Florjan Praprotnik 26.5.2014

# Logic of the original model

Each station has an **exit strategy**: LastStation = .?

**PartInfeed 🡪PartBuffer** (exact: Part Disasembly) (possible MUs: Tray)

* Exit Strategy: LastStation = .?  
   Wait till PartBuffer (compleete) is empty.
* RobotTarget : If LastStation = PartInfeed *then* Target = PartBuffer1… 10

**PartDisasembly 🡪 CNC machines** (possible Mus: Parts,)

* Exit Strategy: LastStation = .?

if .? = PartDisasmelby 1 🡪 waitUntil CNN1 available

Part.PAth = 1;

If .? = PartDisasembly2 🡪 waitUntil CNN1 available

Part.Path = 2;

etc.

* RobotTarget: if LastStation = PartDisasembly1 then Target = CNN1  
   elseif LastStatione = PartDisasembly2 then .. Target = CNN2 etc.

**CNC machines 🡪 Deburring** (robot holding part) [hint: wait for part] (possible Mus: Parts, **Tool**)  
 🡪 ToolBuffer

* Exit Strategy:   
   LastStation = .?  
   **if part**  
   waitUntil Debrurring 1 OR Deburring 2 is available  
   **elseif tool**

**Before a part goes to deburring, Gauging must also be empty (Since the robots tool is ocupied)**

* RobotTarget: if LastStation = CNC1 or CNC2 or.. CNC 10.. then  
   **if part**  
   if Deburring1.empty then Target = Deburring1  
   elseif Deburring2.empty then Target=Deburring2  
   **elseif tool** Target = any tool buffer

**Somehow wait for part!**

**Deburring 🡪 Gauging** (possible Mus: Parts)

* Exit Strategy: LastStation = .?  
   waitUntil Gauging1 or Gaugin2 is available.
* RobotTarget: if LastStation = Deburring1 or Deburring2 then  
   if Cleaning1.empty then Target = Cleaning 1  
   elseif Target=Cleaning2

**Gauging 🡪 Lasering (OP2) and Gauging 🡪 CMM (OP1)** (Possilbe MUs: Parts)

* ExitStrategy: LastStation = .?

if .? = Gauging then  
 if part.Path = 1 or 2 then waituntil CMM1.empty  
 elseif part.path = 3 or 4 then waituntil CMM2.empty

else

waituntil Lasergin1 or Lasergin is empty.

* Robot Target If LastStation = Gauging then  
   if part.OP = 1 then  
   if PAth = 1 or Path = 2 then Target = CMM1  
   elseif Path = 2 or Path = 3 then Target = CMM2  
   elseif part.OP = 2 then

If Lasering1.empty then Target = Lasering1  
elseif Lasering2.empty then Target = Lasering2

**Lasering 🡪 CMM (op2 only anyway)** (Possible MUs: Parts)

* ExitStrategy: LastStation = .?  
   if .? = Lasering then  
   if PAth = 5 or Path = 6 then waituntil CMM3.empty  
   elseif Part.Path = 7 then waituntil = CMM4.empty

elseif Part.Path = 8 then waituntil = CMM5.empty

elseif Part.Path = 9 or Part. Path = 10 then waituntil = CMM6.empty

* RobotTarget: if LastStation = Lasering then  
   if PAth = 5 or Path = 6 then target = CMM 3  
   elseif Part.Path = 7 then Target = CMM 4

elseif Part.Path = 8 then Target = CMM 5

elseif Part.Path = 9 or Part. Path = 10 then Target = CMM 6

**CMMs 🡪 PartAssembly (partBuffer)** (Possible MUs: Parts)

* ExitStrategy: LastStation = .?
* RobotTarget: if LastStatione = CMM1 or CMM2 or … CMM6 then

If part.path = 1 then Target = PartAssembly 1

Elseif part.path = 2 then Target = PartAssembly 2  
 etc… till 10.

