

Opportunistic Rover Science: Finding and Reacting to Rocks, Clouds and Dust Devils

Rebecca Castano, Tara Estlin, Daniel Gaines, Andres Castano, Caroline Chouinard, Ben Bornstein, Robert C. Anderson, Steve Chien, Alex Fukunaga, and Michele Judd

Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109
818-393-3544
firstname.lastname@jpl.nasa.gov

Abstract—The goal of the Onboard Autonomous Science Investigation System (OASIS) project at NASA's Jet Propulsion Laboratory (JPL) is to evaluate, and autonomously act upon, science data gathered by in-situ spacecraft, such as planetary landers and rovers.^{1,2} Using the FIDO rover in the Mars Yard at JPL, we have successfully demonstrated a closed loop system test of the rover acquiring image data, finding rocks in the image, analyzing rock properties and identifying rocks that merit further investigation. When the system on the rover alerts the rover to take additional measurements of interesting rocks, the planning and scheduling component determines if there are enough resources to meet this additional science data request. The rover is then instructed to either turn toward the rock, or to actually move closer to the rock to take an additional, close up, picture. In addition to these hardware integration successes, the OASIS team has also continued its autonomous science research by collaboratively working with other scientists and technologists to identify and react to other scientific phenomena - such as clouds and dust devils. Prototype dust devil and cloud detection algorithms were delivered to an infusion task which has refined the algorithms specifically for Mars Exploration Rovers (MER) and is integrating the code into the next release of MER flight software.

TABLE OF CONTENTS

1. INTRODUCTION.....	1
2. OASIS SYSTEM OVERVIEW.....	2
3. FEATURE EXTRACTION	2
4. DATA ANALYSIS.....	4
4. PLANNING AND SCHEDULING	5
5. SYSTEM TESTING	9
6. RELATED WORK	11
7. SUMMARY	13
REFERENCES	13
BIOGRAPHY	14

1. INTRODUCTION

With the ongoing success of the Mars Exploration Rovers (MER), scientists are starting to acknowledge the benefit that autonomous operations might bring to the table and are now requesting “smarter” rovers with a greater capability for autonomy for future missions.

There are a number of autonomous rover capabilities currently in development for future in-situ missions. One key capability, autonomous onboard science, continues to grow in importance as rover travel distances continue to dramatically increase. OASIS [1, 2, 3], an Onboard Autonomous Science Investigation System, is a JPL-managed project designed to maximize mission science on rover missions with long traverses.

OASIS is designed to operate onboard a rover identifying and reacting to serendipitous science opportunities. It analyzes data the rover gathers, and then prioritizes the data based on criteria set by the science team. At the next opportunity for transmitting data back to Earth, the data is already prioritized – ensuring that the most valuable data is sent first.

As OASIS is working to prioritize the data, it is also searching for specific targets it has been told to find by the science team. If one of these targets is found, it is identified as a new science opportunity and a “science alert” is sent to the planning and scheduling component of OASIS. After reviewing the rover’s current operational status to ensure that it has enough resources to complete its traverse and act on the new science opportunity, OASIS changes the command sequence of the rover.

The rover is instructed to stop its current traverse, locate the rock that triggered the science alert, and take additional data (e.g. color image, closer grayscale image, spectrometer reading) on that rock. In addition, the system now enables the rover to either turn and collect data on the identified target rock or to drive to the target so that closer measurements can be collected. Once it has completed this additional measurement, the rover reverts back to its original plan and continues on its traverse.

¹ 0-7803-9546-8/06/\$20.00© 2006 IEEE

² IEEEAC paper #1478, Version 2, Updated Jan. 3, 2006

The OASIS system includes three primary components. In Section 2, we briefly describe the system and these components. The remainder of the paper emphasizes recent advances in the OASIS system, describing new capabilities in each of the three components in Sections 3-5. Validation of autonomous systems is an ongoing challenge. We describe the ongoing performance testing of the full system in Section 6. In Section 7 we describe work related to the individual components as well as other autonomous systems under development and testing. Finally, a summary and conclusions are presented in Section 8.

2. OASIS SYSTEM OVERVIEW

To assess and subsequently prioritize the scientific value of a set of collected images, we must first extract the information found within the images. A geologist in the field gets information about a site by identifying geologic features including the albedo, texture, shape, size, color, and arrangement of rocks, and features of the topography such as layers in a cliff face. The geologist analyzes and assesses this data, and then takes some action based on the analysis, such as taking a sample or taking some additional measurement of an interesting rock.

In order for scientists to allow an autonomous system to help investigate the traversed region, the system must be able to perform, albeit in a very simple way, these same types of functions. This system thus is used to point out rocks or other objects of interest to the scientist.

There are three major components that comprise OASIS:

- **Extract Features from Images:** Enables extraction of features of interest from collected images of the surrounding terrain. This module both locates rocks in the images and extracts rock properties (features) including shape, texture and albedo.
- **Analyze and Prioritize Data:** Uses the extracted features to assess the scientific value of the planetary scene and to generate new science objectives that will further contribute to this assessment. This module consists of three separate prioritization algorithms that analyze the collected data and prioritize the rocks. A new set of observation goals is generated to gather further data on rocks that either conform to the pre-set specifications by the science team, or are so novel in comparison to the other rocks, that another data measurement may be required.
- **Plan and Schedule New Command Sequence:** Enables dynamic modification of the current rover command sequence (or plan) to accommodate new

science requests from the data analysis and prioritization module. A continuous planning approach is used to iteratively adjust the plan as new goals occur, while ensuring that resource and other operation constraints are met.

