LLM Assistant for heterogeneous multi-robot system dynamic task planning

Miguel Guzmán-Merino[1] and Nils Sören Krause[1]
[1]AI for Sustainable Construction, University of Rostock,
Justus-von-Liebig-Weg 2, 18059 Rostock, Germany
E-mail(s): miguel.merino@uni-rostock.de, nils.krause@uni-rostock.de

Abstract: Multi-robot systems find good implantation in construction applications. The combination of the skills of the robots creates systems with high flexibility. However the combination of these skills is not enough to ensure complex task completion. Task planning needs to face the dynamic behaviors of construction environments. In this paper, a heterogeneous multi-robot system for construction tasks is presented. The system is composed by three robots that present complementary skills and is commanded through natural language requirements, imposing low training requirements for the user. Task planning is managed by a large language model assistant, which receives information about the environment and the state of the system along with the user request. With this information, the assistant provides the execution plan for the robots in a dynamic environment. The system is tested through different situations with real-world tasks, involving extensive interaction with the environment and dynamic planning. Environment recognition, tool manipulation, or obstacle removal are some of the tasks presented in the system.

Keywords: Construction robotics, Human-robot interaction, Large language model, Multi-robot systems, Task planning

Erschienen in Tagungsband 35. Forum Bauinformatik 2024, Hamburg, Deutschland, DOI: 10.15480/882.13508

1 Introduction

Construction environments are complex dynamic scenarios. Constant changes, limited information and unclear references complicate the implantation of robotic systems in construction sites. These problems often required close human assistance during the robotic system performance. Multi-robot systems (MRS) are an alternative to this human interaction. Combining robots it is possible to face situations where single robot systems do not satisfy the requirements. This combination can be between robots with the same features (homogeneous MRS) or different types of robots (heterogeneous MRS). The combination of the robot's skills increases the flexibility of the systems. This flexibility allows MRS to execute complex tasks in dynamic construction environments. Along with the combination of the skills, task planning is an essential part of MRS to address complex tasks. In the task planning process, a complex task is divided into sub-tasks for the robots to perform. Robots coalitions are build if necessary and sub-tasks are allocated to the robots. Environment information and the state of the system are taken into account during the processes. Only through proper task planning it is possible to ensure task completeness. Another important aspect in robotic systems is the human-system interaction. One of the issues included in the human-system interaction is users training requirements. Natural language user-system communication is an interesting solution to this issue. Large Language Models (LLM) can be used to assist in the natural language analysis and understanding. These models also present good performance in information abstraction. Therefore they have been used to face task planning problems. In this paper, a heterogeneous MRS composed by three robots and a LLM Assistant is presented. The LLM Assistant is used to compute the natural language requests from the user and generate the execution plan for the robots to perform. It also takes into account environment information, the state of the system and the robots features. This proposal is detailed in the paper as follows: the related work is presented in Section 2 followed by the description of the robots in the system in Section 3. The system architecture and the integration of the large language model assistant are presented in Section 4. The response of the system in different scenarios is discussed in Section 5. In Section 6, a discussion about the system is presented along with further developments proposals.

2 Related work

MRS present higher flexibility than single robot systems. That makes them more suitable for construction applications. Collaboration between robots allows for instance the exploration of tasks in large working areas [1], [2]. In [3] it is possible to remove the scaffolding process thanks to the collaboration between robots. This task could not be performed by a single robot system. Task planning is an important part of MRS. In [4] they present the main parts that integrate the task planning process as (i) task decomposition, the division of a complex task into sub-tasks for the robots to perform, (ii) coalition formation, to build teams dedicated to the same sub-task, (iii) task allocation, where sub-tasks are assigned to the appropriate robots or team of robots, and (iv) task execution, where the robots in the system execute the sub-task to complete the task. As it will be explained in Section 3, the actions proposed in this paper are single robot actions, therefore coalition formation is not considered. Several solutions for the task planning problem can be found in the literature. Solutions for optimal automatic task decomposition [5], [6] or task allocation [7], [8] focus on one of the parts of the task planning process. However, due to the high interconnection of the parts into the task planning process, studies like [9] target task allocation and planning problems simultaneously. Natural language processing approaches were proposed in [4] to solve the task planning problem. This approach has been successfully implemented in [10], [11]. Common house activities are studied in simulation environments in these papers. In [10] a LLM is trained to solve the task planning problem. They use a heterogeneous robot team to test the generated plans. Compound tasks are presented as the execution of the same task using different agents. In [11] they do not train the LLM previously and only a single agent perform the actions. It is stressed that the untrained LLM can develop accurate plans. In both studies a deterministic environment is presented, where the location of the objects is known and there are no obstacles during the plan execution. The requests are expressed as natural language high-level tasks, implying low or no user training. User training has been identified as one of the challenges for the implantation of technology in construction [12], [13]. The benefit of natural language interaction is demonstrated in [14]. The system identifies the best tools for specific tasks using a probabilistic graph reasoning approach based on the weights derived from the language and computer vision model. The user can interact with the system using natural language prompts, implying no previous user training. In this paper a LLM Assistant is used to solve the task planning problem in non-deterministic scenarios. With this approach it is intended to study whether LLM can be used for dynamic task planning.

