RWA - 1

Building a Manufacturing Robot Software System

March 14, 2022

Students:
Sri Sai Charan Velisetti

Vaishanth Ramaraj

Joseph Pranadeer Reddy Katakam

Pranav Vinay Limbekar

Bharadwaj Chukkala

Instructors:
Z. Kootbally

Group:
Group 2

Semester:
Spring 2022

Course code:
ENPM663

Contents

List of Figures

1 Introduction

- The Challenges faced by the ARIAC environment are handled in this Real World Application Project 2.
- Previously created skeleton structure of the ARIAC challenge package has been modified in accordance with the Challenges being handled.
- The previously created Class and Sequence diagrams have been updated accordingly.

2 Modifications done to Class and Sequence Diagrams

- For the modifications of the class and sequence diagrams, We have added classes for each of the agility challenges and removed some redundant classes to make the whole process a little bit easier to follow through. The classes we defined for agility challenges include member functions and attributes which differentiate the challenges based on the rule that, there is a possibility of all challenges happening at kitting stations and only some happening at assembly stations.
- Steps of sequence diagram modification:
  1. We added the agility challenges to our existing sequence diagram. To do this, we understood that there is much more functionality to the picture.
  2. Firstly when the order is read, our strategy is to check for agility challenge triggers one by one.
  3. We start with insufficient parts challenge first and see if all the parts needed for kitting and assembly exist by checking both the bins and the conveyor. Our next step is to check for triggers for the part flip challenge and perform necessary actions of flipping the part.
  4. Now that we have moved ahead in the pipeline, we check to see for triggers of the faulty part challenge, we used the data from the quality control sensors to trigger this challenge.
  5. The sensor blackout can be triggered in many ways and it cannot be a part of the sequence as such. But the challenge will be tackled accordingly when it gets triggered and the process of kitting and assembly keep going, irrespective of sensors messages.
  6. In-process order change is not an agility challenge where, it is triggered based on the announcement conditions in the trial file. It is not sequential but a possible anomaly, so it is placed above the sequence such that, the challenge will be completed and simultaneously the competition is run.

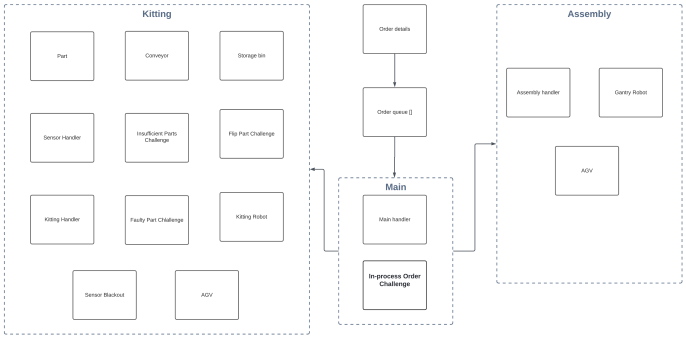

Figure 1: Updated Architecture Diagram

3 Approach to Handle ARIAC Challenges

- Description of the challenges and the approach taken to handle them have been mentioned below.

1. **In-process Order Challenge**:

- **Problem**: This challenge consists of introducing a new order $order1$ while the robots are already working on $order0$. A new order has a priority: If the priority is 3 then this is a higher-priority order. If the priority is 1 then this is a regular order. Order announcements consist of an announcement condition and an announcement value.

- **Approach**: For this we created a node that has functions that store the incomplete and the complete orders. After this a method called process orders will use the data in the attributes to check priority of the new incoming order and based on the announcement condition, the challenge will be triggered. Once the order is triggered, the old order is dropped and the new order is pushed int the priority completion.

2. **Sensor Blackout Challenge**:

- **Problem**: Sensor blackout challenge is where for a finite period of time, all sensors in the factory stop working as to mimic a real time sensor blackout. This challenge is triggered by a condition which is passed in the trial files and the duration it lasts for also is specified in the trial files. basically this information is not conveyed to you. the competitor will know when there is a sensor blackout when cameras and sensors stop reporting the required messages. The competition good practice is such that the robot has to have an internal idea of what next steps it has to take.

- **Approach**: Made according changes in the trial files such that, a sensor blackout is triggered due to a given condition. We created a node such that, when the blackout is triggered and the sensors stop reporting messages on the screen, it is understood that, the challenge has been triggered and the node prints out "Sensor blackout".

3. **Faulty Part Challenge**:

- **Problem**: In this challenge we have to detect the faulty parts that has been spawned on AGVs at the kitting stations. If the faulty parts have been detected on any AGV, the Kitting robot will pick the part and trash it, then replace the part with a similar type. .