3. RECENT FEATURE EXTRACTION RESEARCH:

DUST DEVILS AND CLOUD DETECTION

Extensive effort on the OASIS task has been spent on extracting features relevant to geology as this is a heavy focus of in situ Mars exploration. Over the past year, however, algorithms to observe and opportunistically identify atmospheric phenomena were developed. The atmosphere of Mars is highly dynamic with phenomena including clouds and dust devils observed by the MER rovers. Currently, the MER mission monitors for these events by performing observation campaigns where sequences of images are acquired with the hope of imaging the event in one or more of the frames. There is no guarantee that the phenomena of interest will be captured. For example, only around 10-25% of the cloud campaign images collected have clouds in them. Downloading these images without the phenomena of interest represents an inefficient use of limited bandwidth.

The two most common methods for detecting dust devils are the comparison of two or more spectral bands of the scene and the detection of motion in the scene using a temporal sequence. In OASIS, we have selected the latter as it has application to imagery acquired with the Pancam, the Navcam and either set of Hazcams. In contrast, algorithms that require multiple spectral bands can only be used with the Pancam which is the only camera set equipped with a filter wheel. In theory, detecting motion in the scene is not equal to detecting dust devils as clouds also move. In practice, if image noise can be accounted for, the vast majority of changes in any sequence of images of a static scene of Mars will be caused by dust devils.

For the following descriptions, all the images used in event detection go through a process of border cropping and minimization. The border cropping avoids including border effects from the frame grabber in the estimation of image statistics while image minimization addresses both the constraint of low computational cost and does an initial noise reduction by a factor of $1/k^2$, i.e., pixels are averaged over a $k \times k$ window. In all cases, the standard deviation of the image noise is estimated as a difference of Laplacians [4].

To a great extent, the problem of detecting motion between two images is reduced to taking the difference between the two images and thresholding the result. Values above a

threshold correspond to regions of the images that have changed. If we know beforehand that the feature has a very distinctive albedo (e.g., a dust devil that is bright in a Martian scene that is dark) then the difference of the images produces a large difference in intensity that can be thresholded with confidence. Indeed, motion that is clear to a person can be confidently extracted using this method. The challenge for robust automated detection occurs when the difference in the intensity of the two images, at the location of the change, is comparable in magnitude to the noise of the image. This is the case whenever the dust devil is faint. In such cases, they can usually only be observed by a person when the sequence of images is played as a movie.

For such situations, the threshold cannot be selected easily as it will invariably consider image noise as change (false positive), actual change as noise (false negative) or both. As noise is a function of parameters over which we have no control (e.g., time of day, direction of camera with respect to sun, etc.), a fixed threshold is bound to perform correctly only for a narrow set of conditions.

The detection of faint dust devils in the image takes into account the noise of the image and uses the fact that a dust devil is bounded within a portion of the image. To reduce the noise, we detect changes in image I_i using the average of n images of the sequence, I_o , and I_{io} , the average of the $n-1$ images of the sequence that excludes I_i , i.e.,

$$I_o = \frac{1}{n} \left(\sum_{j=1}^n I_j \right) \text{ and } I_{io} = \frac{1}{n-1} \left(\sum_{j=1}^n I_j - I_i \right).$$

The difference of these images contains the average of the

$$\begin{aligned} I_o - I_{io} &= \frac{1}{n} \left(\sum_{j=1}^n I_j \right) - \frac{1}{n-1} \left(\sum_{\substack{j=1 \\ j \neq i}}^n I_j \right) \\ &= \frac{\left(I_i - \frac{1}{n-1} \sum_{\substack{j=1 \\ j \neq i}}^n I_j \right)}{n} \end{aligned}$$

image noise for all the areas where I_i was equal to the other images and the average of the image noise plus the change for the areas where I_i was different to the other images of the sequence, i.e., the intensity of the change is damped by a factor of $1/n$. Assuming that the major component of the image noise is zero-mean Gaussian noise, then the areas with no change tend to zero while the areas with change do not. Thus, although the intensity of the motion information has been damped, the motion can be detected because the areas with no change tend to zero faster than those with change. To complement this approach we use a threshold biased by the local noise and use blob filters that ensure that there are more than a given number of detections within a local region.

The algorithm was tested on 25 image sequences, all acquired on Mars using the left Navcam of the Opportunity rover. Each sequence had a length that varied between 6 and 20 images. The set of sequences was biased toward faint dust devils. Given these sequences, we analyzed all the possible subsets of a given number of contiguous images for 4, 6, and 8 consecutive images. The results are in Table 1. Figure 1 is a scene with several dust devils. The regions of each dust devil identified using the algorithm described are marked.



Figure 1. Result of motion detection in an image. Two of the dust devils are evident (3rd and 5th red box), while the other three require the sequence to play out to become apparent.

The algorithm was designed to take either a number of images at a time and produce the corresponding result or, to run in batch, taking one image at a time and analyzing large sequences. In the latter case, the average of the images is replaced by a trailing average.

The detection of clouds has limitations similar to those of the dust devil, i.e., clouds that can be easily observed in the sky can be detected easily while those that are extremely faint cannot, as their values approach the noise levels of the image. The set of images from MER available for the design of the algorithm consisted primarily, not of large sequences, like in the dust devil case, but mostly of single images or sequences of up to 3 images each. This prevents a consistent approach to detect clouds using motion. In addition, the motion may be variable with changing winds and other factors. Thus, the detection of clouds is done on individual images.

The approach to detect clouds is to assume that any large changes in the intensity of the sky of the image must correspond to clouds; this assumption holds true with the exception of large changes of local intensity due to zero-mean Gaussian noise (e.g., particularly noticeable at dusk and dawn) and large changes of global intensity due to camera effects (e.g., vignetting-like effect that darkens the corners of the image under low-light conditions). The first step to analyze the sky intensity is to segment the image using the sky detector previously developed under OASIS [5]. The result of the sky detector is used to mask out the ground and, if desired, to buffer an area above the skyline, which avoids illumination effects frequently found near the horizon.