3 Robot system

The proposed system is intended to perform in dynamic environments with no or little previous information. The selection of the robots for the system is based on the combination of their features and the tasks they are likely to perform. A **legged robot** is proposed for environment recognition, remove obstacles and interact with the environment. These robots are appropriate for navigation through irregular terrains like the ones in a construction site. Sensors like RGB camera and LiDAR as well as a manipulator need to be mounted on the robot. A **mobile manipulator** is proposed to perform power-demanding tasks. Tasks like drilling or painting are not suitable for the legged robot manipulator. A mobile manipulator can reach the target location or relocate the robot arm during task execution. To ensure safe navigation, the mobile manipulator platform needs to integrate sensors like LiDAR or depth cameras. Finally a **mobile platform** is proposed for supplies transportation. An extra mobile platform adds flexibility to the system. By using and extra mobile platform it would be possible to perform parallel actions. For example, while the mobile manipulator performs an action in a working area, the mobile platform could travel back and forth to collect supplies. Navigation sensors would also be included in the mobile platform. Table 1 shows the robots' sub-tasks included in the current research. Each sub-task is composed by a robot and information about locations and/or objects. These sub-tasks are used by the LLM Assistant to generate the execution plan. The sub-task implementation is not discussed in this paper.

Table 1: Robots' sub-tasks integrated in the system. Each sub-task specifies the robot that performs it and information about locations or objects

Robot	Sub-tasks
Legged robot	Move (LeggedRobot, location1, location2), Find (LeggedRobot, location), Remove (LeggedRobot, object)
Mobile manipulator	Move (MobileManipulator, location1, location2), Paint (MobileManipulator, location), Drill (MobileManipulator, location)
Mobile platform	Move (MobilePlatform, location1, location2), Carry (MobilePlatform, object)

4 System architecture

There are different sources of information in the system. Figure 1 shows how these sources are integrated in the LLM Assistant to generate a feasible execution plan. Assuming resting state, a execution starts with a user request like: "Paint the left wall in the room number 3", "Drill a hole in the right wall of the office, 1 metre from the floor and 2 meters from the window" or "Sand the floor of the room number 2 and bring paint from the storage room". The request is provided to the LLM

Assistant along with the Environment Information, the Robots' sub-tasks from Table 1 and the System State. The Robots' sub-tasks do not change during the execution. However, the System State and the Environment Information can be updated. The System State includes the information about the location of the robots, or whether they are available. Another kind of information, not included in the current discussion, could be the battery level of the robots. This information is not considered in the current discussion. It is possible to have full, partial or no environment information. This information can proceed from Building Information Models (BIM) or from previous executions. GPT4o is used to implement the LLM Assistant without previous training. The performance of untrained LLM for task planning in deterministic scenarios was tested in [11]. Here the LLM Assistant would be used to solve the task planning problem in non-deterministic scenarios. The Environment Information, the Systems State and the Robots' sub-tasks are provided to the LLM Assistant as context information. No examples of task decomposition or allocation are provided to the LLM Assistant. Some guidance about how to handle the information and how to generate the plans are also part of the context of the LLM Assistant. Once the LLM Assistant has all the context information, the user request is provided. The Assistant's response is an ordered list of sub-tasks. The sub-tasks include information about the robots that perform them. This ensures proper task allocation. The robots updated their state in the system and the environment information during the plan execution. For example, the environment information could be updated with semantic information captured with the Legged Robot camera. It is also possible to update information like the location of an obstacle. The updated information is then communicated to the LLM Assistant. The Assistant generates a new plan if the new information demonstrates that the original plan is no longer feasible. This is how the system captures dynamic changes and how these changes modify the performance of the robots. The scenarios presented in the next section would help to clarify how the system architecture works.

