From mailPool,
at every step loader is
created

:Loader

How the robots could be
initialized with different speeds,
depending on the weight of the
mail item

loadRobots(mailItems, robots)

loop

[for all mailItems]

chooseStrategy(mailItem)

<<interface>>
:ILoadStrategy

Loader creates
an appropriate
loading strategy

alt

[mailItem.weight
<
INDIVIDUAL_
MAX_
WEIGHT]

create

:SingleLoadStrategy

loadRobots(mailItem, robots)

ref

Load Single Robot

[else]

create

:TeamLoadStrategy

loadRobots(mailItem, robots)

ref

Load Team Robots

sd Load Single
Robot

:SingleLoadStrategy

:Robot

loadRobots(mailItem, robots)

alt

[handEmpty]

addToHand(mailItem)

setSingleMoving()

create

:SingleMoving

[!handEmpty]

addToTube(mailItem)

sd Load Team
Robots

:TeamLoadStrategy

robotsAvail[i]:Robot

loadRobots(mailItem, robots)

robotsAvail = availableRobots(robots)

speed = getStepPeriod()

alt

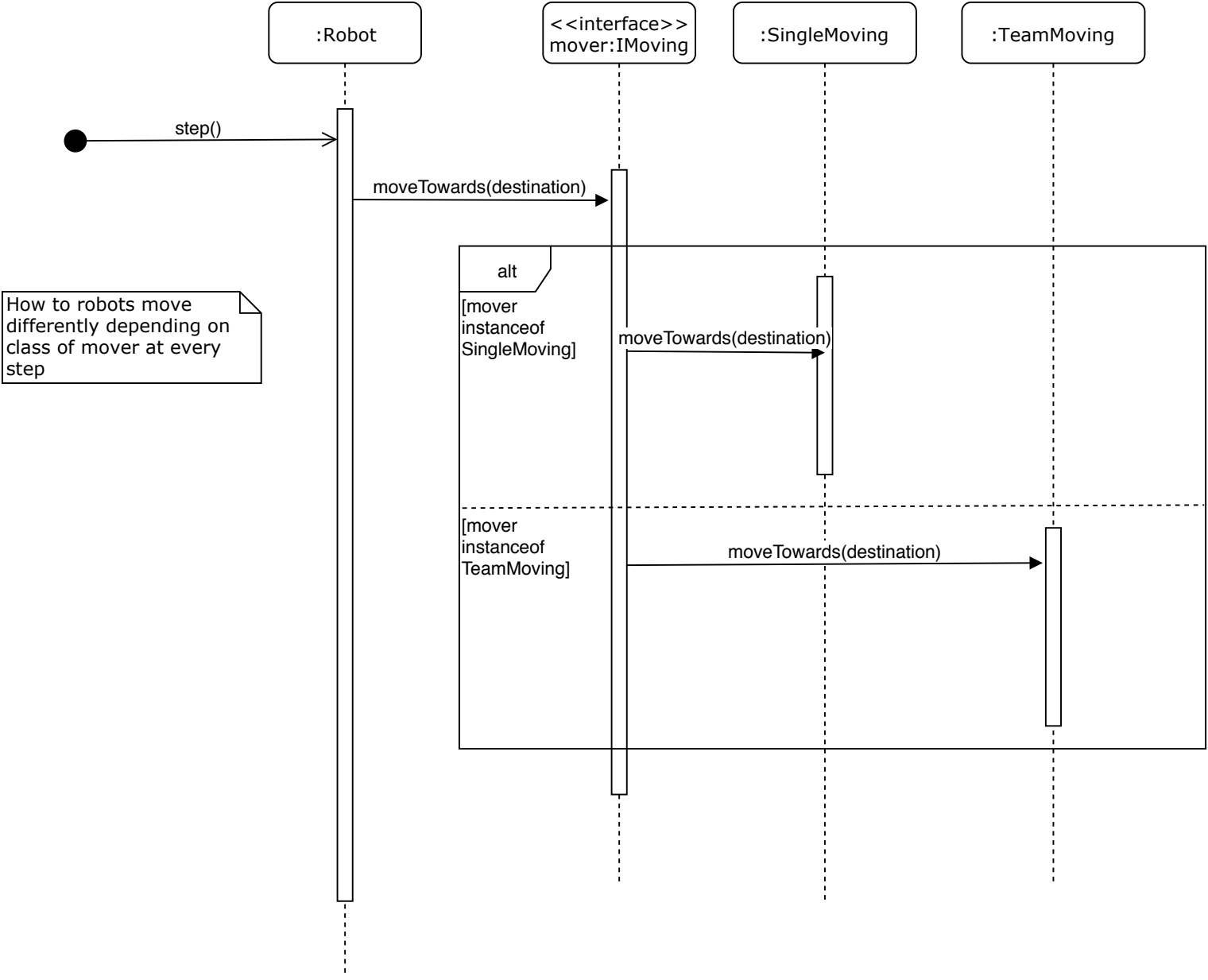
[enoughRobots]

