Autonomous Rovers for Human Exploration of Mars

John Bresina[‡]   Gregory A. Dorais[†]   Keith Golden
David E. Smith   Richard Washington[†]

NASA Ames Research Center, Mail Stop 269-2
Moffett Field, CA 94035-1000
{bresina, gadorais, kgolden, de2smith, richw}@ptolemy.arc.nasa.gov

Abstract

Autonomous rovers are a critical element for the success of human exploration of Mars. The robotic tasks required for human presence on Mars are beyond the ability of current rovers; these tasks include landing-site scouting and mining, as well as emplacement and maintenance of a habitat, fuel production facility, and power generator. These tasks are required before and also during human presence; the ability of rovers to offload work from the human explorers will enable the humans to accomplish their mission. Performing these tasks will require significant advances in rover autonomy and will require improvements in robustness, resource utilization, and failure recovery.

The Pathfinder mission demonstrated the potential for robotic Mars exploration, but at the same time indicated the need for more rover autonomy. The highly ground-intensive control with infrequent communication and high latency limited the effectiveness of the Sojourner rover. Advances in rover autonomy offer increased rover productivity without risk to rover safety.

Towards this end, we are developing an integrated on-board executive architecture that incorporates robust operation, resource utilization, and failure recovery. This work draws from our experience with the Deep Space One autonomy experiment, with enhancements to ensure robust operation in the face of the unpredictable, complex environment that the rover will encounter on Mars.

Our ultimate goal is to provide a complete agent architecture for rover autonomy. The complete architecture will include long-range mission and path planning, self-diagnosis and fault recovery, and continual monitoring and adjustment of execution resources. The architecture will enable robust operation over long ranges of time and distance, allowing rovers to perform complex tasks in a planned and opportunistic manner and serve as an intelligent, capable tools for human explorers.

1   Introduction

Human exploration of Mars in the relatively near future is becoming increasingly likely. NASA has been investigating possible manned missions, and there is widespread public interest in Mars due to factors such as the recent debate about evidence of life on Mars, the success of missions such as Mars Pathfinder, and advocacy by visionaries such as Robert Zubrin [Zubrin and Wagner, 1996].

However, colonizing Mars will not happen by itself; we need to make it happen. In order to make future missions safe, effective and affordable, we must anticipate and develop the technologies that will be needed. In this paper, we discuss the advances in Mars rover software that will be needed to support future missions, and we discuss the steps we are taking to realize those improvements. In particular, we argue that rovers need to be significantly more autonomous.

In the next section, we discuss anticipated missions leading to and following human presence on Mars, and explain why those missions require autonomous rovers. In Section 3, we discuss the current state of the art in rover autonomy and where we would like to see it extended. In Section 4, we discuss the work we are doing to improve rover autonomy.

[‡] Recom Technologies
[†] Caelum Research Corporation

2 Future Mars missions

2.1 Precursor missions

Prior to sending humans to Mars, we need to send machines: to continue learning about conditions on Mars, to test out technologies that will be needed for human missions, and to set up the facilities that will be needed by humans when they first arrive.

We need to learn more about the "Red Planet," both to keep the astronauts safe and to maximize the impact of the first human mission. For example, we still know little about the composition of the Martian soil. We need to learn more — both to guard against unforeseen chemical hazards and to determine whether activities such as growing plants, making bricks or extracting chemicals is feasible. Similarly, a more detailed study of the Martian atmosphere, winds, and dust storms will give us useful information to prepare us for such tasks as in-situ propellant production, deployment of solar arrays, and navigation.

Even more importantly, we need rovers to prepare the way for humans. Rovers may be needed to set up facilities for propellant production, to set up power plants, and to begin resource collection. After humans arrive, rovers will still be useful to augment human capabilities and perform tasks in the hazardous Martian environment.

All of these tasks call for rover autonomy. These missions will run for a long time — some of the upcoming rover missions are expected to last a year — and communication with Earth will be infrequent and with high latency. Controlling the rovers manually from Earth would incur a huge cost and would achieve much lower returns due to the time the rover is idle waiting for instructions. Given that rovers have a limited lifetime, such wasted opportunities translate to a much lower return on our investment.

