Enabling Autonomous Rover Science Through Dynamic Planning and Scheduling[1]

Tara Estlin, Daniel Gaines, Caroline Chouinard, Forest Fisher, Rebecca Castano, Michele Judd, and Issa Nesnas
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109
Phone: 818-393-5375
{firstname.lastname}@jpl.nasa.gov

Abstract— With each new rover mission to Mars, rovers are traveling significantly longer distances. This distance increase allows not only the collection of more science data, but enables a number of new and different science collection opportunities. Current mission operations, such as that on the 2003 Mars Exploration Rovers (MER), require all rover commands to be determined on the ground, which is a time-consuming and largely manual process. However, many science opportunities can be efficiently handled by performing intelligent decision-making onboard the rover itself.

This paper describes how dynamic planning and scheduling techniques can be used onboard a rover to autonomously adjust rover activities in support of science goals. These goals could be identified by scientists on the ground or could be identified by onboard data-analysis software. Several different types of dynamic decisions are described, including the handling of opportunistic science goals identified during rover traverses, preserving high priority science targets when resources, such as power, are unexpectedly oversubscribed, and dynamically adding additional, ground-specified science targets when rover actions are executed more quickly than expected.

After describing our specific system approach, we discuss some of the particular challenges we have examined to support autonomous rover decision-making. These include interaction with rover navigation and path-planning software and handling large amounts of uncertainty in state and resource estimations.

Finally, we describe our experiences in testing this work using several Mars rover prototypes in a realistic environment.

TABLE OF CONTENTS

1. INTRODUCTION

NASA recently demonstrated that mobile robotic craft are a viable and extremely useful option for exploring the surface of other planets. The 2003 Mars Exploration Rovers (MER) have traveled across thousands of meters of terrain and gathered large amounts of valuable scientific data that is being used to answer many questions about the Martian environment. Future missions are being planned to send additional robotic explorers to Mars as well as to the moon and outer planets.

Most mobile robot efforts at JPL and NASA have concentrated on building software infrastructure for navigation, manipulation and control. High-level decision making for these efforts, including for the Mars Pathfinder and MER missions, is performed on Earth through a predominantly manual, time-consuming process. For MER, a ground-based AI planning and scheduling tool was used to support science plan evaluation, however, a command sequence was still manually generated on the ground and uplinked to the rovers.

[2] IEEEAC paper #1405, Version 3, Updated November 12, 2004

One significant problem with this approach of rover operations is that it can result in frequent underutilization of the robotic assets. When a rover encounters a situation that deviates from its uploaded sequence, the fault protection software may attempt some limited resolution methods, such as relying on secondary sensors if primary ones become inoperable. Typically, command sequences are generated using a worst case analysis so that deviations to the sequence are rare (e.g., overestimate time to drive to a new location to ensure that rover arrives by end of day. If the rover performs nominally or better than predicted, then the overestimates result in significant idle time for the rover. If a deviation does occur that cannot be handled by fault protection, the rover enters *safe-mode* and must wait for a new command sequence to be sent from earth. This model of operations results in a significant loss in science return since the rover must remain idle, for hours or days at a time, until new commands are received. Further, new science opportunities can only be taken advantage of after scientists on the ground have been able to review downlinked data. This approach means many opportunities may not be realized, since once identified on Earth, the rover may have already traveled past the object of interest. This case becomes even more prominent as rovers perform longer traverses (e.g., the MER rovers can drive up to 100 meters per day).

A primary objective of our work is to use onboard planning and execution techniques to increase utilization of rover resources by enabling the rover to appropriately respond to unexpected problems and to take advantage of unanticipated opportunities.. The Closed-Loop Execution and Recovery (CLEaR) system is intended to run with little communication with ground. It accepts science and engineering goals and creates a rover command sequence (or plan) that respects relevant constraints, while achieving as many goals as possible. The system executes the produced plan by dispatching commands to the rover's low-level control software and monitoring relevant state information to identify current or potential problems. If problems or new opportunities are detected, the system is designed to handle such situations by using re-planning techniques to add, move or delete plan activities. Through this work, we have also identified a number of challenges for an onboard planning and execution system to not only produce valid plans, but also promote robust and efficient rover behavior. These challenges include properly interacting with the appropriate rover navigation software, handling uncertainty in state and resource estimations, as well as handling dynamic events, such as new science opportunities.

For the past four years, we have spent significant time testing the CLEaR system on several different rovers in the JPL Mars Yard. We will discuss our scenario designs for this testing and give an overview of the results including a discussion of how the system handled major scenario elements. Our main objectives for testing included simulating situations that might arise in future rover missions, (such as the Mars Science Laboratory or MSL mission, planned for launch in 2009), providing feedback on our approach, and identifying future directions that should be investigated. This work was also a major contributor to achieving the NASA Intelligent Systems Program milestone of developing autonomous capabilities for opportunistic science handling [1].

