2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Design of a Multi-robot Coordination System based on Functional Expressions using Large Language Models

Yuki Kato[1], Takahiro Yoshida[1], Yuichiro Sueoka[1], Koichi Osuka[1],
Ryosuke Yajima[2], Keiji Nagatani[2], Hajime Asama[2]

***Abstract*— A system is expected to facilitate coordination among multiple construction machines or robots, enabling them to adaptively perform various tasks in disaster sites and unknown environments. Prior research has generally adopted a model-based approach to designing cooperative behavior. However, it is difficult to adapt to environments and scenarios that cannot be predicted by the model. In recent years, it has been reported that a robot equipped with foundation models can adapt to unknown (open) environments and unpredictable situations. However, there has been little discussion on foundation models for multiple robot systems; a flow that cooperatively handles unexpected events does not exist. In this paper, we propose the system flow that enables multiple robots to adaptively coordinate to unforeseen scenarios based on the functional expressions of each other and environment understanding utilizing *GPT-4* and *GPT-4V*. Through experimentation, we verify that the proposed flow is able to adapt to an unforeseen environment, particularly path obstruction via robot experiments. Furthermore, we examine the validity of the proposed flow by varying the robots' functional expressions and sensor information for the environment.**

I. INTRODUCTION

In unknown environments such as disaster sites or the lunar surface, a system is expected to facilitate coordination among multiple construction machines (robots) to perform various tasks [1], [2]. In these environments, robots may face unforeseen situations, including getting stuck or path obstruction. Prior research generally adopts a model-based approach to designing cooperative behavior [3], [4]. However, it is difficult to adapt to environments and scenarios that cannot be predicted by the model [5].

Recent progress in foundation models [6] has brought high generalization performance in environmental perception and task planning from natural language to robotics. Currently, examples of applying foundation models to open-world environments, such as OK-Robot [7] and GOAT [8], are increasing. On the other hand, few studies have been done for multi-robot systems. There is a lack of discussion about a system flow for multiple robots to adaptively coordinate to unforeseen situations that were not explicitly predicted in advance.

This paper proposes a system flow enabling multiple robots to adaptively address unforeseen situations, leveraging foundation models for functional expressions of each other's robots and environment understanding. The individual flow is illustrated in Fig. 1, and a specific example is shown in Fig. 2. The flow consists of the decision-making (action planning), the robot control (action execution), and the environment understanding to comprehend situations. If an unexpected situation occurs, the flow can handle it in two ways: if a robot can solve it by its own function, the robot addresses it individually. Otherwise, the robot asks for help from other robots for cooperative handling. Notably, the characteristic of the proposed flow is that the decision-making process takes both the functions of individual robot and those of other robots, in addition to the conventional flow mounted on a single robot. This system allows for robots to un-

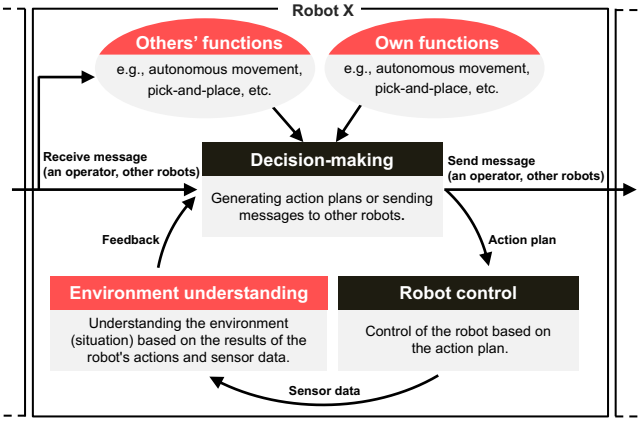

Fig. 1. Proposed flow for adaptive coordination: Making decisions in response to unforeseen situations through environment understanding and the functions of itself and other robots.

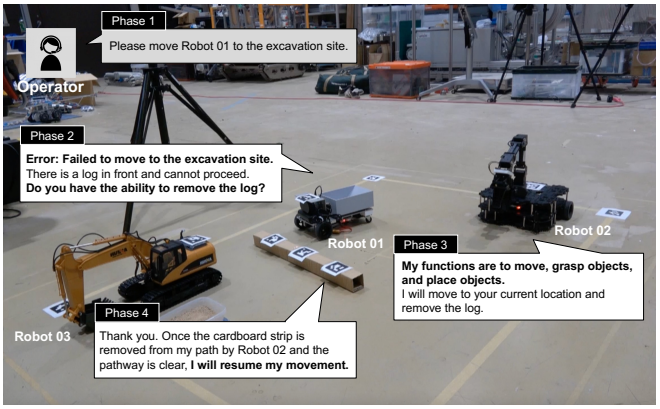

Fig. 2. An example of unexpected scenario (path obstruction): Comprehending the scenario through environment understanding by Robot 01 and helps by removing the obstacle with functional expressions of Robot 02.

