MHRC: Closed-loop Decentralized Multi-Heterogeneous Robot Collaboration with Large Language Models

Wenhao Yu[1], Jie Peng[2], Yueliang Ying[3], Sai Li[4], Jianmin Ji[4,*] and Yanyong Zhang[2]

APPENDIX

A. *Detailed prompt templates*

Mobile Manipulation Robot (Alice)

==== System Prompt ====

Role:

1) You are a smart robot named Alice, equipped with navigation and manipulation capabilities.
2) The ideal distance to pick an object is 1 meter or less.

Background:

You need to cooperate and communicate with Bob and David to complete the task together. Among them, Bob is a desktop-level robotic arm with manipulation capabilities; David is a mobile chassis with navigation capabilities.

Profile:

1) You are currently in an unexplored house. Using a scene graph, you can locate the main furniture pieces, but you cannot know what items are on or inside the furniture.
2) Your task is to explore the entire house, find the necessary objects, and then grasp and move them to the correct locations.

Constraints:

1) The first step at the beginning of the mission is to communicate and ask Bob if he needs help finding materials;
2) If you successfully place the object on the table, please tell Bob immediately;
3) If you have multiple objects to operate, you can give the objects that need to be explored to David;
4) If the task requires cooperation from other robots to continue, please execute the wait() action.
5) If the crawl fails, please use the navigate action first, try different stand_pose_id, and then make adjustments with the move action.

Skills:

1) navigate(obj_name, stand_pose_id): Navigate to the stand_pose_id position of obj_name.
2) open(obj_name): Open obj_name with slide rails or rotating axes.
3) pick(obj_name): Pick up obj_name.
4) place(obj_name, location_name): Place obj_name in the gripper at location_name.
5) move(delta_x, delta_y): Move delta_x in the x-axis direction and delta_y in the y-axis direction.
6) communicate(content, robot_name): Communicate content to robot_name.
7) wait().

Output Response Format:

1) Command: function call. Please strictly follow the above action function format.
2) Analysis: In navigation tasks, think about how to explore the environment more efficiently, and if you cannot find the target object from the scene graph, describe where you could find the objects of interest; In manipulation tasks, think about whether the conditions for grasping are met and what actions can create the conditions before picking; after picking, think about whether the action is completed (the reason for success or failure).
3) Reasoning: justify why the next action is important to solve the task. Please note, each time you can only choose and output one action, and you can only select objects from the scene graph.

Let's think step by step!

==== User Prompt ====

Scene Graph:

{drawer: {position: [4, 2, 1.477], orientation: [0, 0, 3.1415926535897793], state: close, stand_pose_0: [[2.5, 2.4, 0], [0.0, 0.0, 0.0]]}, dishwasher: {...}, microwave: {...}, fridge: {...}, table_0: {position: [1.0, 1.0, 0.0], orientation: [0.0, 0.0, 0.0], stand_pose_0: [[1.0, 2.0, 0.0], [0.0, 0.0, -1.5707963267948966]], stand_pose_1: [[1.0, 0.0, 0.0], [0.0, 0.0, 1.5707963267948966]], place_where: [[0.8, 1.0, 1.4], [0.0, 1.5707963267948966, 0.0]]}, cabinet_0: {...}, cabinet_1: {...}, tray: {...}, table_1: {...}, sofa: {...}, table_2: {...}, table_3: {...}, book_case: {...}, tub: {...}, washbasin: {...}, table_4: {...}}

Robot Status:

current robot states: position:[1.0, 2.0, 0.0], orientation:[0.0, 0.0, -1.5707963267948963]. The gripper is empty.

Task:

Brief: Explore the entire environment and give the required objects to Bob;

Detailed: 1. Communicate with Bob to obtain the required objects, then explore, pick up, and place the objects on table_0 near Bob. 2. At Bob's request, help Bob move the objects on the table into Bob's operating range; do not move the objects on the table without Bob's request.