2.2 First human exploration

When the first humans land on Mars, they will be in the company of rovers — some that were waiting for their arrival and some that arrived with them. These rovers will amplify the productivity of the crew by carrying out mundane or dangerous tasks that the limited crew will not be able to handle alone. Rovers can traverse long distances, gathering samples for the crew to analyze, and can act as "eyes and ears" of the crew, allowing them to "explore" distant terrain, safe from radiation and other perils.

In principle, these tasks could be accomplished by non-autonomous rovers, teleoperated by the crew on Mars or the operations team on Earth. In practice, the crew will be busy with more important tasks and will not have the time to continuously monitor the rovers and control their every move. Thus, autonomous rovers can greatly amplify the productivity of the small crew. It is also advantageous for the rovers to be responsible for their own well-being: to avoid getting damaged, and to diagnose and correct recoverable software and hardware failures. This way, the crew does not have to spend much time babysitting the rovers, and can concentrate on doing science and survival.

2.3 Colonization

Once humans settle on Mars, the tasks for rovers will become more complex. They will be needed for activities such as mining, building habitats and other structures, and maintaining the life-support system of the humans (which includes the rovers themselves). These tasks demand autonomy, since teleoperating rovers to perform such tasks would most likely be more difficult than performing the tasks themselves. The rovers will also need to respond quickly to hazards, such as falling objects.

3 Current and future rover autonomy

3.1 Current rovers

Current rover missions, such as Pathfinder [Mishkin *et al.*, 1998], depend almost entirely on ground-based commanding and employ only enough on-board autonomy to safely follow uplinked commands. If anomalous situations arise, the rover waits for updated commands to be uplinked. In practice, this approach leads to high mission cost and missed science opportunities. The ground operations team for the Pathfinder mission had

to adjust themselves to Mars time for the entire mission and had much more access to expertise in diagnosing and debugging problems than will be possible in future missions. With upcoming missions expected to last a year or more, expecting people to work such hours and maintain such vigilance is not feasible. Furthermore, since almost all of the intelligence behind the Sojourner rover was on Earth, the command sequences that Sojourner executed were quite fragile. There were many cases when something went wrong, forcing the humans on Earth to spend a day or more diagnosing and fixing the problem.

3.1.1 Robust Operation

The impressive success of the Pathfinder mission and in particular the Sojourner rover raises hopes of realizing the vision of rovers roaming the surface of Mars, performing tasks nearly or completely autonomously. At the same time, the mission points out the gaps between that vision and current reality.

Currently, a rover command sequence is specified to the lowest level of detail and leaves few, if any, choices to be made at execution time; hence it admits only one, or a very small number of, valid execution behaviors. This simplifies the execution process, but does not allow execution to be responsive to the dynamic status of the rover and the environment. This inflexibility can cause reduced productivity and execution failures.

For example, in Sol[1] 22 of the mission, Sojourner received the following challenging instruction sequence.

1. Back up to the rock named Soufflé.

2. Place the arm with the spectrometer on the rock.

3. Do extensive measurements on the rock surface.

4. Perform a long traverse to another rock.

At the next communication time, the news came in from Mars. The good news was that the sequence was executed to completion, including the longest traverse ever done in one day, a world's record for Mars. However, there was bad news with the good. The spectrometer data was useless, because the rover stopped short of the rock, and the spectrometer was left hanging out in mid-air rather than placed on the rock. The rock was never again visited.

3.1.2 Resource Utilization

The generation of uplink plans is based on estimated profiles, over time, of capacity and demand for each resource. However, there is great inherent uncertainty in the operating environment and its impact on performance of rover hardware components. For example, the power demand of a traversal is highly dependent on the incline, roughness, and traction of the terrain. Without an accurate estimate of power demand, battery capacity at a given time cannot be accurately predicted. Furthermore, battery capacity will also depend on how much time is spent in shade, which in turn depends on the surrounding landscape. In response to this uncertainty, worst case estimates of resource usage and availability are typically used; however, even tight worse cases bounds may be difficult to predict. Because demand estimates are too pessimistic, execution of such plans often results in reduced rover productivity through wasted time and lost opportunities. On the other hand, overly optimistic estimates of resource usage and availability result in higher risk to basic rover safety and a greater chance of broken plans.

