Cocktail Bot

André Pereira
João Marinho
Maria Teles

Maria

Introduction and motivation

In today's tech-driven era, autonomous systems are transforming industries for the better. One prevalent issue faced by bars is the extended waiting time for intricate cocktails.

Cocktail Bot - the autonomous cocktail robot, aims to revolutionize the cocktail-making process. Using automation we aim to minimize wait times and cost while improving operations efficiency and customer experience.

Purpose and goals

Main goal: Autonomous robot capable of crafting a diverse range of cocktails.

Sub-Goals:

1. **Exploration:** Autonomous gathering of knowledge about the environment. Collect information about storage places.
2. **Parse cocktail recipe:** Determine the objects that compose the recipe.
3. **Collect ingredients:** Discover/Move to ingredient locations, one at a time.
4. **Finish making cocktail:** Return to original location to deliver the cocktail.

Problem description

Exploration

How to get information about the environment?	Explore environment.
What places should the robot explore?	List of points of interest.

Parse cocktail recipe

What ingredients is the cocktail made of?	Query knowledge base.
Where can the ingredients be stored?	Reason about object relations present in the ontology.

Maria

Problem description

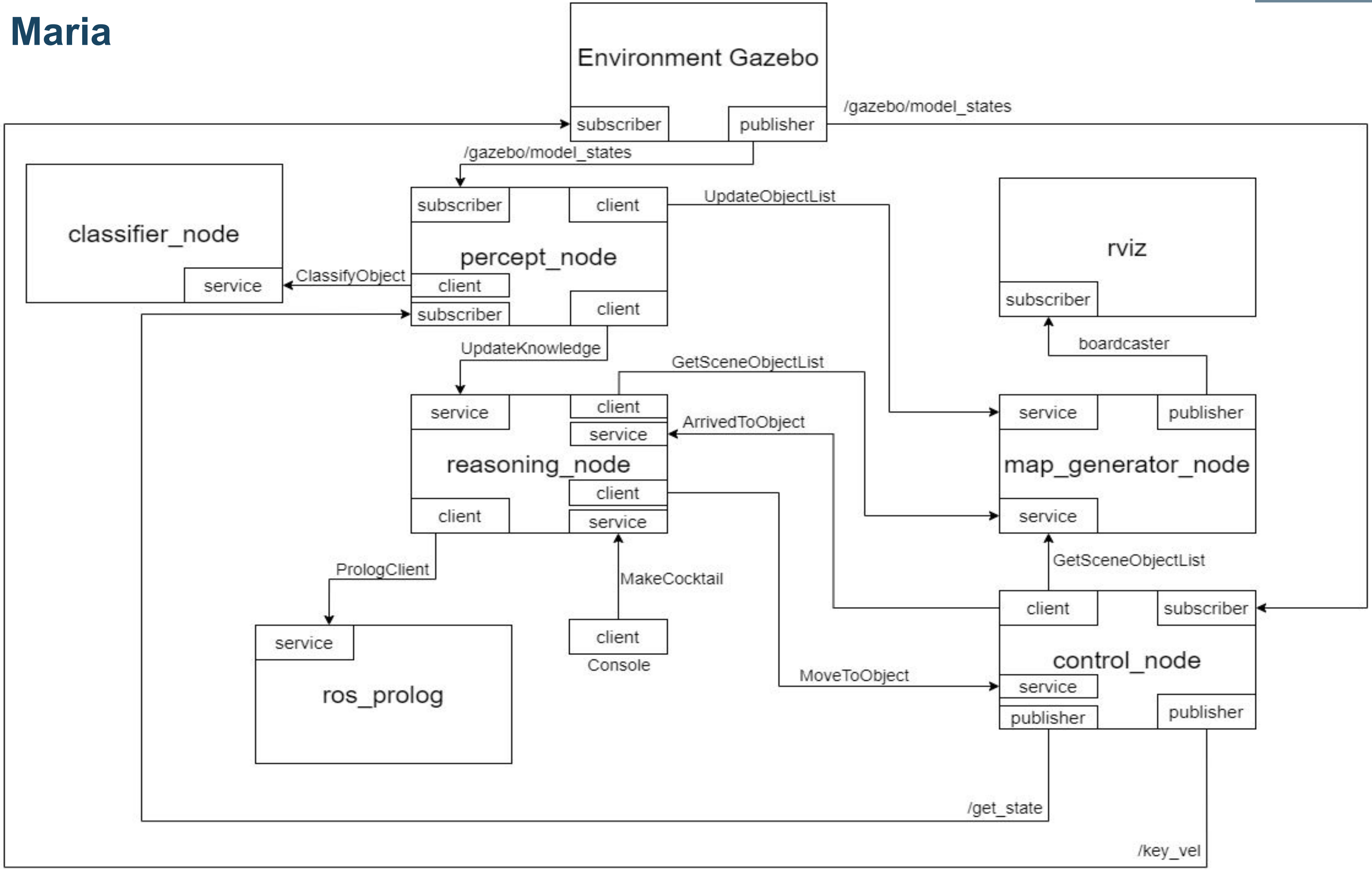
Collect ingredients

Where is the ingredient located?	Access information stored over time.
Where should I go to search for ingredient?	Move to possible ingredient containers.
How do I move to a position?	Navigation system.
How to identify the nearby object?	Perceive nearby object and identify their type.