**PartAssembly 🡪 PartOutfeed:** (Possible MUs: Tray)

* RobotTarget: If LastStation = PartBuffer 1 or… till 10 then Target = Part.offoad.
* Part Attributes
  + LastStation, object
  + Path, integer
  + OP, integer
  + Gripper, string
  + MUType, string
  + Robot
* Tray Attributes
  + LastStation, object
  + Gripper, string
  + MUType, string
  + Robot
* Tool Attributes
  + LastStation, object
  + ~~TimesInCNCCounter, int~~
  + Gripper, string
  + MUType, string
  + Robot
* Other attributes
  + CNCCounter\_1
  + CNCCounter\_2
  + CNCCounter\_3
  + CNCCounter\_4
  + CNCCounter\_5
  + CNCCounter\_6
  + CNCCounter\_7
  + CNCCounter\_8
  + CNCCounter\_9
  + CNCCounter\_10

# Predefined paths for parts

* When a tray is put in a part buffer, it is allready known which CNC and CMM the parts will take. The attribute Path on the parts tell it.
* Compatible devices: OP1 on CNC\_OP1 only, OP2 on CNC\_OP2 only.
* There are 4 CNC-OP1 and 6 CNC-OP2 devices.
* OP1 has 4 part buffers, 4 CNC machines and 2 CMMs on disposal.
* OP2 has 6 part buffers., 6 CNC machines and 4 CMMs on disposal.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PartBuffer | CNC | CMM | OP | Path | Robot |
| 1 | 1 | 1 | 1 | **1** | **1** |
| 2 | 2 | 1 | 1 | **2** | **1** |
| 3 | 3 | 2 | 1 | **3** | **1** |
| 4 | 4 | 2 | 1 | **4** | **1** |
| 5 | 5 | 3 | 2 | **5** | **1** |
| 6 | 6 | ~~3~~  4 | 2 | **6** | **2** |
| 7 | 7 | 4 | 2 | **7** | **2** |
| 8 | 8 | 5 | 2 | **8** | **2** |
| 9 | 9 | 6 | 2 | **9** | **2** |
| 10 | 10 | 6 | 2 | **10** | **2** |

# Other Implementations:

* **Every 20 parts, 3 tools must be changed**
  + ~~Solution: Tools willl have an attribute TimesInCNCCounter. 3 tools change every 20 parts is the same as ALL tools change every 66 parts.~~
  + There will be a table of counters for CNC1, CNC2.. CNC6. When each of those reaches 66, the parts must be set to
  + Exit strategy for Tool:   
     if ?.= CNC1 , waituntil CNCCounter1 = 66

Elseif ?= CNC2, waituntil CNCCounter2 = 66

* GripperChanges: Robot has 3 possible grippers:
  + **Every station must have gripper defined as an attribute.**
  + Robot Exit Strategy: **if** predecessor.gripper /= target.gripper then wait 5 seconds
  + If this part type is different than previous, wait 5 seconds.
* **Pulling:** 
  + Somehow allways prioritize Tools before Parts or Trays. Others will do fine.
  + Block all, go through loop and unblock Tools.
    - If no tools found, unblock All.
* **Batches: one batch consists of 1-5 carriers.**
  + Does parts in trays in one batch all have the same size?
  + Ignored
  + Number the Trays. Every 5th tray is a batch. Tray 1 and tray 5 are checked
  + Maybe also parts would need to be numbered for this.
* Full/reduced inspection on CMM for first and last batch.
  + Batches are not yet defined.
  + Implementing every 50th part.
  + Parts must be numbered then?
  + Parts must have a Batch attribute for this.
* Pick, place, pic kand place takes 10-10-15 seconds.
* Batches:
  + Not enough parts to implement. Idea is the following.
    - I assume one batch has either OP1 or OP2 and not mixed, otherwise..??
    - There are 6 machines operating OP2 parts, which means 6 batches can be working at once (since each batch has its own CNC machine)
    - That means that the robot must have access to containers that come from 6 different batches – which means 6 stations.
    - Either this, or a system should be made that would do the following
    - When the first part of the first container of the first batch comes to the root, robot puts it on the next available PartBuffer (unknown yet)
    - At this point, the PATH is decided.
    - The Robot must now also have acces to 9 different batches, since one batch must be complete on each CNC machine before taking another one.
      * Batched must be numbered
      * Parts inside a container must be numbered
      * Container must be numbered
* The CNC hold 10 pars. **Every 20 trays, 3 must be replaced**. The problem with modeling this is that The tool buffer can not get another crate of tools before it puts theese away. This means that each CNC machine would have to use 2 buffers or each buffer hold 2 trays (one wilt unused tool and the other with used tool).

