MISSION SIMULATION FACILITY: SIMULATION SUPPORT FOR AUTONOMY DEVELOPMENT

Greg Pisanich, Laura Plice, Christian Neukom, Lorenzo Flückiger, Michael Wagner
QSS Group Inc.
gp@email.arc.nasa.gov
Computational Sciences Division
NASA Ames Research Center
Moffett Field, CA

ABSTRACT

The Mission Simulation Facility (MSF) supports research in autonomy technology for planetary exploration vehicles. Using HLA (High Level Architecture) across distributed computers, the MSF connects users' autonomy algorithms with provided or third-party simulations of robotic vehicles and planetary surface environments, including onboard components and scientific instruments. Simulation fidelity is variable to meet changing needs as autonomy technology advances in Technical Readiness Level (TRL). A virtual robot operating in a virtual environment offers numerous advantages over actual hardware, including availability, simplicity, and risk mitigation. The MSF is in use by researchers at NASA Ames Research Center (ARC) and has demonstrated basic functionality. Continuing work will support the needs of a broader user base.

INTRODUCTION

NASA Ames' Mission Simulation Facility (MSF) is designed to address a problem often encountered by researchers in autonomy: how to carry out meaningful testing of autonomy software without a real-world robotic platform. The Mission Simulation Facility offers a simulated testing environment including robotic vehicles, terrains, sensors, and vehicle subsystems. The initial MSF release targets users researching autonomy for Mars rovers; however, the MSF technology solution is applicable to many robotic domains.

The MSF has been developed using a multi-platform distributed architecture that is a specialization of HLA (High Level Architecture, developed by the Defense Modeling Simulation Office[1]), which allows the simulation to be distributed across multiple machines and laboratories. Multi-platform support allows the MSF to easily integrate with existing simulation software developed on Unix, Linux, and Windows platforms. Among other unique qualities, the MSF is being developed with the goal of distributing the software to universities and research groups outside of NASA so that it can be applied to other research applications.

IDENTIFIED NEEDS

The Mission Simulation Facility[2,3] adds value to autonomy research projects by filling several important needs. Researchers in robot autonomy typically focus their resources and expertise on solving a particular problem or developing a specific new approach. Often, research teams do not have time, budget, interest or background in creating software for objective testing. The MSF offers a generic simulation framework intended for technology maturation and mission infusion that is available on a variety of platforms.

Novel autonomy algorithms often begin with limited capability and grow in sophistication as the technology matures. In early stages of development, autonomy software may not be ready for real-world testing. "Tile world" and "blocks world" are examples of domain limitations which define the abilities of early-stage algorithms. The MSF can serve as a bridge between overly simplistic test situations, and the dauntingly complex real world by offering a range of simplifications in models of the vehicle, environment, and onboard equipment. Since all the relevant models are integrated in a modular design, it is possible for users to choose the level of simulation for each component, from stubbed through simple to more sophisticated representations.

Even for robust autonomy software, field time on real robots is not always the ideal testing approach. Robot platforms tend to be very expensive and in high demand among autonomy researchers. Field test opportunities are often rendered unproductive due to delays and problems unrelated to the autonomous control software intended for testing. Software in early stages of research may be constrained by onboard power and processing capability. The MSF can model many features of actual vehicles and real-world terrain. Connectivity to the MSF is available on demand and the algorithm under test can run on the desktop.

Autonomous control software frequently includes branches of reasoning related to hazardous or off-nominal conditions for the robotic vehicle. Executing on a simulated vehicle on virtual terrain offers the opportunity to test portions of code that are difficult to exercise in real situations. Additionally, a readily available simulated vehicle can support numerous repetitions of a test scenario whereas field time on a real rover is very limited.

As autonomy solutions mature further still, there remain obstacles in the development path to operational deployment. Planetary surface exploration missions require technology that is extremely reliable and predictable. The migration path from laboratory to mission is a difficult one in terms of technology, politics, and funding. The MSF hopes to offer a steppingstone toward mission readiness for control software that would otherwise remain unproven.

Figure 1 illustrates conceptually a migration path for planetary missions and the role of the MSF in maturing technology at NASA.