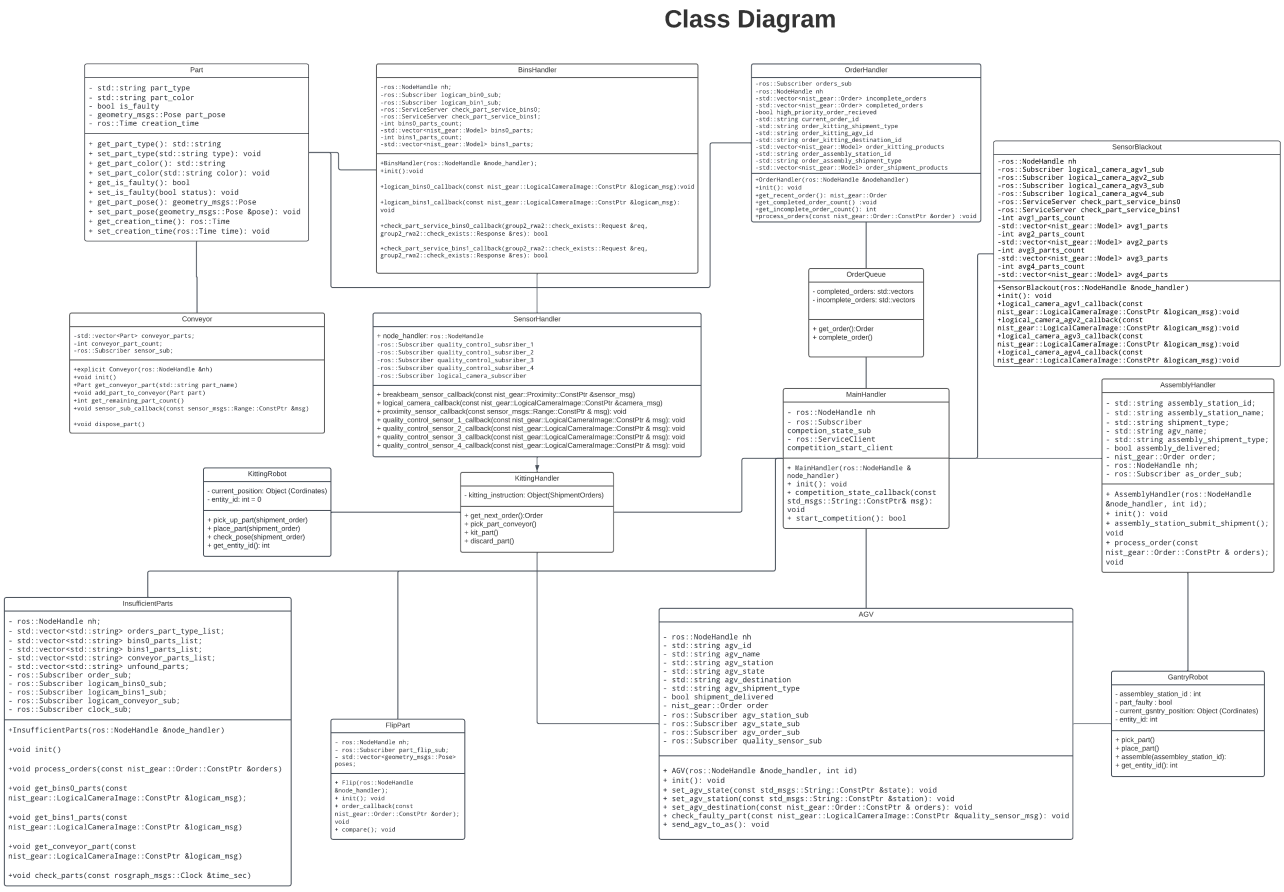

Figure 2: Updated Class Diagram

- **Approach**: Firstly, we subscribed to the topic called /ariac/quality_control_sensor_N. The quality_control_sensor only reports the parts which are faulty. When the part is spawned on an AGV the QC sensor detects it and displays a message on the terminal that "Faulty part detected on an AGV: In order for the QC sensor to detect we have to subscribe to quality_control_sensor_N which will display the published data received from the sensor and will also display the message containing the faulty part pose.

4. **Flip Part Challenge**:

- **Problem**: This challenge is a kitting based one, and needs the user to reorient the part.Firstly the challenge is about flipping a part when the roll value of the part is pi. The part that needs to flip is only the pump in ARIAC 2022. No other part will be asked to reorient. This challenge is triggered when there is change in the quaternion value of any part in the order.