In the following section we outline some key challenges that we have identified for onboard decision-making software. Next, we present our current system approach and explain how this system fits into a larger rover architecture and other supporting software that contributed to our testing. We then describe several Mars rover scenarios that were used to test our system on rover hardware, and describe how our system performed during that testing.

2. CHALLENGES FOR ONBOARD DECISION MAKING

Autonomous rovers have the potential for increasing science return by reducing rover idle time, reducing the need for entering *safe-mode* and dynamically handling opportunistic science events without required communication to Earth. New missions are being designed that will require rovers to support more autonomous endeavors such as long-range traversals, complex science experiments, and longer mission duration. However, autonomy software designers face a number of challenges in providing software to support these types of operations. In this paper, we consider a few key challenges for using planning and execution techniques to provide onboard decision-making capabilities.

To generate and/or modify its own command sequence for carrying out a set of science goals, the onboard software will need to reason about a rich model of resource and temporal constraints. For example, it will need to predict power consumption of variable duration activities such as downlinks and traverses, keep track of available power levels, and ensure that generated plans do not exceed power limitations. When resources are over-taxed, the rover should be capable of making science/resource trade-offs in an effort to produce the highest science return. The rover will also require execution and monitoring capabilities to carry out the generated plan on the rover platform. An execution system must be capable of commanding the control software, collecting state updates from sensors, and smoothly handling activity failures or unexpected events.

Over the course of a mission, the rover will be asked to perform a variety of science operations. The number and scope of these operations are typically limited by the rover onboard resources (e.g., power, memory, lifetime of hardware). Thus, science operations may have varying priorities that indicate their overall mission value. Onboard decision-making software must reason about these priorities and handle new science opportunities in a dynamic and efficient manner. The value of newly identified science observations must be weighed against current resource availability and other scheduled activities.

Sequence generation for rover surface missions also raises a number of interesting challenges regarding spatial reasoning capabilities. One of the dominating characteristics of rover operations is traverses to designated waypoints and science targets. This element is especially important in future missions that intend to explore large geographic areas. Onboard planning and execution software needs to coordinate with several levels of rover navigation software to generate an efficient and achievable rover plan. This coordination will likely include querying a path planner for route information needed to generate a plan of rover activities, using position estimates to track rover progress, and correctly modifying the plan when navigation and obstacle avoidance software cause the rover to move off the predicted route.

Another predominant challenge in developing onboard autonomy software is dealing with the inherent uncertainty in predicting rover navigation and science operations. The difficulty is compounded by the tight resource and time constraints that a rover typically faces. At the resource and temporal level, the estimation of items such as power, memory and even activity duration can be highly uncertain. Rover missions are directed at exploring unknown planetary terrains. Requirements for traversing these new terrains are hard to predict. For instance, it is unknown what type of sand consistency a rover will be traversing, which can dramatically affect the required duration and power for a traverse. Similarly, the duration and resource requirements for science operations can vary as well. These variations could be simple, such as a lower then expected image compression ratio, or more complex, such as a drilling operation taking more power and time than originally estimated.

Furthermore, at the state level, the estimation of rover position is often a constant source of error. The Sojourner rover only used dead-reckoning capabilities to estimate rover position, which produced a position error of roughly 5-10% of distance traveled and an average heading drift of 13 degrees per day of traverse [2]. The MER rovers use more sophisticated techniques to provide position estimation, including a 3-axis gyro and visual odometry. However, these rovers still accrue significant position estimation error and on the ground localization software is often used to recalculate position. Since a large part of a rover schedule consists of rover moves to different locations, the onboard autonomy software must use estimations of position to predict the duration and resource requirements of different operations. If these predications are inaccurate, the autonomy software must be able to continuously modify the schedule to handle changes in expected rover behavior.

3. Planning And Execution For Rover Operations

To address the issues outlined in the previous section, we have developed a system for high-level decision-making capabilities for future Mars rovers. The overall system framework and data flow is shown in Figure 1. This paper primarily focuses on the planning, scheduling and execution element of this framework, which provides autonomous rover command-sequencing capabilities, however other components will also be briefly described.

In this framework, planning, scheduling, and execution techniques are applied to provide initial rover plan generation, plan execution and monitoring, and the continuous modification of that plan based on changing operating context and goal information. These capabilities are provided by the CLEaR (Closed-Loop Execution and Recovery) system [3,4]. CLEaR was developed to pursue a tight integration of planning and execution capabilities. To provides these capabilities, CLEaR closely integrates the CASPER (Continuous Activity Scheduling, Planning, Execution and Re-planning) continuous planner and the TDL (Task Description Language) executive system, which are described further below. Previous versions of the CLEaR framework have been demonstrated for Deep Space Network (DSN) antenna control [5]. Currently CLEaR is being used and extended to provide planning and execution support for planetary rovers.