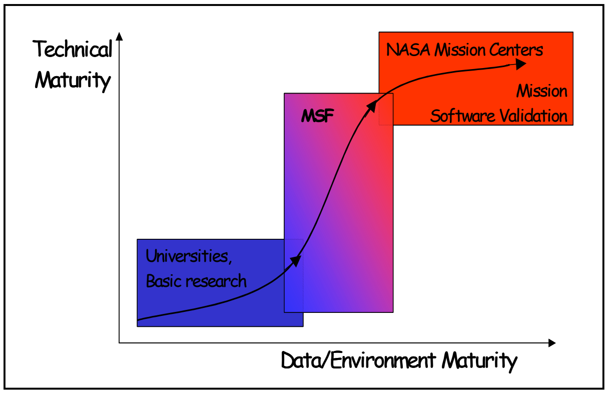
Technical Maturity
NASA Mission Centers
Mission Software Validation
MSF
Universities, Basic research
Data/Environment Maturity

Figure 1: MSF role in technology migration.

Vehicle autonomy solutions are rarely stand-alone products. Robots have multiple mission goals to be accomplished by diverse software components. Researchers focusing on different elements of a complicated mission have limited opportunities to test their software collaboratively with one another. An example at NASA Ames Research Center for Mars rovers is the dichotomy between vehicle autonomy and science autonomy. Science algorithms can assess terrain for promising targets but cannot command vehicle movement; vehicle control software can operate the rover but requires goals and operational priorities as input.

The data interface between science and vehicle software is only one layer of the collaboration problem. Scientists and mission managers tend to have differing perspectives on issues such as risk, mission priorities, resource allocation, and vehicle capabilities. The power and processing resources on board must be shared among competing purposes and equipment. Some functions of both science and vehicle operation may be able to share instruments or data, if software and hardware can be designed for dual-use. MSF can support integrated mission design by allowing science and control software to operate together throughout the development process.

SIMULATION ELEMENTS

The Mission Simulation Facility is a simulation system that represents a diverse collaboration effort. The core technology of the MSF offers a framework for connectivity among modules provided by users or collaborators. Major components of the synthetic world are the terrain surface, environmental conditions, virtual robot, simulated equipment, and graphical display. The MSF design includes technical features essential to support simulation applications. The following describes MSF components, interfaces, and capabilities.

Simulating the ground we walk on is a significant technical challenge. A triangle mesh created from a digital elevation map (DEM) overlaid with realistic textured imaging is a useful structure for portraying a surface in computer graphics and is part of the MSF terrain model. However, researchers in planetary surface robotics require many additional layers of detail. For example, an object-oriented format captures the existence and locations of specific features such as rocks and craters while interaction with scientific instruments is supported with information pertaining to spectrographs and mineral content.

Most of the terrain used in MSF to date comes from collaboration with researchers at NASA's Jet Propulsion Laboratory (JPL). The Simscape project is a server-based provider of artificial, realistic, or real-world terrain data including both physical and science characteristics. Terrain used in MSF simulations can be generated so users can specify characteristics such as rock size and distribution[4]. Specification for artificial terrain may reflect constraints that are exaggerated or over-simplified for specific testing purposes and realistic terrain would be based on knowledge of typical planetary surface conditions. In addition, virtual renditions of real-world sites can be integrated in the MSF database format, allowing simulated robots to drive on synthetic or real-world surfaces. Continuous terrain is available in contiguous patches.

A virtual robotic vehicle offers numerous advantages over real hardware. Depending on the user's research emphasis, autonomy development may be best supported by allowing perfect navigation, instantaneous location changes, unlimited power supply, or perfect sensor readings. In contrast, other researchers may need to introduce navigation errors, locomotor inefficiency, unplanned power shortages, or noisy sensor readings.

Simulated robotic vehicles range from the very simple to the very complex. The MSF provides simple vehicle models based on either kinematics or dynamics. Other vehicle simulations, such as the ROAMS project at JPL[5] can be interfaced with the MSF to provide data related to vehicle subsystems, terrain interactions, and power usage.

The MSF currently provides lighting models in the environment via an interface to the VIZ graphics simulator, including an ephemeris. If needed to support upcoming projects, a weather model, consisting of wind gradients and visibility may be added.

The MSF also provides a general interface to instruments and sensors that interact with a virtual site model, which has been developed through collaborative work with JPL. The MSF team is collaborating with the Instrument Modeling group at JPL to develop a generic command interface for sensor control and a standardized sensor data interface, which will interact with the Simscape server.