# Double infeed tool, infeed part. Why?

Robots sharing the machine makes the "waitUntil" strategy useless. They both pick from the same source therefore Deadlock occurs. Solving the deadlock is not an easy task, therefore the Infeed is split on 2.

# Bugs:

* @.destination = CNC1[1,1] returns an error where as setDestionation(CNC1[1,1],false) does not.
* When the Robot1 Picks up the tray, I tried to set the "Robot = 1" as an atrribute, so that based on this, the Robot will know the target. However, The software apparently executes the Entry Method after the Target Method.
* Allthough the X and Y dimenstion of parallelproc is set to 11.1, SetDestination to [11.1] or similar does not work – does not put the MU on the station. It does however put the [1,1], [2,1], [1,2] and [2,2]. Workaround was a separated proc for CNC.
* Detaching Stations from the PickAndPlace. They still stay on the table and "Teleport" to the destination.
* Not enough parts to model that  
    
  Since it is thoughbput we are after, 3 tool changes every 20 parts is equivalent in CNC-waiting time

\* Pull strategy and Wait at target. Even if I Unblock all the parts with the pull strategy, waitAtTarget does not unblock it.

**Monday, 26. May 2014 – Concept changed to:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PartBuffer | CNC | CMM | OP | Path | Robot |
| 1 | 1 | 1 | 2 | **1** | **1** |
| 2 | 2 | 1 | 2 | **2** | **1** |
| 3 | 3 | 2 | 2 | **3** | **1** |
| 4 | 4 | 3 | 2 | **4** | **2** |
| 5 | 5 | 4 | 2 | **5** | **2** |
| 6 | 6 | 4 | 2 | **6** | **2** |

**28.5.2014 – After the Second meeting.**

CNCTool and CNC have been merget.

**Problem:**

* Every 20 parts, 3 tools are changed. -- relates to:
* Getting Tools in the CNN and keep them there until 66 parts are changed

Solution:

* If I know how many parts are there in CNN and which kinds, I can have a counter nad make Tools waitUnril PartCounter = 66.
* Implement counters: - is another problem. Why?
  + NumMU tells only total number of parts, but not which ones.
  + Waituntil can not be used with the stations inside a proc.
  + Trying to solve this with a simple counter. When MU type part enters the station, +1, when it leaves the station, -1. Problem
  + Problem because ExitControl is not called whenever an MU exits, but when it tries to exit. So it is called several times **regardless of waitUntill** command.
  + Entry control however is executed everytime the part enters, not when it tries to enter. Presumably because the code is set to enter only if the station is available i.e. the robot can not be stuck with the part in front of the full station.
  + Solution. Inseda of putting -1 counter on the exit CNN control, put it on the entry control of the next station.
  + **Works for parts!** But problem – does not work for tool
  + **For reasons unknown**, Assembly station does not execute entry command, when the tool enters it. It executes it only when the main unit, MU enters it.
  + **Solution**, do not use the -1 for the tools, instead know the number of tools with the function NumMU – NumberOfParts.
  + This works, but is ugly.

Still need approval if changing all (10) tools every 66 parts is equal enough to chaning 3 tools every 20 stations.