Once the sky has been segmented, we search for changes in the sky by using an edge detector; strong edges indicate large gradients on the sky that are caused by the presence of clouds. Again, the threshold that determines the value of the edge that corresponds to a cloud is weighted by the noise of the image.

The algorithm was tested using a set of 210 images taken on Mars by Spirit and Opportunity. All of the images contained the sky, and most of the images contained both sky and ground. 47 of the images were images that an MER scientist had labeled as containing clouds, while the remaining 163 images were selected randomly from the set of all MER images that contained the sky, and manually verified as not containing clouds. The images with clouds were further divided into 29 images that contained evident clouds, 13 images that contained soft, hard-to-see wispy clouds and 5 images for which the scientists could not decide if there was a cloud or not. For this set, the algorithm detected 100% of the evident clouds, 100% of the wispy clouds and 60% of the ones in the undecided subset. Likewise, it stated correctly that there were no clouds in the no-cloud set 93.2% of the time. In summary, the algorithm was correct at

Table 1. Results from dust devil detection algorithm.

No. images	n-tuples	+	-	correct	False -	False +
4	279	120	159	237 (84.9%)	10	32
6	228	121	107	190 (83.3%)	14	24
8	180	116	64	155 (86.1%)	10	15

identifying whether or not a cloud was present in 93.3% of the test set; there were 3 false negatives and 11 false positives. A sample image and the clouds that were detected are shown in Fig. 2.

The dust devil detector and cloud detection algorithms were delivered to an infusion task, which further refined the algorithms for MER. The code has been integrated with the MER flight software, and is scheduled to be uploaded to the MER rovers in the spring of 2006 as part of the R9.2 software upgrade, pending final regression testing.

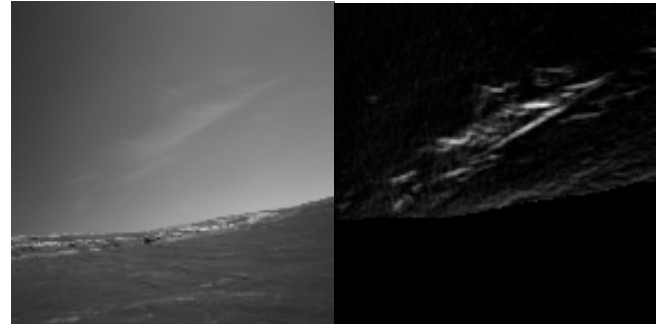


Figure 2. An example of cloud detection. Left image is the original image and the right image is the result of the cloud detection algorithm.

4. DATA ANALYSIS:

INSTRUMENT TARGET SELECTION

A number of rover remote sensing instruments have a very narrow field-of-view and thus require selection of specific focused targets for sampling. Such instruments include mini-TES, LIBS, and infrared point spectrometers. The typical scenario for selecting targets for these instruments is

to manually select the targets using data that has been previously downloaded. This means that targets can only be selected from the site at which the rover is in at the start of a day (for which data has already been downloaded the night before).

After a traverse day, samples from the new site could be collected by ‘blindly’ targeting at the end of the day in addition to collecting samples the following day after the science team has reviewed the imagery of the new site. Further, during a traverse, samples from these instruments can only be collected using blind sampling. By analyzing data onboard as the rover traverses and at the end of its traverse, targets for these instruments can be identified automatically. We have implemented a method for automatically selecting rock targets for sampling at the end of a traverse. This could be used, for example on the Mars Science Laboratory (MSL) to select targets for the ChemCam instrument to sample.

Two scenarios that lead to two different levels of targeting are being explored. In the first scenario, an image (Mast Cam – color or Navcam – grayscale) is taken and analyzed onboard. Regions on the surface that are rocks are identified and one or more rock regions selected as targets for other instruments. In the second level, a Mast Cam image is taken and analyzed, but in this case in addition to identifying the location of rocks, some properties of the rocks are estimated and a priority for the rock targets is determined based on the extracted properties. The second scenario is most useful when there are more rocks in a scene than can immediately be analyzed.

The approach is to first identify the rocks in the scene using the rock finder in the feature extraction component. Points on these identified rocks are then selected for targeting. The problem is distinctly different from accurately determining the outline of a rock because the result is the selection of a single point on each rock.

A prototype method has been tested on images forming the *Legacy* panorama and the *Mission Success A* panorama - both from the Spirit rover. The performance of the target selection algorithm was validated by using hand-labeled rocks as the ground truth. There were 22,079 hand-labeled rocks in the Mission Success A panorama and 17,596 hand-labeled rocks in the Legacy panorama. For this experiment, both the Mission Success A and Legacy panoramas were divided into 1024 x 1024 sub-images. Within each sub-image, a point was selected as the primary target. It was then assessed whether or not this point correctly represented a rock. We compared the method to random point selection.

The results from Table 2 show that for the *Mission Success A* panorama, the target point selected correctly belonged to a rock in 70% of the sub-images,. In contrast, if a point in the image was randomly selected there was only a 9%

chance of selecting a rock pixel. This represents a nearly 8X improvement in the probability that a desired target (rock) will be selected.

	<i>Mission Success A</i>	<i>Legacy</i>
Percent of images for which primary target was a rock	70	66
Percent of randomly selecting a rock pixel	9	12

Table 2. Accuracy of target selection algorithm.

5. PLANNING, SCHEDULING AND EXECUTION

Once the data analysis software has identified a set of new science targets, these targets are passed to onboard planning and scheduling software that can dynamically modify the current rover plan in order to collect the new science data. This component takes as input the new set of science requests, the current rover command sequence (or plan), and a model of rover operations and constraints. It then evaluates what new science tasks could be added to the current plan while ensuring other critical activities are preserved and no operation or resource constraints are violated.