Also available are methodologies for managing time. The MSF provides a central representation of simulation time, and through the HLA time services, is able to synchronize the operation of the participating components of a simulation. The current version supports time-stepped simulations with a settable time step. It is also possible to set the date and time of the simulation, which allows the ephemeris model to compute the correct sun angle.

Tools for collecting and analyzing data from simulation runs are fundamental to the evaluation of autonomy software. The HLA communication layer and the tools developed for it provide easy collection of all information exchanged among the MSF components. One of the first qualitative evaluation tools used in the MSF is a 3D viewer (Viz), allowing the user to observe robot behaviors in a simulated world and to see from camera viewpoints. The same viewer can display abstract object parameters including the torque on motors, the field of view of a camera, or the data from an instrument. Additional generic displays and viewers are being planned that can be easily reconfigured to display simulation data in a way that will support analysis of systems under development.

The MSF will include methods for representing and implementing failures and uncertainty. It is often useful to assert the failure of a component to test the decision-making algorithm under failure mode. As an example, the placement of a robotic arm on a rock for taking an instrument measurement is very complex, and it is not necessary to simulate all of the details of such a scenario in order to test the execution of a plan to autonomously gather science data on rocks. What is needed is simply the result of the execution – success or failure. The MSF also supports failures according to statistical distributions. Executing multiple runs on the same scenario can generate a number of possible outcomes, allowing full exercise of autonomous capabilities.

TECHNICAL APPROACH

The MSF provides a software framework for the development of autonomous software and autonomy technology for robotic vehicles. The MSF's transport layer is built on HLA[6] for easy integration of autonomy and simulation modules. Execution can be on single machine or distributed among multiple processors and the system runs on a variety of computer platforms. An example configuration in Figure 2 outlines the basic architecture of MSF.

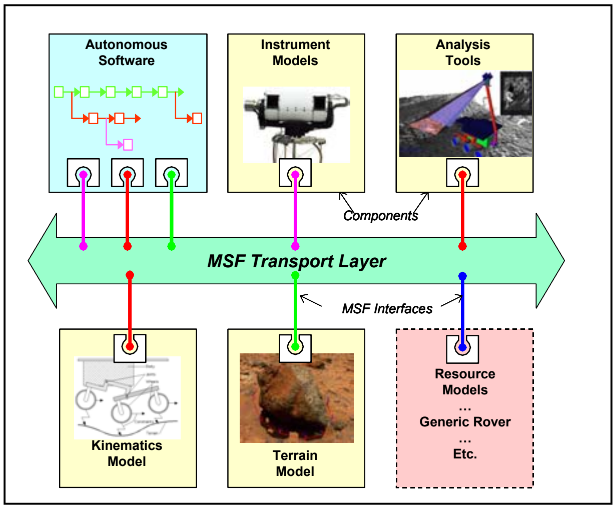

Figure 2: MSF Architecture Overview.

Since the MSF is designed to be flexible, significant development effort focused on generic descriptors and interfaces. A simulated vehicle may contain many conceptual representations in a single application. A vehicle description may include the vehicle's physical characteristics (size, mass properties, configuration), and appearance for graphical display. Another way to characterize a vehicle is with a data dictionary which represents the vehicle's capabilities in terms of input and output as well as the subsystems on board (for example, power supply or independent payload models). Additionally, the model may also include a functional description of the operation of the vehicle that could be understood by an intelligent controller.

Similarly, the virtual environment must be defined in terms relevant to many perspectives: dynamic interactions with the vehicle, appearance in graphical output, characteristics related to sensors and instruments, features that are meaningful to human users of the system, abstract functional descriptions, and any significant changes with time. In the MSF's approach to creating a simulated world, all the user input definitions are maintained in a file structure that eliminates redundant information so that changes made in one place will be reflected throughout the system.

While the MSF is designed for compatibility with the Coupled Layer Architecture for Rover Autonomy[7] (CLARAty) class definitions for robotic vehicles, API definitions in MSF do not require users to have access to CLARAty.