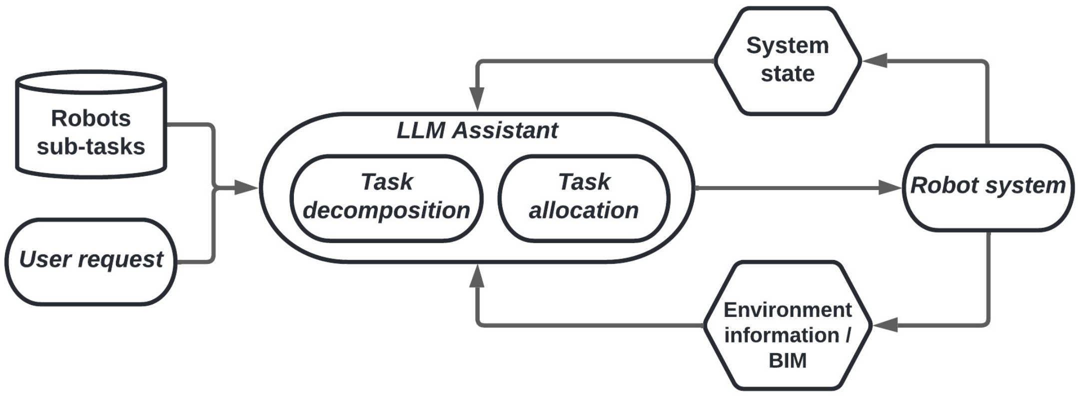

Figure 1: System architecture. The LLM Assistant generates the execution plan for the robots to perform. To do it the Environment information, the System state and the Robots' sub-tasks are provided to the LLM Assistant along with the User request.

5 System analysis

Three different scenarios are proposed to test the system performance. Sub-tasks from Table 1 are common to all the scenarios and do not change. The environment information is treated here as locations information and no time variable is assumed. Each step in the execution plan starts and ends with the corresponding sub-task execution. It is intended to analyse the system performance with the following scenarios: (i) There is partial knowledge of the environment and the state of the system is updated. (ii) An obstacle does not allow the task completeness. (iii) A situation with a parallel sub-tasks execution is proposed.

5.1 Partial known environment

In the scenario proposed in Table 2 there is no information about the "left wall in the room 3" in the Environment information. A Find() sub-tasks is proposed in the point 2 of the Plan 1 to find the missing location before the Paint() sub-task. The context provided to the LLM Assistant is good enough to generate a logic sub-tasks sequence. Following Plan 1 it is possible to complete the User request. The update of the System State interrupts the execution of the plan. The LLM Assistant generates the Plan 2 in response. It is proposed to find the Paint location in the Store location in steps 1-3. Steps 4-6 describe how the Legged Robot takes the paint can from the Paint location and how the Mobile platform carries it. Steps 7-11 describe the replacement of the paint in the Mobile manipulator. The second plan shows the main flow needed to complete the task, but it is possible to detect some missing information or wrong steps. In Step 4 the Legged robot removes the paint can for the paint can location, but there is no step to put the can in the Mobile platform. In the step 11, the structure of the sub-task does not correspond with the right sub-task structure. Although Plan 2 is not fully correct, some interesting facts should be taken into account.

Table 2: Partial known environment scenario. The Room 3 left wall location is missing and an update of the system state forces the generation of a new plan.

Environment information	Room 1 location, Room 3 location, Room 3 right wall location, LeggedRobot location, MobileManipulator location, MobilePlatform location, Store location
System State	LeggedRobot(operative), MobileManipulator(operative), MobilePlatform(operative)
User request	Paint the left wall in the room 3
Plan 1	**1**. Move (LeggedRobot, LeggedRobot location, Room 3) **2**. Find (LeggedRobot, left wall) **3**. Move (MobileManipulator, MobileManipulator location, Room 3) **4**. Paint (MobileManipulator, left wall)
System State	Low paint level
Plan 2	TASK: Refill paint for MobileManipulator Subtask List: **1**. Move (LeggedRobot, LeggedRobot location, Store location) **2**. Find (LeggedRobot, Paint location) **3**. Move (LeggedRobot, Store location, Paint location) **4**. Remove (LeggedRobot, Paint can) **5**. Move (MobilePlatform, MobilePlatform location, Paint location) **6**. Carry (MobilePlatform, Paint can) **7**. Move (MobilePlatform, Paint location, Room 3 left wall location) **8**. Move (LeggedRobot, Paint location, Room 3 left wall location) **9**. Remove (LeggedRobot, Empty paint can from MobileManipulator) **10**. Move (LeggedRobot, Room 3 left wall location, Store location) **11**. Carry (MobilePlatform, Full paint can to MobileManipulator)