Finish making cocktail

Did I pick up all the ingredients?	Successfully reached all ingredients.
How do end the task?	Move back to original station.

Maria



Environment Gazebo
subscriber
publisher
/gazebo/model_states
/gazebo/model_states
classifier_node
service
ClassifyObject
subscriber
client
percept_node
client
subscriber
client
UpdateObjectList
rviz
subscriber
boardcaster
UpdateKnowledge
GetSceneObjectList
service
client
service
ArrivedToObject
reasoning_node
client
client
service
service
publisher
map_generator_node
service
GetSceneObjectList
PrologClient
MakeCocktail
client
subscriber
control_node
service
publisher
publisher
service
ros_prolog
client
Console
MoveToObject
/get_state
/key_vel

Method and implementation
(Nodes)

Perception: Collects information from the environment.

- **Gazebo Subscriber:** Gather knowledge about the nearby objects.
- **Classifier Client:** Object classification.
- **Update Object List Client:** Update object instance mapping.
- **Update Knowledge Client:** Update knowledge base instances.
- **Controller Subscriber:** Get current robot state.

Classification: Identify the type of object.

- **Classifier Service:** Classifies objects based on their characteristics.

Maria

Method and implementation
(Nodes)

Reasoning: Fulfills the cocktail request by telling the robot where to go.

- **Cocktail Service:** Responsible to process the cocktail request.
- **Update Knowledge Service:** Receives instances to update knowledge base.
- **Prolog Client:** Interacts with prolog knowledge base.
- **Move to Object Client:** Informs the controller which object to move to.
- **Arrived to Object Service:** Know when the robot has arrived to the target.
- **Get Object Client:** Requests information (pose) about a certain object.

Method and implementation
(Nodes)

Mapping: Stores information about the object position.

- **Rviz Publisher:** Broadcast object position to rviz node.
- **Get Object Service:** Provides the requested object(s) poses.
- **Update Object List Service:** Updates the map with the new object and respective pose.

Method and implementation
(Nodes)

Controller: Commands the robot to reach its destinations.

- **Move to Object Service:** Handles movement requests to the robot.
- **Get Object Client:** Requests information about the object pose.
- **Gazebo Subscriber:** Receives information about the current robot pose.
- **Controls Publisher:** Publishes movement controls to the robot.
- **State Publisher:** Publishes the current robot state.

João

Reasoning description & implementation
(Deductive reasoning)

- Verify if robot has picked up an ingredient/finish cocktail. **(Propositional Logic)**
  - Distance from target to robot < 1.5 ?
  - Collected all ingredients ?
- Get recipe ingredients. **(First Order Logic)**
  - Premise: Margarita contains [Tequila, Lime, Salt, Ice].
  - Conclusion: We only need those ingredients for a margarita.
- Find the containers that store the ingredients. **(Description Logic)**
  - Premise: Tequila is a subclass of Drink.
  - Premise: Drinks are stored in a drink storage.
  - Conclusion: Tequila is stored in a drink storage.

João

Reasoning description & implementation
(Predicates)

cocktail_recipe/2

Defining cocktail recipes and their ingredients.

?Cocktail_name

?Cocktail_ingredients

cocktail_recipe(margarita, ['Tequila', 'Lime', 'Salt', 'Ice']).

get_instances_for_cocktail/4

Obtain the instances for a given cocktail.

+Cocktail % Name of the cocktail

-Ingredients % List of ingredients

-Ingredient_inst % List of instances for each ingredient

-Alternative_inst % List of container instances for each ingredient

João

Reasoning description & implementation
(Predicates)

get_instances_for_class/3

Return the instances for a given item class.

