Safe Long-Range Travel for Planetary Rovers through Forward Sensing

Conference Paper · January 2013

16 authors, including:

Yashodhan Nevatia
Universidad de Navarra
17 PUBLICATIONS **303** CITATIONS
SEE PROFILE

Yang Gao
University of Surrey
157 PUBLICATIONS **1,798** CITATIONS
SEE PROFILE

Souhila Mili
University of the West of Scotland
1 PUBLICATION **8** CITATIONS
SEE PROFILE

Roland Sonsalla
Deutsches Forschungszentrum für Künstliche Intelligenz
18 PUBLICATIONS **66** CITATIONS
SEE PROFILE

SAFE LONG-RANGE TRAVEL FOR PLANETARY ROVERS THROUGH FORWARD SENSING*

Y Nevatia(1), F Bulens(1), J Gancet(1), Y Gao(2), S Al-Mili(2), R Kandiyil(2), RU Sonsalla(3), M Frische(3), T Vögele(3), E Allouis(4), K Skocki(5), S Ransom(6), C Saaj(2), M Matthews(2), B Yeomans(2), L Richter(6), TP Kaupisch(7)X

(1)Space Applications Services, Leuvensesteenweg325, Zaventem, 1932 Belgium,
Email:firstname.lastname@spaceapplications.com
(2)Surrey Space Centre, university of Surrey, Surrey, GU2 7XH UK, Email:firstinitial.lastname@surrey.ac.uk
(3)DFKI Robotics Innovation Center, Robert-Hooke-Strasse 5, 28359 Bremen, Germany,
Email:firstname.lastname@dfki.de
(4)EADS Astrium, Gunnels Wood Road, Stevenage, SG1 2AS UK, Email:firstname.lastname@astrium.eads.net
(5)Astri Polska, Bartycka 18a, 00-716 Warsaw, Poland, Email:firstinitiallastname@cbk.waw.pl
(6)Liquifer Systems Group,Obere Donaustrasse 97-99/1/62, 1020 Vienna, Austria , Email:srconsultancy@freenet.de,
lrichter@liquifer.com
(7)DLR, Königswinterer Strasse 522-524, 53227 Bonn, Germany, Email:firstname.lastname@dlr.de
XTP Kaupisch was affiliated with DFKI Robotics Innovation Center during his involvement in the work reported here

**This work was supported in part by the European Commission through the SPACE theme of the FP7 Programme, under Grant Agreement 284419*

ABSTRACT

Despite the increase in autonomous capabilities of mobile robots, planetary exploration rovers are severely limited in their operational paradigms. Current rover operations, especially the traverse operations, involve day to day planning and simulation with human involvement for the estimation of potential dangers faced by the rover. Additionally, due to the large investment represented by the rovers, operations tend to be carried out in a cautious manner, with minimal autonomous deliberation. Such limits on the daily traverse, though motivated by concerns of safety and overall success, are not sufficient for future missions.

This paper describes an operations concept for safe, long range traverses for planetary rovers based on forward sensing of terrain trafficability, and presents the software architecture enabling such operations.

1. INTRODUCTION

The Forward Acquisition of Soil and Terrain data for Exploration Rover (FASTER) system has been conceived keeping in mind the potential traverse requirements of future missions, primarily a future Mars Sample Return Mission requiring lengthy traverses with minimal science operations.

It leverages the operation of a lightweight, highly mobile scout rover as a forward sensor of the primary rover, ascertaining terrain trafficability and identifying potential soil hazards. This increases the safety of the primary rover, allowing additional autonomy with the goal of achieving faster traverses overall.

Section 2 will provide an overview of the system concept, before Section 3 describes the applicable operation concept enabling safe long range traversal. Section 4 goes on to describe the software architecture corresponding to the operation concept. Section 5 describes the environment that will be used to validate the software system.