3.1.3 Failure Recovery

Current rovers have very limited capabilities for recovering from faults or anomalous situations. The rover's response to most execution failures is to halt all activity and wait for the ground operation team to determine what went wrong and uplink a recovery plan. Depending on the nature of the anomalous situation and the quality of the downlinked information for the purposes of diagnosis, this ground-based recovery process can cost days of rover idleness and lost science opportunity. For example, a rover traverse may fail with a high wheel current combined with the wheel encoder showing no movement. In this case the wheel may be stuck on a rock or the encoder may be broken; the ground team will need to uplink diagnostic sequences

---

[1] Martian day

to determine which is the case, and then recovery sequences to remove the rover from the rock if that is the problem. Valuable science opportunities are lost during this time. Even transient problems such as an overheated motor can cause plan failure, where perhaps a brief pause to cool down would be sufficient to remedy the problem.

3.2   The next generation

The next generation of rovers, the first of which is expected to launch in 2003, will be more flexible than Sojourner, but will be shy of full autonomy. Instead of simple command sequences, the rovers will execute complex contingency plans, which tell the rover explicitly what to do if something goes wrong. They will also execute plans more robustly, so minor problems such as motor overheating do not cause failure. Finally, they will be able to identify and diagnose internal faults, and recover from simple failures.

To imagine how these rovers will behave, consider again the problem of backing up to a rock, deploying a sensor, gathering data and moving on. We can imagine what a smart rover would need to do in this case. First of all, the operation of backing up on Sojourner was brittle because it used a simple "try three times" strategy to back up. We can imagine a more robust operation of backing up until contact, with some timeout to avoid an insurmountable problem. The backing up operation should include the ability to try different approach paths if obstacles block the planned route; the rover should be able to take a moment to let an overheating motor cool down rather than abandoning the operation; the rover should notice that a wheel seems to be malfunctioning and pulling the rover off-path, and shift control algorithms to compensate for that. Certainly the rover should notice contact or not before doing hours of measurements, and have alternative plans in case it cannot make contact with the rock despite its best efforts. While it is performing measurements, the rover should monitor its energy level to make sure that it will be have enough energy left to send its data to Earth at the beginning of the morning, and potentially cut short its monitoring if it ends up in the shade of a larger rock and cannot charge its battery enough to complete the task and communicate.

A smart rover that behaves as described above is beyond the current state of the art in deployed missions. However, the technology needed to construct such a rover is within reach using artificial intelligence technology in development today. We are working toward that end by using components of the Remote Agent, a system developed for autonomous spacecraft, and applying them to rovers.

3.3   Future vision

In the future, we would like to see rovers that are capable of full autonomy. These rovers will take very high level instructions from human operators and will be able to achieve those goals with no further supervision, even in dynamic and uncertain environments. These rovers will be self-diagnosing and self-repairing, and will be capable of detecting gradual degradation, adjusting internal parameters accordingly, and subjecting themselves to preventive maintenance to avoid catastrophic failure. For example, motors are subject to wear out over time and solar panels accumulate dust and are gradually damaged by UV. Rovers will be able to automatically replan when unexpected problems or opportunities arise.

We are building toward this vision in our research on autonomy. While we are not there yet, it is reasonable to expect such capabilities in the rovers that accompany humans in exploration of Mars.

4   Increasing rover autonomy

To achieve our goal of greater rover autonomy, and to work toward the goal of full autonomy, we are focusing on the following capabilities:

- Robust operation via execution and monitoring of flexible, contingent mission plans

- Optimal resource utilization via continuous resource capacity assessment, demand prediction and dynamic allocation re-scheduling

