**Diverter Agent Design**

**Data**

Feeder feederAgent

Lane laneAgent1 (top lane)

Lane laneAgent2 (bottom lane)

Lane laneAgent (current lane)

GUIDiverter guiDiverter

**public** **enum** DiverterDirection

{

*TOP*,

*BOTTOM*

}

DiverterDirection orientation;

**public** **enum** DiverterStatus

{

*orientation\_change\_requested*,

*changing\_orientation*,

*none*

};

DiverterStatus status;

MyPart currentPart;

Queue<MyPart> nextParts = **new** LinkedList<MyPart>();

**public** **enum** PartStatus

{

*not\_here\_yet*,

*not\_done*, *doing*, *done*,

*waiting\_to\_pass*, *okay\_to\_pass*;

};

**private** **class** MyPart

{

Part part;

PartStatus status;

**public** MyPart(Part part, PartStatus status)

{

**this**.part = part;

**this**.status = status;

}

}

**Messages**

msgSetDiverterOrientation(DiverterDirection direction)

{// sent from GUI panel

**this**.status = DiverterStatus.*changing\_orientation*;

**this**.orientation = direction;

**if** (direction == DiverterDirection.*TOP*)

{

laneAgent = laneAgent1;

stateChanged();

}

**else** **if** (direction == DiverterDirection.*BOTTOM*)

{

laneAgent = laneAgent2;

stateChanged();

}

}

msgGuiDoneChangingOrientation()

{// sent from GUILane

**this**.status = DiverterStatus.*none*;

stateChanged();

}

msgPartReady(Part part)

{// sent from Feeder

nextParts.add(**new** MyPart(part, PartStatus.*not\_here\_yet*));

stateChanged();

}

msgHereIsPart(Part part)

{// sent from Feeder

**if** (part.equals(nextParts.remove().part))

currentPart = **new** MyPart(part, PartStatus.*not\_done*);

**else**

print("v ERROR\n" +

"ERROR: Feeder gave me the wrong part!" +

"^ ERROR");

stateChanged();

}

msgGuiDoneMovingPart()

{// sent from GUILane

currentPart.status = PartStatus.*done*;

stateChanged();

}

msgReadyToReceive(Part part)

{// sent from Lane

currentPart.status = PartStatus.*okay\_to\_pass*;

stateChanged();

}

msgIAmFull()

{// sent from Lane

currentPart.status = PartStatus.*waiting\_to\_pass*;

stateChanged();

}

**Scheduler**

**if** (**this**.status == DiverterStatus.*orientation\_change\_requested*)

{

**this**.status = DiverterStatus.*changing\_orientation*;

changeDirection();

**return** **true**;

}

**if** (currentPart == **null**

&& nextParts.size() == 1

&& **this**.status == DiverterStatus.*none*)

{

prepareToReceive();

**return** **true**;

}

**if** (currentPart != **null**

&& nextParts.size() > 1)

{

tellFeederHoldOn();

**return** **true**;

}

**if** (currentPart != **null**)

{

**if** (currentPart.status == PartStatus.*not\_done*)

{

addPart(currentPart);

**return** **true**;

}

**if** (currentPart.status == PartStatus.*doing*)

{

movePart(currentPart);

**return** **true**;

}

**if** (currentPart.status == PartStatus.*done*)

{

notifyNextAgent();

**return** **true**;

}

**if** (currentPart.status == PartStatus.*okay\_to\_pass*)

{

passPart(currentPart);

**return** **true**;

}

}

**return** **false**;

**Actions**

prepareToReceive()

{

feederAgent.msgDiverterReadyToReceive(nextParts.peek().part);

}

tellFeederHoldOn()

{

feederAgent.msgHoldOn();

}

addPart(MyPart p)

{

p.status = PartStatus.*doing*;

guiDiverter.msgDoAddPart(p.part.guiPart);

}

movePart(MyPart p)

{

guiDiverter.msgDoMovePart(p.part.guiPart);

}

notifyNextAgent()

{

laneAgent.msgPartReady(currentPart.part);

}

passPart(MyPart p)

{

laneAgent.msgHereIsPart(p.part);

currentPart = **null**;

}

changeDirection()

{

guiDiverter.msgDoChangeOrientation(**this**.orientation);

}