Progress Evaluation: Milestone 6

SRCIS

Search and Rescue Coordinated Intelligence Systems

Team Members
Yavanni Ensley, yensley2022@my.fit.edu
Younghoon Cho, ycho2021@my.fit.edu
Jaylin Ollivierre, jollivierre2022@my.fit.edu

Faculty Advisor & Client
Dr. Thomas Eskridge, teskridge@fit.edu , Harris Center for Science & Eng.

Milestone 6 task matrix:

Task	Completion	Yav	Young	Pop
Target Sharing	*100%*	50%	25%	25%
Multi-agent Coordination	*100%*	60%	20%	20%
Integrating custom ROS Driver w/ UAV	*100%*	40%	60%	0%
Integrating custom ROS Driver w/ quadruped	*100%*	40%	0%	60%
Concurrent Map Sharing	*100%*	60%	20%	20%
Mission Command Structure	*100%*	40%	0%	60%
Target engagement algorithm	*100%*	33%	33%	33%
Full-dress Demo	*100%*	33%	33%	33%
Performance Statistics	*100%*	33%	33%	33%

Task Discussion

Target Sharing

All agents can share target locations. Previously, the UAV SLAM algorithm was properly functioning. The algorithm we were using was ineffective for the UAV's camera. Instead of using the one we created, we found the UAV's built-in SLAM algorithm worked fine on our system. The navigation library for the quadruped was incomplete, but we were able to reconfigure it to display the global costmap. The SLAM algorithm works because the costmap provides the information needed for path planning, including the optimal distance to the target.

Multi-agent Coordination

We were able to get the custom ROS driver running on both the quadruped and UAV. The two platforms are now available in the web application and accessible to the user. The UGV requests aerial reconnaissance, and the UAV will relocate to hover over it. The quadruped stands in a corner of the map and scans for potential target movement near a high avenue of approach. When a quadruped spots the target, it notifies the team and requests assistance from the UGVs to intercept the target's next movement.

Integrating custom ROS Driver w/ UAV and quadruped

The same issues mentioned in **Target Sharing** were restricting the custom ROS driver from running on the UAV and the quadruped. Once we corrected these issues, we were able to

get the custom ROS driver running fine. This driver connects the platforms to the abstraction layer (the biggest part of this project)

Concurrent Map Sharing

Agents can scan the environment more quickly when more agents are present. The agents can explore different parts of the map and add their known locations and obstacles to a shared map to help other agents stay situationally aware without incurring additional exploration time. When locating a target, each agent can find its best path to it and work together to trap it or intercept its movement. Concurrent map sharing also allows them to know where the other is on the map, so when an agent requests assistance, the agent can respond accordingly without needing to request information.

Implementing Mission Command Structure

We prompted the agents with a breakdown of how to work as a team so they could change their behavior to be better teammates. The implementation of the mission command will be a cycle among the agents to distribute workload. Each agent will get a turn to check the chat history and decide to request help, continue exploring, or assist a teammate. Anytime the mission operator has something to insert in the cycle, their request will take priority before the next agent's turn.

Full Demo

The demonstration includes a full run-through of a search-and-rescue operation. We emphasize human-robot teamwork and continuous compositional control, as these are major features of the project as a whole. Another feature we draw a lot of attention to is the shared map; without it, the project's efficiency component does not exist. We also highlight the web application because it is the user's *only* way to interact with the system, so we spent time in our demo discussing it and its purpose. For the urban search-and-rescue operation, we created a mock urban environment with boxes and lab materials to simulate a city for our robotic platforms to navigate. This environment consists of walls, buildings without surrounding obstacles, a rigid boundary, tunnels with light, and tunnels with minimal light. All of these different obstacles create unpredictable routes. To polish things, we recorded a commercial-style demo to showcase our project's features while making it visually appealing and engaging.

Performance Statistics

To support the argument that our project makes search-and-rescue missions more efficient, we conducted time trials to measure how quickly the UGVs could map the mock environment. Time was our dependent variable, and the number of UGVs scanning the area was the independent variable. The correlation we drew from our results was: As the number of UGVs increases, their efficiency in scanning the map increases, so it's better to have multiple working together. Teamwork prevails, even before we added any of the other platforms to assist.

Member Contributions:

Yavanni Ensley
During the final stretch, I needed to fully integrate all robotic platforms with the abstraction layer so we could record our demonstration. I wanted to ensure everything was working properly with the UGVs before adding the drones and quadrupeds, since we knew the custom driver already worked on the other platforms. To reduce video latency and load on our project system, I switched our video streaming to WebRTC. The UGVs' exploration algorithm runs more smoothly now as well, since I adjusted the algorithm and the pathfinding prioritization. I ironed out the performance tweaks in the demonstration to prepare us for the showcase.