In our system framework, CLEaR handles the following functionality:

- Creating an initial plan based on an input set of goals
- Maintaining resource, temporal and other rover operability constraints
- Executing a plan by interacting with basic and low-level rover control functionality (e.g., navigation, vision)
- Monitoring plan execution to ensure plan objectives are met
- Dynamically modifying the current plan based on plan activity, state and resource updates
- Performing plan optimization to reason about soft constraints and goal priorities
- Handling dynamically identified science goals (called *science alerts*) that are generated through onboard data analysis

CLEaR's primary objective is to provide a tightly coupled approach to coordinating goal-driven and event-driven behavior. Many past approaches have followed a three-level architecture style where the planning and executive

Figure 2: Sample rover plan displayed in planner GUI. Plan activities are shown in upper portion of window, where bars represent the start and end time of each activity. State and resource timelines are shown in bottom portion of the screen and show the effects of the plan as time progresses. Time is depicted as advancing from left to right.

processes are treated as *black box* systems. This is in contrast to how CLEaR enables the planner and executive to interact with each other and more effectively share the responsibility for decision making. In part this is managed through shared plan information and continual updates of state being made available to both the planner and executive. CLEaR also provides heuristic support for deciding when certain plan conflicts should be handled by the planner vs. the executive. For instance if a rover gets off track during a traverse, the reaction of the planner and executive need to be coordinated. If the executive believes it can resolve the navigation delay within the planned time constraints it will manage the plan changes. However, once the executive identifies that the repair will require more time or resources than allotted by the planner, it will then fail the task, which will result in the planner using its global perspective to fix the problem.

Planning in CLEaR is provided by the CASPER continuous planning system [6]. Based on an input set of science goals and the rover's current state, CASPER generates a sequence of activities that satisfies the goals while obeying relevant resource, state and temporal constraints, as well as operation (or flight) rules. Plans are produced using an *iterative repair* algorithm that classifies conflicts and resolves them individually by performing one or more plan modifications. CASPER also monitors current rover state and the execution status of plan activities. As this information is acquired, CASPER updates future-plan projections. Based on this new information, new conflicts and/or opportunities may arise, requiring the planner to re-plan in order to accommodate the unexpected events. An example of a plan in the CASPER GUI that was executed during a rover demonstration is shown in Figure 2.

The executive functionality in CLEaR is performed by the TDL executive system [7]. TDL was designed to perform task-level control for a robotic system and to mediate between a planning system and low-level robot control software. It expands abstract tasks into low-level commands, executes the commands, and monitors their execution. It also provides direct support for exception handling and fine-grained synchronization of subtasks. TDL is implemented as an extension of C++ that simplifies the development of robot control programs by including explicit syntactic support for task-level control capabilities. It uses a construct called a *task tree* to describe the tree structure that is produced when tasks are broken down into low-level commands.

Currently, CLEaR has a separate planner and executive and thus does share similarities to other three-layer architecture approaches. However, as compared to these approaches where planning is typically done in a batch fashion and takes on the order of minutes to hours, this integration uses a continuous planning approach, where plans are updated and repaired in a matter of seconds. This enables CLEaR to use planning techniques at a finer timescale for tracking the progress of plan execution, quickly identifying potential problems in future parts of the plan, and responding accordingly. As we expect minor portions of the plan to change frequently, we use a lightweight plan runner to dispatch activities to the executive a few seconds before the task's scheduled *start time*. This approach differs from the more common batch approach of turning the entire plan over to the executive for execution. Executive techniques are then used in only reactive situations or at times where procedural reasoning is preferred. In the Discussion section, we discuss other steps we have taken towards tighter integration that have been tested in simulation.

Another way that CLEaR differs from previous approaches is in how the delegation between the planner and the executive is managed. We have primarily taken a planning centric approach to this management. The planner handles the decision of when an activity should be mapped to a task for execution as well as when to perform re-planning. The re-planning process is driven by applying and propagating updates to the plan, and then taking corrective actions to address any conflicts or opportunities that may arise. Re-planning can also be performed synchronously with any already executing tasks. Once a planning activity has been mapped to an executive task for execution, control over that one task is given to the executive. The executive may then perform further task expansions as a result of updates and/or exception handling. The executive also provides task completion status back to the planner by either marking an activity as complete or failed. A task is marked as completed when the executive decides the task has met its objective, or marked as failed upon concluding that given constraints provided by the planner cannot (or even might not) be met.

Science Alerts

To handle opportunistic science, we extended CLEaR to recognize and respond to *science alerts*, which are new science opportunities detected by onboard science-data analysis software. For example, if a rock is detected in navigation imagery that has a previously unseen color or texture, a science alert may be generated to take additional measurements of that rock.

Currently, science alerts can have different levels of reaction from the CLEaR system. The most basic reaction is to adjust the rover plan so that the rover holds at the current position and the flagged data is sent back to Earth for further analysis at the next communication opportunity. The next level of reaction is to collect additional data at the current site before transmitting back to Earth. Further steps include having the rover alter its path to get closer to objects if interest before taking additional measurements. These operations would provide new data that could not be obtained through analysis of the original image.