[1]Graduate School of Engineering, Osaka University, Japan kato.y@dsc.mech.eng.osaka-u.ac.jp
[2]Graduate School of Engineering, The University of Tokyo, Japan

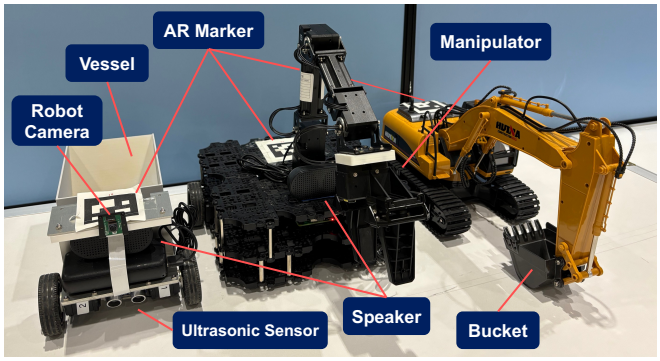

Fig. 3. Overview of the actual robots used in experiments (left: Robot 01, center: Robot 02, right: Robot 03).

derstand unexpected events and for robots to communicate the functions to one another in natural language, which leads to emergence of adaptive and cooperative behaviors. In addition, the proposed flow not only improves the performance of handling unexpected events but also promises to facilitate the cooperation with human operators through adopting communication in natural language.

In this paper, we consider the path obstruction for cooperative earthmoving by a group of autonomous construction robots. First, we prepare Robot 01 for autonomous sediment transportation, Robot 02 for manipulating objects, and Robot 03 for sediment excavation. After designing the individual flow for each robot, we set the unexpected scenario where Robot 01 encounters an obstacle on its way to the excavation site. We verify whether such situations can be collaboratively handled by multiple robots, with environmental perception of Robot 01, asking for help from Robot 02, functional understanding of Robot 02, and the actual obstacle removal by Robot 02. Furthermore, we examine the validity of the proposed flow based on the functional expressions and environment understanding by varying the robots' functional expressions and sensor information for the environment.

This paper is organized as follows. In Sec. II, we introduce the related works and discuss the originality of this study. In Sec. III, we design the individual flow and the actual robot system. In Sec. IV, we present the validity of the flow via cooperative adaptation to an unexpected event in the actual experiments, and also discuss the effect of changing the robots' functional expression and sensor information for the environment. In Sec. V, we summarize the paper and discuss future plans.

II. RELATED WORKS

A. Foundation Models for Robotics

Foundation models refer to a set of models trained on broad datasets, adaptable to a wide range of downstream tasks [6], including large language models (LLMs) [9]–[11], vision-language models (VLMs) [9], [12]–[14]. The characteristic of foundation models is the high generalization ability to tasks and environments, a result of pre-training on massive and diverse datasets.

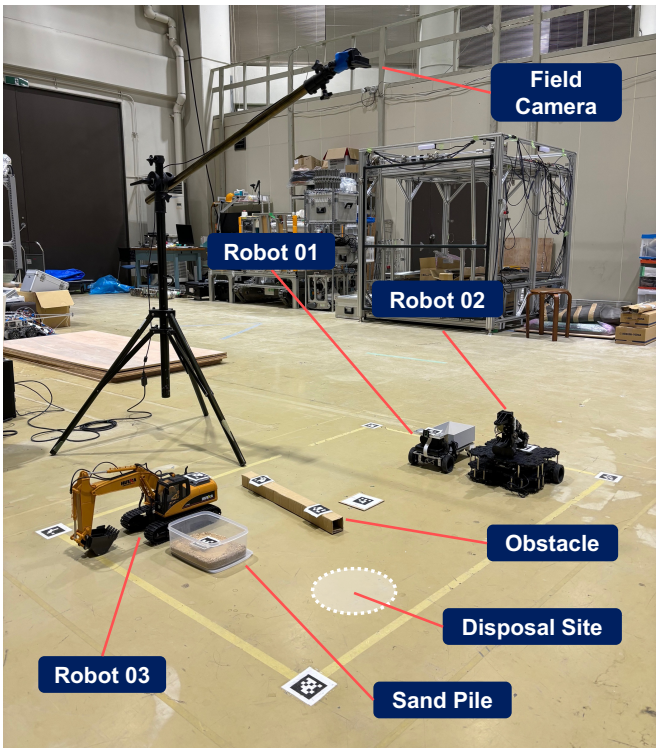

Fig. 4. Experimental setting: an obstacle as a cardboard strip, Robot 01, Robot 02, and Robot 03 positioned within an area, the field camera located at the top of the field.

In recent years, robotic researchers utilized the generalization ability, using LLMs as task planning and perception [15]. For example, RT-1 [16] applies VLMs for recognition and generation of action plans by using camera images and textual inputs.

In addition to the usage of LLMs for task planning and VLMs for perception, we further develop a system that enables cooperative handling of unexpected scenarios, allowing robots to communicate in natural language and adaptive action planning based on the functions of individual and those of others. A cooperative system of multiple robots is also expected to be more parallel and highly adaptive than an individual robot system.

B. Swarm Robotics

Swarm robot systems [17], [18] are inspired by the collective behavior of living organisms to develop a flexible, robust, and scalable multi-robot system. Although prior works have attempted to engender emergent behaviors through the interaction of the robots with each other and with the environment [5], [19]–[21], they have not yet achieved a system that can perform adaptive cooperative behavior in response to unexpected situations. Here, we consider the realization of adaptive and flexible cooperation through the understanding of unexpected situations and the functional expressions of robots, utilizing the generalization ability and high adaptive capability of the foundation models.