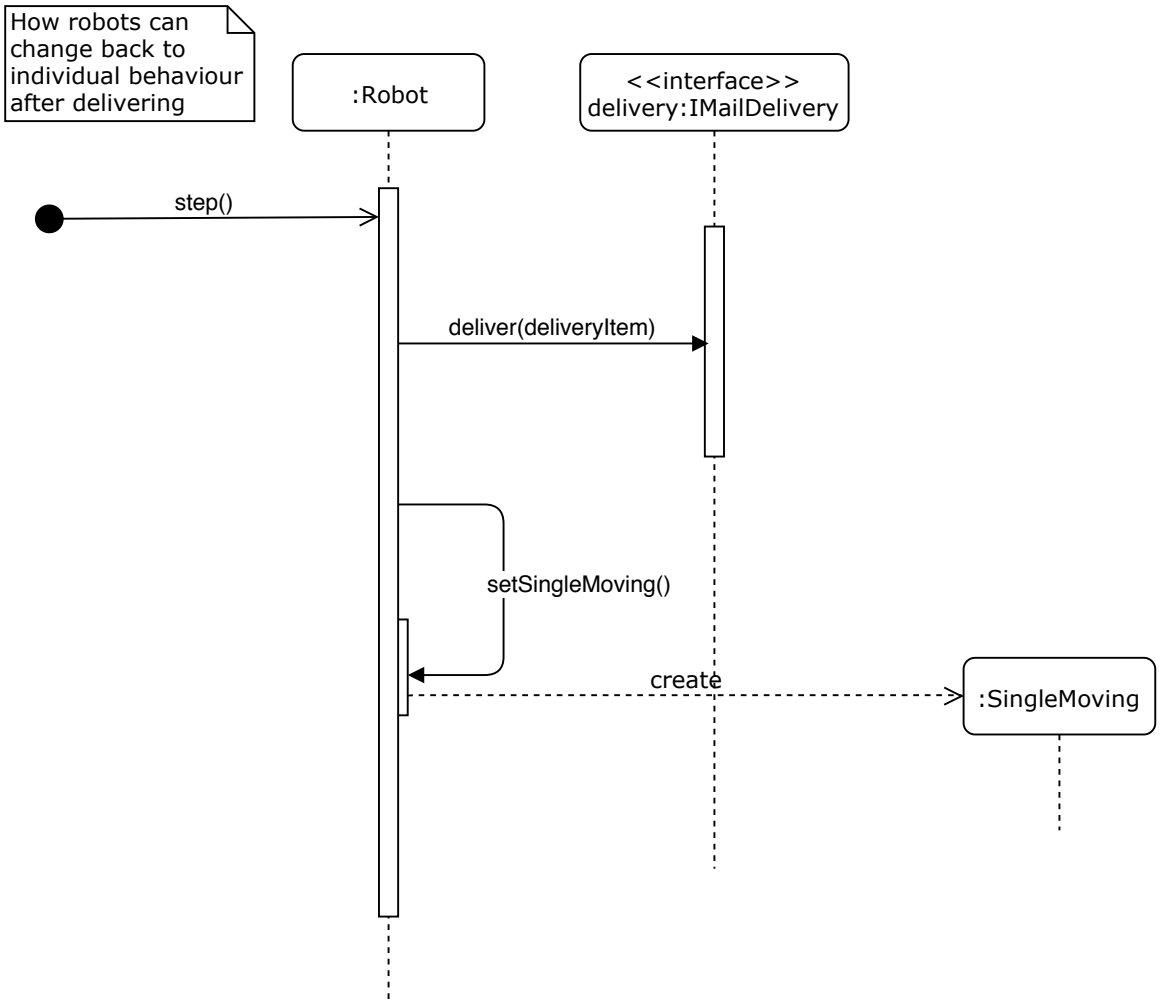
addToHand(mailItem)

setTeamMoving(speed)

create

:TeamMoving


:Robot
<<interface>>
mover:IMoving
:SingleMoving
:TeamMoving
step()
moveTowards(destination)
alt
[mover
instanceof
SingleMoving]
moveTowards(destination)
[mover
instanceof
TeamMoving]
moveTowards(destination)
How to robots move
differently depending on
class of mover at every
step


How robots can
change back to
individual behaviour
after delivering
:Robot
<<interface>>
delivery:IMailDelivery
step()
deliver(deliveryItem)
setSingleMoving()
create
:SingleMoving