2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

ROSMC: A High-Level Mission Operation Framework for Heterogeneous Robotic Teams

Ryo Sakagami[1], Sebastian G. Brunner[1,2], Andreas Dömel[1], Armin Wedler[1], and Freek Stulp[1]

***Abstract*— Heterogeneous teams of multiple mobile robots will be important for future scientific explorations of extraterrestrial surfaces or hazardous areas. Mission operation in such harsh, unknown environments poses diverse challenges. Robots need to cooperate autonomously due to the large network latency to the ground station while operators need to adapt the ongoing mission flexibly based on new discoveries obtained during execution. Furthermore, shared situational awareness between operators and roboticists is highly required to deal with execution failures promptly. To overcome these challenges, this paper proposes the high-level mission operation framework ROSMC. The concept of mission synchronization to robots enables continuous mission adaptations and future planning by operators while robots execute the mission autonomously. The ROS-based GUIs enable operators to intuitively create and monitor the mission for robots as well as to communicate with roboticists smoothly. The proposed framework was evaluated by a pilot study with a simulator and demonstrated at a Moon-analogue field on Mt. Etna in Sicily, Italy, involving 3 robots and around 70 researchers for 4 weeks.**

I. Introduction

In the field of scientific exploration in harsh environments or extraterrestrial surfaces, the use of heterogeneous teams of robots systems has several advantages over a single robotic system, including *efficiency*, *robustness*, and *versatility*. Because a team of robots can execute skills in parallel, the total amount of time consumption for a mission is reduced (*efficiency*). Even if one of the robots in a team suffers from an accident, other robots with different but overlapping capabilities could substitute the one with malfunction (*robustness*). Finally, a combination of heterogeneous robots such as ground and aerial vehicles enables a mission that is not achievable by a single robot (*versatility*). Due to these advantages, we expect a heterogeneous team of robots to improve the quality and the speed of scientific explorations.

To orchestrate a heterogeneous robotic team, this paper proposes the open-sourced tool ROS Mission Control (ROSMC)[1], a high-level mission operation framework based on ROS [1]. Several technical challenges are overcome by ROSMC. A first one is that mission execution by robots should be autonomous due to the large network latency. In Mars exploration scenarios, for example, the communication delay can be up to 20 minutes. Under such situations, step-by-step human interventions are time-consuming and hinder efficient mission execution. Therefore, robots should demonstrate self-reliance and be able to execute mission specifications without human support. Most importantly, the mission specification should exploit collaboration to take advantage of the *versatility* of heterogeneity.

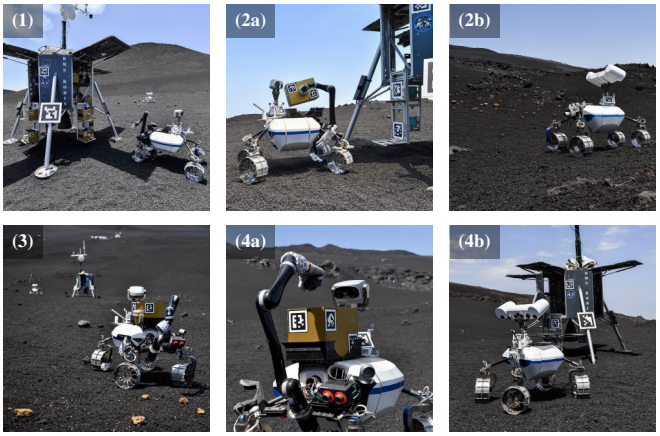

Fig. 1: Snapshots of the real sample-return mission at a Moon-analogue site on Mt. Etna. (1) LRU1 [2], [3] and LRU2 [2], [4] started next to the lander; (2a) LRU2 picks the sample box from the lander while (2b) LRU1 identifies a POI for sample return; (3) LRU2 navigates to the POI while LRU1 explores an unknown area; (4a) LRU2 collects a stone while (4b) LRU1 drives back to the lander and improves map quality by loop closures.

A second challenge, which poses a contradictory requirement to the first challenge, is that the ongoing mission should be flexibly adaptable to maximize scientific achievements. New, unforeseeable scientific opportunities often emerge from the in-situ discoveries and they need to be flexibly incorporated into the original mission.

These two contradicting challenges are tackled by a *mission synchronization* concept. This specifies the entire mission by two sections: the non-synchronized one the operators prepare as a blueprint and the synchronized one the robotic team autonomously executes. This concept enables seamless adaptations on the synchronized section while planning future tasks on the non-synchronized section.