2. SYSTEM OVERVIEW

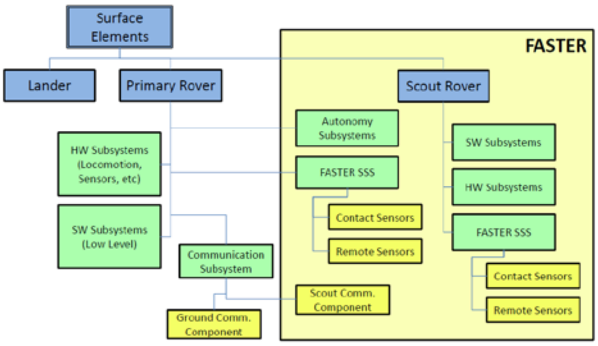

Figure 1. Elements of the FASTER system concept

As shown in Fig. 1, the FASTER system is based on the following elements or subsystems in addition to the primary mission rover:

Scout Rover: A lightweight scout rover capable of traversing terrain that would be dangerous for the primary mission rover, enabling forward sensing. Further information on the proposed scout rover design can be found in [1].

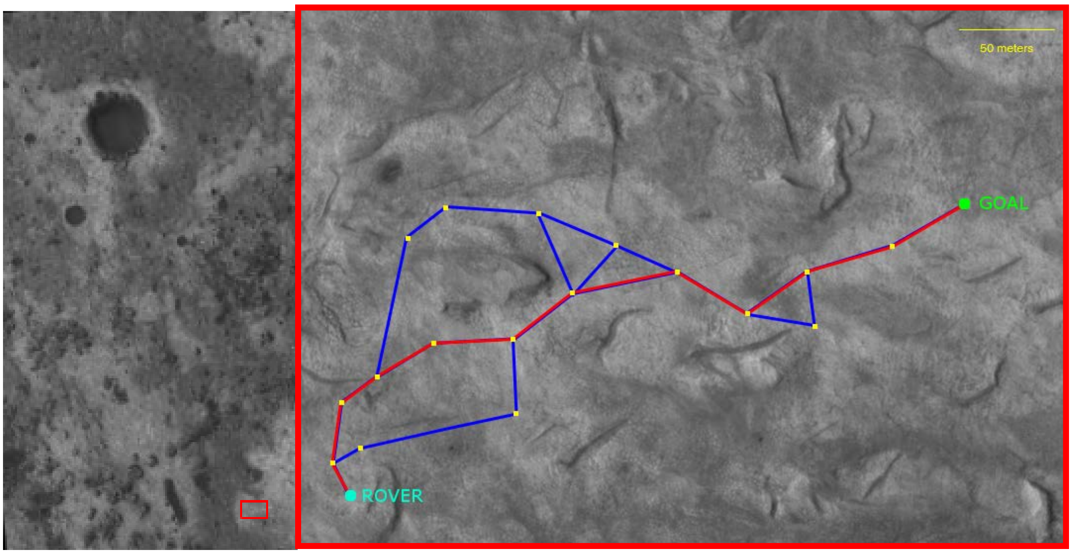

Figure 2. Set of possible paths across a section of the Mawrth Vallis (preferred path highlighted in red) [Images are from the HiRISE Online Image Viewer, [2]]

Soil Sensing System (SSS): Soil sensors form a critical component of the FASTER system. They include remote sensing capabilities allowing the visual detection of hazards from camera images as well as contact sensors for both rovers. The soil sensor system also includes a data fusion module that produces aggregated estimates of terrain trafficability that are classified as 'GO'/'NO-GO'/'MAYBE' decisions. Further information on various sensors considered for the FASTER Soil Sensing System can be found in [3].

Autonomy and Cooperation: Software subsystems on both rovers enabling the use of the scout rover as a forward sensor and the implementation of the operation concepts described below.

While the FASTER system has been designed and sized keeping in mind potential requirements from a future European Mars Sample Return Mission, the general concept can be applied to any future rover mission requiring rapid traverse of large distances.