* **So, to get the number of parts, I need:**
  + Entrance control of Deburring
  + Entrance control of CNN
* To get the number of tools I need.
  + Entrance control of CNN
  + Entrance control of the AssemblyTool Station – however no increments (absolute numbers only!).

**Problem: destination(@target, TRUE) –** wait at station is blocked when Pulling mechanism is used.

Problem: **Batches**:

Parts in a Tray in a Batch. All parts from the same batch should be used on the same CNN machined.

Walkthrough.

* First batch goes in the infeed.
* The CNN1 and CMM3 was decided by the first part
* No other CNN or CMM machine will be used for the parts on this batch
* If Robot does not have acces to another batch, that can use a different CNN machine, all other CNN machines will wait.  
    
  **Solution for this:**
* Robot must have acces to different batches.
* How many? As many as there are CNN machines so they can work silmutaiounsly.
* Problem – Part limit.

~~29.5.2014 – after meeting~~

~~Implementing Double gripper:~~

~~Double gripper implemented, however since the CNC was forced to have Tools AND Parts in the station, CNC1.ready (function which informs if any parts from the station are ready to exit), becomes useless, since tools are always waiting to get out. NOT.~~

~~Go to CNC1 only if the part is ready to leave. But don’t go emptyhanded~~

~~So go to CNC1 only if the part is ready to leave, but go to the PartBuffer first.~~

~~Now reverse logic is, exit strategy for PartBuffer (when will the part call the robot):~~

~~First part in the part buffer. Main difference: CNC is empty. I am trying no to get the part blocked, so waitUntil is the only way for this.~~

~~waitUntil (CNC is empty and toolchange = FALSE)~~

~~or (CNC1 has 1 part, Deburring station is empty, and CNC previous part wants to leave )~~

~~The CNC.ready is not legal to put in the waituntil statement. Again everything needs to be restructured.~~

**~~Second strategy:~~**

~~Try to move the part only if there is no tool change and there either is 0 or 1 part in the CNC~~

**The hard nut of the Double gripper: Workaround implementation:**

Part must not leave the part buffer to CNC station, until either CNC has 0 parts or the part in the CNC is ready to leave.

* Part in PartBuffer will ask to leave if number of parts is less than 2 (so 1 or 2).
* When the part in PartBuffer askes to leave, the robot can be anywhere.
* The part must be allowed (pulled) to enter the robot only if the part in the CNC is ready to leave or the number of parts in the CNC is 0.
  + If parts in CNC is 0 – can be unblocked anytime, CNC is waiting.
  + If Parts is 1 – only part that tries to leave the CNC is allowed to unblock the part in the Part buffer.  
    If the part in the part buffer calles the Pulling method and number of parts is 1, then unblock the Part from PartBuffer.
* **Logic not suffice**: why?
  + Part can leave the CNC station only if the Deburring station is empty.   
    At the deburring station, the robot must hold the part, which means that:
  + Nothing can enter the Deburring staton, until the successor of the deburring station (Gauging is empty).
* **In other words.** A part can not leave the part buffer until the part in CNC1 is ready to leave and the Deburring and Gauging stations are both empty.  
  It must somehow be prioritized that:
  + The Gaugin station must be served first
  + Then the cleaning station
  + Then the CNC

This implemented. Now the next hard nut: Tools want to leave, however…

PartBufDIS1 is unlocked either with:

- himself, if CNN is empty and he is full

- With part in CNN, if he determins to go out and all the following stations are available.

What if he is not full and CNN is empty (in between tool change, tray Change).

**After the tool change**

- The Robot is stuck in front of the CNC because there is no one at the CNC to unblock the PartBuffer. However, when the part tray is changed, there is also no one to call the PartBuffer. In that case, the numbe of CNC = 0 and number of parts in Part buffer is 10, the part buffer unlocks himself.

The double gripper now resolved for 1 CNC station with static route.