+Class % Name of the class

-Class_inst % List of instances for the class

-Alt_inst % List of container instances for the class

get_storage_for_class/2

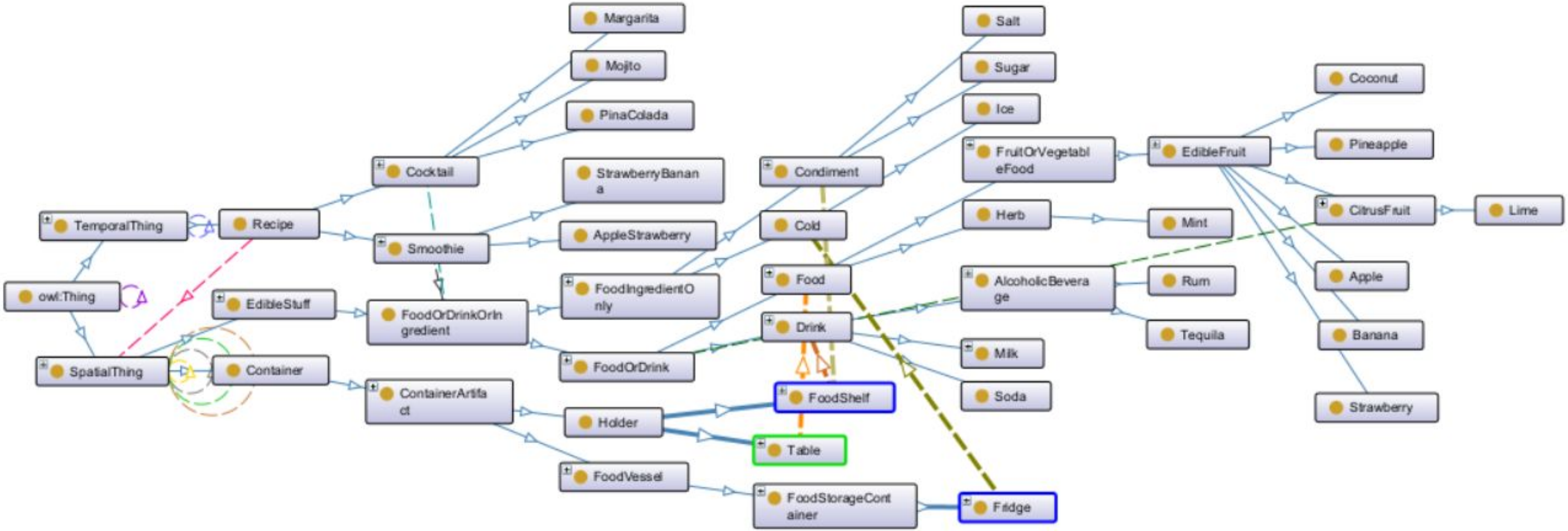
Return the storage class for a given item class.

+Class % Class name

-Storage % Storage related to the class

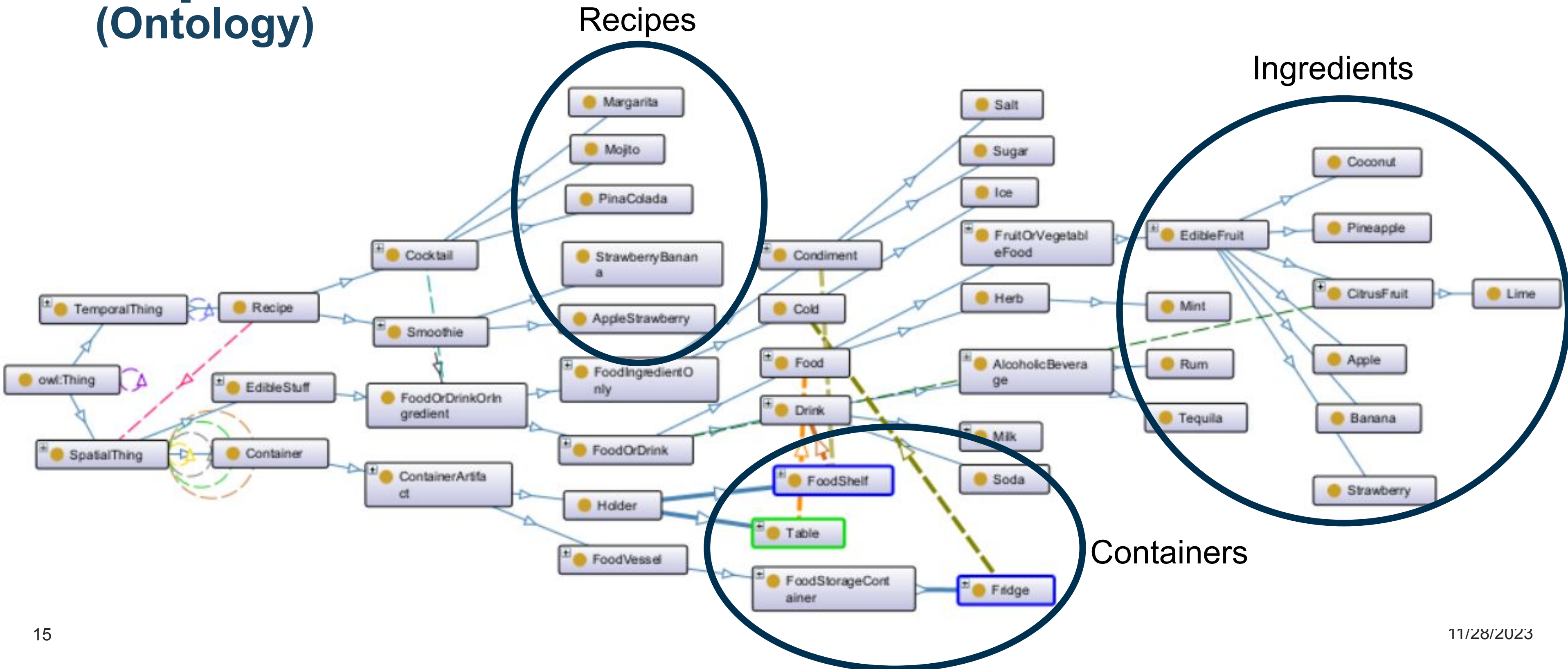
André

Reasoning description & implementation
(Ontology)



