Conveyor Template

- Data
  - List<Part> partsbeforepopup = new ArrayList<Part>();
  - Enum ConveyorStatus {moving, stopped, passingparttopopup}
  - Enum ConveyorEvents {respondtoquery}
  - List<ConveyorEvents> events
  - ConveyorStatus state;
  - Private string name;
  - Private final int MAX_PARTS_ALLOWED = 2
  - Machine nextmachine
  - Machine priormachine
  - Private final int conveyorindex
  - Private final int frontsensorindex
  - Private final int rearsensorindex
  - Boolean priormachineholding
  - Boolean loadintomachinefinished

- Messages
  - MsgMachineCannotAcceptPart() {
    state = ConveyorStatus.stopped
    fireStopConveyorEvent();
    stateChanged();
    }
  - MsgMachineCanAcceptPart() {
    state = conveyorStatus.moving
    fireStartConveyorEvent()
    stateChanged();
    }
  - MsgCanYouAcceptPart() {
    events.add(ConveyorEvents.respondtoquery)
    stateChanged();
    }
  - MsgHereIsNewPart(ConveyorFamily cf, Part part)
    parts.add(part);
    stateChanged();
    }

- Actions
  - RespondToPriorMachine() {
    if (parts,size() < MAX_PARTS_ALLOWED && state ==
    ConveyorStatus.moving) {
    priormachine.msgConveyorCanAcceptPart();
    }
    else {
    priormachine.msgConveyorCannotAcceptPart();
    priormachineholding = true;

```
    }
    stateChanged();
  }
```
- PassPartToMachine() {
```
    machine.msgHereIsNewPArt(this, parts.get(0));
    stateChanged();
  }
```
- AskMachineIfItCanTakePart() {
```
    machine.msgCanYouMachinePart()
  }
```
- FireStartConveyorEvent() {
```
    transducer.fireevent(Tchannel.CONVEYOR,
        Tevent.CONVEYOR_DO_START, temp);
  }
```
- FireStopConveyorEvent() {
```
    transducer.fireevent(Tchannel.CONVEYOR,
        Tevent.CONVEYOR_DO_STOP, temp);
  }
```

- Scheduler
  - If (events.size() > 0) {
```
    if (priormachineholding) {
      if (parts.size() < MAX_)PARTS_ALLOWED && state ==
            ConveyorStatus.moving) {
        priormachine.msgConveyorCanAcceptPArt()
        priormachineholding = false;
      }
    }
    if (events.size() > 0) {
      if (events.get(0) == conveyorEvents.respondtoquery) {
        events.remove(0)
        respondToPriorMachine()
      }
    }
  }
```

Machine Template

- Data
  - Private ConveyorAgent priorconveyor;
  - Private String name;
  - Private Part[] crrentpart;
  - Private enum MachineEvent {respondtoconveyor, conveyoronhold, donewithpart, notdonewithpart, passtonextfamily}
  - Private List<MachineEvent> events
  - Private ConveyorAgent nextconveyor;
  - Private final int machineindex = 2
  - Private Boolean priorconveyorholding

- Messages
  - MsgConveyorCanAcceptPart() {
    events.add(MachineEvent.passtonextfamily)
    stateChanged();
    }
  - MsgHereIsNewPart(ConveyorFamily cf, Part part) {
    currentpart[0] = part
    stateChanged();
    }
  - MsgCanYouMachinePart() {
    events.add(MachineEvent.respondtoconveyor)
    stateChanged();
    }
  - MsgNextFamilyCanAccept() {
    events.add(MachineEvent.passtonextfamily);
    stateChanged();
    }
- Actions
  - RespondToConveyorPartPassQuery() {
    if (currentpart[0] == null) {
    priorconveyor.msgMachineCanAcceptPart();
    }
    else {
    priorconveyor.msgMachineCannotAcceptPart();
    priorconveyorholding = true
    }
    stateChanged();
    }
  - ExecuteMachineAction() {
    if (currentpart[0].getRecipe().charAt(machineindex) == '1') {
    transducer.fireEvent(Tchannel.MANUAL_BREAKOUT,
    Tevent.WORKSTATION_DO_ACTION, null)
    }
    else {
    nextconveyor.msgCanYouAcceptPart();
    }
    stateChanged();
    }
  - PassPartToNextFamily(Part p) {
    transducer.fireEvent(Tchannel.MANUAL_BREAKOUT,
    Tevent.WORKSTATION_RELEASE_GLASS, null)
    }
- Scheduler
  - If (events.size > 0) {
    if (events.get(0) == MachineEvent.respondtoconveyor ||
    events.get(0) == MachineEvent.conveyoronhold) {

```
    events.remove(0)
    respondToConveyorPartPassQuery();
  }
  else if (events.get(0) == MachineEvent.passtonextfamily) {
    events.remove(0);
    passPartToNextFamily(currentpart[0])l
  }
  else if (events.get(0) == MachineEvent.notdonewithpart) {
    events.remove(0)
    executeMachineAction();
  }
}
```