CASPER Planner

Planning and scheduling capabilities are provided in OASIS by the Continuous Activity Scheduling, Planning and Re-Planning (CASPER) system [6, 7]. CASPER provides a generic planning and scheduling application framework that can be tailored to specific domains. Its components include:

- An expressive modeling language to allow the user to naturally define the application domain,
- A constraint management system for representing and maintaining domain operability and resource constraints,
- A set of search strategies and repair heuristics,
- A temporal reasoning system for expressing and maintaining temporal constraints,
- A graphical interface for visualizing plans, and
- A real-time system that monitors plan execution and modifies the current plan based on activity, goal, state and resource updates.

CASPER employs a *continuous planning* technique where the planner continually evaluates the current plan and modifies it when necessary based on new state and resource information. Rather than consider planning a batch process, where planning is performed once for a certain time period and set of goals, the planner has a current goal set, a current rover state, and state projections into the future for that plan. At any time an incremental update to the goals or current state may update the current plan. This update may be an unexpected event (such as a new science opportunity) or a current reading for a particular resource level (such as power). The planner is then responsible for maintaining a plan consistent with the most current information. And since things rarely go as expected during planetary surface operations, the planner stands ready to continually modify the plan.

CASPER models a spacecraft's resources (and states) while also defining domain constraints and hardware functionality. The model includes information on the environment, such as the timeframe of each plan, the types of instruments available on the rover, as well as information on rover constraints which may detail the minimum and maximum usage of onboard resources or transitions from possible execution states to renewal states at particular times of day. A model of rover activities is given to

CASPER, each of which estimates the predicted amount of resources and time which will be used, while also stating which hardware components will be required while performing the science. Some components may be atomic and cannot be used by two activities at the same time, while others are aggregate and may be used by a limited number of activities at any given time.

A plan consists of a set of grounded (i.e., time-tagged) activities that represent different rover actions and behaviors. Activities can be at different levels of abstraction, where low-level activities are typically sent to an executive system for execution on the rover hardware. For example, a plan typically contains several *traverse* activities that move the rover between different locations in order to visit science targets. Rover state in CASPER is modeled by a set of plan timelines, which contain information on states (such as rover position) and resources (such as power).

Timelines are calculated by reasoning about activity effects and represent the past, current and expected state of the rover over time. As time progresses, the actual state of the rover drifts from the state expected by the timelines, reflecting changes in the world. State updates are relayed back from sensors and the rover control software. As these

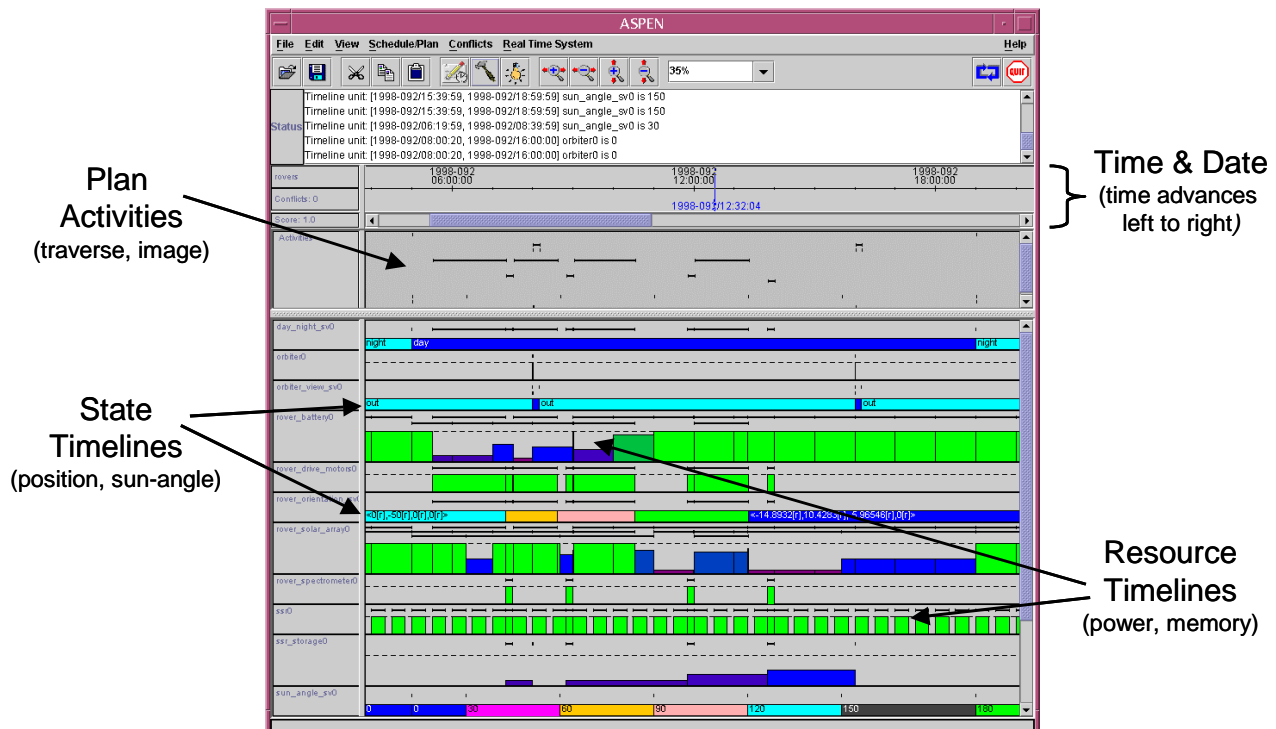


Figure 3: Sample rover plan displayed in planner GUI. Plan activities are shown as bars in upper portion of window, where bars represent the start and end time of each activity. State and resource timelines are shown in bottom portion of screen and show the effects of the plan as time progresses. Time is depicted as advancing from left to right.

updates are received, CASPER updates the relevant timeline models with actual state values, resource values, activity completion times, etc. Each of these updates may introduce problems into the current plan, such as a power oversubscription due to a long traverse or an instrument being in the incorrect position to perform a particular science reading. These problems (or plan conflicts) cause CASPER to perform plan modifications to bring the plan back into sync with the current state and set of goals. An example of a plan in the CASPER GUI is shown in Figure 3.