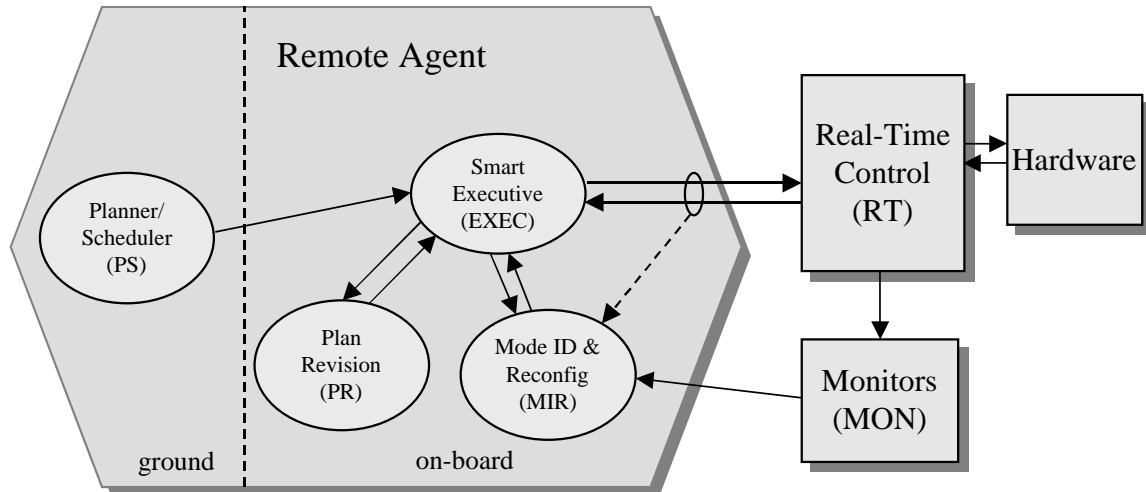- Advanced failure recovery via active sensing/testing and reasoning from first principles.

Figure 1: Remote Agent architecture embedded within rover real-time software.

To realize these capabilities, we are applying components from the Remote Agent (RA) architecture to the rover domain.

In this section, we describe the RA and how it is being enhanced to meet the particular requirements of the rover domain. We describe how the RA is embedded in the overall system, provide an overview of the components of the RA, and describe its high level functioning.

4.1   Architecture Overview

The Remote Agent (RA) is an architecture designed for intelligent control of complex systems. The RA has been applied to the requirements of spacecraft control [Muscettola *et al.*, 1998, Bernard *et al.*, 1998], but since it is composed of heterogeneous, state-of-the-art, general-purpose components, it is appropriate for many of the characteristics of rover control as well.

The relationship between the RA and the system in which it is embedded is portrayed in Figure 1. When viewed as a black box, RA sends out commands to the *real-time control* software (RT). RT provides the primitive skills of the autonomous system, which take the form of discrete and continuous real-time estimation and control tasks. RT responds to high-level commands by changing the mode of a control loop or state of a device and sending a message back to RA when the command has completed.

In addition, the status of all RT control loops are passed back to RA through a set of *monitors* (MON). The monitors discretize the continuous data into a set of qualitative intervals based on trends and thresholds, and pass the results back to RA. The abstraction process is fast and simple, involving discretizing a continuous variable using thresholds on an absolute or relative scale.

The RA comprises three major reasoning components: a temporal planner/scheduler (PS), a smart executive (EXEC), and a model-based diagnosis and reconfiguration system (MIR). In the rover domain, PS is a ground-based planner, which is given high-level goals corresponding to the mission goals and generates a schedule. Some limited plan revision (PR) capabilities are on-board, and we envision that more of the planning work will migrate on-board as both computational power and the need for autonomy increase. The schedule is sent to EXEC on-board, which decomposes the general schedule into lower-level spacecraft commands that implement the schedule elements and respect the constraints between them. As described above, these commands are sent to RT, with results coming back via MON into MIR. MIR's *mode identification* layer, MI, infers the system state from the monitored information and updates the state for EXEC. If commands fail or schedule constraints are violated, EXEC tries to recover using retries or local recoveries, potentially calling MIR's *mode reconfiguration* layer, MR, to produce a recovery plan. PR adapts the plan as the situation changes.

4.1.1 Planner

Throughout a mission, detailed mission operations plans must be constructed, validated, and uplinked to a spacecraft or rover. Currently a mission operations plan takes the form of a rigid, time-stamped sequence of low-level commands. Unfortunately, there is uncertainty about many aspects of task execution: exactly how long operations will take, how much power will be consumed, and how much data storage will be needed. Furthermore, there is uncertainty about environmental factors that influence such things as rate of battery charging or which scientific tasks are possible. In order to allow for this uncertainty, current plans are based on worst-case estimates and contain fail-safe checks. If tasks take less time than expected, the spacecraft or rover just waits for the next time-stamped task. If tasks take longer than expected, they may be terminated before completion. In fact, all non-essential operations may be halted until a new command sequence is received. All of these situations result in unnecessary delays and lost science opportunities.

PS can actively plan for, and take advantage of, possible contingencies. Thus, if an operation takes longer than a certain amount of time, or the power remaining drops below a specified value, a different pre-planned sequence of operations can be performed. Building contingency plans is, in general, intractable, and so contingency planners tend to be slow. [Draper *et al.*, 1994, Pryor and Collins, 1996, Weld *et al.*, 1998]. To overcome this problem, PS uses *Just in Case* planning.

