

# Robotics Object to Context Report

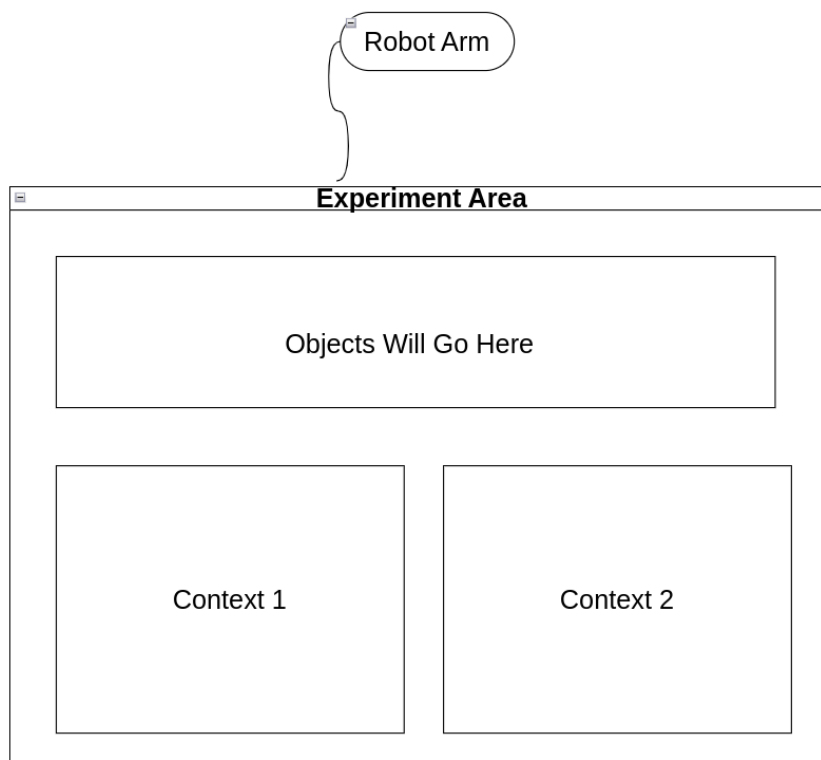
Rafael Hidalgo and Jesse Parron

## Recap

During our last discussion, we deliberated on using a robot arm to organize objects based on their relevant contexts. For instance, a toy might belong in a children's bedroom, while scissors might fit in an office. The plan involves:

1. Recognizing and identifying concepts (like pictures or room bins).
2. Recognizing objects.
3. Organizing these objects according to the most fitting context.

Jesse has been essential collaborating with me due to his expertise with robotic arms. Below is our current visualization diagram of object and context placements in our experiment zone. This arrangement can be modified as needed.



## Status of Implementation

### Robot Gripper

Initially, we faced a hurdle linking the robotic arm to my object detection code on Google Colab. While webcams previously connected without issues, the robotic arm's camera posed a problem.

Consequently, we managed to integrate my object detection and context identification codes directly into the robot's computer. Now, the robot can efficiently detect and identify objects and contexts.

We're focusing on perfecting the robot's object-grasping ability, which proved more challenging than anticipated. We've employed a pre-existing program, which, like ours, uses bounding boxes for object identification. However, it differs in its identification approach; it uses YOLO while mine utilizes detectron2. YOLO achieves near-normal video FPS, but detectron2 lags, dropping to 1-2 FPS. This lag possibly hampers the robot's grasping efficiency. Currently, Jesse is working to adjust the robot gripper's coding to suit detectron2.

One consideration we can have, to address the low FPS issue, GPU on the robot's computer can perhaps be enabled. Detectron2 seems optimized for GPU over CPU use. Although the present robot computer only uses its CPU, it does have an available GPU, which it does not use due to incompatibility issues between Linux and Nvidia. If a work around is found for this issue, then this could lead to potential improvements.

Also, Jesse has mentioned to me that another consideration we should take is choosing what objects are most compatible with the robot gripper. Some objects like remote control or computer mouse can be grabbed well by the gripper, but flat objects like any utensils are most likely not able to be picked up.

### Object to Context Logic

After refining the robot gripper, we'll focus on the logic determining object placements. I'm developing a system using ConceptNet's common sense knowledge to deduce an object's ideal location. By applying a breadth-first search in ConceptNet, we can trace all paths from a location to an object. ConceptNet assigns weights to specific connections between nodes, indicating their relevance. The robot could then make informed decisions based on these weights and paths, backed by explainable AI.

Below is a glimpse of the paths my code generates thus far, demonstrating how the robot might deduce the connection from a house to a calculator:

Paths from house to calculator:

house (AtLocation) -> computer (IsA) -> calculator | Weight: 8.00

house (AtLocation) -> table (RelatedTo) -> calculator | Weight: 2.00

house (AtLocation) -> computer (RelatedTo) -> dekatron/n (RelatedTo) -> calculator | Weight: 8.00

house (AtLocation) -> furniture (RelatedTo) -> table (RelatedTo) -> calculator | Weight: 6.32

house (AtLocation) -> bed (RelatedTo) -> table (RelatedTo) -> calculator | Weight: 4.00

Please also find attached an example of the ConceptNet ontology below.

If you have any questions, please feel free to ask.