(i) The LLM Assistant keeps track of the previous events. For instance, the Legged robot and the Mobile platform are moved to the "Room 3 left wall location" where the Mobile manipulator was painting. (ii) There is a coherent sequence of actions. The dependency between sub-tasks is taken into account during the plan. For instance, it is planned to remove the empty can before placing the full one. (iii) Sub-tasks are correctly assigned to robots. The legged robot is used to remove the paint can and the mobile platform is used to carry it.

5.2 Unknown obstacle

Table 3 presents a scenario where the request contains the same information as in Table 2, but presented in a different way, "Paint room 3 left wall". The Room 3 left wall location is included in the environment information in this case. The first plan generated by the LLM Assistant does not take into account whether there is a door in the way or not. Just when the Mobile Manipulator finds an obstacle and updates the environment information, Plan 2 is generated in response. Find() and Remove() sub-tasks refer to the Room 3 door in this plan. It is interesting to stress that the updated information does not specify location about the door. In spite of Plan 2 execution completes the user request, an error is presented. In the parallel sub-task proposed in step 4, there is no need of moving the Legged Robot to the room 3 left wall location after the obstacle was removed.

Table 3: Unknown obstacle scenario. All the information about environment is known. The environment information is updated when an obstacle is found and a new plan is generated in response.

Environment information	Room 1 location, Room 3 location, Room 3 right wall location, Room 3 left wall location, LeggedRobot location, MobileManipulator location, MobilePlatform location, Store location
System State	LeggedRobot(operative), MobileManipulator(operative), MobilePlatform(operative)
User request	Paint room 3 left wall.
Plan 1	**1**. Move (MobileManipulator, MobileManipulator location, Room 3 location) **2**. Move (MobileManipulator, Room 3 location, Room 3 left wall location) **3**. Paint (MobileManipulator, Room 3 left wall).
Environment information	Door closed.
Plan 2	**1**. Move (LeggedRobot, LeggedRobot location, Room 3 location) **2**. Find (LeggedRobot, Door location) **3**. Remove (LeggedRobot, Room 3 door) **4**. Move (MobileManipulator, MobileManipulator location, Room 3 location) // Move (LeggedRobot, Room 3 location, Room 3 left wall location) **5**. Move (MobileManipulator, Room 3 location, Room 3 left wall location) **6**. Paint (MobileManipulator, Room 3 left wall).

5.3 Parallel sub-task execution

The requested task in Table 4 includes two sub-tasks that can be executed at the same time. All the locations are known in this scenario and the Legged robot is involved in a Find() sub-task. In Table 4 Plan 1, the Carry() and the Move() sub-tasks are executed at the same time. The Mobile platform carries paint when moving to the room 1. Nevertheless the main parallels sub-tasks are not properly planned. The main idea of parallel sub-task is to execute actions without dependencies at the same time. In this scenario the robots could move simultaneously to the desired locations and perform the Drill() and Carry() sub-tasks in parallel. An example of the expected plan is proposed at the end of Table 4.

Table 4: Parallel sub-task execution scenario. The user request includes two tasks that can be executed in parallel.

Environment information	Room 1 location, Room 3 location, Room 3 right wall location, LeggedRobot location, MobileManipulator location, MobilePlatform location, Store location
System State	LeggedRobot(finding), MobileManipulator(operative), MobilePlatform(operative)
User request	Drill a hole room 3 right wall and bring paint to room 1.
Plan 1	**1**. Move (MobileManipulator, MobileManipulator location, Room 3 right wall location) **2**. Drill (MobileManipulator, Room 3 right wall location) **3**. Move (MobilePlatform, MobilePlatform location, Store location) **4**. Carry (MobilePlatform, paint) // Move (MobilePlatform, Store location, Room 1 location)
Alternative plan (not generated by Assistant)	**1**. Move (MobileManipulator, MobileManipulator location, Room 3 right wall location) // Move (MobilePlatform, MobilePlatform location, Store location) **2**. Drill (MobileManipulator, Room 3 right wall location) // Carry (MobilePlatform, paint) **3**. Move (MobilePlatform, Store location, Room 1 location)