3. OPERATION CONCEPT

While the availability of a free ranging scout rover potentially enables a number of scenarios with greater scientific return, the FASTER operation concept focusses on the 'traverse phase' of missions. This phase, the identified long range traversal required in sample fetch missions with minimal science to be performed, is addressed as three components:

- Ground Planning,
- Global Path Planning, and
- Waypoint Traversal.

3.1. Ground Planning

A large amount of terrain data of the Martian surface is available, ranging from low resolution and accuracy contour maps from the United States Geological Survey to high resolution maps from the High Resolution Stereo Camera (HSRC) on Mars Express and HiRise imager on the Mars Reconnaissance Orbiter. This information can be used by Ground Control to perform preliminary planning of traverse operations in regions that have been mapped.

The operation concept selected comprises the ground control team using such available terrain data for the identification of potential paths across clear terrain, avoiding large obstacles and geological features (such as crevices) and avoiding lengthy detours.

Each planned path would be in the form of a sequential list of way points, with a potential straight line path identified between two consecutive waypoints. Multiple paths would then be combined into a single directed graph structure, with each waypoint represented as a node in the graph structure and the edges weighted based on estimated cost of traversal. This graph would form the traversal command sent to the primary rover, and is referred to as the *traverse graph*.

Such commands are expected to cover traversal till the next potential telecommand possibility - potentially distances of hundreds of meters, however they could also be used to provide longer plans for use as contingencies.

Fig. 2 shows a sample set of paths across a section of the Mawrth Vallis.

3.2. Global Path Planning

Global Path Planning involves basic operations on the traverse graph once received by the primary rover, primarily executing graph searches.

At the start of each traverse phase, we assume that the rover location is present as a node in the graph. The best path is computed as a list of waypoints – which is then used for iterative waypoint to waypoint navigation as described in Section 3.3 below. At the end of each successful waypoint traversal, the corresponding estimated cost is updated.

If a particular edge is found as non-traversable, modification of the traverse graph is required – edges between the respective nodes are removed. If the primary rover is not at an existing waypoint, a new waypoint is added to the graph, connected to the previous waypoint visited by the rovers and possibly other waypoints that are in the immediate vicinity (excepting the waypoint that triggered the traversal failure).

This is followed by a new search for a path to the target location. If no path is found the rovers will nominally wait for updated commands from mission control, but could also be instructed to carry out alternate navigation modes as contingency actions.

3.3. Waypoint Traversal

Waypoint Traversal performs the core of the traversal actions based on a mode of navigation similar to the *motion-to-goal* and *boundary-following* behaviours described in [4].

Starting from a waypoint, the rovers turn till they are facing the next waypoint. As a straight line path can be assumed between consecutive waypoints, the rovers are then facing along the notional trajectory.

Navigational sensors (presumed to be stereo cameras) on both rovers are used to generate a detailed terrain map of the region directly ahead of the rovers, with the known relative positions and techniques for map optimization used to build a combined map. As the next waypoint could be further away than is visible in the generated map, the rovers move towards it iteratively, with the map extensions created when the mapped region is traversed.

A path for the Primary Rover is then planned from the current position leading towards the next waypoint. Apart from a simple geometric analysis, remote visual soil sensing data from the FASTER SSS, when available, is used as additional inputs for obstacle locations. As ground planning might have been performed with low resolution information, the straight line path to the next way point may be obstructed by obstacles resulting in no possible paths towards the waypoint. In such cases, the rovers can attempt to circumnavigate the detected hazard. This is achieved by the rovers turning away from the obstacle. However the rovers are permitted to turn only a limited amount in the circumnavigation efforts, preventing the rover from moving away from the waypoint. Once past the obstruction – or on reaching the end of the mapped region – the rovers turn towards the waypoint and restart the sense-plan-move cycle.