Initial Plan Generation

The responsibility of the planning and scheduling system in OASIS consists of both initial plan generation and dynamically modifying the plan during execution in response to unexpected events. For initial plan generation, a set of prioritized science goals can be provided to the planner and represent science targets that are being requested by mission scientists on Earth. Initial plan generation produces a plan for a certain time period (e.g., on Martian day) and typically more goals are provided that can be handled during that time period due to time or resource limitations. There are several methods of deciding which science goals to include in the initial plan, based on priority and other cost function parameters such as distance between targets and the sun angle at certain times of day. In our current system, we use a “strict priority” method as a model for generating the initial plan and the type of action to take when the plan must change. A strict priority method states that higher priority goals are always more desirable than lower priority goals, no matter how many lower priority targets could be included in a plan for even one higher priority target. Other rules or methodologies could also easily be adopted and have been used for previous tests.

To guarantee an optimal initial plan, based on our specified criteria, we use Depth First Branch and Bound (DFBnB) to order the set of science goals. To respect the strict priority rule, plans with the largest number of high priority targets are scored the highest. Bounding occurs when the priorities of the remaining goals to be added to the search tree’s “branch” are not as high as the priorities of the goals of the best plan found so far, when the accumulated distance cost of the “branch” is higher than the best plan found, and when the current “branch” oversubscribes time and/or resources. Temporal constraints on science goals are also handled during this process. The result is a conflict-free plan with as many of the highest priority targets included as possible.

Target ordering uses the shortest distance, fits in a limited plan timeframe, and uses only the amount of resources that are initially allowed. The DFBnB algorithm can also be easily adjusted for many different bounding criteria. For example, one future change might be to relax the strict priority algorithm specifically for plans with temporally constrained activities. If the highest priority target not

included in the current plan is constrained to start at 14:00 and the current time is 10:00, we may want to try to add other lower priority targets that could fit in the plan now, rather than wait for the opportunity to add-in the high priority target. With this expansion comes the need to define a timing tolerance for how close in time to the higher priority target we can be to allow this relaxed behavior.

Iterative Repair

To repair problems and ensure plan consistency during plan execution, CASPER uses an iterative repair algorithm, which classifies plan conflicts and attacks them individually. Conflicts occur when a plan constraint has been violated where this constraint could be temporal or involve a resource, state or activity parameter. Conflicts are resolved by using one or more plan modifications such as moving, adding, or deleting activities. One example of a conflict is when a new science activity oversubscribes a resource such as power or memory. Possible resolutions to this conflict might be moving the science activity to a part of the plan that doesn’t oversubscribe that resource, deleting the science activity, or moving and/or deleting other contributing activities. The decisions made are based on the conflict types and the built-in heuristics.

Continuous Optimization

When a deletion of a science goal is required due to unforeseen events during plan execution, the goal’s status is changed, but it is not permanently deleted from the plan. The goal becomes “requested”. Goals that could not be initially included in the plan due to time or resource limitations are also in this state. When favorable events occur which free up resources and time, these goals may be added back to the plan through the use of the continuous optimization procedure. This procedure is always trying to improve the score of the current plan by adding-back science goals or adding-in newly discovered goals. As each activity in the plan finishes, the state of the plan may be a little more or a little less accommodating to new goals. To ensure conflicts on hard constraints are always resolved, an unconflicted plan’s score is always better than a conflicted plan. Thus plan conflicts are always resolved before continuous optimization attempts to improve the plan further by adding additional science goals.

Handling Science Alerts

As described previously, onboard data analysis can use collected data, such as hazard camera images, to detect interesting rocks. When an interesting rock is discovered, a “science alert” is sent to the planner. The planner then attempts to add-in the appropriate traverse and image activities needed to achieve the opportunistic science while maintaining the state of the existing plan. See Fig. 4 for an example of an image taken in response to a science alert from a detected rock. In this example, the analysis system

was set to detect rocks of light albedo.



Figure 4. Sample image that was taken in response to a science alert on the JPL FIDO rover

In our current system implementation, we typically assign science alerts a lower priority than any initial (or ground-specified) science target. Science alerts are treated differently, because they are considered supplemental science. Further, science alerts must be handled in a short timeframe. If they are not planned for quickly, the rover could move far past them, creating a more difficult problem to solve. The current constraints and state of the plan may also make adding-in the science alert infeasible. To ensure the quick handling of science alerts, alerts have an expiration time. If the planner cannot add the science alert's activities into the plan within this time limit, the alert is removed from the list of potential goals. The continuous optimization cycle is then allowed to work on other types of plan improvement.

Previously, all science alerts kept the rover in the same physical location, only turning to face the new target. The planner can now change the planned path and plan for the rover to drive closer to the new target to better accommodate the data collection at the new goal. These drive-to science alerts introduce new problems, including where to insert them into the planned path, since the current time might not be the best position for them. They are inserted into their most optimal ordering based on shortest path distance, in most cases. If the new goal is close enough to an already scheduled goal, the same location is used for both goals, and the higher priority goal will be achieved first.

Executive

To provide executive capabilities in OASIS, we are using the Task Description Language (TDL) [8] executive. TDL expands abstract tasks into lower-level commands, executes the commands, and monitors their execution. It also provides direct support for exception handling and fine-

grained synchronization of subtasks. The TDL model also contains information about the rover and its environment. This includes constraint information, such as what preconditions must be true for an activity to begin execution, ways to decompose activities into commands based on current state, and exception handlers for some situations.

