Architecture for RWA #3:

In the ARIAC competition there are three types of tasks in the orders, information on a task (Kitting Task, Assembly Task, or Combined Task). Each task consists of at least one part of a specific color and type. Orders are published on the topic '/ariac/orders'. For RWA#3 We will only perform Kitting Tasks. The only challenge we are considering is high-priority orders, which must be completed before low priority orders. The priority of orders with other part information is given in the config trial file 'rwa3_spring2024.yaml'. Steps in the RWA#3 are as follows:

1. The chronology of the task is first the competition state is checked using topic '**/ariac/competition_state**' if the state is in **READY** state then the competition is started using service '**/ariac/start_competition** '

2. Listen to topic '/ariac/orders' to receive orders, upon receiving first order we check its priority and for storing the order we have chosen a dequeue so that we can easily pop the high priority order first after the order is submitted and it will be faster than the list. If the order priority is LOW, then append left in dequeue else right in the dequeue, then wait for 15 seconds t_{0}.

3. Then read and store the second order similar to the previous step. After appending in dequeue, wait for 15 seconds t_{0}. Ship and submit the high priority order first and then pop left the dequeue so that we know we completed the order. Then wait for another15 seconds t_{0}.

4. Then Ship and Submit LOW priority order and pop right the dequeue, then to check if all orders are submitted check if the dequeue is empty, if yes then go ahead and check competition state and if the state is ORDER_ANNOUNCEMENTS_DONE then end the competition and if the state is not this then go back to fulfill order step.

5. If the dequeue is not empty, wait for orders.

To see t
The above is a broader overview of the steps, services we use and topics used can be seen in the diagram.

CCS /Activity diagram:(drive link:
https://drive.google.com/file/d/1_gj35kmAWOyAzIyHL4h3RVK294ynK14V/view?usp=sharing)

Start Commands

Competitor Control
System (CCS)

START → start trial → Wait until competition state is set to READY state

start competitor control system(CCS)

Start competition

using service '/ariac/start_competition'

Wait for orders

order published on topic '/ariac/orders' ← Recieve and store the first order

Order priority

HIGH → Append in dequeue left side

LOW → Append in dequeue left side

Wait 15seconds t_{0}

Recieve and store the second order

Order priority

HIGH → Append in dequeue left side

LOW → Append in dequeue left side

Wait 15seconds t_{1}

Fulfill orders

lock tray service: /ariac/agv{n}_lock_tray Move AGV serviice : /ariac/move_agv{n} ← Ship High priority order

submit by service: /ariac/submit_order Check topic : /ariac/agv{n}_status ← Submit high priority order

POP the deque left

Wait 15seconds t_{0}

lock tray service: /ariac/agv{n}_lock_tray Move AGV service : /ariac/move_agv{n} ← Ship High priority order

submit by service: /ariac/submit_order Check topic : /ariac/agv{n}_status ← Submit high priority order

POP the deque right

Check if dequeue is empty ← All orders submitted?

No

Yes → Competition state set to ORDER_ANNOUNCEMENTS_DONE

No

Yes → End Competition

STOP