In case a path cannot be found, the path from the current (or last) waypoint to the next one can be identified as non-traversable.

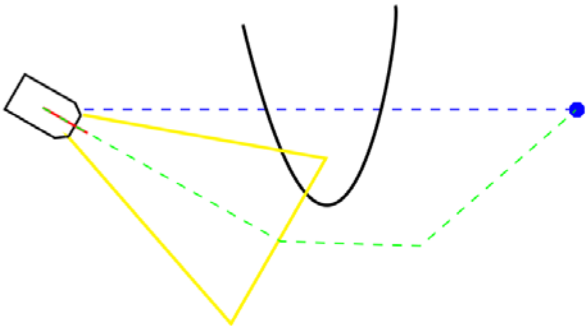

Figure 3. Circumnavigation of an obstacle in the straight line path

On a path being found, the scout rover begins forward sensing operations. It moves along the planned path, deploying the on-board contact soil sensor suite with the SSS returning updated trafficability assessments. Once the scout rover has advanced, the primary rover follows deploying on-board continuous sensors to verify terrain trafficability.

When the scout rover reaches the end of the planned path, as mentioned above the scout rover turns towards the next waypoint. A further set of images from the scout rover navigation cameras is used to extend the detailed terrain map, serving as input for another iteration of local path planning with the final location of the previous trajectory as a start point.

One important consideration here is that the scout rover always operates within in line of sight of the primary rover. This is essential as it allows a robust, drift free relative localization between the two rovers enabling forward trafficability assessment.

At any time, the FASTER SSS could reach a ‘NO-GO’ trafficability assessment based on either of the rover sensors resulting in the invalidation of the planned local path. If non-traversability is determined on the basis of the primary rover sensors, the scout rover returns to the primary rover and the planning of a new path is attempted. If the scout rover sensors trigger the negative assessment, an attempt is made to plan an alternate path for that segment. In this case, if no alternate path is found the scout returns to the last path endpoint it traversed, and two new nodes are added – corresponding to the locations of both rovers. Depending on the optimal global path found, either the primary rover proceeds to the location of the scout using the planned local path or the scout returns to the primary rover location.

4. SOFTWARE ARCHITECTURE & AUTONOMY

The operations concept described above in Section 3 is targeted through the partial implementation of the E4 level of autonomy as defined in the ECSS standards [5]: execution of goal oriented mission (traversal) operations on-board.

Due to potential limitations on the scout rover, the majority of the autonomy is focussed on the primary rover, with the scout rover treated as a remote, mobile sensor that is capable of following a path provided to it by the primary rover as well as basic health monitoring.

Fig. 4 shows the software architecture for the primary rover, identifying the different software subsystems.

Task Planner: A symbolic task planner supporting goal based planning (and replanning) of tasks and contingency actions. Further described in Section 4.2.

Health Management: A representative fault detection and recovery subsystem based on offline analysis of potential faults and the corresponding indicators and corrective actions.

Task Execution Controller(s): On-board procedure execution engines supporting the execution of pre-defined sub-tasks.

GNC: The Guidance, Navigation and Control subsystem performs all the path planning, mapping and self-localization tasks. Further described in Section 4.3.

Data Management: A representative data handling subsystem which is responsible for dispatching and maintaining shared data between the subsystems, as well as preparing telemetry for transmission.

Scout Localization: A computer vision subsystem to localize the scout rover in camera images, allowing drift free localization of the scout rover. Further described in Section 4.4.

Soil Sensor System SW Chain: Subsystem interfacing and implementing parts of the FASTER SSS software, able to provide classified trafficability results.