The prioritization of scientific activities and mission specification should be done by experts in the exploration domain, e. g. geologists. A third challenge is that interfaces should thus be intuitive to use also for operators without a robotics background. Furthermore, in case of a critical robotic failure, the operators should be able to communicate with roboticists effectively which part of the mission caused the problem.

This challenge is addressed by graphical user interfaces (GUIs) that enable intuitive mission creation and monitoring. The noticeable advantage of our framework is that the GUIs

[1] DLR (German Aerospace Center), Institute of Robotics and Mechatronics, Münchner Str. 20, 82234 Wessling, Germany, {firstname.lastname@dlr.de}. [2] At DLR at the time of development of ROSMC.

[1] https://github.com/DLR-RM/rosmc

are based on the ROS architecture, decreasing the communication burden between the operators and the roboticists.

We demonstrated ROSMC with a pilot study using a simulator and deployed it with the actual system during the ARCHES demo mission [5]: the exploration scenario on a Moon-analogue terrain with two different rovers, an aerial drone, a lander, and multiple payload boxes (see Figure 1).

II. Related Work

ROSMC falls into the research field of *mission specification*, how a mission can be defined for multiple robots including their coordination, and *a graphical tool for mission creation and monitoring*, the interface for operators to monitor a robotic team and transfer their intentions to it.

A. Mission Specification

MissionLab [6], [7] specifies a mission as a set of possible states, while M2PEM [8], [9] does as a flow of activities. In both approaches, each mission block (a state or an activity) is assigned to one or several robots. While it renders the whole mission easy to grasp, the execution sequence of each robot is obscured and flexible mission adaptation for a specific robot becomes less tractable.

In SeMo [10], a mission for the team has different modes, each of which has listen, report, and action plans to enable multitasking. Each plan hierarchically consists of services that robots offer. Unlike ROSMC, actual code needs to be compiled after each modification.

A recent formal language approach to specify skills and missions for a single robot can also be found in [11], although scalability for multiple robots is not discussed.

PROMISE [12] specifies a mission in a domain-specific language translatable into a temporal logic. The mission hierarchically consists of basic robotic tasks provided by PsALM tool [13]. A global mission consists of the local mission per robot. Cooperations are specified by event triggers.

Our ROSMC specifies an entire mission as a composition of local missions per robot, similar to PROMISE. Contrary to it, the specification of concurrency, fallbacks, and event handlers is handled by robotics experts with a task programming tool interfacing with ROSMC.

B. Graphical Tools for Mission Creation and Monitoring

The GUI of MissionLab [6] represents states and their transitions by boxes and arrows, respectively. Choosing suitable triggers and transitions yet requires robotics knowledge, restricting the GUI mainly for roboticists.

M2PEM [8] also provides a graph-based mission editor employing special graphical operators to construct logics. MOCU [14] is integrated with it to increase situational awareness with its 2D map, a telemetry table, and a pop-up window triggered by events that robots encounter.

We can find outstanding interfaces in CHMI^2 [15], [16] and IMPACT [17], [18]. Both provide a 2D tactical map displaying robots as icons on a satellite image. CHMI^2 shows the mission progress in a timeline. In IMPACT, active high-level tasks are highlighted by corresponding waypoints. Furthermore, IMPACT employs easy-to-understand icons to show the current robot status on the map.

To the best of our knowledge, however, none of the above provides visualization of local 3D environments and possibility to interact with objects around robots. This feature is essential for specifying and monitoring manipulation tasks.

III. Framework Concepts

The requirements derived from the challenges described in Section I are summarized as follows; 1) the operators should be able to flexibly adapt the uploaded mission while making future plans, 2) the mission should be uploaded to the robots for autonomous execution, 3) the mission specification can exploit inter-robot collaboration, and 4) even non-roboticists should be capable of all the mission operation activities. In this section, we describe the concepts employed in ROSMC to satisfy these requirements.

A. Mission Specification

In robotic space operation, the primary interest of the operators is to determine scientific activity plans. From this fundamental aspect, we specify a mission as a combination of skills rather than as a collection of robotic states whose transitions are triggered by external events. The state-orientated specification also has practical limitations of the intractably large space of states and events in unknown environments.

The mission data structure is shown in Figure 2. The entire *mission* of the robotic team consists of single-robot missions (*tasks*), each of which consists of a sequence of parameterized high-level *skills*. The skill provides a functional interface to the actual execution routines/controllers programmed by the robotics experts. The mission operators instantiate these skills with *parameters* (bool, integer, float, string, list, dictionary, and euler 6D pose) and use them to compose the task. A *library* associates a *robot* with the skills that it can execute.