André

Reasoning description & implementation
(Ontology)

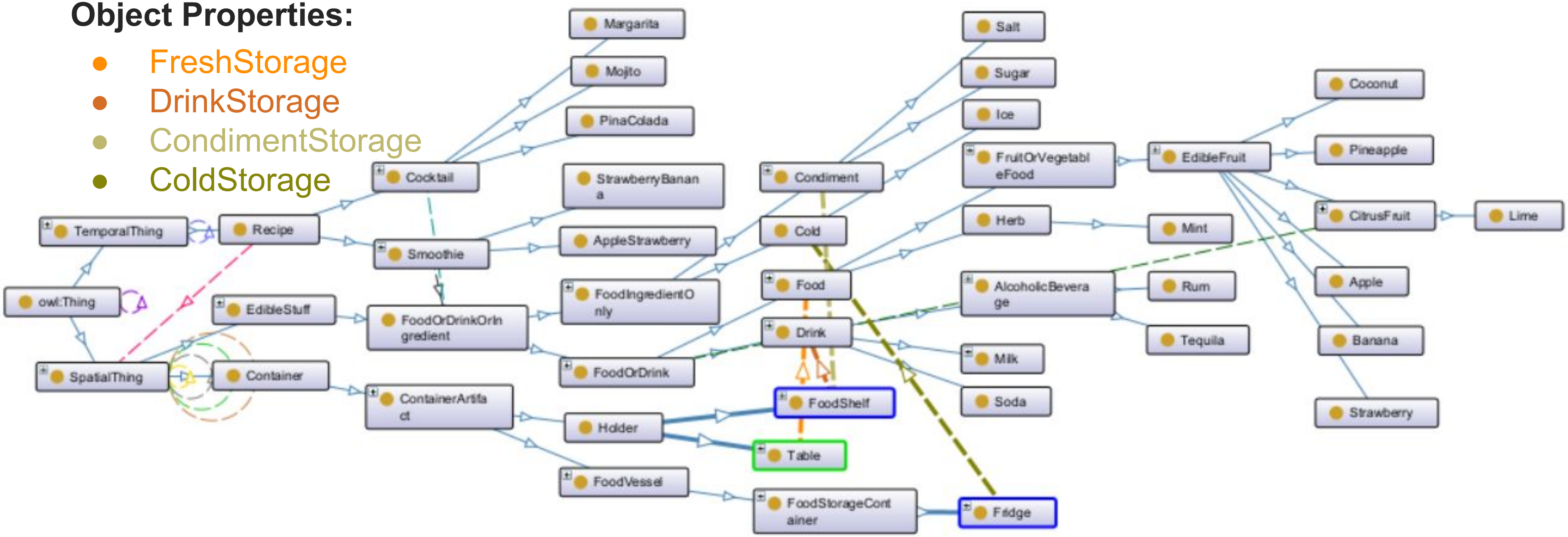


André

Reasoning description & implementation
(Ontology)

Object Properties:

- FreshStorage
- DrinkStorage
- CondimentStorage
- ColdStorage



André

Reasoning description & implementation
(Inductive reasoning)

- Identify type of environment objects. **(Decision Tree)**
  - Premise: An apple's average mass is 165g.
  - Premise: Apple's have an oval shape.
  - Conclusion: All ovals objects with an average mass of 165g are apples.