The planner maintains the scheduled plan activities and when their start time falls within a certain time window, sends them to the executive for execution. The executive receives activities from the planner and translates them into the appropriate rover commands, often breaking up one activity into several smaller tasks which will be monitored for progress and completion. The executive receives command status updates as well as state and resource information and can attempt to resolve local problems with smaller tasks, but will notify the planner when the task cannot be accomplished and a global, planner-level fix is required. Updates on activities, states and resources are also relayed to the planner since they may cause a need for further plan adjustment. The idea of separate control allows the planner to work on optimizing the future state of the plan while the executive monitors the progress of the current state [9, 10].

Path Planning

To provide spatial reasoning capabilities to CASPER and TDL, we are using a global path-planning module, which provides rover route information to both the planner and the executive based on a map of the rover's surrounding environment. This module is intended to give a global perspective of the rover's anticipated path as opposed to the local perspective that would be considered by obstacle avoidance software. We are assuming that for most rover operations some global map information would be available through orbital or descent imagery, or from panoramic imagery generated onboard the rover itself. We are also assuming this map information may be incomplete and certain terrain features and/or obstacles may be missing.

The path planner is queried for two main pieces of information. The first piece is estimated distances between science targets or other designated traverse waypoints. The second piece is a list of intermediate-waypoint coordinates that can be used to direct the rover's traverse to a particular target. Path-distance information is used by the planner and executive to estimate duration and power required for rover traverses between targets. Intermediate waypoints are used to track the rover's progress during a traverse. To provide path planning information to our system, we are currently using the D* (Dynamic A*) path planner, which produces paths in partially known or changing environments using an efficient and optimal algorithm [11].

Functional Layer

In order to interact with low-level rover hardware and necessary control software, the planning and executive components of OASIS are integrated with the Coupled Layered Architecture for Robotic Autonomy (CLARAty) [12], 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. For this work CLARAty has provided software for obstacle avoidance, navigation, vision, locomotion, and rover pose estimation. Through CLARAty, OASIS has been tested with several JPL rover platforms, including Rocky 8, and FIDO.

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 and many different opportunistic science events as well as items such as limited time and resources, resource usage uncertainty causing under or oversubscriptions of power and memory, large variations in traverse time, temporally constrained science goals, and unexpected obstacles blocking the rover's path.

Our testing scenarios typically consisted of a random 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 5 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 (shown in Figure 5 as the larger circles) represented targets that would be communicated by scientists on Earth. These targets were typically prioritized and for most 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.

A primary scenario element was dynamically identifying and handling opportunistic science events. For these tests, we concentrated on a particular type of event per scenario, which was either finding rocks with a high albedo measurement (i.e., light or white-colored rocks), or finding rocks with a high eccentricity value combined with a medium albedo (i.e., grey, elongated rocks). These settings were an example of using the data analysis algorithm to

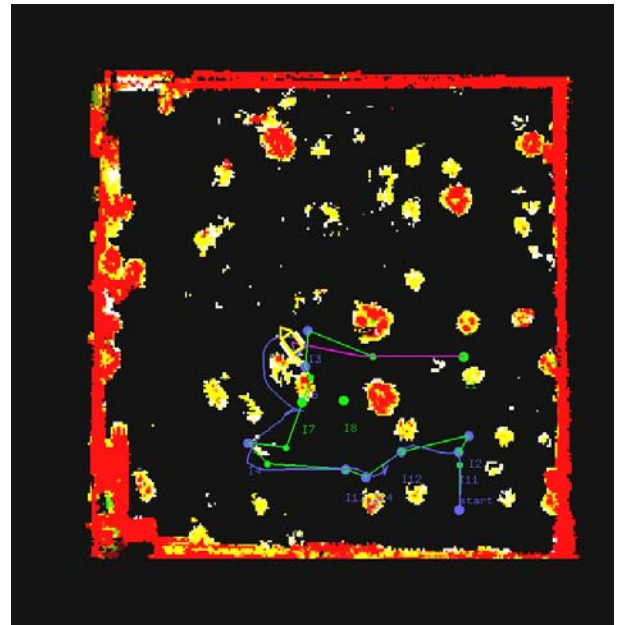


Figure 5: Sample scenario shown in the Grid Visualization Tool (GrViT). Green lines show the planned path of the rover. Blue lines show the real path, and pink lines show the path that is currently executing.

generate science alerts based on a 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 at or above 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 would be acquired. A sample image that was taken in response to a high albedo measurement science alert was shown previously in Figure 4.

Other important scenario elements included adding or deleting ground-specified science targets based on resource under or over-subscriptions. For instance, in some tests, the rover covered distances more quickly 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 activities, which often caused a power over-subscription, where enough power was not being preserved for later plan activities. 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 while still adhering to required resource and state constraints.

Testing in Simulation

Since testing with rover hardware can be an expensive and time-intensive process, we ran a large number of tests with the planning and execution system 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 multiple 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 statistics automatically gathered, including information 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 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 was shown in Figure 5. 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 addition to testing in simulation, a large number of tests were run in the JPL Mars Yard (shown in Figure 6) using different rover hardware platforms. For the past year the FIDO rover (shown in Figure 7) 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 subsystem 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 rock-finding 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 of these components to the onboard operating system (VxWorks) was not complete.



Figure 6: The JPL Mars Yard with terrain of various difficulties.



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

Tests in the Mars Yard have consisted of 20-50 meter runs over a 100 square meter area with many obstacles that cause deviations in the rover's path. A number of successful tests have been run on the rovers with random combinations of science targets, including science targets selected by observing MER scientists. 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.

Most recently, we have performed tests in the newly expanded Mars Yard. Figure 8 demonstrates the OASIS system identifying highly eccentric rocks with medium albedo. For this test, FIDO is placed at one end of the rock field (8.a.) and begins a long-range traverse to a science goal 15 meters north of the rover's current position. As FIDO traverses the terrain, rocks within the navigation images are identified and classified (8.b., 8.c., and 8.d.). Feature extraction detects a grey, elongated rock from the set of rocks in one image and sends an alert to the planner. The planner accommodates the alert by stopping the rover and inserting a new traverse activity and a new image activity to satisfy this opportunistic science goal. The rover drives closer to the discovered rock (8.e.), and captures an additional image (8.f.). FIDO then continues navigating the terrain toward its primary goal, continuously searching for pre-determined, scientifically interesting rocks, while the planner and executive manages the rover's current state.