While there are several studies on multiple agents applying LLMs to task planning [22], the approaches usually adopt a

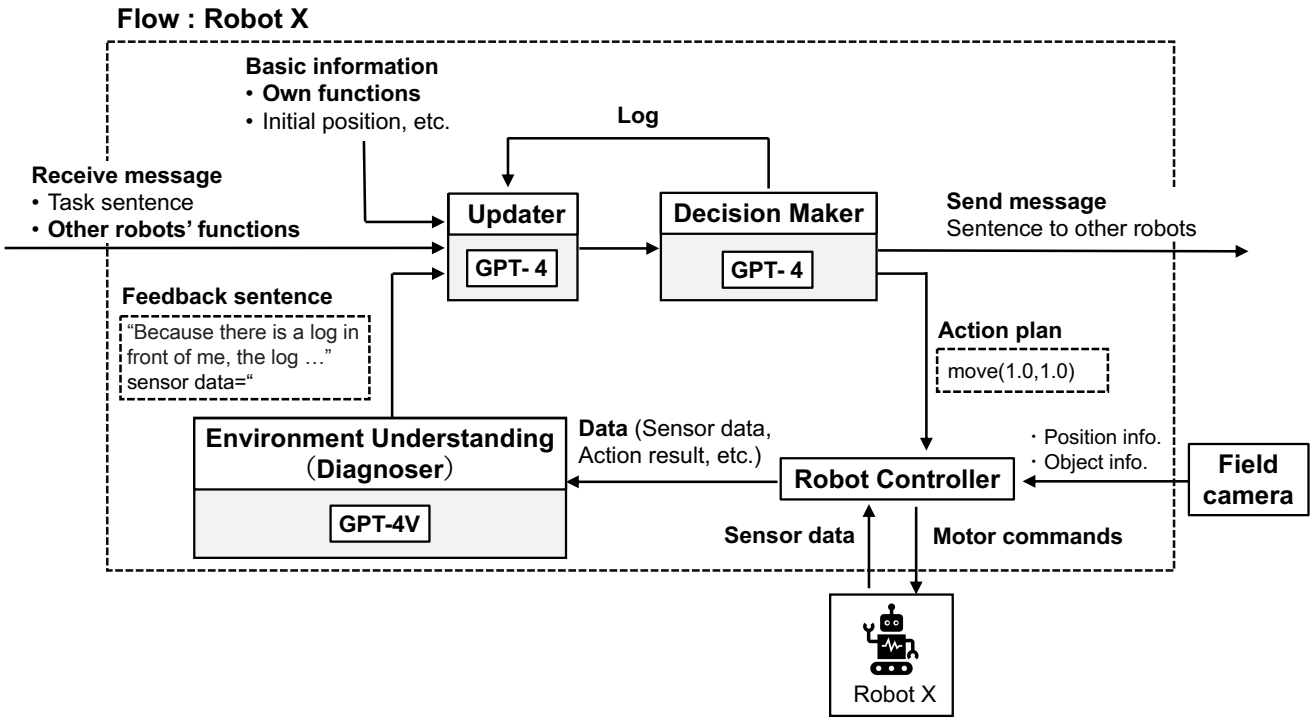

Fig. 5. Individual flow composed of four modules: updater, decision maker, robot controller, and diagnoser. The updater sorts important information from messages received from operators, other robots, and basic information, including feedback results. The decision maker determines actions, such as sending messages to others. The robot controller manages actions according to the action plan, and the diagnoser comprehends the current situation.

centralized structure where a single LLM allocates actions to each agent. Our proposed flow, however, implements an autonomous decentralized manner using foundation models for all robots, as shown in Fig. 1. Although Zhang et al. [23] present a distributed system, our approach additionally emphasizes the embodiment and functionality of the robots. The system also enables us to design scalable systems and to cooperatively adapt to unexpected situations with nearby robots.

III. Development of Actual Robot System and Individual Flow

In this section, we initially describe the development of the actual robots and that of the experimental environment. We also explain the functions of the robots and their sensor information. Subsequently, we design the flow of robots using GPT-4 and GPT-4V. We also describe the four modules in the flow, and the input-output information of each module.

A. Design of the Robots

An overview of the robots is shown in Fig. 3. Robot 01 on the left side in Fig. 3 is a toy robot of the dump truck; Robot 02 in the center of Fig. 3 is a mobile manipulator robot; and Robot 03 on the right side in Fig. 3 is a backhoe toy robot. Robot 02 is based on ROBOTIS TURTLEBOT3 Waffle Pi equipped with OpenMANIPULATOR-X, and Robot 03 is also based on Huina Tongli1580 RC.

Robot 01 is designed with a vessel for loading and disposing soil, as well as moving. Robot 02 is equipped with the manipulator for picking up and releasing objects, as well as moving. Robot 03 is equipped with the bucket for excavating soil. Additionally, Robot 01 is equipped with an ultrasonic sensor, a robot camera, a speaker, and AR markers; Robot 02 is equipped with a speaker and AR markers; and Robot 03 is equipped with AR markers.

B. Construction of the Experimental Environment

An overview of the experimental environment is shown in Fig. 4. The experimental field is a rectangular field with dimensions of 1.0 meters in width and 1.5 meters in length, equipped with a field camera mounted on a tripod. The three robots, a sand pile, and an unknown obstacle are placed in the field. Their positions can be obtained by AR markers pasted to them by the field camera.