Plan Optimization

To reason about goal priorities and other soft constraints we used the CASPER optimization framework to continually search for a more optimal plan. User-defined preferences are used to compute plan quality based on how well the plan satisfies these preferences. Optimization proceeds similar to iterative repair. For each preference, an optimization heuristic generates modifications that could potentially improve the plan score.

One key area where plan optimization was used was to take advantage of extra time or resources in the schedule. Since traverse times and rover resource usage are difficult to predict, it is often the case that a rover operation takes less time or power than expected. For instance, a traverse could take much less time than expected due to sand consistency or a reduced number of obstacles. For these cases, the optimization framework was used to dynamically add in additional science goals to the plan that could not be fit in the original plan due to time and resource constraints. This capability enables the scenario where scientists on the ground specify a number of prioritized science goals, but not all of them may be achievable due to limited rover resources. However, some goals may be fit in as time proceeds due to less resource being used than predicted.

OASIS also uses the optimization framework to decide how to respond to science alerts. Because it may not be possible to accommodate all alerts, a science alert is represented as an optional goal, which indicates its achievement is not mandatory but may improve the plan's optimization score if included in the plan. Before attempting to handle a science alert, CASPER protects the current plan by saving a copy before optimization. If the quality has not increased after a certain time limit, the previous plan is restored. If CASPER can handle a new science alert (e.g., by adding additional science measurements) without causing other negative affects, such as resource over-subscriptions or the deletion of ground-specified science goals, then the new plan that accommodates the science alert is used.

We created a set of plan modification functions that are invoked when the optimizer attempts to satisfy a science alert. How the plan is modified depends on the type of alert that is considered. When science alert is received that requires holding at the current position until data is communicated with earth (called a *stop and call home* alert), the planner alters the plan to remove any non-engineering criticial activities and wait for the next communication opportunity, If activities are currently executing, the planner requests the executive components of CLEaR to abort them. If activities are scheduled in the future, the planner deletes then and resolved any inconsistencies created by these deletions.

To handle a science alert that requests additional measurements (called a *data sample request* alert), the planner must generate a plan that achieves the new goals without deleting existing activities or causing conflicts that cannot be resolved (e.g., scheduling more activities than can be executive over a certain time window). To handle a data

Figure 3: Rocky 8 rover (left), FIDO rover (middle), Rocky 7 rover (right)

sample request, the planner must be able to add a new science observation and a new move command to correctly place the rover in position to take the observation.

4. SCIENCE DATA ANALYSIS

The Feature Extraction and Data Analysis modules, shown in Figure 1, are responsible for onboard science alert generation. Together with the planning and scheduling component, these capabilities comprise the OASIS onboard science system [8]. OASIS enables the rover to perform onboard analysis of collected science data and to trigger science alerts if interesting science opportunities are detected. For instance, if a rover is performing a long traverse, OASIS can analyze navigation images as they are taken to search for interesting rocks or other terrain features that the rover is passing.

As shown in Figure 1, new science data is first processed by the Feature Extraction component. Currently, we have focused on analyzing rocks within image data, but plan to expand to other types of data, such as spectrometer measurements. Images are broken down by first locating individual rocks, and second, by extracting a set of rock properties (or features) from each identified rock. Extracted rock properties are then passed to the Data Analysis component of the system. This component consists of different prioritization algorithms, which analyze the data by searching for items such as rocks with features that match pre-known signatures of interest, which have been identified by scientists on Earth, or novel rocks (i.e., outliers) that have not been seen in past traverses. If the Analysis component detects new science opportunities of significant interest, it will generate a *science alert* that is sent to the planning, scheduling and execution module.

5. CLARATY ROBOTIC ARCHITECTURE

The planning, scheduling, and execution component is also integrated with the Coupled Layered Architecture for Robotic Autonomy (CLARAty) [9], which is being developed at JPL in response to the need for a robotic control architecture that can support future mission autonomy requirements. CLARAty provides a large range of basic robotic functionality and simplifies the integration of new technologies on different robotic platforms. Through CLARAty, the CLEaR system has been tested with several JPL rover platforms, including Rocky 7, Rocky 8, and FIDO, which are shown in Figure 3.

To run realistic scenarios with rover hardware, a number of supporting pieces of software were used. These components were provided through the CLARAty architecture and could run on the relevant JPL rover platforms. This software includes the Morphin navigation system [10], which enables the rover to avoid obstacles and navigate to specified waypoints and a position estimation algorithm, which uses an IMU (Inertial Measuring Unit) to estimate rover attitude (roll, pitch and heading) and wheel odometry to estimate linear velocity.

