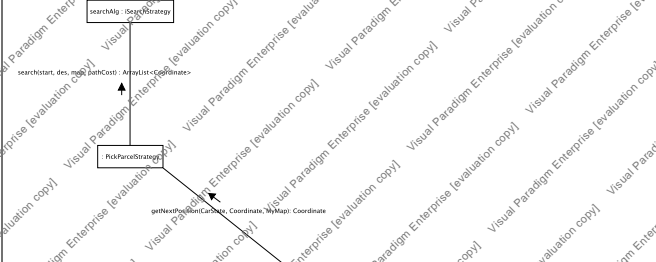
sd.mycontroller.strategies.ExploreStrategy.getNextPosition(CarState, Coordinate, MyMap) - Communications



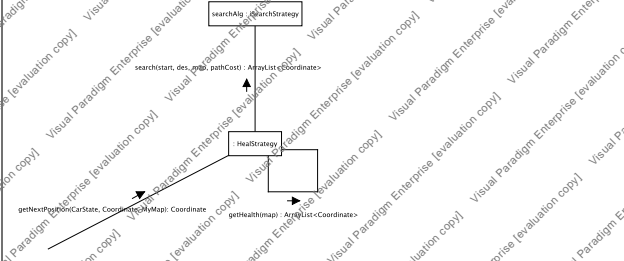
sd.mycontroller.strategies.ExitStrategy.getNextPosition(CarState, Coordinate, MyMap) - Communications



sd.mycontroller.strategies.PickParcelStrategy.getNextPosition(CarState, Coordinate, MyMap) - Communications



sd.mycontroller.strategies.HealStrategy.getNextPosition(CarState, Coordinate, MyMap) - Communications



sd.mycontrollerstrategies.CompositeControllerStrategy.reachable(CarState, Coordinate, MyMap) - Communications

ControllerStrategyFactory : ControllerStrategyFactory

[!strategies.containsKey(state)]: getInstance() : ControllerStrategyFactory

[!strategies.containsKey(state)]: addToStrategy(state, strategy) : void

CompositeControllerStrategy

[!strategies.containsKey(state)]: getStrategy(carState, type, pathCost) : iControllerStrategy

: ControllerStrategyFactory

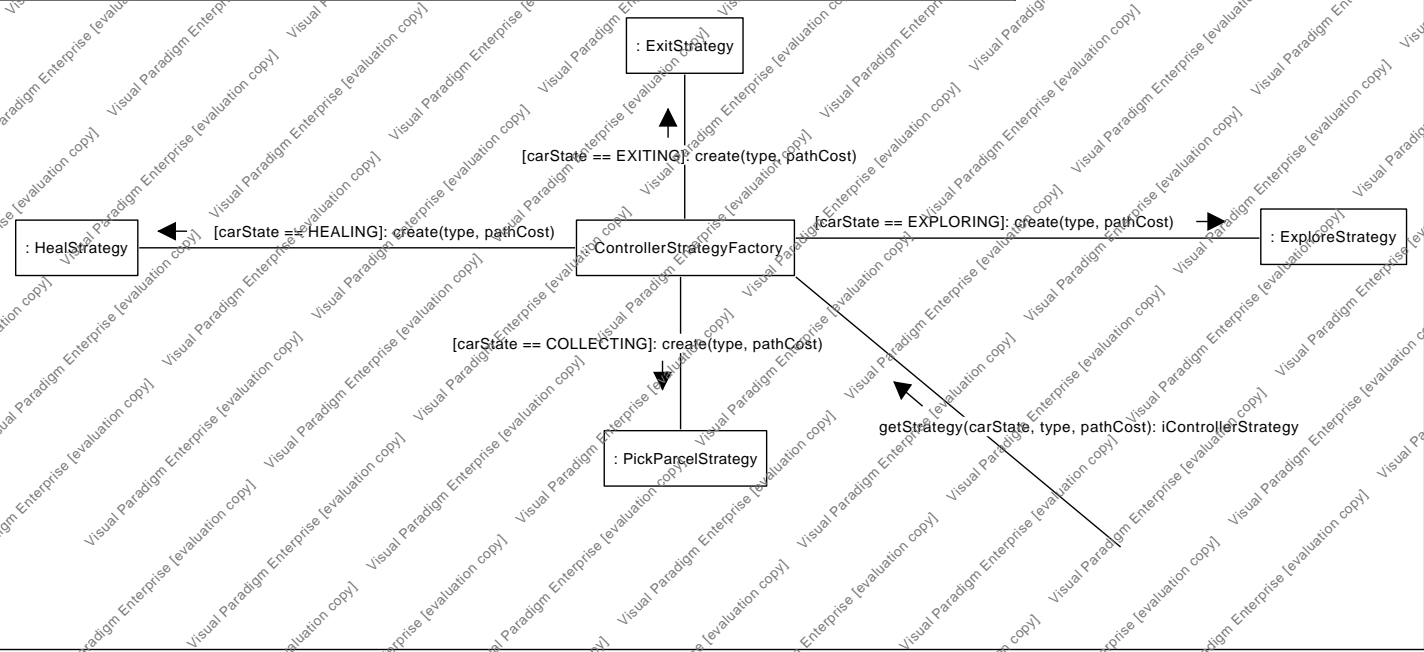
reachable(CarState, Coordinate, MyMap): boolean

get(state)

reachable(CarState, Coordinate, MyMap): boolean

strategies : java.util.HashMap

: iControllerStrategy



searchAlg: ISearchStrategy

search(start, des, map, pathCost) : ArrayList<Coordinate>



: ExitStrategy



getNextPosition(CarState, Coordinate, MyMap): Coordinate

sd mycontroller.strategies.ExploreStrategy.getNextPosition(CarState, Coordinate, MyMap) Communications

unseenPos : java.util.ArrayList

create()
[for next in nextPos][map.getExploreMap()[next.y][next.x] == 0]: add(next)

getPosAround(currPos, map) : ArrayList<Coordinate>

getExploreMap() : int[][]