Each robot obtains its own position coordinates from the field camera, as well as the names and position coordinates of other robots and objects it can observe, specifically targets within a 120° field of view and up to a distance of 0.7 meters. The information acquired by the field camera will be distributed autonomously by GPS and sensors on each robot in the future.

C. Design of Individual Flow

In this subsection, we design an individual flow mounted on each robot to adapt to unexpected scenarios cooperatively utilizing GPT-4 and GPT-4V. The designed flow is illustrated in Fig. 5. In Fig. 1, Decision making corresponds to the decision maker module, Robot control corresponds to the

TABLE I
SKILL FUNCTIONS

Robot 01 Functions	**Arguments**	**Descriptions**
move	destination	Move to your destination by setting the coordinates of your destination.
dispose_of_soil	coordinate	The coordinates of the place where you want to dispose of the soil.
send_a_message	target, content	Send a message to a robot sharing the same network connection in English.
Robot 02 Functions	**Arguments**	**Descriptions**
move	destination	Move to your destination by setting the coordinates of your destination.
pick	object coordinates	You pick up and hold what is in front of you.
place	object name	You place what you are holding in front of you.
send_a_message	target, content	Send a message to a robot sharing the same network connection in English.

robot controller module, and Environment understanding corresponds to the diagnoser module. We also design the updater module to efficiently arrange the information, considering the constraints of the volume of information that LLMs can process.

The structure of the flow is described in detail. The flow is comprised of four modules: the updater which arranges important information from basic information, received tasks and messages, action results, and the logs sent from the decision maker, the decision maker which determines the robot's action plan based on information sent from the updater, the robot controller which controls the robot action according to the action plan, and the diagnoser which understands its own situation and surrounding environment from the data sent from the robot controller. The decision maker and updater adopt GPT-4 [9]. For the diagnoser, GPT-4V, the vision-language model capable of interpreting image information, is employed to comprehend the surrounding environment [9].

1) Updater: This module is designed to arrange important information from basic information, received tasks and messages, feedback of action results, and the log data of the decision maker. The updater arranges information regarding the basic information of the robot itself and surrounding objects, the current task, and messages from the other robots, including the names and functions. The arranged information is sent to the decision maker.

The basic information includes the characterization of the robot, including its name, functions, and performance, articulated in natural language. This information is given to foundation models to make appropriate inferences for a particular situation or task. We set the basic information for Robot 01 and Robot 02 as follows.

【Robot 01】

- Name: Robot 01
- Organization: Working robot
- Functions: {Loading soil and sand, Discharging soil and sand, Movement}
- Performance: {Movement speed: ◎, Loading of soil and sand: ◎}

【Robot 02】

- Name: Robot 02
- Organization: Working robot
- Functions: {Picking up the object, Placing the object, Movement}
- Performance: {Movement Speed: ◎}

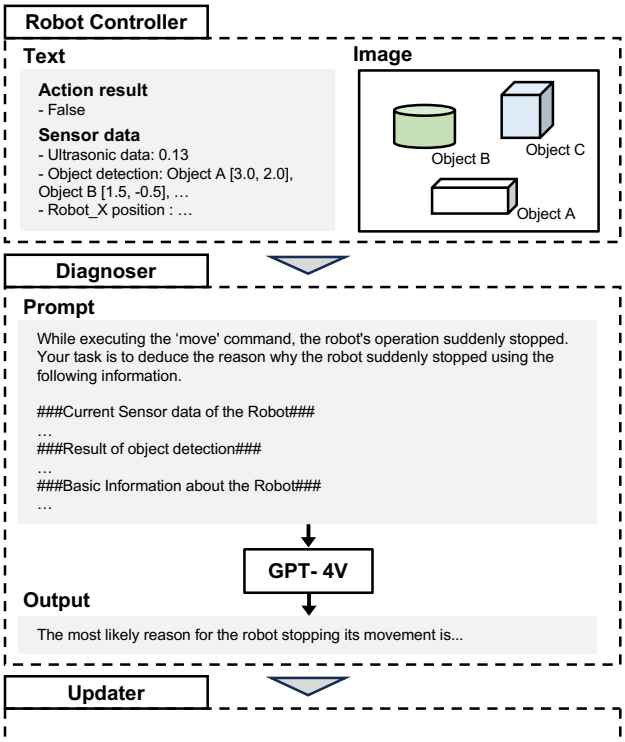

Fig. 6. Detailed explanation of the diagnoser (environment understanding process): Analyzing the reason for the action result being false, using image and sensor data sent from the robot controller.

2) Decision Maker: This module is designed to generate action plans for robots (such as move, send a message). We prepare skill functions, as shown in Table I, corresponding to the functions designed for the robots. A skill function with arguments is output as the robot's action plan.