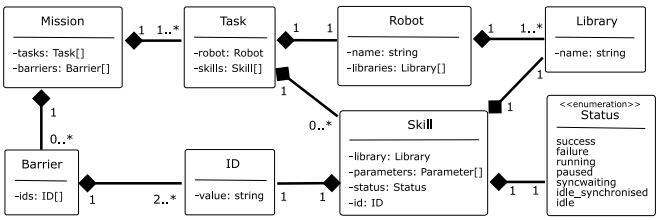

Fig. 2: Simplified class diagram of the mission data structure.

For skill specification, we refer to RAFCON [19] for its rich model. Skills are modeled as hierarchical, concurrent state machines to achieve complicated autonomous decision-making behaviors for robots. Due to its hierarchical structure, skills can be modularized in an arbitral granularity, ranging from the lowest level to trigger capabilities to the highest one to interface with ROSMC. The concept of Resource-Aware Task Nodes [20], which models resources, pre-conditions, and effects, enables optimization of skill execution. Provided with these concepts, the skills could implement local autonomy to tackle task-level challenges on multi-robot systems. This exempts operators from mission-irrelevant management

workload such as avoiding collisions with other robots or sharing manipulated object information among robots.

ROSMC allows only a sequential order of skills, meaning that the operators cannot create branching nor loops in the mission specification. This contributes to simplifying the mission specification to satisfy the fourth requirement while keeping the disadvantage minimal. In the scientific exploration domain, decision-makings are often based on various, non-programmable domain knowledge and thus branching does not benefits the mission specification. In case looping is necessary (e. g. collect 3 stones), exposing the iteration number as an input parameter of the skill should suffice.

B. Mission Synchronization

ROSMC employs the *mission synchronization* concept, illustrated in Figure 3, to address the first and second (contradictory) requirements. This concept is to upload/delete the mission to/from the robots part by part. With this concept, the entire mission is specified by two sections: the synchronized one the team of robots autonomously executes being followed by the non-synchronized one for operators to prepare a blueprint.

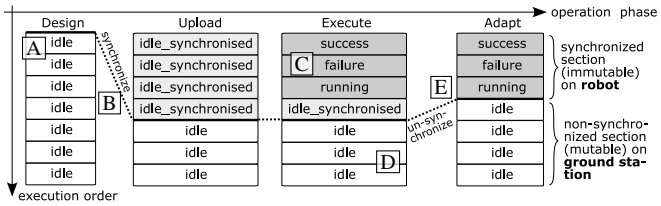

Fig. 3: Graphical example of the mission synchronization concept.

A mission is initially created all as the non-synchronized section at the ground station A. The operators then choose the first set of skills and synchronize this section to the robot B. Since the synchronized section is autonomously executed by the robots C under the network latency, this part is protected against modifications by the operators. The non-synchronized section, however, remains on the ground station and still editable while execution D.

This scheme of uploading a mission and planning next activities is common practice in space-robot operations [21]. What is unique to ROSMC is that modifications on the synchronized section can be performed seamlessly with future planning. As long as the synchronized skills are not yet executed, the operators can make a request (named *unsynchronization*) to delete them from the robot and prepend them at the beginning of the non-synchronized section E. In this way, the operators can easily check the consistency of the mission while adapting the unsynchronized skills and creating the future ones simultaneously. Furthermore, the operators do not suffer from the network latency except for the moment of the unsynchronization request, increasing the efficiency of mission adaptations.

This concept is achieved by the *status* of skills in the mission specification. The skills with the *idle* status are considered as the non-synchronized section and the rest as the synchronized one. The skills with *success* and *failure* are completely immutable since they are execution logs. The operators can stop the execution if the skill is *running*, *paused*, or *synchwaiting*. The stopped skill and those synchronized but not yet executed have the status *idle_synchronized*, which can be unsynchronized by request.

One of the planetary exploration scenarios where this concept is effective is that the sample acquisition destination is revealed to be not traversable by a rover after arriving nearby. Since orbital images are not satisfactory to predict the exact terrain conditions [22], the initial plan could not guarantee scientific activities at the planned location. The mission could be adapted in various ways: by updating the targeted location, by inserting other scientific activities than sample acquisition, or by skipping the location and letting the rover proceed with the rest of the mission. In either case, the mission synchronization concept provides flexibility for operators to reflect their intentions swiftly.