The Just in Case (JIC) technique was originally developed to generate contingent observation schedules for automated telescopes [Drummond *et al.*, 1994]. The basic idea is to take an existing schedule and look for the places where it is most likely to fail. The JIC scheduler then generates alternative schedules for each of those situations. The JIC scheduler starts with a sequence of tasks, where each task must be performed in a certain temporal window. However, there is uncertainty in how long a particular task may take, and this can lead to potential failures of the schedule. For example, an execution failure could result if one task finished sufficiently late that the next task's start window has already passed.

The JIC approach is the most straightforward way to introduce contingency planning into mission operation planning. Using this idea, a plan is examined to determine the most likely places that it could fail. As with telescope scheduling, one source of potential failures is due to uncertainty in task duration — a task might take longer than expected, preventing its successor from starting in the required time window. Alternatively, a task might take less time than expected so the spacecraft or rover would sit idle until the beginning of the time window for the next task. If these failures were sufficiently likely a contingent plan could be built for these situations (just as in JIC scheduling for telescopes).

For spacecraft and rover operations, uncertainty about other resources can also lead to potential failures in a plan. For example, if a task utilizes more power than expected, or the battery has not charged as much as expected, there may be insufficient power available for the subsequent tasks. However, there may be enough power left to do other useful tasks. JIC could also be employed to generate contingent plans for such situations. Equally useful would be contingent plans for situations where more of a resource is available than expected. In this case, a JIC contingent plan could take advantage of the unexpected surplus to perform additional tasks, or perform tasks that are more power hungry.

4.1.2 Smart Executive

EXEC is responsible for interpreting the command sequence coming from the planner or an external source (e.g., ground control), decomposing high-level actions into low-level commands for the real-time system RT, checking run-time resource requirements and availability, and performing fault recovery in coordination with the fault recovery system MR.

The input to EXEC is a contingent command sequence. The contingencies are represented as a branching plan structure, plus a library of contingent plan fragments and suffixes that are invoked on plan failure, unexpected opportunities, or conditions such as resource shortfall or component degradation. At each point in time, EXEC may have a choice of multiple possible plan steps corresponding to the eligible plan branches or plan library elements. EXEC chooses the plan step with the highest estimated expected utility (computed over the remainder of the plan). This utility is initially computed by the ground planner PS, but may be updated by the plan revision component PR at run time to reflect changes in resource availability, system state, or the environment.

Each of the plan steps may be a high-level action that needs to be decomposed into low-level commands

for RT. This decomposition uses constructs for robustly executing commands, including catching failure conditions and performing local recoveries and retries. Thus low-level conditionality may be represented at a level below the granularity of the planner, reducing the computational complexity of the planning problem while guaranteeing the correct execution semantics. For example, the seemingly simple command "move to a point 5 meters straight ahead" may include the ability to pause if a motor overheats, try multiple routes if obstacles block the path, and switch to alternative control algorithms if a wheel becomes blocked.

EXEC extends its own robust activity mechanism using the fault recovery mechanisms of MIR. EXEC may request a recovery plan from MIR (see Section 4.1.3 below), and it will integrate that plan in with its currently executing plan.

EXEC also monitors run-time usage and availability of resources. PS constructs its plans using expected resource availability profiles, but these may change at execution time. For example, a power profile for the day will depend on expected solar exposure for the solar arrays, but this may change based on long traverses on tilted terrain (more or less power available depending on the angle), dust storms, shadows, etc. A resource manager component takes the resource usage expectations for the plan's activities and the current best information for resource availability, and signals potential (future) and real (current) conflicts based on that information. EXEC responds to those signals as to other failures, using its fault response mechanism.

4.1.3 Mode identification and Reconfiguration

MIR is a discrete, model-based controller that uses a single, declarative model of the rover for both mode identification (inferring the internal state) and reconfiguration [Williams and Nayak, 1996] (changing the system to a more desirable state). Like EXEC, MIR runs as a concurrent reactive process. MIR itself contains two components, one for *Mode Identification* (MI) and one for *Mode Reconfiguration* (MR).