The process of generating action plans is composed of two steps based on Chain-of-Thought prompting [24], [25]. In the first step, the decision maker confirms the received tasks and messages, environmental information, the functions of the robot itself and those of the other robots, and the log sent from the updater. It then decides whether to execute an action by the robot itself or to communicate with other robots. In the second step, it determines the function and its arguments leveraging function calling of GPT-4. For example, if the robot is assigned a movement task to a specific coordinate (1.0, 1.0), it outputs the function: 'move(1.0, 1.0)', referring to Table I. Along with the framework, the decision maker

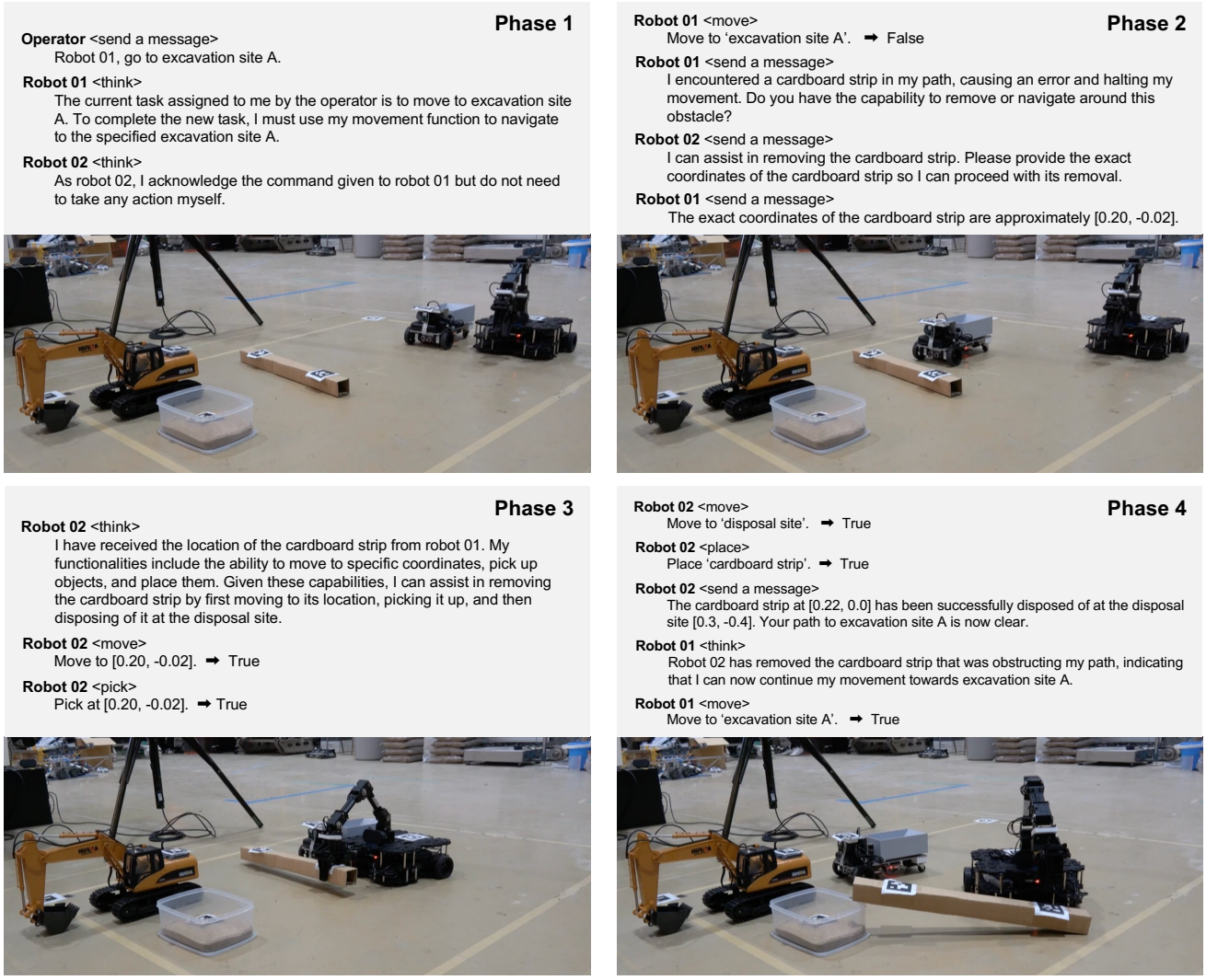

Fig. 7. Experimental verification of adaptive handling in an unexpected scenario: Encountering an obstacle during travel, communicating with Robot 02 for assistance, removing the obstacle by Robot 02, and resuming travel.

generates the action plan for the robots. The log of the thought process is also sent to the updater for the improvement of the decision-making in the future. For the text of the prompt, see Appendix.

3) Robot Controller: This module is designed to execute the skill functions listed in Table I, based on the action plans generated by the decision maker. For example, when the move action plan is selected as an input for Robot 01, a program written in Python is executed to make a path to the destination. A program is also executed to give angular velocity commands to the robot's motors so that the robot follows its path. In the case of the disposal of soil action plan, a program is executed for the robot to operate its vessel and dispose of the soil. When the pick action plan is selected as an input for Robot 02, a program that solves inverse kinematics is executed, enabling the robot to manipulate its manipulator and grasp an object at the specified coordinates. For the place action plan, a program is executed for Robot 02 to operate its manipulator and place the object held by the robot.

Furthermore, the robot controller sends the sensor data of the robot and the action result to the diagnoser. As mentioned in Sec. III-B, the sensor data are the camera images, the ultrasonic sensor value, its own position coordinates, and the object information sent by the field camera. The action result is denoted as either true or false, reflecting the success or failure of the action plan. The robot implements a collision avoidance algorithm for safety. If Robot 01 performs the action plan of move and the value of the ultrasonic sensor is less than a certain value, the robot stops the action and is set to send the false result to the diagnoser.