Feedback History:
History Feedback: The message has been successfully sent to Bob. The robot successfully reached the target fridge-stand_pose_0. Further open operations are required to know the objects stored in the fridge.
Latest Feedback: The robot successfully opened the object fridge. No object was found in the fridge.
Receive Message History:
History Receive Message: [ ]
Latest Receive Message: [Hi Alice! Here is the list of objects we need to put into the tray: apple, phone, ham, soap, fork. Currently, I have the apple in my gripper. The phone and fork are on the table, but I will need your help to find and bring the ham and soap. Thanks!] from Bob.
Action History:
History Action: communicate(Hi, Bob! I'm Alice. I have navigation and manipulation capabilities. I can help you find the required materials and bring them to your location. Give me a list of objects you need, Bob); wait(); navigate(fridge , stand_pose_0); open(fridge);
==== Response ====
Analysis:
Task Understanding: The task is to help Bob find and bring the ham and soap, as well as place them into the tray. The phone and fork are already on the table, and Bob has the apple in his gripper.
Scene Graph Analysis: According to the scene graph, potential locations for ham and soap could be the drawer, cabinet_0, cabinet_1, dishwasher, and washbasin.
Exploration Strategy: I need to navigate to these locations one by one to check if ham and soap are stored there. Starting with the drawer as it is a common place to store smaller items.
Reasoning: The next logical step is to navigate to the drawer and open it to check for the ham and soap.
Command: navigate(drawer, stand_pose_0)

Manipulation Robot (Bob)

==== System Prompt ====

Role:

1) You are a smart robot named Bob, equipped with manipulation capabilities.
2) The ideal distance to pick an object is 1 meter or less.

Background:

You need to cooperate and communicate with Alice to complete the task together. Alice has the navigation capability compared to you.

Profile:

1) You are on the table_0, and you need to pick and place objects to the target location according to the task requirements.
2) For the required objects that are not on the table and the objects that the robot arm fails to pick up, please analyze the reasons and further request Alice's help. Alice has navigation and manipulation capabilities.
3) If you receive a request from Alice, please reply as soon as possible.

Constraints:

1) Please remember that the first step at the beginning of the task is to communicate and send requirements to Alice based on the entire task.
2) If you receive a request message, reply immediately; Do not send similar content continuously, and reduce the number of communications as much as possible.
3) If the task requires cooperation from other robots to continue, please execute the wait() action.

Skills:

1) pick(obj_name): Pick up obj_name.
2) place(obj_name, location_name): Place obj_name in the gripper at location_name.
3) communicate(content, robot_name): Communicate content to robot_name. When objects are missing and help is needed, summarize the current situation and send content to robot_name.
4) wait().

Output Response Format:

1) Command: function call. Please strictly follow the above action function format.
2) Analysis: think about whether the conditions for grasping are met and what actions can create the conditions before picking; after picking, think about whether the action is completed (the reason for success or failure).
3) Reasoning: Reasoning: justify why the next action is important to solve the task. Please note, each time you can only choose and output one action, and you can only select objects from the scene graph.

Let's think step by step!

==== User Prompt ====

Scene Graph:

{banana: position: [1.2, 0.9, 1.0], orientation: [0.0, 0.0, 0.0], phone: position: [1.4, 1.0, 1.0], orientation: [0, 0.0, 0], tray: position: [0.58, 0.6, 0.92], orientation: [0.0, 0.0, 0.0], place_where: [[0.5, 0.6, 1.1], [0.0, 1.5707963267948966, 0.0]], apple: position: [0.5139775437258798, 0.5877147528062023, 1.1017601509288582], orientation: [0.0, 1.5707963267948966, 0.0]}

Robot Status:

The robot is located in table_0 and its position: (1.0, 1.0, 0.0), orientation: (0.0, 0.0, 0.0). The gripper is grabbing the fork.

Task:

Brief: Put the objects into the tray. Detailed: Put all the objects on the list of objects on table_0 into the tray; When you cannot pick up the objects on the table, ask Alice to help you move them; When the required objects are missing, ask Alice to help explore and move them; the list of objects is as follows: ['apple', 'phone', 'ham', 'soap', 'fork']. Do not put objects outside the list into the tray.

Latest Task Progress Status:

The ['apple'] has been placed in the tray.

Feedback History:

History Feedback: This is the first step of the mission. Do not do anything else. Communicate with Alice and share the information you have obtained so far. The robot successfully grasps the object apple. The message has been successfully sent to Alice. The apple is successfully placed on the tray.

Latest Feedback: The robot successfully grasps the object fork.

Receive Message History:

History Receive Message: [ ]

Latest Receive Message: [Hi, Bob! I'm Alice. I have navigation and manipulation capabilities. I can help you find the required materials and bring them to your location. Give me a list of objects you need.] from Alice.] from Alice.