Learning Implementation

Data: Object label and 8 characteristics

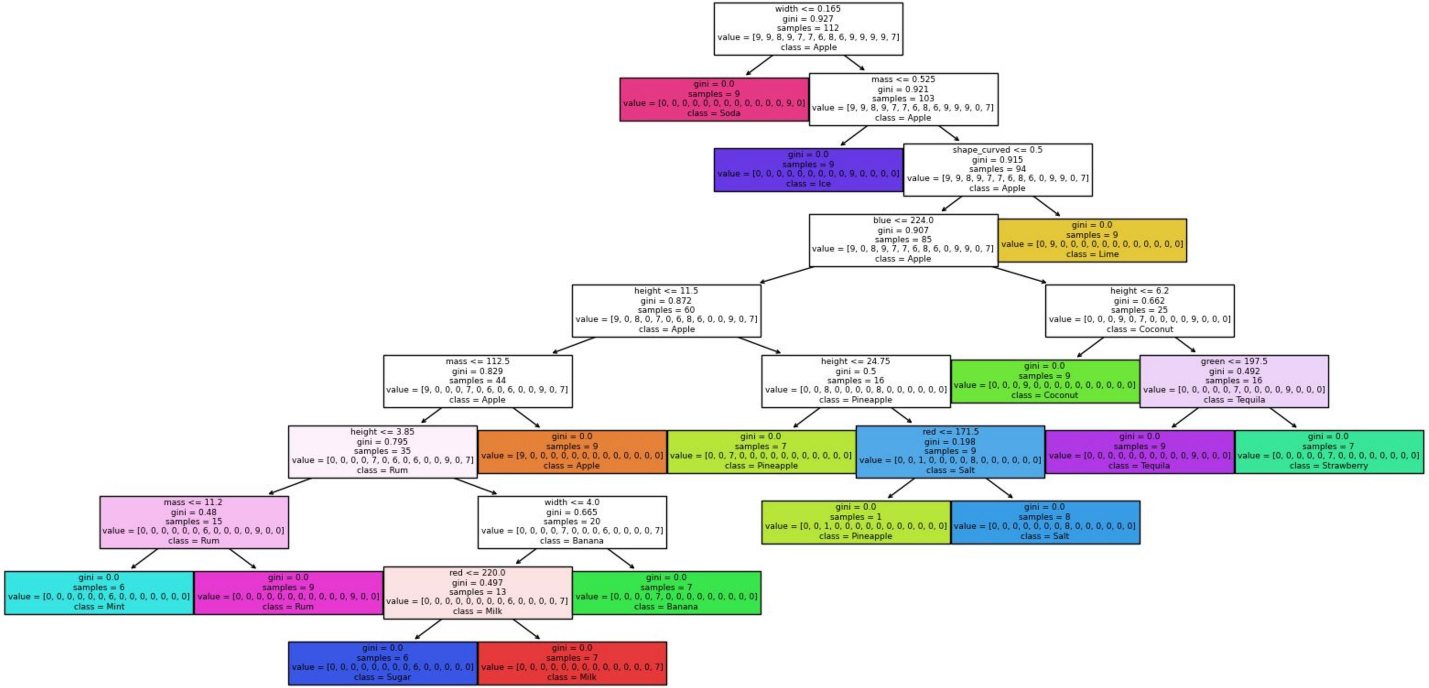
- 140 rows.
- 14 unique classes.
- Generated with OpenAI large language model.

label	mass	width	height	shape	red	green	blue	alcohol
Apple	150	7	8	round	255	0	0	0
Apple	130	6.5	7.5	oval	220	20	60	0
Apple	180	7.5	8	irregular	255	165	0	0
Apple	140	6	7	round	0	128	0	0
Apple	160	7	7.5	oval	165	42	42	0
Apple	170	6.5	7	round	255	69	0	0
Apple	150	6.8	7.2	oval	218	165	32	0
Apple	140	6.2	7	round	255	223	0	0
Apple	160	6.7	7.3	oval	204	51	51	0
Apple	175	7.2	7.8	round	255	182	193	0
Lime	80	5	6	oval	0	255	0	0
Lime	60	4.5	5	round	0	128	0	0
Lime	90	5.5	6.5	oval	34	139	34	0
Lime	70	4.8	5.5	round	50	205	50	0
Lime	75	5	5.8	oval	0	255	127	0
Lime	85	4.5	7	cylinder	0	100	0	0
Lime	95	5.2	6.2	oval	173	255	47	0
Lime	78	4.7	5.7	round	144	238	144	0
Lime	80	5	6	oval	255	69	0	0
Lime	65	4.2	5	round	0	255	0	0
Pineapple	1200	20	30	oval	255	223	0	0
Pineapple	1400	22	35	oval	255	165	0	0
Pineapple	1100	18	28	oval	255	99	71	0
Pineapple	1000	15	25	cylinder	255	215	0	0
Pineapple	1300	21	32	oval	255	140	0	0

André

Learning Implementation

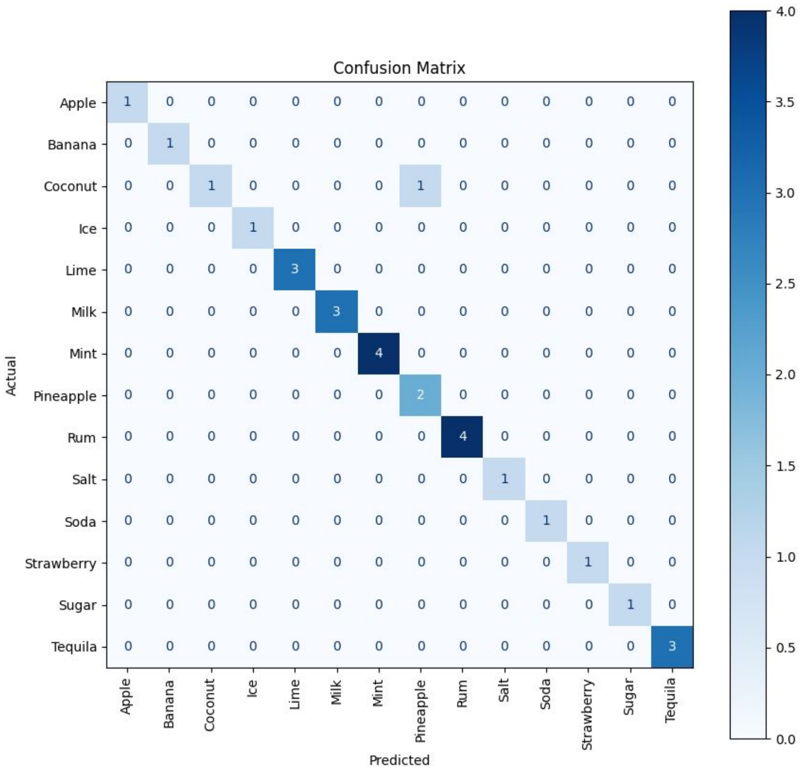
Approach: Decision Tree (sklearn)