[unseenPos.size() == 0]: create()
[unseenPos.size() == 0]*[for coord in seenWorld]: add(new Coordinate(i, j))

candidate : java.util.ArrayList

[unseenPos.size() > 0]: search(start, des, map, pathCost) : ArrayList<Coordinate>
[unseenPos.size() == 0]: search(start, des, map, pathCost) : ArrayList<Coordinate>

getNextPosition(CarState, Coordinate, MyMap): Coordinate

searchAlg : SearchStrategy

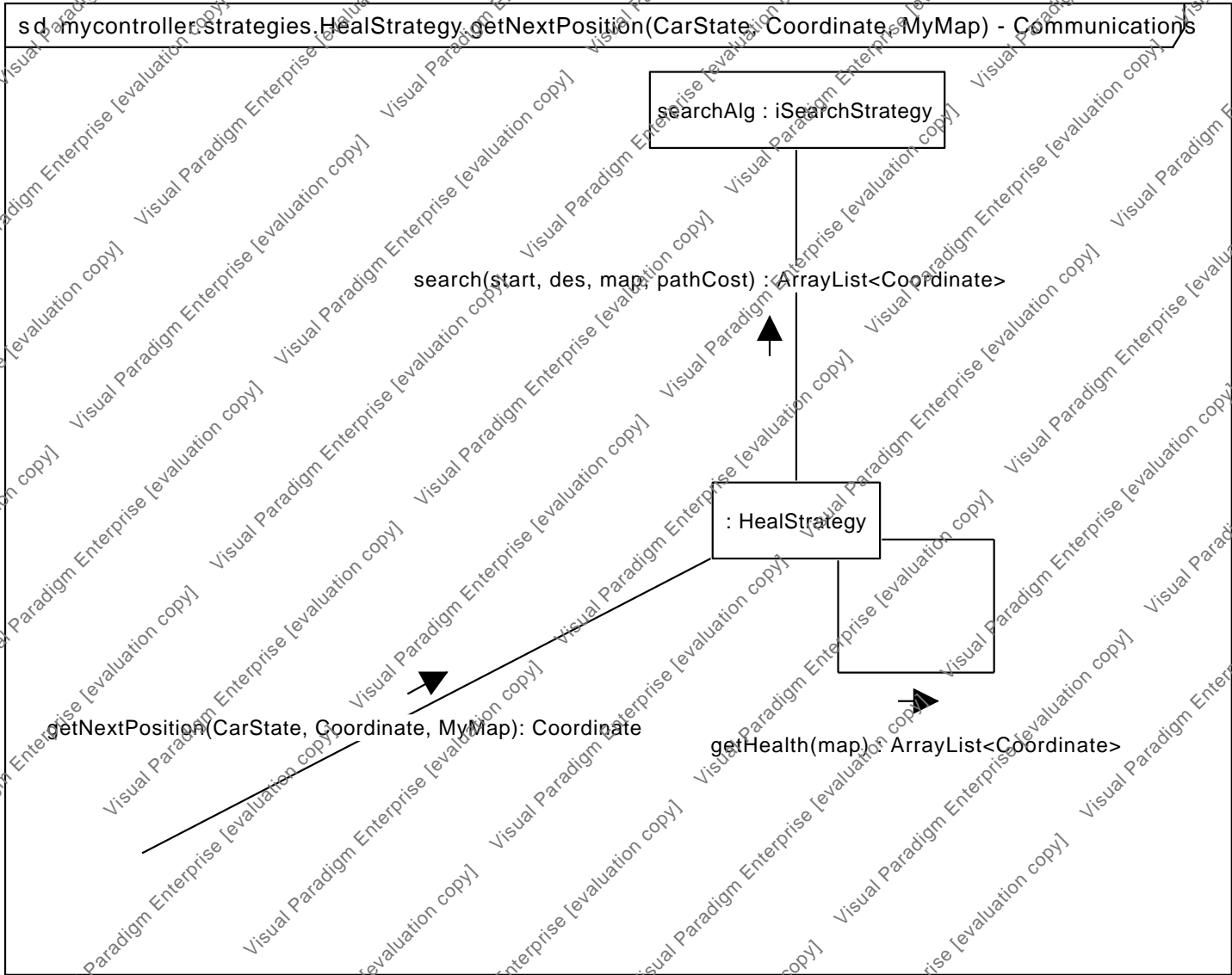
searchAlg : ISearchStrategy

search(start, des, map, pathCost) : ArrayList<Coordinate>

: HealStrategy

getNextPosition(CarState, Coordinate, MyMap): Coordinate

getHealth(map) : ArrayList<Coordinate>



searchAlg: iSearchStrategy

search(start, des, map, pathCost) : ArrayList<Coordinate>

: PickParcelStrategy

getNextPosition(CarState, Coordinate, MyMap): Coordinate



searchAlg : iSearchStrategy

search(start, dest, map, pathCost) : ArrayList<Coordinate>

: PickParcelStrategy

reachable(CarState, Coordinate, MyMap): boolean