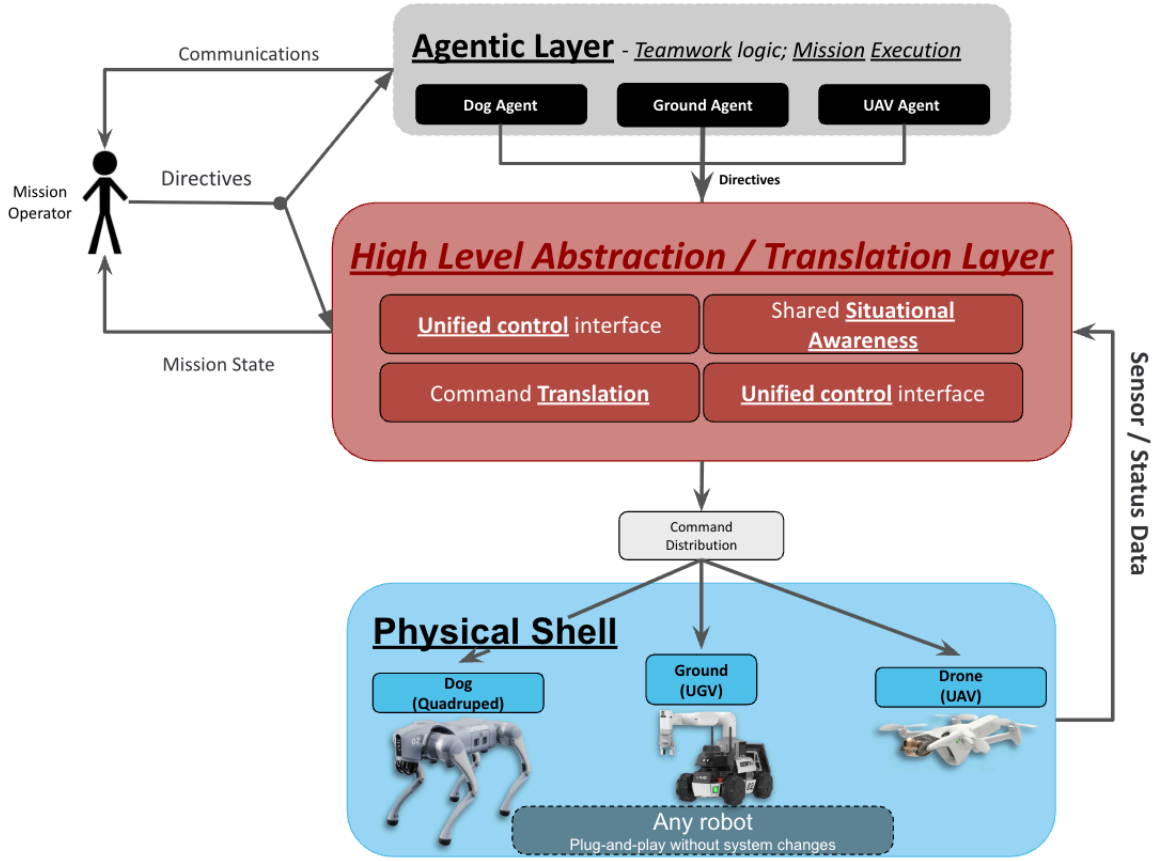
Younghoon Cho
For the final milestone, I was responsible for integrating the UAV with the custom ROS driver system. During development, we encountered a technical issue where the Docker setup was not communicating with the correct target. We resolved this by removing Docker Desktop and switching to Docker Engine. In addition, I created a digital map of the lab for our poster based on the real-world environment, including walls and tunnels. I created the user and developer manual and divided the work to ensure efficient document distribution. I was also responsible for ensuring the team adhered to the criteria provided by Dr. Chan.

Jaylin Ollivierre
I identified the missing navigation library configurations for the quadruped to map the environment and use the SLAM algorithm for pathfinding. Once I completed this task, we were able to use the custom ROS driver on the quadruped. Another contribution I made during this milestone was designing our poster and final diagrams. I made sure the poster was showcase-ready, professional, visually appealing, and easy to gather information from. I was responsible for creating and editing our video demonstration for the final project. We worked to "Brand" this project, so I wanted the demo to be an experience and put us ahead of the curve by making it as authentic as possible. I assisted in the user and developer manuals, both contributing and making final edits to ensure their readiness.