6. SYSTEM TESTING

To evaluate our system we performed a series of tests both in simulation and using rover hardware in the JPL Mars Yard. These tests covered a wide range of scenarios that included the handling of multiple, prioritized science targets, limited time and resources, opportunistic science events, resource usage uncertainty causing under or over-subscriptions of power and memory, large variations in traverse time, and unexpected obstacles blocking the rover's path. We also performed a final demonstration in October of 2004 that incorporated a large number of these elements and used the JPL FIDO rover.

Our testing scenarios typically consisted of a number of science targets specified at certain locations. A map was used that would represent a sample mission-site location where data would be gathered using multiple instruments at a number of locations. Figure 4 shows a sample scenario that was run as part of these tests. This particular map is of the JPL Mars Yard. The pre-specified science targets represented targets that would be communicated by scientists on Earth. These targets were typically prioritized and for many scenarios constraints on time, power or memory would limit the number of science targets that could be handled. A large focus of these tests was to improve system robustness and flexibility in a realistic environment. Towards that goal we used a variety of target locations and consistently selected new science targets and/or new science target combinations that had not been previously tested.

Another primary scenario element was dynamically identifying and handling opportunistic science events. For these tests, we concentrated on a particular type of event, which was finding rocks with a high albedo measurement (i.e., light or white-colored rocks). This setting was an example of using the data analysis algorithm for target signature, where a particular terrain signature is identified as having a high interest level. If rocks were identified in hazard camera imagery that had a certain interest score, then a science alert was created and sent to the planner. Science alerts would typically come in during rover traverses to new locations, but it was also possible for them to come in while the rover was at a science target location due to a small lag caused by image processing time. If a science alert was detected the planner attempted to modify the plan so an additional image of the rock of interest was acquired.

Other important scenario elements included adding or deleted ground-specified science targets based in resource under or over-subscriptions. For instance, in some tests, the rover covered distances faster than expected and the planner was able to add in additional science targets that could not be fit into the original plan. Conversely, in other tests, the rover used more power than expected during traverses (or science measurements), which eventually caused a power oversubscription. The planner resolved this situation by deleting some lower priority science targets. Unexpected energy drops during a traverse could also be handled by the executive, which detects the shortfall and stops the current traverse if there is not enough energy to complete it. In all cases, the planning and execution system attempts to preserve as many high priority science targets as possible with current resource and time settings.

Testing in Simulation

Since testing with rover hardware can be an expensive and time-intensive process, we ran a large number of tests in simulation using a relatively simple simulator. This simulator could execute rover sequence commands and simulate their effects at a coarse level of granularity. For instance the simulator handled items such as rover position changes and energy usage over straight-line movements, but did not simulate obstacle avoidance or rover kinematics. Another capability that was used in simulation was triggering science alerts at pre-set or random times. This capability helped in evaluating the planner's capacity to correctly handle different opportunistic science scenarios.

To easily run and evaluate large numbers of tests, we also invested in a testing infrastructure, which allowed tests to be run offline and automatically gathered statistics, including items such as number of plan conflicts found and resolved, plan generation and re-planning time, number of goals satisfied, overall plan traverse distance and plan

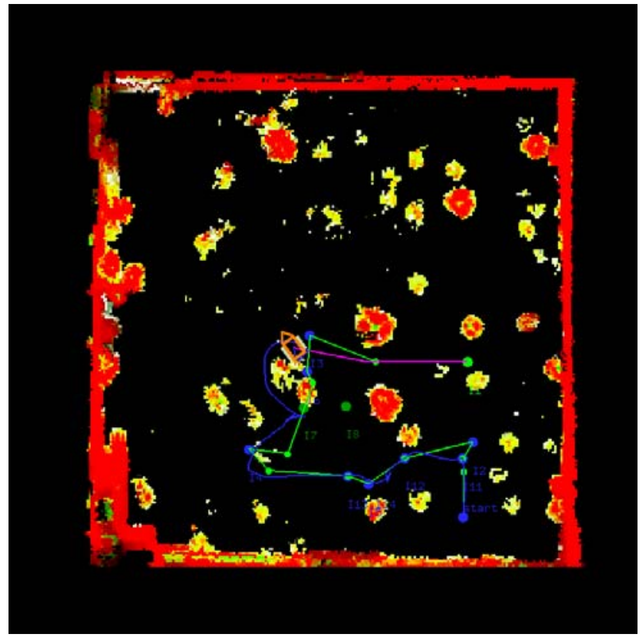

Figure 4: Sample plan shown in the Grid Visualization Tool (GriViT). Green lines show the planned path of the rover. Blue lines shown the real path, and pink lines show the path that is currently executing.

optimization scores. This testing infrastructure also enabled the automatic creation of mpeg movies that showed plan changes using snapshots of a plan visualization tool. This tool showed the results of plan generation and execution on an overhead map of the world, and could be used for both simulated and hardware testing. An example plan snapshot displayed by this tool is shown in Figure 4. Planning and execution results were evaluated by examining gathered statistics and by viewing created mpegs to flag incorrect or non-optimal behavior.