7. RELATED WORK

The Autonomous Sciencecraft Experiment (ASE) [13] has demonstrated the capability of planning and data analysis systems to autonomously coordinate behavior of the EO-1 Earth orbiting satellite. ASE can also detect and respond to new science events, however it uses different detection and analysis algorithms. The Remote Agent Experiment (RAX) [14] was flown on the NASA Deep Space One (DS1) mission. It demonstrated the ability of an AI planning, execution and diagnosis system to respond to high-level spacecraft goals by generating and executing plans onboard the spacecraft. However, RAX did not incorporate data analysis to identify new science targets and used a batch approach to planning. Furthermore, since RAX and ASE were applied to spacecraft, neither handle issues associated with the uncertainty of surface navigation.

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.

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) [15]. 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,



a. FIDO rover at start of plan.



b. A stereo image captured during a navigation traverse.



c. The same stereo image with rocks outlined using the OASIS rock detection algorithm.



d. The OASIS feature extraction identifies an elongated, grey rock.



e. FIDO navigates to and performs an image at the identified target rock



f. The resultant data sample request (DSR) image.

Figure 8: The FIDO rover searches for rocks of medium albedo and highly eccentric shape.

the rover must be halted to wait for communication with ground.

Other similar approaches include Atlantis [16], 3T [17], and a robotic control architecture developed at the LAAS-CNRS lab [18] which all use a deliberative planner and an executive (or sequencing component) on top of a set of reactive controllers. These approaches have distinctly separate planning and execution techniques, 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. SUMMARY

Onboard autonomy and science data analysis will have a significant impact on future landed missions. Two elements of the OASIS system are being considered for use on current and future rover missions. The OASIS system is continuing to expand its capabilities for opportunistic science by increased functionality in the feature extraction, data analysis and planning and scheduling components. For acceptance of an autonomous system, thorough testing of a wide range of possible scenarios is necessary. We have found that testing in simulation can have extensive benefits.

Although the behavior of the rover in the field is not thoroughly modeled, many more scenarios can be explored in simulation. The system can be exercised more extensively in a number of ways that cannot be done with the full hardware system. Simulation testing led to the identification of several issues that were then addressed before hardware testing, making the hardware testing more effective.

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.

Finally, our work this year could not proceed without the support and advice of scientists outside of the team, in particular: Albert Haldemann (JPL) and Matt Golombek (JPL).

REFERENCES

- [1] Rebecca Castano, Robert Anderson, Tara Estlin, Dennis Decoste, Forest Fisher, Daniel Gaines, Dominic Mazzoni, and Michele Judd, "Rover Traverse Science for Increased Mission Science Return," *Proceedings of the 2003 IEEE Aerospace Conference*. Big Sky, Montana, March 2003.
- [2] Rebecca Castano, Michele Judd, Tara Estlin, Robert Anderson, Lucas Scharenbroich, Lin Song, Dan Gaines, Forest Fisher, Dominic Mazzoni, and Andres Castano, "Autonomous Onboard Traverse Science System," *Proceedings of the 2004 IEEE Aerospace Conference*, Big Sky, Montana, March 2004.
- [3] R. Castano, M. Judd, T. Estlin, R. C. Anderson, D. Gaines, A. Castaño, B. Bornstein, T. Stough, and K. Wagstaff, "Current Results from a Rover Science Data Analysis System," *IEEE Aerospace Conference*. Big Sky, MT. March 2005.
- [4] John Immerkaer, "Fast Noise Variance Estimation, Computer Vision and Image Understanding, 64:2," Sep 1996, pp. 300-302.
- [5] Anderson, R. C., R. Castano, M. Judd, T. Estlin, D. Gaines, D. Mazzoni, F. Fisher, B. Bornstein, A. Castano, L. Scharenbroich, L. Song, M. Gilmore, "Maximizing Rover Science Return Through Autonomous Onboard Data Analysis", *American Geophysical Union Fall Meeting*, Dec. 2003, *Eos Trans. AGU*, 84(46), Fall Meet. Suppl., Abstract P41B-0408.
- [6] Tara Estlin, Forest Fisher, Daniel Gaines, Caroline Chouinard, Steve Schaffer, and Issa Nesnas. Continuous Planning and Execution for a Mars Rover. Proceedings of the Third International NASA Workshop on Planning and Scheduling for Space. Houston, TX, Oct 2002.
- [7] Steve Chien, Russell Knight, Andre Stechert, Rob Sherwood, and Gregg 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.
- [8] R. Simmons and D. Apfelbaum, "A Task Description Language for Robot Control," *Proceedings of the Intelligent Robots and Systems Conference*, Vancouver, CA, October 1998.
- [9] T. Estlin, D. Gaines, C. Chouinard, F. Fisher, R. Castano, M. Judd, R. Anderson and, I. Nesnas, "Continuous Planning and Execution for an

Autonomous Rover,” *Proceedings of the 2005 IEEE Aerospace Conference*, Big Sky, Montana, March 2005.