MI is the sensing component of MIR's model-based reconfiguration capability. Based on sensor readings and the commands executed on the rover, MIR uses its model of the rover to infer the most likely current state. MI also provides a layer of abstraction to the executive: it allows the RA to reason about the state of the rover in terms of component modes, rather than in terms of low level sensor values.

MR serves as a *recovery expert* to EXEC, taking as input a *recovery request*, and returning a sequence of operations that, when executed starting in the current state, will move the executive into a state satisfying the properties required for successful execution of the failed activity.

MIR uses algorithms adapted from model-based diagnosis [de Kleer and Williams, 1987, de Kleer and Williams, 1989] to provide the above functions. MIR extends the basic ideas of model-based diagnosis by modeling each component as a finite state machine, and the whole rover as a set of concurrent, synchronous state machines. Modeling components as finite state machines allows MIR to effectively track state changes resulting from executive commands. Modeling the rover as a concurrent machine allows MI to effectively track concurrent state changes caused either by executive commands or component failures.

Interaction with the environment In the past, MIR has been used to model systems, such as spacecraft, in which the environment outside the spacecraft can be effectively ignored. Spacecraft follow known trajectories, free of obstacles, in which the external environment can be reduced to a few simple variables, such as the relative position of the Earth. On rovers, interaction with the environment is central to many of the possible faults. Dust accumulates on the solar panels, the rover passes into the shadow of large rocks, or gets caught on small ones. In order to handle failures involving external factors, we need to add the capability to reason from first principles about the rover and its interaction with the environment. The proposed reasoning approach will use a combination of model-based deduction and hybrid simulation.

Active sensing and testing In support of this reasoning, we will add the capability to perform active sensing and testing in order to narrow the candidate situation assessments (diagnoses) and in order to evaluate the utility of alternative recovery plans. The sensor information that MI can passively acquire is not always adequate to allow an unambiguous diagnosis. In general, it may be ambiguous whether a given component has failed, or the sensor responsible for measuring that component has failed. If an encoder indicates that a wheel drive motor isn't turning, or is turning too slowly, that could be a sensor error; the encoder may be skipping counts or entirely dead. The true state of the rover may be determined by

performing experiments designed to eliminate certain diagnoses. For example, if the wheel does not appear to be turning, the rover could try backing up to see if the wheel is caught on a rock. It may also look at the motor currents to see if the motor appears to be stalled, which could indicate a stuck wheel or seized bearings. If it can't establish that the motor is stalled, it could try turning only that wheel, and use other sensors to detect movement. If so, that would indicate that the encoder has failed.

In support of this active testing, the rover can make use of its models, both to determine when there are multiple competing diagnosis and to identify activities it can perform that will rule out or confirm certain hypotheses. Reasoning about the information to be gained by executing actions exceeds the ability of the original MIR system, but we are working to provide that capability.

5   Conclusions and Related work

We believe that autonomous rovers will be an essential part of a program of human exploration of Mars. We have described our efforts underway to make rovers more autonomous, building on earlier work for autonomous spacecraft. We are developing an integrated on-board executive architecture for robust operation, focusing in particular on problems of contingent planning, run-time resource management, and active sensing. Much work remains to be done, but we believe the ultimate goal of autonomous Mars rovers can be realized.

The motivation and examples that drive the rover-specific research come from the Pathfinder mission [Mishkin *et al.*, 1998], which will remain the reference against which all near-term rover work will be compared. The work described in this paper is directed towards future NASA missions to Mars, but it is first being tested on the NASA Ames Marsokhod rover [Christian *et al.*, 1997]. Other robotics work specifically designed for space applications includes the JPL Long Range Science Rover project [Volpe *et al.*, 1997], CMU Nomad project, and the LAAS IARES project [Chatila *et al.*, 1995].

The technology builds on the Remote Agent Experiment on the Deep Space One mission, where the Remote Agent is a technology experiment that will control the spacecraft during a week-long period [Muscettola *et al.*, 1998, Bernard *et al.*, 1998]. Other agent architectures for controlling real-time systems include CIRCA [Musliner *et al.*, 1993] and 3T [Bonasso *et al.*, 1995, Bonasso *et al.*, 1997]. The attitude and articulation control subsystem (AACS) on the Cassini spacecraft [Brown *et al.*, 1995] has explicit software modules for context-dependent command decomposition, resource management, configuration management, and fault protection, and provides an example of the state of the art in deployed spacecraft autonomy.