C. Skill Barrier

A *skill barrier* specifies which skills should be executed synchronously, satisfying the third requirement. As shown in Figure 2, a mission contains *barriers* in addition to the tasks. Each barrier consists of more than one *IDs* uniquely given to instances of the skills. The skills whose IDs are specified in the barrier are meant to be triggered at the same time. Although each robot executes the synchronized skills in the task self-reliantly, it always checks if the ID of the skill is included in any barrier. If yes, the robot awaits the other required robots until they are ready to execute the skill. Only after all the robots specified by the skill barrier are ready, they trigger the skill execution. The synchronization *during* the skill execution should be dealt with by the functional layer [23] or the world model, i. e., should be programmed by the robotics experts.

IV. Implementation

The implementation overview is shown in Figure 4. The mission server maintains data and communicates with the 3D GUI, the command GUI, and the robots. RAFCON [19] is employed for skill implementation and execution on the robots. A world model [4] allows the robots to exchange world knowledge and to collaborate during skill execution.

The system boundary is designed so that the operators and the roboticists could work on different architectural layer. This could potentially allow for a different paradigm of robotic space operation, e. g., letting astronauts on the Mars orbit be operators being supported by roboticists on Earth.

A. Mission Server

The mission server communicates mission data via ROS services so that the clients are informed in case of communication failures and able to rollback to consistent states safely[2]. The mission server could also be a static agent in our multi-robot SLAM framework to compute the map and

[2]Although ROS1 is used for prototypical development, migration to ROS2 with more reliable communication backend is planned.

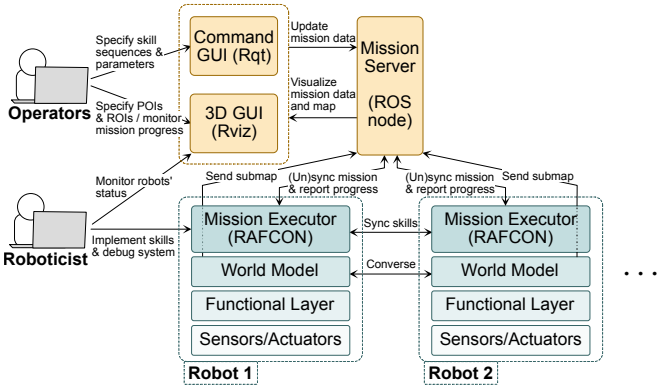

Fig. 4: Overview of the ROSMC implementation. The orange boxes represent the ROSMC components.

the poses of the robots [24], [25][3]. Each agent exchanges so-called submaps with lower bandwidth and lower frequency, keeping communication resource requirements low.

B. 3D GUI

The 3D GUI, as shown in Figure 5, enables intuitive skill parametrization by interactive markers and enhances situational awareness. The GUI is implemented on Rviz [26] so that the roboticists can also visualize on their own Rviz instances the decisions made by the operators and vice versa.

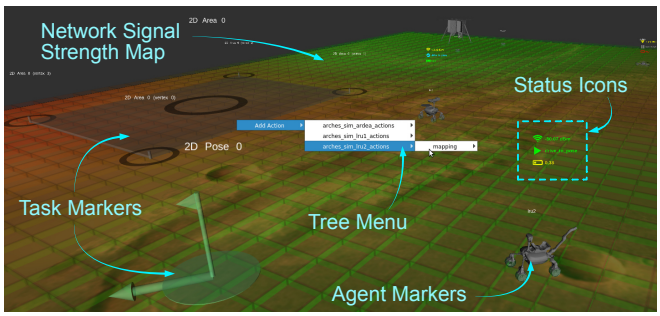

Fig. 5: Screenshot of the 3D GUI.

Three types of ROS interactive markers [27] are implemented; the dynamic agent markers, the static agent markers, and the task markers. The dynamic agent markers visualize robots that can be commanded by operators. Static agent markers visualize stationary robots such as payload boxes or a lander. The task markers can be created freely and represent points/regions of interests (POIs/ROIs) for a skill parametrization. The tree menu appears when clicked and enables operators to add skills to the mission.

Status icons are implemented using pictograms provided by jsk_rviz_plugins[4]. They visualize the Wi-Fi signal strength, the remaining battery energy, and the skill execution status above the robots. Necessary telemetry topics are subscribed through fkie_multimaster package[5].

Although Rviz is originally designed for visualizing data from a single robot, our SLAM framework and the ROS multi-master setup make it feasible for multiple robots.

[3]Alternatively, ROSMC could directly visualize the map from one of the robots since our SLAM framework does not require a central server.

[4]https://jsk-visualization.readthedocs.io/

[5]http://fkie.github.io/multimaster_fkie/

C. Command GUI

The Command GUI, as shown in Figure 6, functions as the main interface to build and monitor a mission. This is implemented as a plugin of Rqt[6] so that other arbitrary plugins can be combined into a single window. The Command GUI consists of three panels; a schedule panel, a parameter panel, and a control panel.