4) Diagnoser: This module is designed to understand the surrounding environment from the sensor data including the action result from the robot controller. The detailed flow of the diagnoser is illustrated in Fig. 6. If the action result is true, it sends the true message to the updater. In contrast, if the action result is false, it understands the environment using the GPT-4V by integrating the camera images and the sensor

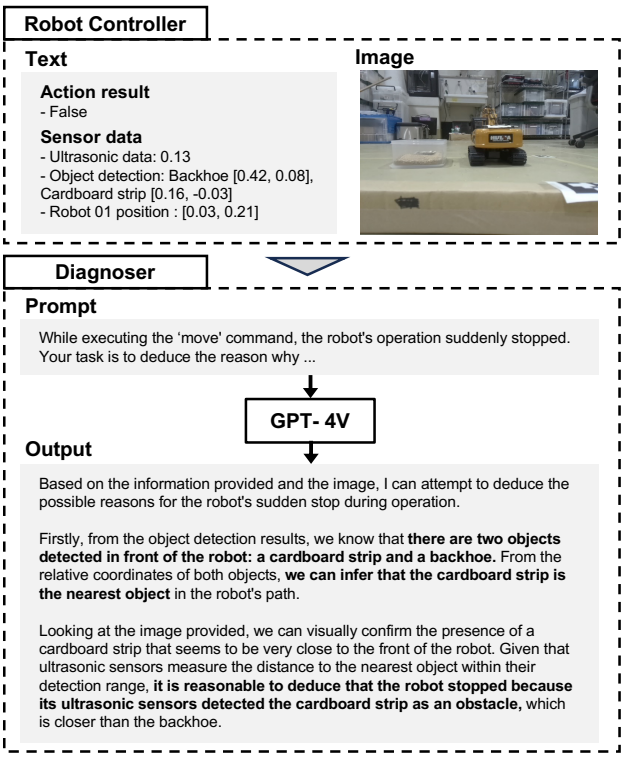

Fig. 8. A result of the diagnoser's understanding process: comprehension of the cause of action being false by summarizing the sensor and image data.

data. For example, if the move action is false, the diagnoser makes inferences from the camera images, the robot's state (action result: false), ultrasonic sensor value, its own position coordinates, and object information in front of the robot. The inference result, such as encountering a cardboard strip in the robot's path that causes an error and halts movement, is sent to the updater.

IV. Experiments and Discussions on Adaptive Coordination

In this section, we verify the proposed flow through experiments. Specifically, we investigate whether the robots can adaptively coordinate to an unexpected situation, such as path obstruction on a travel route, based on the functional expressions of the robots and environment understanding. Furthermore, we examine the validity of the proposed flow by varying the function expressions and the sensor information for the environment.

A. Experimental Setup

In the experiment, we prepared an unexpected scenario where Robot 01 encountered an obstacle on its traveling path to the excavation site A to receive sediment from Robot 03. The obstacle was a cardboard strip. The position coordinates of the excavation site A and the disposal site were set to be given. However, none of the robots knew about the obstacle's existence or its position. Each robot knew only the names of the other robots; the functions, shapes, and additional information were not given.

TABLE II
Experiment Success Rates

Experiment	Success Rate (%)
Experiment 1	80
Experiment 2	0
Experiment 3	40

B. Experimental results on adaptive coordination

Fig. 7 shows an experimental result. In the initial state (phase 1), an operator ordered Robot 01 to proceed to the excavation site A. The robot 01 decided to travel to the excavation site A by using the movement function.

In phase 2, Robot 01 stopped its movement by encountering the obstacle detected by the ultrasonic sensor. This led to the action result being false, triggering the diagnoser to understand the environment. The flow is illustrated in Fig. 8. Firstly, by integrating the action result, ultrasonic sensor information, field camera information, and robot camera information which perceives the cardboard strip and the backhoe (Robot 03) in the front, the robot made decisions that it stopped because its ultrasonic sensor detected the cardboard as an obstacle. Robot 01 then decided to send a message to Robot 02 for help; *I encountered a cardboard strip in my path, causing an error and halting my movement. Do you have the capability to remove or navigate around this obstacle?* Robot 02 replied to Robot 01; *I can assist in removing the cardboard strip. Please provide the exact coordinates of the cardboard strip so I can proceed with its removal.*

In phase 3, Robot 02 removed the cardboard strip after asking for its location. It could be found that Robot 02 planned and executed a series of actions: moving to the location of the cardboard strip, picking it up, and then removing it.

Finally, in phase 4, Robot 02 accomplished the removal of the cardboard strip. Robot 02 communicated to Robot 01 by sending a message to complete the obstacle removal. After receiving the message from Robot 02, Robot 01 restarted its movement towards the excavation site A. Robot 01 had successfully finished the action.

The experimental results indicated that Robot 01 correctly comprehends the situation of action being false. Robot 01 also recognized that it lacked the functions to solve the problem on its own and asked Robot 02 for help. After removing the obstacle by Robot 02's function, Robot 01 completed its travel to the destination. Therefore, we can say that our proposed flow adaptively coordinates to unexpected scenarios based on environment understanding and its own and other robots' functional expressions.