- **Approach**: Firstly, we subscribed to the topic called `/ariac/orders` from which we are retrieving the pose data of the part. The order message consists of the pose in a quaternion form. We are accessing the x value and w value from the orientation block of the pose message. We are finding the value of pi in quaternion so that we can compare it and check if the part needs to be flipped. Now for every part that comes in an order, we will check if it needs to be flipped, and when there is a change in the quaternion value which matches to the roll

value being pl, the challenge will be triggered and message will be sent to the kitting robot which will take necessary actions to flip the part as necessary.

5. **Insufficient Part Challenge**:

- **Problem**: As the name conveys, this challenge handles a scenario where there are not enough parts for completing a kitting Shipment. Doing so will pre-check the required parts before the actual kitting operation, thus avoiding any hassle mid-operation. This challenge only occurs at the Kitting Station. The challenge is triggered when either the parts cannot be found in work-cell or parts are found but are in insufficient quantity. The result of handling this challenge is to submit the shipment even if not complete.
- **Approach**: Handling this challenge is done in 3 steps. After retrieving the order information from it's topic, the part_type of all the parts required for kitting shipment are stored. Depending on the kitting shipment and the AGV used, the respective 4 bins are checked for parts using the logical_camera_bin and the part_type of all the parts in the bins are stored. Now, the parts required for kitting shipment are checked with the parts currently present on the bins. If all the required parts are present, then it is concluded that there are sufficient parts to move forward with the Kitting operation and submitted. If found that required parts are missing from the bins, then for 10 simulation seconds those missing parts are searched for on the conveyor belt. Even after 10 simulation seconds, if the parts are not to be found, then a message saying "Insufficient parts" is displayed and the shipment is submitted.

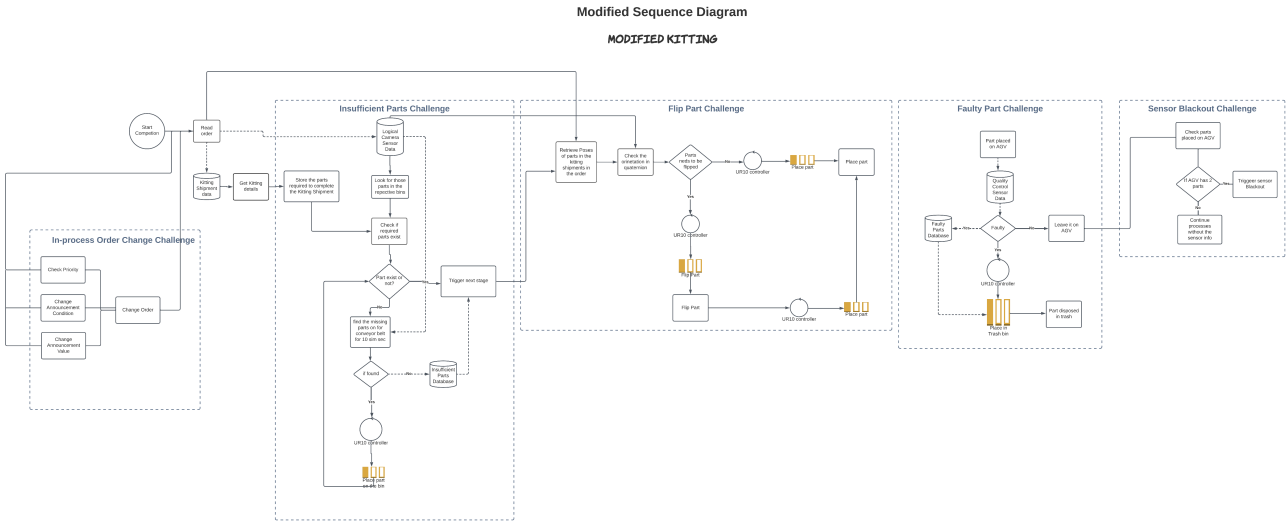

Figure 3: Updated Sequence Diagram

4 Project Contribution

S.No.	Work	Group Members
1	Understanding ARIAC Challenges	All
2	In-process Order Challenge	Vaishanth
3	Sensor Blackout Challenge	Charan
4	Faulty Part Challenge	Pranav
5	Flip Part Challenge	Bharadwaj
6	Insufficient Product Challenge	Joseph
7	Updating Class Diagram	All
8	Updating Sequence Diagram	All
9	Documentation	All
10	Report	All

5 Resources

- ARIAC Agility Challenges
- ARIAC Sensors Info
- NIST Gear Plugin Files