Fig. 6: Screenshot of the command GUI showing a mission for a team of two robots.

The schedule panel is an array of lists of skills. A panel in a column corresponds to a dynamic robotic agent. Each column has two panels in a row. The upper one and the lower one shows the synchronized and the non-synchronized section of the mission (see Section III-B), respectively. Each panel visualizes a sequence of skills and skill barriers in the mission. The color of the rows (e. g. red) represents the execution status (e. g. failure). The visualization enables operators to grasp concurrent activities of the robots at once, which is particularly beneficial to optimize execution.

The parameter panel shows the parameter details of a skill selected in the schedule panel. Operators can edit the parameter values by typing directly, moving its task marker in the 3D GUI, or selecting a different marker. Modifications are synchronized between the GUIs via the mission server.

In the control panel, operators can add/delete skills to/from the non-synchronized section. The (part of) non-synchronized section can be selected and synchronized to the robots. Since we employ a parallel robotic architecture, each robot can be controlled independently and thus the buttons to start/pause/stop execution are available for each robot.

V. Case Studies

The presented framework was deployed onto both a simulated and a real heterogeneous team of multiple robots. With the simulator, we conducted a pilot study to evaluate the usability and intuitiveness. The scalability to the real system was demonstrated during the ARCHES demo mission [5].

A. Heterogeneous Robotic Team

Table I shows our heterogeneous robotic team. The flying robot Ardea [28], [29] serves as a fast scout of areas that are difficult to access by rovers. LRU1 [2], [3] can investigate terrain with science cameras whereas LRU2 [2],

[6]http://wiki.ros.org/rqt

TABLE I: Heterogeneous robots and their characteristics.

	System	Capabilities
Dynamic	**Ardea** [28], [29] 	• Explore and map terrain quickly from above • Up to 10 minutes of outdoor flight time (104 Wh battery)
	LRU1 [2], [3] 	• Explore and map terrain from ground • Analyze the terrain with science cameras in different spectral filters
	LRU2 [2], [4] 	• Explore and map terrain from ground • Manipulate and carry payload boxes • Collect stone and sand samples
Static	**Lander** [3] 	• Recharge batteries of the other robots • Provide computational resource • Act as a navigation landmark
	Payload Boxes [3], [30] 	• Expand network range (Wi-Fi boxes) • Store samples (sample boxes) • Take measurements (low-frequency radio array boxes) • Supply electrical power to other boxes when stacked vertically (power boxes)

[4] can manipulate and transport the payload boxes and collect samples with the robotic arm. Inside the standardized casing, the payload boxes contain scientific instruments or infrastructure equipment. A global landmark for all robots is established by the lander, which serves as a base station and defines an on-site coordinate frame.

B. Pilot Study with the Simulator

Simulation Setup: We designed a Mars surface sample collection scenario. The task given to the participants is to explore an unknown area, discover regions ideal for sampling stones (RSSs), identify, pick, and bring stones to the lander (see Figure 7). The goal is to obtain scores which are given for each stone when brought to the lander. Stones can be identified only inside the RSSs, whose location and size are predefined but unknown to the participants. Each RSS has its maximum score of stones that can be identified there.

A noticeable aspect of heterogeneity of the team is that the score of stones increases if LRU1 uses the science cameras. If LRU1 and LRU2 collaboratively identify a stone, one with the maximum score can be identified. Without collaboration, however, only a stone with the minimum score (globally set to 0.1) can be identified.

Each robot has simulated constraints on the battery capacity and the Wi-Fi connection. The robots consume the battery energy during the mission and can recharge themselves at the lander. Mission adaptation is only possible when the robots are inside the network coverage, which can be expanded by deploying the Wi-Fi boxes using LRU2. Since the purpose of the simulation is not for simulating precise controlling of wheels, arms, nor propellers, we used Gazebo [31] with a limited physics accuracy and ignored collisions.

Procedure: The pilot study had nine participants, three of which had no robotics background. First, they were given 15 minutes to read through the instruction document to acquire knowledge on this simulated exploration scenario. Second, they were instructed on how to use the GUIs in 10 minutes. Third, the investigator demonstrated how each robotic skill works in the simulator, which took 15 minutes. Fourth, the participants performed the mission, which took around half an hour. No strict time limitation was set so that the participants did not feel time pressure. Finally, the participants were asked to fill in two questionnaires; the *questionnaire for the subjective consequences of intuitive use* (QUESI) [32] and the NASA *Task Load Index* (TLX) [33].