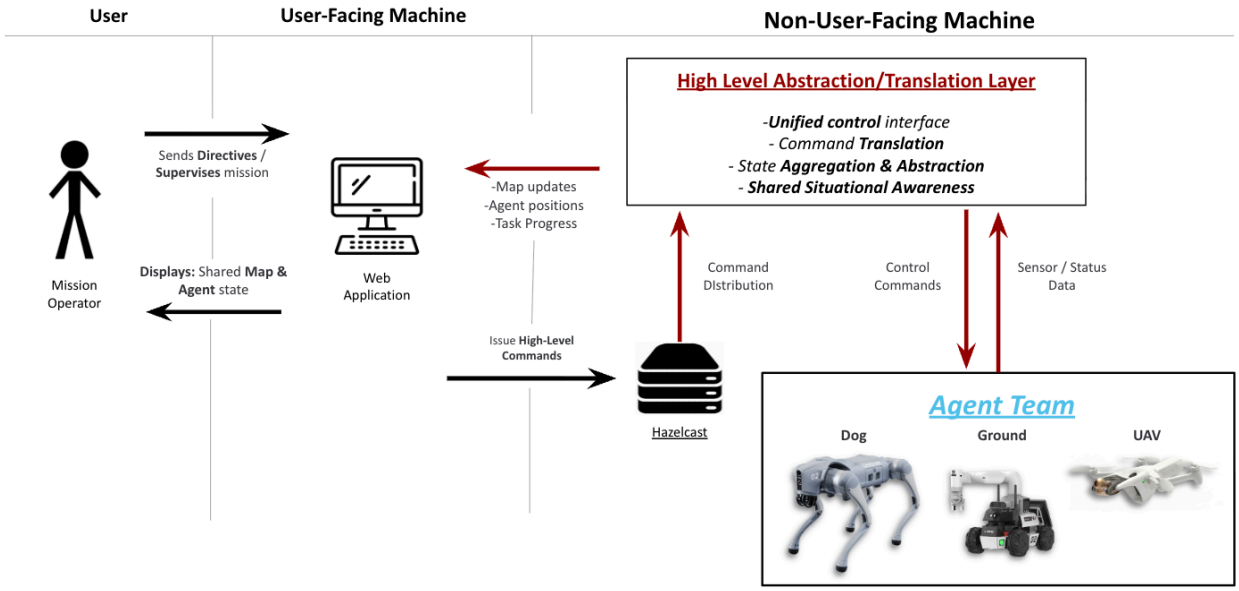
Lessons Learned

1. Time management and estimation are important skills. You will likely expect how long something will take, but you will also be able to account for when things do not go as planned.
2. When working with multiple systems, it is crucial to ensure that all software versions and configurations are identical to maintain consistency. Using Docker greatly simplified this process for us.
3. Communication is very important. You may be running into an issue someone else has experienced, so asking for help or, at a minimum, communicating when you are stuck, can save loads of time.
4. Do not be afraid to delegate things that are common to the whole group. If no one takes responsibility, it may not get done, and the whole group suffers because no one assumed responsibility.
5. Preventative checks and maintenance on all equipment are important. From the start, we learned that our UGVs needed a special wheel bracket because the axles were prone to snapping. Robots can be fragile systems that require careful handling, regular inspection, and preventive checks and maintenance. Testing our equipment early, often, and for long periods of time saved us from having significant problems toward the end of our project.
6. Goal decomposition is the key to success. When thinking about how to start, you cannot think about the end goal. While knowing the end goal is important, creating a minimum viable product with very basic expectations is imperative to success. Following an agile process model, you make changes as you go because a waterfall approach *would* be more costly and yield little progress. From the start, our advisor explained the decomposition and the smaller pieces we should focus on before reaching our end goal.
7. There is value in working both synchronously and asynchronously. Do not be afraid to spend time in the lab without your group mates. In the same vein, it is sometimes important to have everyone present to get on the same page. Our best lab sessions came from all of us being present, brainstorming together, asking questions, and assisting when a problem arose.

Advisor Diagram



Instructor Diagram



Advisor Section

Date(s) of meeting(s) with Client during the current milestone:

- See faculty advisor dates below

Client feedback on the current milestone

- See faculty advisor feedback below

Date(s) of meeting(s) with Faculty Advisor during the current milestone:

- 4/15/26

Faculty Advisor feedback on each task for the current Milestone

Faculty Advisor Signature:  Date: 4/20/2026