**Overview of HRI Robot Controller Structure**

The methods developed allow tracking of progress through an assembly task to be done. The task is primarily undertaken by a human user with requirements for robot interaction at specific points, namely bringing parts as required and removing completed products.

A PostgreSQL database is used to keep track of users, possible actions, actions required to complete a specific task, the current action being performed by different users and an episodic memory of events.

Each user has a node which connects to 3 shimmer IMUs, reads their data and feeds it into the action classifier. This outputs the current action prediction for the user every 0.5s to the /CurrentAction topic.

The UsersNode keeps track of the state of each user. Each user object has the current task they are performing which is loaded from the relevant SQL table. The /CurrentAction for the user is read and stacked together. These are then grouped if they are the same into the user state tracker. If a new action is recognised, the previous action is uploaded to the episodic memory. The previous action is also compared to the next expected action required to complete the user’s task. If it matches, then the next expected action is updated to the next one requiring the users input. The current state table is then update along with publishing to /CurrentState.

The robot controller takes each user’s current state and looks ahead in the task list to see when the next robot action is required. The time is determined from the default times for each action and the time since the current action started. If this time is less than a threshold, the node publishes to the /RobotMove topic with the action it wishes the robot to perform.

The hri\_static\_demo node takes the action to perform from /RobotMove and performs the required set of movements to complete the action. To avoid repetitions it will not perform the same action twice in a row, a limitation that may be addressed in future.

Testing methods may be done at different levels. When using the simulation or robot without gripper, the gripper\_demo\_test is required to simulate the confirmation of gripper actions to the hri\_static\_demo node. The robot\_controller\_test may be used to test the robot control aspects without the user aspects by simply publishing the required robot movements sequentially. The fake\_imu\_publisher gives fake /CurrentAction predictions which may be used for testing without the need to connect the shimmer sensors. This sequentially publishes a set of expected action classifier outputs for completion of the task.

The im\_recog\_realsense vision recognition is used with the realsense D435i camera to output vision recognition predictions. The YOLOv3-tiny algorithm has been fine tuned on a dataset containing the box sides, screws, screwdriver, allenkey, hands, faces, various lego bricks. The node publishes the class, xy centre position, xyxy corner positions and estimated distance to each object detected on the /ObjectStates topic. In a single frame each object has one message, the Header.seq number is the same for each object in the same frame.

The count\_screws node takes the vision recognition output and for each box side in the frame counts the number of screws with centre positions lie within the bounding box of the side. This then publishes the side type and screw count for each side in the frame.

Diagram

Description automatically generated