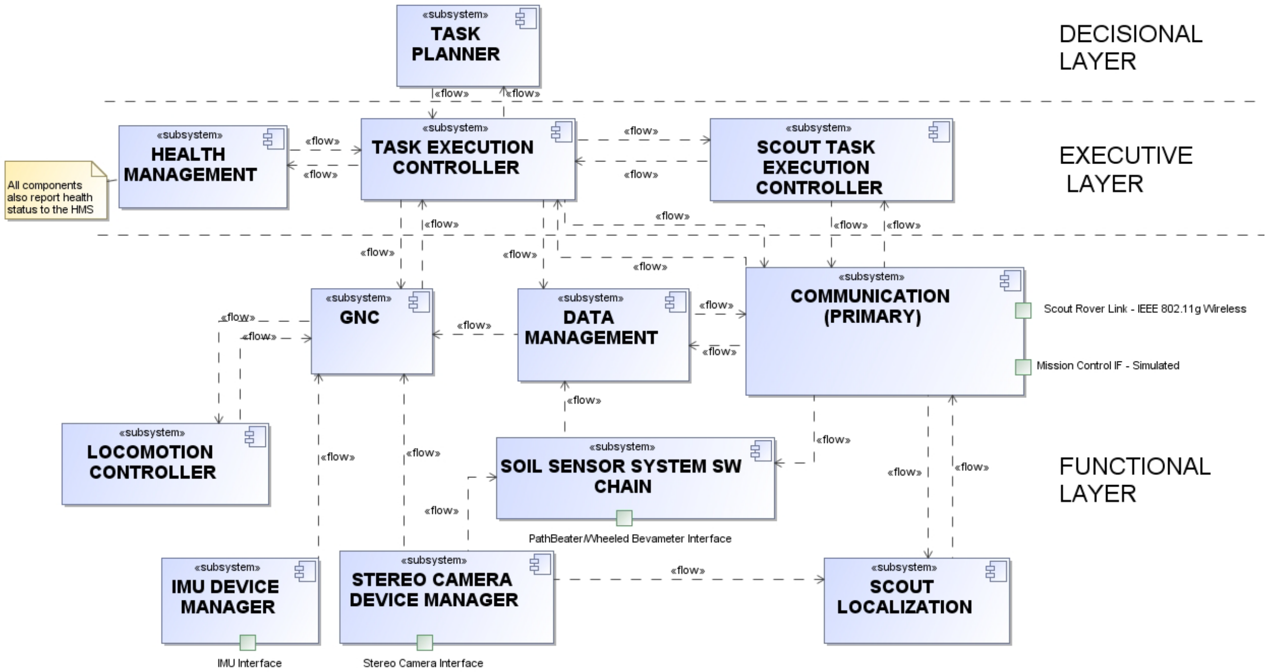

Figure 4. Proposed software architecture for the primary rover

Communication: Subsystem responsible for communication between the rovers, as well as providing representative functionality for communication with mission control.

Locomotion Controller: A motion controller for the primary rover, capable of following simple paths and trajectories that have been planned by the GNC.

Device Manager(s): Subsystems providing interfaces to various primary rover sensors.

4.1. Software Framework

The software subsystems for the primary rover have been implemented using a combination of the popular Generator of Modules ($G^{en}oM$) and Robotic Operating System (ROS) frameworks.

The $G^{en}oM$ [6] framework, previously deployed as a framework for planetary rovers as described in [7], is used to define the software subsystems interfaces and handle communication between the subsystems.

The subsystems themselves are designed to leverage the popularity of the ROS framework [8], enabling quick prototyping of functionality through the re-use of open source algorithmic implementations.