Results and Discussion: The average duration of the mission was 26.8 minutes. All participants were able to make the robots bring at least one stone sample back to the lander. In particular, two non-roboticists and three roboticists gathered 1.8 or more scores of stones in the 26.8 minutes, discovering the two highly valuable RSSs on the right bottom corner of the map. They operated the robotic team ideally; they deployed all the three robots for exploration, disclosed all the RSSs, identified their maximum score by LRU1, and focused on taking the two most valuable stones by LRU1 and LRU2 collaboratively. The intuitive GUIs of ROSMC helped the participants to compose, monitor, and adapt a mission smoothly and effectively.

The results of the QUESI shown in Figure 8 also support these qualitative analyses. In all the sub-scales of QUESI (ranging from 0 to 5; the higher the better), the lower bound of the 95% confidence intervals of the non-roboticists' results lie above the lower bound of the roboticist with the score of 0.5 as a margin. This implies that ROSMC has good intuitiveness and enabled non-robotics experts to achieve good performance.

On the other hand, the results of the NASA-TLX questionnaire show the tendency that non-roboticists experienced more difficulty than the roboticists. Even though the results are distributed due to the small number of samples, the subscale "frustration" clearly showed that the non-roboticists had more stress than the roboticists.

C. Analogue Mission with Real Robots on Mt. Etna

The main aim of the ARCHES moon-analogue demonstration mission on Mt. Etna in Sicily was to demonstrate the advantages of using semi-autonomous heterogeneous teams of robots for exploration and sample return, and their successful control from a base station [34].

The first experiment focused on returning stone and sand samples (see Figure 1) while the second one aimed to deploy the network of the low-frequency radio array (LOFAR) boxes in rough terrain [35], [36]. In both scenarios, LRU1 and LRU2 started next to the lander (1)[7]. While LRU2 picked the suitable payload box from the lander shelf (2a), LRU1 explored the uncovered areas and defined POIs for the scientific activities (2b). LRU2 then drove to the POIs (3) and

[7]Due to strong winds, Ardea could not be deployed during the mission.

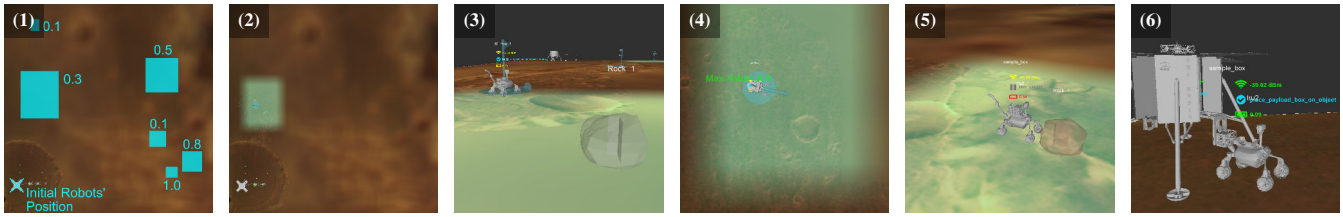

Fig. 7: Overview of how the mission progresses in the pilot study; (1) initial state with the pre-defined locations of the RSSs and their maximum score of stones; (2) an RSS is discovered; (3) LRU2 identifies a stone; (4) LRU1 reveals the maximum stone score of an RSS; (5) LRU2 collects the stone; (6) LRU2 brings the stone back to the lander.

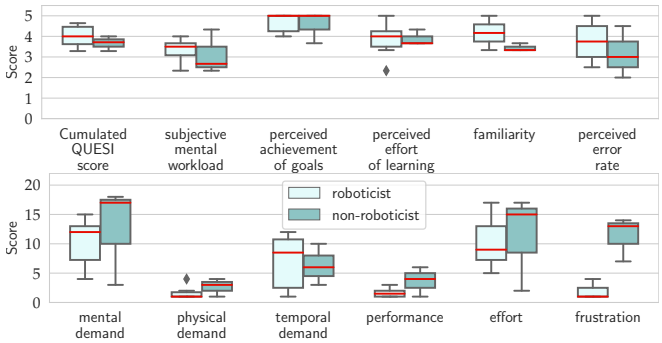

Fig. 8: Results of the scale scores from the questionnaires of QUESI (upper), whose score ranges from 0 (worst) to 5 (best), and NASA-TLX (lower), whose score ranges from 0 (best) to 20 (worst).

collect sand and stone samples or deploy the LOFAR boxes (4a). In the meantime, LRU1 further explored unknown areas (3) while visiting the lander periodically to improve the map precision (4b). The steps above were repeated until satisfactory scientific results are obtained.