Learning Implementation

Accuracy: 96.43%

lable	precision	recall	f1-score	support
Apple	1	1	1	1
Banana	1	1	1	1
Coconut	1	0.50	0.67	2
Ice	1	1	1	1
Lime	1	1	1	3
Milk	1	1	1	3
Mint	1	1	1	4
Pineapple	0.67	1	0.80	2
Rum	1	1	1	4
Salt	1	1	1	1
Soda	1	1	1	1
Strawberr	1	1	1	1
Sugar	1	1	1	1
Tequila	1	1	1	3

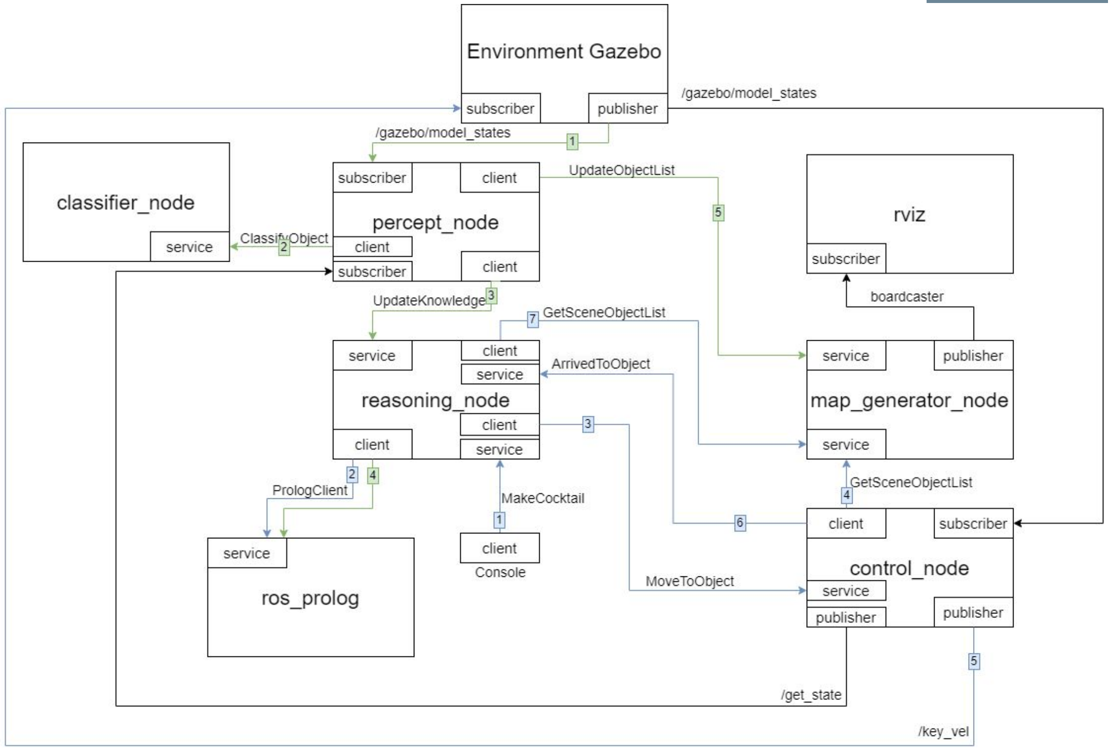


Robotics Implementation

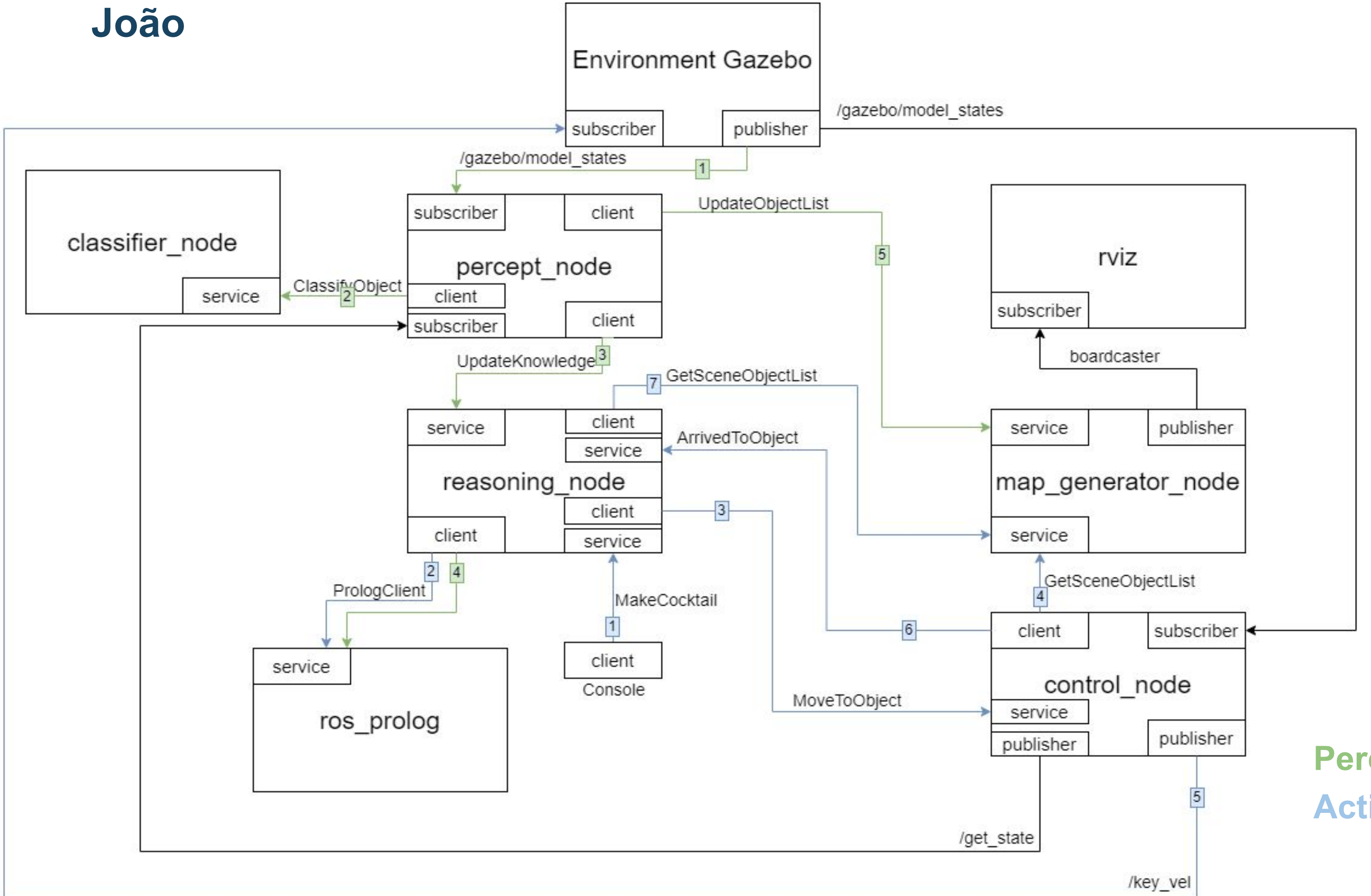
Input: Service call with recipe name, e.g.:
- margarita
- mojito
- pina_colada

Outcome:
- Reason about ingredient locations
- Move to ingredient instances