Five more experiments were conducted, and the success rate was 80% as shown in experiment 1 of Table II. Here, we define the success case as Robot 01 traveling to the excavation site A. In the unsuccessful case, we observed incorrect task planning in Robot 02. Robot 02 performed the action plan for pick before moving to the cardboard piece, resulting in the failure in action because the distance

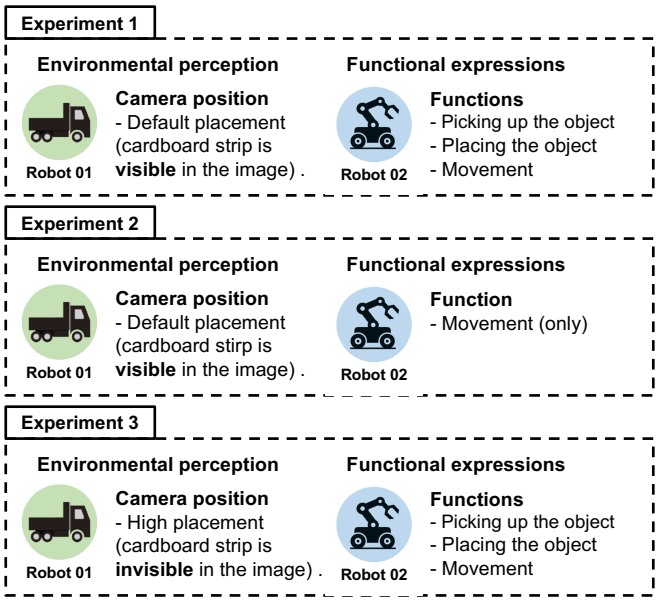

Fig. 9. Comparative experimental setup: experiment 1 represents the baseline setup, experiment 2 modifies the functionality of robot 02, and experiment 3 adjusts the camera position of Robot 01.

to the object was too far. This failure caused the system to mistakenly assume that the cardboard strip was an unpickable object, and robot 02 was unable to remove the cardboard strip, resulting in Robot 01 not reaching the excavation site.

There are several possibilities to improve the success rate by modifying the few-shot learning method [26]. In this study, only the same few-shot learning was performed for the move action in order to keep the conditions the same for Robot 01 and Robot 02. For complicated task planning such as robot 02's obstacle removal (sequence of move, pick, move, and place), we will improve the performance of task planning by applying few-shot learning according to the robot's functions.

C. Effect of functional expressions and sensor information

In this section, we examine the validity of the proposed flow by changing the robot's functional expressions and robot's camera information. With the experiment in the previous subsection as the experiment 1, two additional experiments were conducted under the conditions shown in Fig. 9; the experiment 2 varied the functional expressions of the robots, and the experiment 3 varied the robot's camera information. Experiments were performed five times each, and the success rates were calculated in Table II. By comparing the results, we show the validity of the proposed flow with the consideration of both functional expressions and environment understanding.

1) Effect of functional expressions: To assess the effectiveness of adaptive coordination to unforeseen environments based on the functional expressions, we examined the experiments by changing the functions of Robot 02. By eliminating the function of picking and placing, we conducted experiments five times, identified as the experiment 2, by setting the robot 02 information as follows.

【Robot 02】

- Name: Robot 02
- Organization: Working robot
- Functions: {Movement}
- Performance: {Movement Speed: ◎}

The experimental results are shown in Table II. The experiment 1 performed the success rate of 80%, whereas the experiment 2 performed 0% success rate. In the experiment 2, it was not confirmed that Robot 02 removed the cardboard strip and cleared a pathway. The above results indicate that the explicitly functional expressions of robots are effective in demonstrating the adaptive cooperation.

2) Effect of sensor information: To examine the effect on obstacle detection using the robot's camera, we adjusted the robot's camera 50 mm higher than its original position in the setup of the experiment 1, thereby ensuring the cardboard strip was not detectable, and performed five trials under these modified conditions. These trials, identified as the experiment 3, were then demonstrated in comparison to the experiment 1 to evaluate the effect of environmental information.

The results are shown in Table II. The experiment 1 achieved an 80% success rate, while the experiment 3 decreased to 40%. In the experiment 3, the robot mistakenly judged from the camera image that the backhoe was obstructing its path, resulting in incorrect assumptions. Consequently, the robots 01 and Robot 02 judged that it was difficult to remove the backhoe which resulted in the decrease of the success rate.

The results imply that a lack of comprehensive real-world information (e.g., invisible obstacles) results in incorrect judgments about the environment. Detailed research on sensor information is needed in the future in order to perform effective inference in real environments.

V. CONCLUSIONS

This paper proposed the flow enabling multiple robots to adaptively coordinate to unforeseen situations, leveraging foundation models for functional expressions of each other's robots and environment understanding. After developing the individual flow for adaptive coordination and the robot system, it was confirmed through experiments on actual robots that the robots can cooperatively respond to an unexpected situation in which a travel route is obstructed. Furthermore, we examined the validity of the proposed flow by varying the robots' functional expressions and sensor information for the environment. From the results, it can be expected that the emergence of adaptive cooperative behavior can be achieved by explicitly designing functions into robots.

In the future, we plan to evaluate the versatility and stability of the proposed flow for various scenarios.

APPENDIX

To enhance the reproducibility of our research, we describe the prompts entered into the system. As space is limited, this paper highlights the prompt for the decision maker, crucial for generating collaborative behaviors.