The prominent challenge of these missions is that such complicated activities must be scheduled flexibly since the mission specification is highly affected by the actual robot status as well as by the in-situ scientific achievements.

One roboticist per robot stayed in the base camp located around $400\,\mathrm{m}$ away from the robotic team. They monitored the status of the robots and performed recovery remotely in case of unexpected software issues.

Only a single operator controlled the entire mission. The LOFAR mission operation took place at the base camp but in a physically isolated container from the roboticists. The sample-return mission was operated from the real ground station, which is $23\,\mathrm{km}$ away from the mission site.

Results and Lessons Learned: Overall, the robotic team achieved the mission goals with only a single operator and a few roboticists despite the complexity of the systems and tasks. In the sample-return mission, the robots successfully collected three stones and three sand samples from three different locations. During the LOFAR mission, a power box and a LOFAR box were transferred and deployed successfully to the designated place. In total, the robots explored and mapped around $1500\,\mathrm{m}^2$ of an unknown area and manipulated 10 objects in seven hours. Even though the robots executed over 20,000 state machines, the number of the skills executed by the mission control was only 58, reducing the operation workload decently.

Major mission adaptations took place to strike the right balance between navigation uncertainty and scientific activities. Since the localization becomes uncertain after the long distance of travel, the robots have to visit the lander as a global landmark. Nevertheless, if POIs that are scientifically interesting lie near the rovers or on the way to the lander, the opportunities for scientific activities should not be missed. The operator was able to swiftly re-schedule the navigation, exploration, and scientific activities of the both rovers by arranging the prioritization of them. This was due to the strong support of the mission synchronization concept.

Although the robots experienced navigation and manipulation failures, the shared situational awareness over the ROS-based architecture enabled the recovery activity by the roboticists and the mission continuation by the operators at the same time. One missing feature of ROSMC we found would be useful is to let robots retry the skill that failed.

All mission aims were achieved, and it was considered an overall success. From the project perspective, the mission was a large integration test of all hardware and software components, involving about 30 developers. As ROSMC plays a central role for the mission specification and task communication to robots, the overall success implies that ROSMC was successful in this role. The aim of the mission was not to test ROSMC independently, or conduct user studies focussing on ROSMC only. This was therefore done in the simulation experiments conducted before the mission, as described in Section V-B.

VI. Conclusion

This paper presented ROSMC, a high-level mission control tool for a heterogeneous team of multiple robots. The main feature of this tool is the mission synchronization concept to make the autonomous execution and the flexible adaptation of a mission compatible. The ROS-based interfaces not only enables intuitive mission specification but also enhances shared situational awareness with the roboticists. The tool is available on `https://github.com/DLR-RM/rosmc`.

Acknowledgment

We thank the Mobile Robots Group at DLR-RMC and the ARCHES & ROBEX teams for support with the systems and experiments, and the anonymous reviewers for insightful suggestions. This work was supported by the Helmholtz Association, project alliance ROBEX (contract number HA-304) and project ARCHES (contract number ZT-0033).

REFERENCES

[1] M. Quigley, *et al.*, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, vol. 3, 2009.

[2] M. Schuster, *et al.*, "Towards autonomous planetary exploration: The lightweight rover unit (LRU), its success in the SpaceBotCamp challenge, and beyond," *Journal of Intelligent & Robotic Systems*, 2017.

[3] A. Wedler, *et al.*, "First results of the ROBEX analogue mission campaign: robotic deployment of seismic networks for future lunar missions," in *68th International Astronautical Congress (IAC)*, 2017.

[4] P. Lehner, *et al.*, "Mobile manipulation for planetary exploration," in *IEEE Aerospace Conference*, 2018.

[5] M. J. Schuster, *et al.*, "The ARCHES space-analogue demonstration mission: Towards heterogeneous teams of autonomous robots for collaborative scientific sampling in planetary exploration," *IEEE Robotics and Automation Letters (RA-L)*, vol. 5, 2020.

[6] D. C. MacKenzie, R. C. Arkin, and J. M. Cameron, "Multiagent mission specification and execution," in *Robot colonies*, 1997.

[7] F. J. S. Rodríguez, *et al.*, "The complete integration of MissionLab and CARMEN," *International Journal of Advanced Robotic Systems*, vol. 14, 2017.

[8] J.-P. de la Croix, *et al.*, "Mission modeling, planning, and execution module for teams of unmanned vehicles," in *Unmanned Systems Technology XIX*, vol. 10195, 2017.