6 Conclusion

In this paper a heterogeneous MRS assisted by a LLM for task planning has been presented. The proposed robots for the system and their features have been discussed. It has also been shown how the proposed system architecture can handle multiple information sources. The analysis of the scenarios demonstrates that LLM Assistants are good candidates to solve dynamic task allocation and planning problems. Although the proposed plans are not accurate enough, interesting features has been exposed. The LLM Assistant can dynamically elaborate plans while keeping previous relevant information into account. The generated plans also present a logic sequence of action. Further development like include new sub-tasks into the robots would help to generate more accurate plans. It is also proposed the implementation of reinforcement learning methods in combination with the LLM Assistant. The system would also be recreated in simulation. A more precise analysis of the system will be possible including time variable and the implementation of sub-tasks.

References

[1] X. Zhang, M. Li, J. H. Lim, *et al.*, “Large-scale 3d printing by a team of mobile robots”, *Automation in Construction*, vol. 95, pp. 98–106, Nov. 2018. DOI: https://doi.org/10.1016/j.autcon.2018.08.004.

[2] A. Adel, “Co-robotic assembly of nonstandard timber structures”, *42nd Annual Conference of the Association for Computer Aided Design in Architecture, ACADIA 2022*, pp. 604–613, Oct. 2023.

[3] S. Parascho, I. Han, S. Walker, A. Beghini, E. Bruun, and S. Adriaenssens, “Robotic vault: A cooperative robotic assembly method for brick vault construction”, *Construction Robotics*, vol. 4, pp. 117–126, Dec. 2020. DOI: 10.1007/s41693-020-00041-w.

[4] Y. Rizk, M. Awad, and E. W. Tunstel, “Cooperative heterogeneous multi-robot systems: A survey”, *ACM Computing Surveys (CSUR)*, vol. 52, no. 2, pp. 1–31, Apr. 2019. DOI: 10.1145/3303848.

[5] B. Fu, W. Smith, D. Rizzo, M. Castanier, M. Ghaffari, and K. Barton, "Robust task scheduling for heterogeneous robot teams under capability uncertainty", *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 1087–1105, Apr. 2023. DOI: 10.1109/TRO.2022.3216068.

[6] A. Messing, G. Neville, S. Chernova, S. Hutchinson, and H. Ravichandar, "Grstaps: Graphically recursive simultaneous task allocation, planning, and scheduling", *The International Journal of Robotics Research*, vol. 41, no. 2, pp. 232–256, Nov. 2021. DOI: 10.1177/02783649211052066.

[7] F. Quinton, C. Grand, and C. Lesire, "Market approaches to the multi-robot task allocation problem: A survey", *Journal of Intelligent and Robotic Systems*, vol. 107, no. 2, pp. 1–29, Feb. 2023. DOI: 10.1007/s10846-022-01803-0.

[8] Y. Emam, S. Mayya, G. Notomista, A. Bohannon, and M. Egerstedt, "Adaptive task allocation for heterogeneous multi-robot teams with evolving and unknown robot capabilities", *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7719–7725, Sep. 2020. DOI: 10.1109/ICRA40945.2020.9197283.

[9] F. Faruq, D. Parker, B. Laccrda, and N. Hawes, "Simultaneous task allocation and planning under uncertainty", Jan. 2019, pp. 3559–3564. DOI: 10.1109/IROS.2018.8594404.

[10] S. S. Kannan, V. L. N. Venkatesh, and B.-C. Min, "Smart-llm: Smart multi-agent robot task planning using large language models", *ArXiv*, 2024.

[11] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents", *International Conference on Machine Learning*, pp. 9118–9147, Mar. 2022.

[12] M. Yahya, Y. Hui, A. Md. Yassin, R. Omar, R. Robin, and N. Kasim, "The challenges of the implementation of construction robotics technologies in the construction", *MATEC Web of Conferences*, vol. 266, p. 05 012, Jan. 2019. DOI: 10.1051/matecconf/201926605012.

[13] M. S. R. Espinoza, M. A. C. Contreras, K. M. A. Estela, C. J. N. Varillas, and G. F. R. Torpoco, "Robotics in building construction: Advantages and barriers", in *2023 Congreso Internacional de Innovación y Tendencias en Ingeniería (CONIITI)*, Oct. 2023. DOI: 10.1109/CONIITI61170.2023.10324131.

[14] L. Fischer, S. Hasler, J. Deigmoeller, *et al.*, "Which tool to use? grounded reasoning in everyday environments with assistant robots", *CogRob@ KR*, pp. 3–10, 2018.