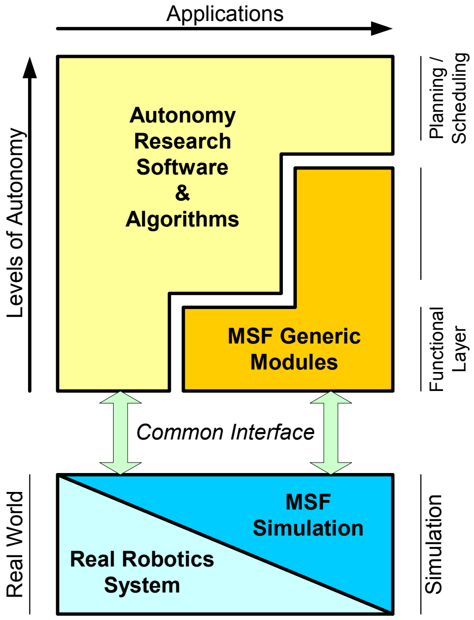

Figure 3: Varying levels of abstraction.

Well-defined interfaces allow interchangeability of real hardware with simulated components. Developers can port their product from the MSF to real platforms without having to maintain separate interfaces. Another advantage of presenting the user with a clean API is easy extensibility to new software elements.

Figure 3 illustrates the modular design approach of the MSF, which allows users to customize the simulation to include the layers of abstraction appropriate for the testing situation. For autonomy research, which includes capabilities ranging from abstract planning and scheduling all the way to detailed functional commands, the MSF offers interfaces directly to the robotic platform. In cases where research software focuses on high-level decision making only, the MSF provides intermediate layers of abstraction between the autonomous component and the robot.

An important feature of the MSF is the capability to provide varying levels of simulation fidelity. With software for high level reasoning, such as planning or resource allocation, the module under development might require only summary status information as input: command completion, for example. In cases involving only high-level abstractions, there is no need to simulate detailed hardware functionality; a simple "stub" will do. In other cases, such as fault diagnosis or science data processing, there may be a need for much higher fidelity in the simulated vehicle and its interactions with the environment.

CASE STUDIES

1. Simulation support: K9 rover mission demonstration

Researchers at NASA Ames Research Center (ARC) are developing autonomous software that can be applied to planetary rovers. One of the rovers owned by the research lab is a six-wheeled rover, with rocker-boogey suspension, named K9, which is among the family of rovers developed by JPL. Figure 4 shows a virtual version of the K9 rover driving on a synthetic terrain. The rover has the capabilities to maneuver to targets rocks, place instruments on the rocks using its robotic arm, and to take scientific measurements[8,9].

The rover is running an on-board flexible plan executive (Conditional Executive) accepting conditional sequences. The plan feeding the executive is produced by a ground based contingent planner. The planner generates sequences containing multiple branches of execution in order to optimize the achievement of mission goals based on conditions encountered by the rover during execution.

Because of the high demand for using the single prototype rover for testing the various autonomy software modules, an MSF simulation of the test scenario was created to free the autonomy developers from being dependent on the actual rover.

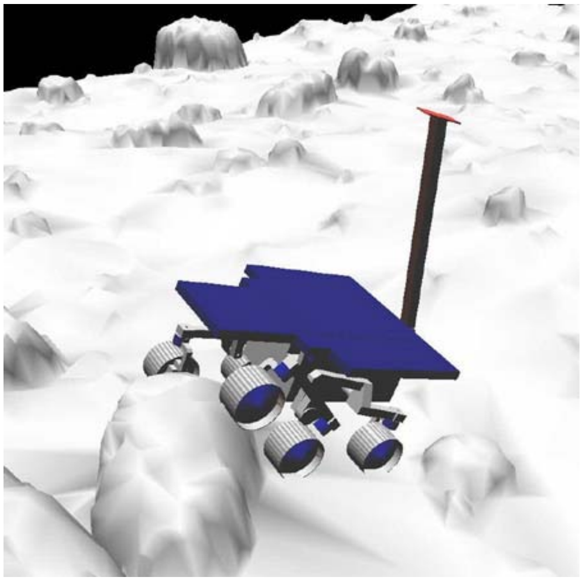

Figure 4: Virtual K9 driving over a synthetic rock.

Configuration - The following component models (see Figure 5) participated in the simulation: ROAMS (integrated models of rover kinematics, obstacle avoidance, and power management), Viz (visualization tool), specialty rock detector, Exec (Conditional Executive). The components exchanged commands and states through the MSF transport layer. The terrain used in the simulation was a 100x100 meter patch that included rolling hills, part of a large crater, and a random distribution of rocks of varying size.