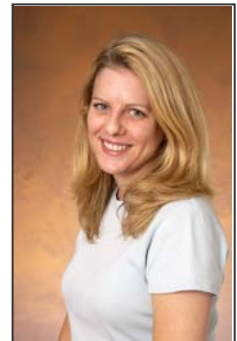
- [10] F. Fisher, T. Estlin, D. Gaines, S. Schaffer, C. Chouinard, R. Knight, “CLEaR: Closed Loop Execution and Recovery – A Framework for Unified Planning and Execution,” *Technology and Science IND News Issue* 16, pg. 15-20, September 2002.
- [11] Anthony Stentz, “Optimal and Efficient Path Planning for Partially-Known Environments,” *Proceedings of the IEEE International Conference on Robotics and Automation*, San Diego, CA, May 1994.
- [12] 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.
- [13] Chien, S., R. Sherwood, D. Tran, B. Cichy, G. Rabideau, R. Castano, A. Davies, D. Mandl, S. Frye, B. Trout, S. Shulman, D. Boyer, “Using Autonomy Flight Software to Improve Science Return on Earth Observing One,” *Journal of Aerospace Computing, Information, & Communication*, April 2005.
- [14] 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.
- [15] 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.
- [16] E. Gat, “Integrating planning and reacting in a heterogeneous asynchronous architecture for mobile robots,” in *SIGART Bulletin* 2, 1991, 70-74
- [17] 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.
- [18] 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. Rebecca Castano, 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 dissertation in the area of computer vision. Dr. Castano 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 Millennium Program's ST6 project. Dr. Castano 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.



Dr. Tara Estlin, OASIS Planning and Scheduling team lead, senior member technical staff. Dr. Estlin is a member of the Artificial Intelligence Group at the Jet Propulsion Laboratory where she performs research and development of planning and scheduling systems for rover automation and multi-rover coordination. Dr. Estlin is currently the PI of the “An Onboard Scientist for Multi-Rover Scientific Exploration” and the “Integrated Resource and Path Planning” projects. These efforts are developing capabilities for onboard rover-command generation, resource planning and scheduling, and data analysis for single and multiple rovers. She is also a member of the “Robotic Autonomy Architecture” project and is a past member of the Long-Range Science Rover (LRSR) project, which developed the Rocky 7 and Rocky 8 rovers. Dr. Estlin has been at JPL since 1998. 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.



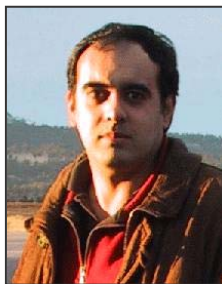
Dr. Robert C. Anderson, OASIS Science team lead, senior member technical staff. Dr. Anderson attended Old Dominion University in Norfolk, Virginia, where he received his Bachelor of Science degree in geology in 1979. In 1985, he received a Master of Science from Old Dominion University in geology with an emphasis on structural geology and mapping tectonic features surrounding the Tharsis region of Mars. In 1995, he received a Doctor of Philosophy from the University of Pittsburgh in geology with an emphasis on visible and near infra-red remote sensing. His Ph.D. research was centering on mapping Quaternary surfaces and soils around the Whipple Mountains of southwestern Arizona. Dr. Anderson worked on the successful Mars Pathfinder Project as science support for the Mineralogy and Geochemistry Science Operations Group. Currently Dr. Anderson is the Investigation Scientist for the Rock Abrasion Tool (RAT) and science support for Mission Operations on the Mars 03 mission. He works closely with the FIDO rover team and is presently the Science Team lead on the Onboard Autonomous Science Investigation System (OASIS) project. Dr. Anderson's research is centered on unraveling the geologic history of Mars.



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.



Dr. Andres Castano received his B.S. and M.E.E. degrees from the University of Los Andes, Bogota, Colombia, and his M.S. and Ph.D. degrees from the University of Illinois at Urbana-Champaign, all in Electrical Engineering. At Urbana-Champaign he worked at the Beckman Institute in visual servo-control of manipulators and omni-directional cameras. In 1998 he joined the Information Sciences Institute at the University of Southern California where he was the lead roboticist of the Conro project on reconfigurable robots. In 2000 he joined the Machine Vision Group at the Jet Propulsion Laboratory where he



currently works in feature detection and recognition. His research interests are robotics, computer vision and graph theory.

Caroline Chouinard is a member of the Artificial Intelligence Group at the Jet Propulsion Laboratory in Pasadena, California where she performs software development for planning and scheduling systems. She has contributed to the many stages of the rover's decision-making life-cycle, including initial plan generation, scheduling, execution, plan optimization and re-planning for events. She received a B.B.A in International Business and Finance in 1996 and a B.A. in Computer Science in 2000 both from the University of Texas at Austin. Her current interests include cognitive science, artificial intelligence, and graphics.



Ben Bornstein is a research programmer in the Machine Learning Systems group at the Jet Propulsion Laboratory in Pasadena, CA. He enjoys bringing machine learning techniques and considerable hacking (programming) skills to bear to solve problems in geology, bioinformatics, and systems biology. He has designed and implemented software systems for several Caltech biology labs, the Institute for Genomics and Bioinformatics (IGB) at UC Irvine, USC Children's Hospital, and JPL's Mars Exploration Rover (MER) project. He is also the inventor of and lead developer for libsbml, an open-source library for the Systems Biology Markup Language (SBML). Ben received a B.Sc. in Computer Science from the University of Minnesota Duluth in 1999 and is currently pursuing a M.Sc. in Computer Science at the University of Southern California.



Dr. Steve Chien is Technical Group Supervisor of the Artificial Intelligence Group and Principal Computer Scientist at the Jet Propulsion Laboratory, California Institute of Technology where he leads efforts in automated planning and scheduling for space exploration. Dr. Chien is also an Adjunct Associate Professor with the Department of Computer Science of the University of Southern California and a Visiting Scholar at UCLA. He holds a B.S. with Highest Honors in Computer Science, with minors in Mathematics and Economics, M.S., and Ph.D. degrees in Computer



Science, all from the University of Illinois.

Dr. Alex Fukunaga is a researcher in the Artificial Intelligence Group at JPL. His research interests are in artificial intelligence and combinatorial optimization. He received a Ph.D. in computer science from UCLA.



Michele Judd, P.E., OASIS Research Manager and senior technical staff member of JPL's Science 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.