Testing with Rover Hardware

In additional to testing in simulation, a large number of tests were run in the JPL Mars Yard (shown in Figure 5) using different rover hardware platforms. For the past year and for the final demonstration, the FIDO rover (shown in Figure 3) was used for the majority of tests. FIDO is a terrestrial, advanced technology prototype rover similar to the Mars Exploration Rover (MER) rovers on the surface of Mars. FIDO's mobility sub-system consists of a six-wheel rocker-bogie suspension capable of traversing over obstacles up to 30 cm in height. All demonstrated software has been designed to run onboard the rover, however during testing, only functional-level CLARAty modules, such as navigation and vision, and the OASIS rockfinding software were run onboard FIDO. Other modules, including the planning and execution module and the analysis module, were run on offboard workstations that communicated with the rovers using Wireless Ethernet since a port to the onboard operating system (VxWorks) was not complete.

Figure 5: The JPL Mars Yard with terrain of various difficulties.

Tests in the Mars Yard typically consisted of 20-50 meter runs over a 100 square meter area with many obstacles that cause deviations in the rover's path. Most rocks in the Mars Yard are dark in color, thus we brought in a number of whiter rocks to trigger science alerts during rover traverses. Science measurements using rover hardware were always images, since other instruments were not readily available (e.g., spectrometer). However different types of measurements were included when testing in simulation.

Testing and Demonstration Results

Testing in simulation and with real hardware provided important steps in the evaluation of our system. Many bugs were caught early through simulated testing, but others did not surface until significant runs had been performed on rover hardware. Furthermore, running with hardware often allowed a perspective that was difficult to attain through simulated testing. For example, the accuracy of rover turns towards new science opportunities was much easier to judge when running with hardware.

As a final test of our system, we performed a several hour long demonstration in October 2004. This demonstration covered the elements previously presented in this section. Further, the combination of science targets used had not been previously tested with. This set also included a science target that was selected that day by a present MER scientist. Rocks intended to cause science alerts were also placed in new locations not previously used. Overall, the demonstration was very successful. Two scenario runs were performed. Both had multiple targets with time or resource constraints preventing all targets from being included in the initial plan. In the first run a number of science alerts were correctly identified and handled. This run also had an additional science target added dynamically in the run due to the rover traveling faster than estimated. In the second run, lower priority targets were deleted due to more power being used in early traverses than expected. The software presented in this paper (planning, scheduling, execution, feature extraction and data analysis) operated correctly in all cases and caused no undesirable behavior. In general, the rovers operated fully autonomously and traveled over 40 meters.

7. Related Work

A number of planning and executive systems have been successfully used for robotic applications and have similarities to the approach we describe in this paper. Most of these approaches have used some combination of planning and execution, however they differ in not only the behavior of these individual components, but also in how these systems interface with each other and with other system modules.

The Remote Agent Experiment (RAX) [11] was flown on the NASA Deep Space One (DS1) mission. It demonstrated the ability of an AI system to respond to high-level spacecraft goals by generating and executing plans onboard the spacecraft. The Autonomous Sciencecraft Experiment (ASE) [12] has demonstrated the capability of planning and data analysis systems to autonomously coordinate behavior of the EO-1 Earth orbiting satellite. ASE also can also detect and respond to new science events, however it uses very different detection and analysis algorithms. Furthermore, since RAX and ASE were applied to spacecraft, neither handle issues associated with the uncertainty of surface navigation.

Another approach directed towards rover command generation uses a Contingent Planner/Scheduler (CPS) that was developed to schedule rover-scientific operations using a Contingent Rover Language (CRL) [13]. CRL allows both temporal flexibility and contingency branches in rover command sequences. Contingent sequences are produced by the CPS planner and then are interpreted by an executive, which executes the final plan by choosing sequence branches based on current rover conditions. In this approach, only the executive is onboard the rover; planning is intended to be a ground-based operation. Since only a limited number of contingencies can be anticipated, our approach provides more onboard flexibility to new situations. In the CRL approach, if a situation occurs onboard for which there is not a pre-planned contingency, the rover must be halted to wait for communication with ground.

Other similar approaches include Atlantis [14], 3T [15], and a robotic control architecture developed at the LAAS-CNRS lab [16] which all use a deliberative planner and executive (or sequencing component) on top of a set of reactive controllers. These approaches have distinctly separated planning and execution techniques, and have not closely interacted with navigation software used for rover missions., and are not integrated with onboard analysis system for dynamically identifying new goals.

8. CONCLUSIONS AND FUTURE WORK

This paper discussed a number of challenges for using planning, scheduling and execution techniques to provide autonomous rover capabilities for future NASA missions. We described our approach for using an onboard decision-making system and explain how it provides capabilities for sequence generation, execution, monitoring, re-planning, sequence optimization, and opportunistic science handling. Through a series of tests in simulation and on rover platforms, we have demonstrated our systems ability to robustly respond to unexpected problems and take advantage of unforeseen opportunities, thus achieving higher utilization of rover resources.