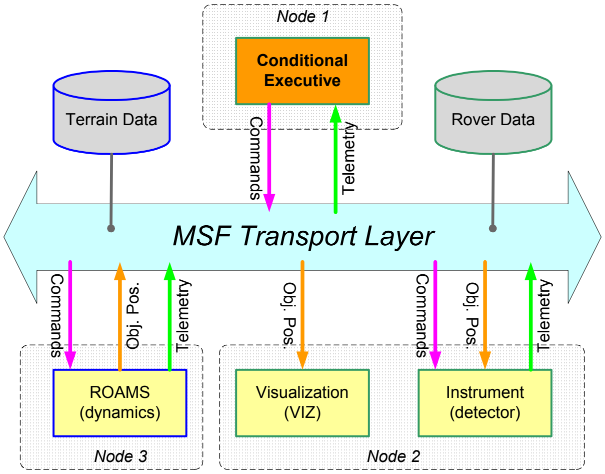

Figure 5: Component models used in the ARC simulation.

A typical scenario – In a typical demonstration mission simulation scenario (see Figure 6), the NASA Ames K9 rover acquires a set of panoramic images of the site. These are downloaded and processed off-board to create a virtual terrain model of the environment, rendered in Viz. The mission controllers explore the virtual world, choose rock targets of interest, and assign a utility to the measurements of the rock. The information is provided to the contingent planner software, which generates a plan that maximizes the expected utility subject to constraints on time and power consumption. The planner generates a sequence, with contingencies, and uploads the plan to the conditional executive component.

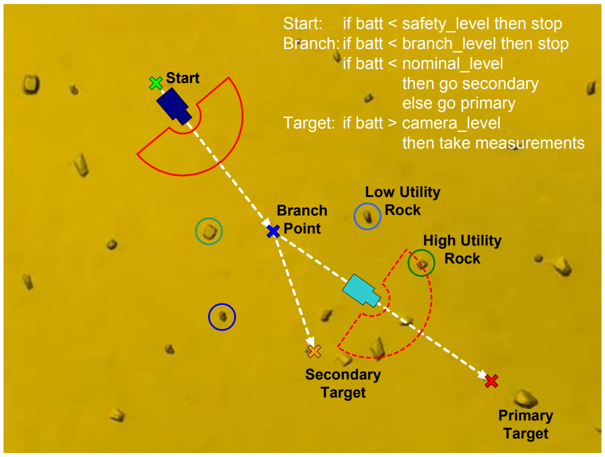

Figure 6: Sample demo scenario.

This is the starting point for the MSF simulation. The Conditional Executive then attempts to execute the plan, which involves commanding the rover to drive to each set of targets, deploying instruments and taking a number of measurements. The rocks visited and the measurements taken are dependent on a number of conditions, which can include battery level, time constraints, and success of the instrument measurements. The branch point in Figure 6 represents such a choice between possible alternatives. The Conditional Executive also considers floating contingencies during the execution, which can be triggered by external events. The specialty rock detector module provided this trigger, simulating serendipitous science opportunities along the rover path.

The results of this experiment allowed for the evaluation of the autonomous software at two levels:

Executive behavior – Although some of the models in the simulation were relatively simple, they were sufficient to provide useful feedback to the executive.

The simulation allowed for analysis of the behavior of the executive, particularly the branching and floating contingency mechanisms.

Planner generation – The simulation provided a set of components that was rich enough to create interesting scenario runs, to test different versions of the plans generated by the planner, and to detect potential inconsistencies in these plans. In addition, it was possible to explore the effect of variability on the scenarios by using the obstacle avoidance component in ROAMS. By varying the obstacle height, the rover had to select different paths to reach its destination, which in turn affected the power consumption.

2. Mars Smart Lander Executive Controller

The Apex project at NASA Ames[10] is developing an executive controller for the Mars Science Laboratory (MSL) mission in collaboration with the engineering design process for the vehicle. The Apex development environment for autonomy software presents detailed insight into the behavior of software during execution.

The problem – For autonomy technology to gain acceptance for widespread use on remote exploration missions, it must demonstrate robustness, reliability, and effectiveness. In addition, there is a need to communicate hardware and software requirements among all developers.