Imagine a scenario where multiple robots work together to accomplish a assigned task. Your task is to think as a robot with the functions that will be described later.

Your name is {robot_name}. {robot_name} can exchange messages and use its own functions. In case of a problem, check the functions of other targets connected to the network, then cooperate to solve the issue. If the assigned task is accomplished, report to the entity that made the request and wait for the next one.

Below is the basic information provided for the accomplishment of this task. It includes information about yourself as well.

###Basic information (JSON format)###
{basic_information}

###Constraint Conditions###
- Firstly, express your thoughts using the following format. Then, based on these thoughts, execute the provided function.
<think>1. Verify the current assigned task (or any assigned task) that you are working on. 2. Organize information about what other robots connected to the network can do, their functions, the current progress of the task, and what you last did. 3. Decide which actions should be taken next. 4. Determine if communication with other robots is necessary, especially for verifying their functions, upon task completion, or when cooperation is needed. </think>

ACKNOWLEDGMENT

This research was partially supported by JSPS KAKENHI Grant Number 21K14183, 21H05104a and JST [Moonshot R&D][Grant Number JPMJPS2032] and JST ACT-X Grant Number JPMJAX22A9.

REFERENCES

[1] K. Nagatani, *et al.*, Innovative technologies for infrastructure construction and maintenance through collaborative robots based on an open design approach, *Advanced Robotics*, vol. 35, no. 11, pp. 715–722, 2021.
[2] B. Sherwood, Principles for a practical moon base, *Acta Astronautica*, vol. 160, pp. 116–124, 2019.
[3] J. Chen, M. Gauci, W. Li, A. Kolling, and R. Groß, Occlusion-based cooperative transport with a swarm of miniature mobile robots, *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 307–321, 2015.
[4] S. Nouyan, R. Groß, M. Bonani, F. Mondada, and M. Dorigo, Teamwork in self-organized robot colonies, *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 695–711, 2009.
[5] L. Garattoni and M. Birattari, Autonomous task sequencing in a robot swarm, *Science Robotics*, vol. 3, no. 20, p. eaat0430, 2018.
[6] R. Bommasani, *et al.*, On the opportunities and risks of foundation models, *arXiv preprint arXiv:2108.07258*, 2021.
[7] P. Liu, Y. Orru, C. Paxton, N. M. M. Shafiullah, and L. Pinto, Ok-robot: What really matters in integrating open-knowledge models for robotics, *arXiv preprint arXiv:2401.12202*, 2024.
[8] M. Chang, *et al.*, Goat: Go to any thing, *arXiv preprint arXiv:2311.06430*, 2023.
[9] OpenAI, Gpt-4 technical report, *arXiv preprint arXiv:2303.08774*, 2023.
[10] H. Touvron, *et al.*, Llama: Open and efficient foundation language models, *arXiv preprint arXiv:2302.13971*, 2023.
[11] A. Chowdhery, *et al.*, Palm: Scaling language modeling with pathways, *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1–113, 2023.
[12] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra, Detecting twenty-thousand classes using image-level supervision, in *European Conference on Computer Vision*. Springer, 2022, pp. 350–368.
[13] A. Radford, *et al.*, Learning transferable visual models from natural language supervision, in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.
[14] J. Li, D. Li, C. Xiong, and S. Hoi, Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, in *International conference on machine learning*. PMLR, 2022, pp. 12 888–12 900.
[15] K. Kawaharazuka, T. Matsushima, A. Gambardella, J. Guo, C. Paxton, and A. Zeng, Real-world robot applications of foundation models: A review, *arXiv preprint arXiv:2402.05741*, 2024.
[16] A. Brohan, *et al.*, Rt-1: Robotics transformer for real-world control at scale, *arXiv preprint arXiv:2212.06817*, 2022.
[17] M. Schranz, M. Umlauft, M. Sende, and W. Elmenreich, Swarm robotic behaviors and current applications, *Frontiers in Robotics and AI*, p. 36, 2020.
[18] E. Şahin, Swarm robotics: From sources of inspiration to domains of application, *International workshop on swarm robotics*, pp. 10–20, 2004.
[19] R. O' Grady, R. Groß, A. L. Christensen, and M. Dorigo, Self-assembly strategies in a group of autonomous mobile robots, *Autonomous Robots*, vol. 28, pp. 439–455, 2010.
[20] M. Rubenstein, A. Cabrera, J. Werfel, G. Habibi, J. McLurkin, and R. Nagpal, Collective transport of complex objects by simple robots: theory and experiments, in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, 2013, pp. 47–54.
[21] M. Rubenstein, A. Cornejo, and R. Nagpal, Programmable self-assembly in a thousand-robot swarm, *Science*, vol. 345, no. 6198, pp. 795–799, 2014.
[22] S. S. Kannan, V. L. Venkatesh, and B.-C. Min, Smart-llm: Smart multi-agent robot task planning using large language models, *arXiv preprint arXiv:2309.10062*, 2023.
[23] H. Zhang, *et al.*, Building cooperative embodied agents modularly with large language models, *arXiv preprint arXiv:2307.02485*, 2023.
[24] J. Wei, *et al.*, Chain-of-thought prompting elicits reasoning in large language models, *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 824–24 837, 2022.
[25] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, Large language models are zero-shot reasoners, *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.
[26] T. Brown, *et al.*, Language models are few-shot learners, *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.