References

[Bernard *et al.*, 1998] D. E. Bernard, G. A. Dorais, C. Fry, E. B. Gamble Jr., R. Kanefsky, J. Kurien, W. Millar, N. Muscettola, P. P. Nayak, B. Pell, K. Rajan, N. Rouquette, B. Smith, and B. C. Williams. Design of the remote agent experiment for spacecraft autonomy. In *Proceedings of the IEEE Aerospace Conference*, Snowmass, CO, 1998. IEEE.

[Bonasso *et al.*, 1995] R. P. Bonasso, D. Kortenkamp, D. Miller, and M. Slack. Experiences with an architecture for intelligent, reactive agents. In *Proceedings of IJCAI-95*, 1995.

[Bonasso *et al.*, 1997] R. P. Bonasso, D. Kortenkamp, and T. Whitney. Using a robot control architecture to automate space shuttle operations. In *Proceedings of IAAI-97*, pages 949–956, 1997.

[Brown *et al.*, 1995] G.M. Brown, D.E. Bernard, and R.D. Rasmussen. Attitude and articulation control for the Cassini spacecraft: A fault tolerance overview. In *14th AIAA/IEEE Digital Avionics Systems Conference*, Cambridge, MA, November 1995.

[Chatila *et al.*, 1995] R. Chatila, S. Lacroix, T. Simeon, and M. Herrb. Planetary exploration by a mobile robot: mission teleprogramming and autonomous navigation. *Autonomous Robots*, 2(4):333–344, 1995.

[Christian *et al.*, 1997] D. Christian, D. Wettergreen, M. Bualat, K. Schwehr, D. Tucker, and E. Zbinden. Field experiments with the Ames Marsokhod rover. In *Proceedings of the 1997 Field and Service Robotics Conference*, December 1997.

[de Kleer and Williams, 1987] J. de Kleer and B. C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32:100–117, 1987.

[de Kleer and Williams, 1989] J. de Kleer and B.C. Williams. Diagnosis with behavioral modes. In *Proceedings of IJCAI-89*, pages 1324–1330, August 1989.

[Draper *et al.*, 1994] D. Draper, S. Hanks, and D. Weld. Probabilistic planning with information gathering and contingent execution. In *Proc. 2nd Intl. Conf. AI Planning Systems*, June 1994.

[Drummond *et al.*, 1994] M. Drummond, J. Bresina, and K. Swanson. Just-in-case scheduling. In *Proceedings of the 12th National Conference on Artificial Intelligence*, 1994.

[Mishkin *et al.*, 1998] A. H. Mishkin, J. C. Morrison, T. T. Nguyen, H. W. Stone, B. K. Cooper, and B. H. Wilcox. Experiences with operations and autonomy of the Mars Pathfinder microrover. In *Proceedings of the IEEE Aerospace Conference*, Snowmass, CO, 1998. IEEE.

[Muscettola *et al.*, 1998] N. Muscettola, P. P. Nayak, B. Pell, and B. C. Williams. Remote agent: To boldly go where no AI system has gone before. *Artificial Intelligence*, 103(1/2), August 1998. To Appear.

[Musliner *et al.*, 1993] D. Musliner, E. Durfee, and K. Shin. Circa: A cooperative, intelligent, real-time control architecture. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(6), 1993.

[Pryor and Collins, 1996] L. Pryor and G. Collins. Planning for contingencies: A decision-based approach. *J. Artificial Intelligence Research*, 1996.

[Volpe *et al.*, 1997] R. Volpe, J. Balaram, T. Ohm, and R. Ivlev. Rocky7: A next generation Mars rover prorotype. *Journal of Advanced Robotics*, 11(4), December 1997.

[Weld *et al.*, 1998] D. S. Weld, C. R. Anderson, and D. E. Smith. Extending graphplan to handle uncertainty & sensing actions. In *Proceedings of AAAI-98*, pages 897–904, 1998.

[Williams and Nayak, 1996] B. C. Williams and P. P. Nayak. A model-based approach to reactive self-configuring systems. In *Proceedings of AAAI-96*, pages 971–978, 1996.

[Zubrin and Wagner, 1996] R. Zubrin and R. Wagner, editors. *The case for Mars: The plan to settle to Red Planet and why we must*. The Free Press, 1996.