The solution – The Apex-MSF simulation addresses the need for interaction between hardware and autonomy software developers with a high-fidelity simulation of the autonomous vehicle operating in a realistic environment. Aside from being a planning and execution software product, Apex is also able to access and display models of the hardware, which allows hardware and autonomy software developers running a simulation and to ask what-if questions and make on-the-fly software changes.

For example, in a situation where an onboard sensor indicates overheating of a component, autonomy software designed to respond to the anomaly must demonstrate the proper response. The Apex-MSF simulation allows developers to examine a detailed hardware model simultaneously with a trace of the reasoning process. Insight into both hardware and autonomy software behavior allows experts to verify that autonomous handling of anomalies is appropriate, reliable, and valuable to a mission.

Looking simultaneously at detailed hardware models and autonomy software greatly enhances the communication between collaborating hardware and software engineers and increases productivity and reliability of the software. In addition, hardware and mission specialists may accept incorporation of autonomy software more readily if they have better understanding of the inner workings of the software.

3. Antarctic Operations Simulation

The MSF team is participating in a proposal for a field test that will explore the use of Autonomous robotic operations to search for subsurface life in Antarctica. This proposal is for a prototype rover that will attempt to cross the Antarctic in 2007. The MSF is developing robotic models, systems, dynamics, terrain and environmental models for the development of a planner and executive for this project. The MSF will also be enhanced to provide plan design and verification, traverse visualization, and science data interpretation during actual operations.

CONTINUING DEVELOPMENT

The MSF project has accomplished its preliminary goals. As a research project in its own right, MSF has demonstrated that the technological approach is effective. The demonstration of the K9 rover simulation, which was completed in the spring of 2003, validated the distributed architecture running several component modules. This accomplishment is significant for MSF technology, and the MSF is currently deployed and in use by autonomy developers at Ames Research Center. The goals for the coming year emphasize efforts aimed at making MSF accessible to autonomy developers external to NASA.

Technical capabilities to be added include greater flexibility in introducing failures and uncertainty into simulated scenarios, more robot-specific features for the K9 and other robots, and improved handling of time and synchronization. User support elements to be addressed in the coming year include the ability to analyze data collected during execution, graphical user interfaces, and easier management of distributed processes for starting and ending a run.

The MSF can offer valuable support in the collaboration between researchers in planetary science and developers of vehicle autonomy. The inclusion of scientific expertise is an area where collaboration remains difficult. While an autonomous rover's capabilities represent significant technical achievement in and of themselves, from the perspective of scientific research, the rover is a means to a greater end. To planetary science, the rover is a remote, mobile laboratory that represents a rare and unique opportunity to collect data. Autonomy technology stands to improve the ability to collect scientific data on remote exploration missions. The coordination of scientific goals and technological capabilities is a critical gap in the formulation of autonomous exploration missions.

As the Mission Simulation Facility expands to support multiple users, it will allow science and autonomy researchers to collaborate in the development of autonomous science missions. Prioritization of operational objectives, utilization of common resources, formulation of mission parameters, optimization of hardware capabilities, shared understanding of data relevance, and communication protocols for software applications are examples of areas where science and autonomy researchers can collaborate effectively.

Until now, the reasoning behind stated scientific objectives has not been fully available to the autonomy design process and the potential capabilities provided by autonomy technology have been difficult to incorporate in the basic experimental design for scientific research. By providing a mission description capability much like the existing vehicle and site descriptors, the MSF will be able to capture the interaction between scientific and autonomy researchers in a way that can be codified, stored, recalled, and modified during the collaboration process.

The potential collaborative capabilities of the MSF will enable autonomy developers to formulate a set of goals for the autonomous planner or controller that are based on realistic priorities, constraints, and/or heuristics used by scientific researchers. Scientists will have the opportunity to explore the capabilities and limitations of the autonomous rover as a mobile asset in pursuing research. Both parties will be able to participate in developing exploration missions that gain maximum benefit of the available resources and circumstances. The mission design process will benefit by having a means to benchmark, compare, and develop mission designs.

The MSF project is taking actions to increase its engagement with autonomy research in the university community. Basic research projects tend to be lower on the TRL scale[11] and do not always require vehicle simulations with detailed subsystems. Smaller, more streamlined components will be available as alternatives to the greater sophistication needed by further advanced, better funded projects.