In future work, we plan to extend our capabilities for opportunistic science handling to include adding observations for different types of science instruments and performing close-contact measurements for high priority alerts. We also intend to extend our system to handle area surveying, where all rocks (or other terrain features) within a certain area would be properly examined and catalogued. Finally, we plan to investigate how our system can efficiently handle activity failures and exceptions. For instance, our system would handle science operations failing in different fashions such as an unsuccessful data acquisition (e.g., an over-exposed or miss-targeted image frame or an unsuccessful grasping of a rock).

ACKNOWLEDGEMENTS

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. This work was sponsored by the Intelligent Systems Program of CICT and the Mars Technology Program.

REFERENCES

[1] R. Morris, T. Estlin, L. Pederson, and B. Williams, “Demonstrating Robotic Autonomy in NASA’s Intelligent Systems Project,” To appear at *The Eighth ESA Workshop on Advanced Space Technologies for Robotics and Automation*, Noordwijk, the Netherlands, Nov 2004.

[2] A. Mishkin, J. Morrison, T. Nguyen, H. Stone, B. Cooper, B. Wilcox, "Experiences with Operations and Autonomy of the Mars Pathfinder Microrover," *Proceedings of the 1998 IEEE Aerospace Conference,* March 21-28 1998, Snowmass at Aspen, Colorado.

[3] T. Estlin, F. Fisher, D. Gaines, C. Chouinard, S. Schaffer, I. Nesnas, “Continuous Planning and Execution for an Autonomous Rover,” *Proceedings of the Third International NASA Workshop on Planning and Scheduling for Space,* Houston, Texas, October 2002.

[4] F. Fisher, T. Estlin, D. Gaines, S. Schaffer, C. Chouinard, R. Knight, “CLEaR: Closed Loop Execution and Recovery – A Framework for Unified Planning and Execution,” Technology and Science IND News Issue 16, pg. 15-20, September 2002.

[5] F. Fisher, R. Knight, B. Engelhardt, S. Chien, and N. Alejandre, “A Planning Approach to Monitor and Control for Deep Space Communications,” *Proceedings of the 2000 IEEE Aerospace Conference*, Big Sky, Montana, March 2000.

[6] S. Chien, R. Knight, A. Stechert, R. Sherwood, and G. Rabideau, “Using Iterative Repair to Improve the Responsiveness of Planning and Scheduling,” *Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling*, Breckenridge, CO, April 2000.

[7] R. Simmons and D. Apfelbaum, “A Task Description Language for Robot Control, “ *Proceedings of the Intelligent Robots and Systems Conference,* Vancouver, CA, October 1998.

[8] R. Castano, M. Judd, T. Estlin, R. C. Anderson, L. Scharenbroich, L. Song, D. Gaines, F. Fisher, D. Mazzoni, and A. Castano, “Autonomous Onboard Traverse Science System,” *Proceedings of the 2004 IEEE Aerospace Conference.* Big Sky, Montana, March 2004.

[9] I.A. Nesnas, A. Wright, M. Bajracharya, R. Simmons, T. Estlin, Won Soo Kim, "CLARAty: An Architecture for Reusable Robotic Software," SPIE Aerosense Conference, Orlando, Florida, April 2003.

[10] C. Urmson, R. Simmons, I. Nesnas, "A Generic Framework for Robotic Navigation," *Proceedings of the IEEE Aerospace Conference*, Montana, March 2003.

[11] A. Jonsson, P. Morris, N. Muscettola, K. Rajan, and B. Smith, "Planning in Interplanetary Space: Theory and Practice, " *Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems,* Breckenridge, CO, April 2000.

[12] R. Sherwood, S. Chien, D. Tran, B. Cichy, R. Castano, A. Davies, G. Rabideau, "Operating the Autonomous Sciencecraft Experiment," *Proceedings of the International Conference on Space Operations* (SpaceOps 2004). Montreal, Canada. May 2004.

[13] J. Bresina, K. Golden, D. Smith, and R. Washington, "Increased Flexibility and Robustness of Mars Rovers," *Proceedings of the International Symposium, on AI, Robotics and Automation for Space*, Noordwijk, The Netherlands, June 1999.

[14] E. Gat, "Integrating planning and reacting in a heterogeneous asynchronous architecture for mobile robots," in *SIGART Bulletin 2,* 1991, 70-74

[15] R. Bonasso, R. Firby, E. Gat, D. Kortenkamp, D. Miller, and M. Slack, "Experiences with an Architecture for Intelligent, Reactive Agents," *Journal of Experimental and Theoretical Artificial Intelligence Research*, 9(1), 1997.

[16] R. Alami, R. Chautila, S. Fleury, M. Ghallab, and F. Ingrand, "An Architecture for Autonomy," *International Journal of Robotics Research*, 17(4) April, 1998.