- Return to base station

Perception: Proximity to objects.



João



Perception

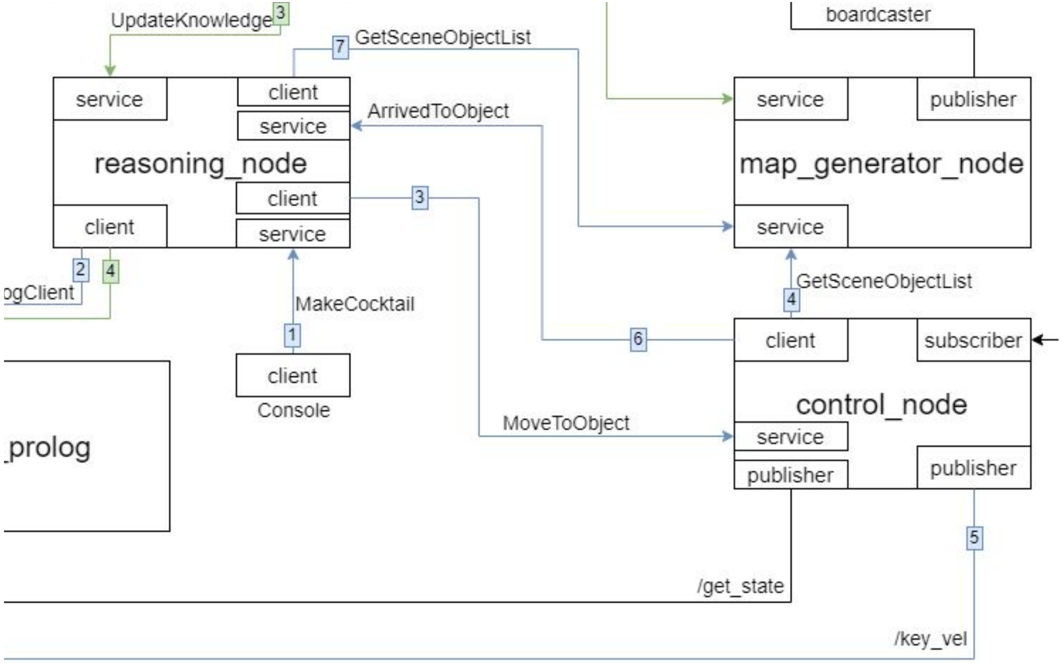
Action

11/28/2023

Robotics Implementation

Action loop:

1. Receive recipe from client.
2. Get recipe ingredients and containers instances from knowledge base.
3. Request to move to an object instance.
4. Get object's pose.
5. Calculate movement commands and send to robot.
6. Inform reasoner of robot arrival to target position.
7. If ingredient is not known, verify if any new instance is discovered and if it is close enough to pick up.