A promising growth path for MSF technology is support of mission operations. Current Mars exploration missions require considerable decision time among human experts to determine the next action. Real mission data rendered for a virtual rover could be a valuable tool for mission planners to assess command decisions and predict hardware and software behavior.

Further identified growth applications of MSF include several vehicle domains. While the initial focus has been on planetary rovers, the MSF architecture is designed to support a variety of mission platforms. Underwater vehicles, unmanned fliers, and spaceborne robots are all suitable candidates for autonomy and share similar identified needs to those described for Mars rovers. There are numerous potential missions for autonomous vehicles that could be researched in simulation. Deep sea exploration, offshore structure maintenance, navigation under ice, hazardous site reconnaissance, space construction, and even the Jovian moon Europa involving atmospheric, ice, and water exploration, are all candidates for simulation in MSF.

SUMMARY

This paper identifies several of the challenges in the development of autonomous systems. To address these needs, the Mission Simulation Facility is an architecture and simulation framework for the development and testing of autonomy. The MSF uses an abstraction layer to provide communication among models of vehicles, the environment, and any simulated equipment or subsystems needed. The distributed architecture connects components running on multiple machines and is compatible with several computer platforms. The MSF has demonstrated its capabilities and utility in an application consisting of a planner and executive executing a mission goal for an autonomous Mars rover. We look forward to continuing to develop the MSF and gaining new insights through its application to additional missions and domains for autonomy.

ACKNOWLEDGEMENTS

None of this work would be possible without the support of Dr. Butler Hine and the IS program. We would also like to acknowledge the collaborative contributions of several researchers: Dr. Abhinandan Jain and his staff at the DARTS lab, Jet Propulsion Laboratory; Dr. Michael Freed and his APEX team at NASA Ames; and Dr. Rich Washington, Dr. Dave Smith, Dr. Larry Edwards, and Anne Wright and their work in autonomy and robotics at NASA Ames. We are also indebted to Dr. Meemong Lee and Dr. Dan Katz at the et Propulsion Lab.

REFERENCES

[1] IEEE Standard for Distributed Interactive Simulation – Application Protocols (Revision and redesignation of IEEE Std 1278-1993). Designation: 1278.1-1995

[2] L. Flückiger and C. Neukom. A new simulation framework for autonomy in robotic missions. In *Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, EPFL, Lausanne, Switzerland (pp 3030 – 3035).

[3] L. Edwards, L. Flückiger, and R. Washington. VIPER: Virtual intelligent planetary exploration rover. In *Proceedings of i-SAIRAS 2001*, 2001.

[4] R. W. Gaskell, J. B. Collier, L. E. Huseman, and R. L. Chen. Synthetic environments for simulated missions. In *Proceedings of the IEEE Aerospace Conference*, Big Sky, MT, March 2001.

[5] J. Yen, A. Jain, and J. Balaram. ROAMS: Rover Analysis, Modeling and Simulation. In *Proceedings of i-SAIRAS 1999*, pages 249–254, 1999.

[6] F. Kuhl, R. Weatherly, and J. Dahmann. *Creating Computer Simulation Systems: An Introduction to the High Level Architecture*. Prentice Hall, 2000.

[7] Nesnas, I., R. Volpe, T. Estlin, H. Das, R. Petras, D. Mutz. Toward Developing Reusable Software Components for Robotic Applications, In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2001.

[8] L. Pedersen, M. Bualat, D. Lees, D.E. Smith, R. Washington, "Integrated Demonstration of Instrument Placement, Robust Execution and Contingent Planning," *in Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics and Automation for Space*, 2003.

[9] R. Washington, K. Golden, J. Bresina, D. E. Smith, C. Anderson, and T. Smith. Autonomous rovers for Mars exploration. In *Proceedings of The 1999 IEEE Aerospace Conference*, 1999.

[10] M. Freed, Managing multiple tasks in complex, dynamic environments. In *Proceedings of the 1998 National Conference on Artificial Intelligence*, Madison, WI, 1998.

[11] C. P. Graettinger, S. Garcia, J. Siviy, R. J. Schenk, P. J. Van Syckle, Using the Technology Readiness Levels Scale to Support Technology Management in the DoD's ATD/STO Environments. Technical Report, Software Engineering Institute (SEI), Carnegie Mellon University, CMU/SEI-2002-SR-027, September 2002.