BIOGRAPHY

Dr. Tara Estlin *is a senior member of the Artificial Intelligence Group at the Jet Propulsion Laboratory where she performs research and development of planning and execution systems for rover automation and multi-rover coordination. Dr. Estlin is currently the manager of the Intelligent Decision Making for Rover Operations task and the An Onboard Scientist for Multi-Rover Scientific Exploration task. These efforts are developing capabilities for onboard rover-command generation, resource planning and scheduling, execution, and data analysis for single and multiple rovers. Dr. Estlin is Deputy Task Manager and a Cognizant Engineer for the CLARAty Rover Technology Testbeds task, which is a large effort at JPL that supports development of a generic robotic architecture and supports testing and validation of technology for the 2009 planned rover mission of Mars Science Laboratory. She received a B.S. in computer science in 1992 from Tulane University, an M.S. in computer science in 1994 and a Ph.D. in computer science in 1997, both from the University of Texas at Austin.*

Dr. Daniel Gaines *is a senior member of the Artificial Intelligence Group at JPL. His research interests are in integrated planning and execution and in machine learning to improve planning. Dr. Gaines received a Ph.D. in Computer Science from the University of Illinois at Urbana-Champaign. Before coming to JPL, Dr. Gaines was an Assistant Professor in Computer Science at Vanderbilt University. His work at JPL is primarily focuses on planning and execution techniques for planetary exploration rovers.*

Caroline Chouinard *is a software engineer at NASA's Jet Propulsion Laboratory. She has developed a variety of software components, including test simulators, visualization tools, and path planning packages for autonomous rover domains on the CLEaR team in cooperation with several other JPL teams. As part of the Artificial Intelligence Group, she also contributes to the development of automated planning and scheduling systems. She has enjoyed working in both the business software industry, specializing in AI related natural language processing tools, and also the scientific community, working on rovers. She received her B.B.A. in International Business and her B.A. in Computer Science, both from the University of Texas at Austin.*

Forest Fisher *Forest Fisher is a member of the technical staff in the Integrated Sequence System Engineering group at the Jet Propulsion Laboratory, California Institute of Technology. Forest is a Mission Sequence Engineer for the Mars Reconnaissance Orbiter, which is scheduled to launch in August 2005. Previously, Forest performed research and development of automated planning, scheduling and execution systems for autonomous rover operations and ground communication station automation. His research interests are in the areas of: monitor and control, and planning, scheduling and execution. Mr. Fisher was the PI of the "Unified Planning and Execution" project and the "Closed Loop Execution and Recovery" (CLEaR) project. He was the Software and Automation Lead for the Deep Space Terminal (DS-T) task. Mr. Fisher has been at JPL since 1995. He holds a B.S. in Computer Science from the University of Texas at Austin, and is in progress of a M.S. in Computer Science at the University of Southern California.*

Dr. Rebecca Castaño*, Supervisor, Machine Learning Systems (MLS) group at JPL, OASIS Technical team lead. She received her Ph.D. in Electrical Engineering from the University of Illinois with her dissertion in the area of computer vision. Dr. Castaño has been advancing the state* *of the art in onboard science analysis methods for the past five years and has been lead author on numerous publications in this field. From 1999 - 2001, Dr. Castano served as the application lead for phenomenological computational field geology efforts in the MLS Group. She is currently the technology lead for science data processing for the Autonomous Sciencecraft Experiment on the New Millenium Program's ST6 project. Dr. Castaño is also the Team Lead of the Onboard Autonomous Science Investigation System (OASIS) project and the subtask lead for Data Analysis on the automated Multi-rover Integrated Science Understanding System (MISUS). Her research interests include machine learning, computer vision and pattern recognition.*

Michele Judd, P.E*., OASIS Research Manager and senior technical staff member of JPL's Earth and Space Sciences Division. Ms. Judd received her B.S. in Petroleum Engineering from Stanford University and her M.L.A. in Organizational Development from Southern Methodist University. She currently manages two additional research tasks in earthquake modeling and simulation, "Numerical Simulations for Active Tectonic Processes: Increasing Interoperability and Performance" and "Complexity Computational Environments: Data Assimilation SERVO Grid." A nationally certified Quality Manager and licensed professional engineer, Ms. Judd has over 15 years of experience in project management, both within JPL and for Mobil Oil.*

Issa A.D. Nesnas*, Ph.D. is the principal investigator for the visual tracking task and the principal investigator for the Architecture and Autonomy Research collaborative task. His research interests include sensor-based robot control and software and hardware architectures for robotic systems. Issa received a B.E. degree in Electrical Engineering from Manhattan College, NY, in 1991. He earned the M.S. and Ph.D. degrees in Mechanical Engineering from the University of Notre Dame, IN, in 1993 and 1995 respectively. Prior to joining JPL in 1997, he worked as a senior engineer at Adept Technology Inc. Issa is a member of Eta Kappa Nu and Tau Beta Pi National Honor Societies.*