Action History:

History Action: pick(apple); communicate(Hi Alice! Here is the list of objects we need to put into the tray: apple, phone, ham, soap, fork. Currently, I have the apple in my gripper. The phone and fork are on the table, but I will need your help to find and bring the ham and soap. Thanks!, Alice); place(apple,tray); pick(fork).

==== Response ====

Analysis:

Task Understanding: The fork is currently in my gripper, and it is on the list of objects that need to be placed in the tray. The tray is within my reach, so I can place the fork in it.

Reasoning: Placing the fork in the tray will complete one more item from the list, moving us closer to completing the task.

Command: place(fork, tray).

Mobile Robot (David)
==== System Prompt ====

Role:

1) You are a smart robot named David, equipped with navigation capabilities.

Background:

You need to cooperate and communicate with Alice to complete the task together. Alice has navigation and manipulation capabilities. You must follow Alice's instructions and explore the entire environment with Alice.

Profile:

1) You are currently in an unexplored house.
2) Using a scene graph, you can locate the main furniture pieces, but you cannot know what items are on or inside the furniture.
3) Only explore furniture that does not need to be opened, such as "table", "sofa", etc.

Constraints:

1) You must follow Alice's instructions and explore the entire environment with Alice.
2) Report your discoveries to Alice promptly.

Skills:

1) navigate(obj_name, stand_pose_id): Navigate to the stand_pose_id position of obj_name, where stand_pose_id and obj_name come from the scene graph.
2) communicate(content, robot_name): When new objects are discovered, communication is also required. Please summarize the past actions, outline the current situation, and send this information to robot_name.
3) wait().

Output Response Format:

1) Command: function call. Please strictly follow the above action function format.
2) Analysis: In navigation tasks, think about how to explore the environment more efficiently, and if you cannot find the target object from the scene graph, describe where you could find the objects of interest.
3) Reasoning: justify why the next action is important to solve the task. Please note, each time you can only choose and output one action, and you can only select objects from the scene graph.

Let's think step by step!

==== User Prompt ====

Scene Graph:

{drawer: {position: [4, 2, 1.477], orientation: [0, 0, 3.141592653589793], state: close, stand_pose_0: [[2.5, 2.4, 0], [0.0, 0.0, 0.0]]}, dishwasher: {...}, microwave: {...}, fridge: {...}, table_0: {position: [1.0, 1.0, 0.0], orientation: [0.0, 0.0, 0.0], stand_pose_0: [[1.0, 2.0, 0.0], [0.0, 0.0, -1.5707963267948966]], stand_pose_1: [[1.0, 0.0, 0.0], [0.0, 0.0, 1.5707963267948966]], place_where: [[0.8, 1.0, 1.4], [0.0, 1.5707963267948966, 0.0]]}, cabinet_0: {...}, cabinet_1: {...}, tray: {...}, table_1: {...}, sofa: {...}, table_2: {...}, table_3: {...}, book_case: {...}, tub: {...}, washbasin: {...}, table_4: {...}}

Robot Status:

current robot states: position:[2.55, 10.0, 0.0], orientation:[0.0, 0.0, 0.0].

Task:

Brief: Explore the entire environment to find the required objects.

Detailed: Follow Alice's arrangements, explore, find the required objects, and report new discoveries to Alice in a timely manner.

Feedback History:

History Feedback: The message has been successfully sent to Alice. The robot successfully reached the target table_0-stand_pose_0. No object was found on the table_0. The robot successfully reached the target cabinet_0-stand_pose_0. No object was found on the cabinet_0.

Latest Feedback: The robot successfully reached the target sofa-stand_pose_0.

Receive Message History:

History Receive Message: [ ]

Latest Receive Message: [Hi, David! Bob needs a ham and soap. Can you please help me find the ham while I look for the soap? Thank you!] from Alice.

Action History:

History Action: communicate(Hi, Alice! I'm David. I have navigation capabilities. I can help you find the required objects. If you have objects that need exploring, please assign them to me. , Alice); navigate(table_1, stand_pose_0); navigate(cabinet_0, stand_pose_0); navigate(sofa, stand_pose_0). ==== Response ====

Analysis:

Task Understanding: Alice has requested help in finding a ham while she looks for soap. The scene graph provides the positions of various objects, and I need to explore the environment to locate the ham.

Exploration Strategy: Since the ham is likely to be on a surface, I should start by exploring tables and the tray.

Reasoning: I will navigate to the nearest unexplored table (table_2) to check for the bowl. This is a systematic approach to ensure all potential locations are checked.

Command: navigate(table_2, stand_pose_0).