[9] K. Otsu, *et al.*, "Supervised autonomy for communication-degraded subterranean exploration by a robot team," in *IEEE Aerospace Conference*, 2020.

[10] H. Hong, *et al.*, "SeMo: Service-oriented and model-based software framework for cooperating robots," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, 2018.

[11] C. Lesire, D. Doose, and C. Grand, "Formalization of robot skills with descriptive and operational models," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[12] S. García, *et al.*, "High-level mission specification for multiple robots," in *Proceedings of the 12th ACM SIGPLAN International Conference on Software Language Engineering*, 2019.

[13] C. Menghi, *et al.*, "PsALM: Specification of dependable robotic missions," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 2019.

[14] P. Candela, A. Xydes, and A. Nans, "Designing an operator control unit for cooperative autonomous unmanned systems," in *Unmanned Systems Technology XIX*, vol. 10195, 2017.

[15] Y. Lim, *et al.*, "Human-machine interfaces and interactions for multi UAS operations," in *Proceedings of the 31th Congress of the International Council of the Aeronautical Sciences (ICAS)*, 2018.

[16] ——, "Cognitive human-machine interfaces and interactions for multi-UAV operations," in *18th Australian International Aerospace Congress (AIAC): Defence Science and Technology (DST) International Conference on Health and Usage Monitoring (HUMS): 27th International Symposium on Space Flight Dynamics (ISSFD)*, 2019.

[17] M. Draper, *et al.*, "Intelligent multi-unmanned vehicle planner with adaptive collaborative/control technologies (IMPACT)," in *19th International Symposium on Aviation Psychology*, 2017.

[18] G. L. Calhoun, *et al.*, "Human-autonomy teaming interface design considerations for multi-unmanned vehicle control," *Theoretical issues in ergonomics science*, vol. 19, 2018.

[19] S. G. Brunner, *et al.*, "RAFCON: A graphical tool for engineering complex, robotic tasks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[20] ——, "Autonomous parallelization of resource-aware robotic task nodes," *IEEE Robotics and Automation Letters (RA-L)*, 2019.

[21] L. Cheng, *et al.*, *Opposite Ends of the Spectrum: Cassini and Mars Exploration Rover Science Operations*.

[22] D. Gaines, *et al.*, "Productivity challenges for mars rover operations," in *Planning and Robotics Workshop of ICAPS*, 2016.

[23] R. Volpe, *et al.*, "The CLARAty architecture for robotic autonomy," in *IEEE Aerospace Conference*, 2001.

[24] M. J. Schuster, "Collaborative localization and mapping for autonomous planetary exploration: Distributed stereo vision-based 6D SLAM in GNSS-denied environments," Ph.D. dissertation, University of Bremen, 2019.

[25] M. J. Schuster, *et al.*, "Distributed stereo vision-based 6D localization and mapping for multi-robot teams," *Journal of Field Robotics*, 2018.

[26] H. R. Kam, *et al.*, "RViz: a toolkit for real domain data visualization," *Telecommunication Systems*, vol. 60, 2015.

[27] D. Gossow, *et al.*, "Interactive markers: 3-d user interfaces for ros applications [ros topics]," *IEEE Robotics & Automation Magazine*, vol. 18, 2011.

[28] M. G. Müller, *et al.*, "Robust visual-inertial state estimation with multiple odometries and efficient mapping on an MAV with ultra-wide FOV stereo vision," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[29] P. Lutz, *et al.*, "ARDEA—an MAV with skills for future planetary missions," *Journal of Field Robotics*, vol. 37, 2020.

[30] G. Tsakyridis, *et al.*, "Power system analysis and optimization of a modular experiment carrier during an analog lunar demo mission on a volcanic environment," *Acta Astronautica*, vol. 155, 2019.

[31] N. P. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, 2004.

[32] A. Naumann and J. Hurtienne, "Benchmarks for intuitive interaction with mobile devices," in *Proc. 12th Int. Conf. Human-Computer Interaction with Mobile Devices and Services*, 2010.

[33] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research," in *Human Mental Workload*, 1988, vol. 52.

[34] A. Wedler, *et al.*, "Preliminary results for the multi-robot, multi-partner, multi-mission, planetary exploration analogue campaign on mount etna," in *72nd International Astronautical Congress (IAC)*, 2021.

[35] ——, "Finally! Insights into the ARCHES lunar planetary exploration analogue campaign on Etna in summer 2022," in *73rd International Astronautical Congress (IAC)*, 2022.

[36] E. Staudinger, *et al.*, "Enabling distributed low radio frequency arrays – results of an analog campaign on Mt. Etna," in *IEEE Aerospace Conference*, 2023.