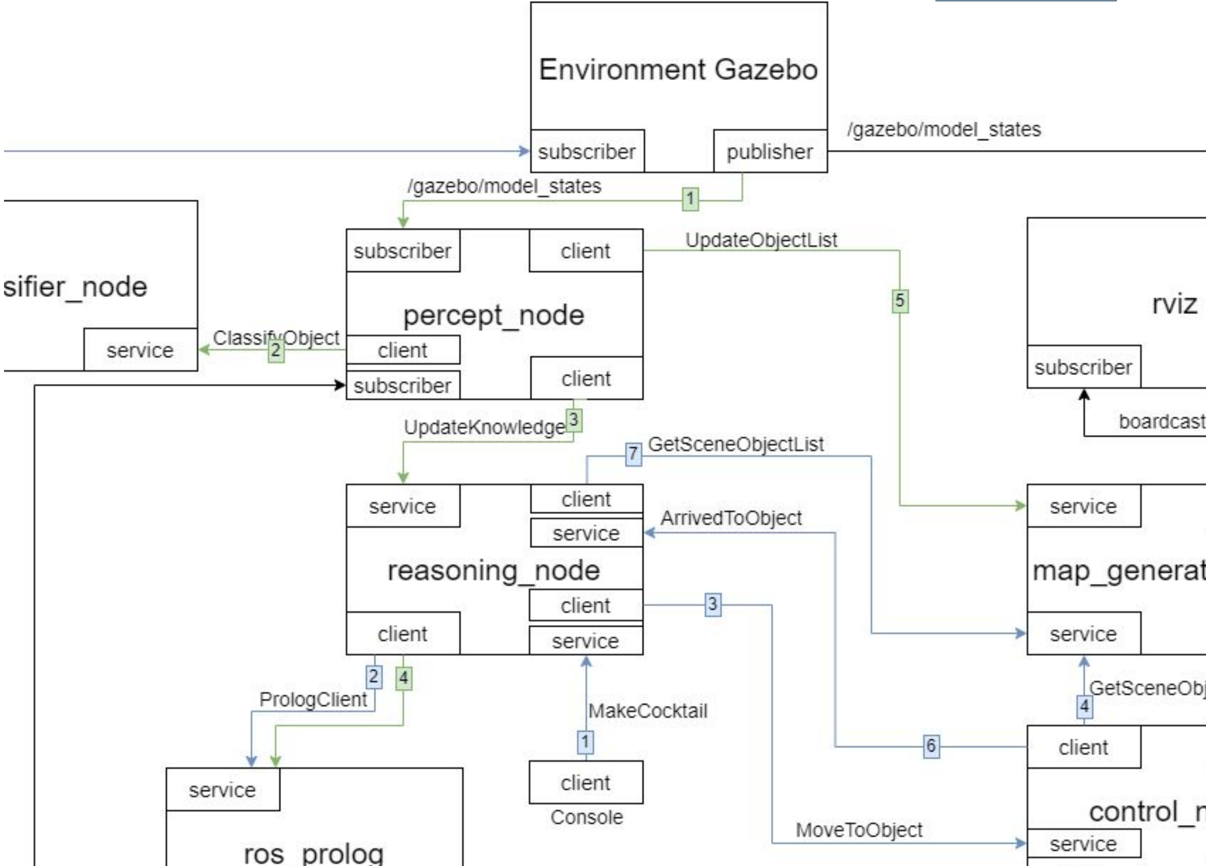


João

Robotics Implementation

Perception loop:

1. Perceive objects near the robot.
2. Classify the newly found object instance based on their characteristics.
3. Inform reasoner of new object instance.
4. Store object instance in knowledge base.
5. Store object pose in mapping node.



João

Video demo

Maria

Limitations

Scope of the project:

- Grasping objects
- Image processing

Implementation:

- Impossible to avoid obstacles (robot decides path in a straight line)
- The robot is not able to perceive any ingredient during the exploration phase (even though some may be close to it)
    - This also implies that if the robot misses a container it may never find certain ingredients

Conclusion

The problem description highlighted intricate decision-making processes for autonomous cocktail creation, encompassing exploration, perception, and reasoning:

- Integration of perception and reasoning showcased the complexity necessary for successful autonomous operations;
- Machine learning, specifically a decision tree approach, emphasized the significance of data-driven decision-making;

The Cocktail Bot project demonstrated the integration of robotics, automation, and machine learning, emphasizing the practical relevance of these principles in addressing real-world challenges, making it a valuable learning experience in today's tech-driven landscape.

CHALMERS
UNIVERSITY OF TECHNOLOGY

CHALMERS
UNIVERSITY OF TECHNOLOGY