4.2. Task Planner

The task planner that is incorporated on the joint deliberative layer of the primary rover and ground station is mainly responsible for planning and monitoring joint high level tasks of the primary-scout rover system. The symbolic task planner which is based on Hierarchical Task Network (HTN) planning allows the generation of plans from multiple, concurrent action sequences and the validation of upcoming command sequences with respect to the required and available resources. An interleaved planning mechanism supported by the HTN planner [9] helps in calling external programs and the task planner is also able to query functional layer modules for generating new tasks, such as getting the estimated distance to reach a point of interest based on the progress of the scout rover. The task planner receives information from the Task Execution Controller which executes on-board validation procedures and also collects the cost estimates of different tasks from the subsystems. The task planner ensures that sufficient resources are available to execute tasks and priority is always given to communication and monitoring tasks. Since the task planner is based on time line planning all the subtasks receives the deadline from the main tasks to meet the strict temporal constraints. An effective technique to encode all the ordering of the tasks and subtasks so as to meet the temporal constraints of the planning domain can be done using Allen’s Algebra [10]. In case of failure in the planned operations the planner is able to schedule contingency plans. The interleaved planning mechanism helps in replanning while the contingency plan is executed and planner can return to nominal operations if replanning is not successful.

4.3. Guidance, Navigation & Control

The GNC subsystem performs the core traversal related tasks for the FASTER system.

Self Localization: Accurate localization of the rover is needed to successfully follow planned paths. The self localization module for the primary rover uses both continuous localization as well as localization correction to achieve accurate position estimates. Continuous localization is based on visual and inertial odometry, and is intended to be used while following paths. As odometry based methods have an unbounded drift, this is complemented with a mechanism to correct the drift. Apart from allowing operators at mission control updating the pose estimates, an approach best on matching high resolution local maps created by the rover with low resolution elevation maps from orbiters is being investigated. While such an autonomous drift correction mechanism might not be very accurate, it would provide for a means to constrain the previously unbounded drift in odometry estimates.

Mapping: The mapping module of the GNC component is responsible for merging 3D information from navigation cameras (stereo cameras) mounted on both rovers into a single elevation map of the local environment. This is performed utilizing the assumption that the Scout Localization subsystem provides an accurate estimate of the scout pose. 3D point clouds generated from both rovers are first edited to remove any ‘outlier’ elements that correspond to the other rover. Then, using the relative localization estimate as a starting value, the Iterative Closest Point algorithm is used to further refine the relative positions. The combined point cloud is then filtered to form an elevation map that can be used for local path planning. The same process is repeated when maps are extended, taking the last created elevation map as a point cloud, and optimizing the relative position of new 3D data.

Path Planning: Path planning activities cover both global planning as well as local path planning. While global path planning tasks are based on well-known graph operations as described in Section 3.2, local path planning is based on D* planning utilizing the generated local elevation map and a cost function that takes the soil trafficability estimates into consideration. Optimal trajectories fitting the selected local paths are then used to drive the primary rover.

4.4. Scout Localization

The Scout Localization subsystem is a critical component of the software system, accurately localizing the scout rover using images from the primary rover cameras. By limiting scout rover operations to within line of sight of the primary rover, it is ensures that there is always a good estimate of the scout rover position – allowing the use of the scout rover as a forward sensor.

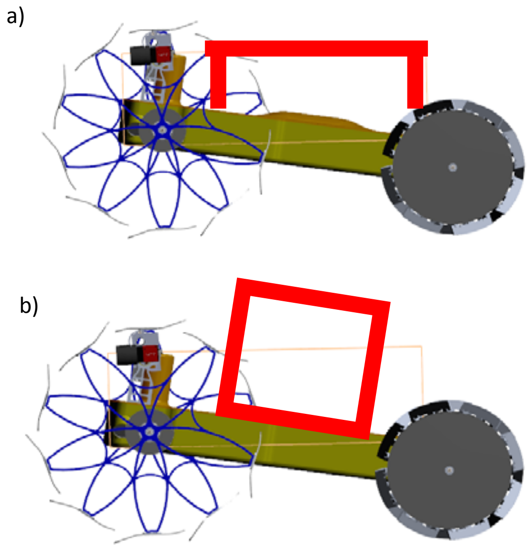

Figure 5. Scout marker configurations (a) single marker, b) cube markers)

Two approaches for the scout localization are currently under study – Markerless Tracking and Markered Tracking. Markerless Tracking is based on the recognition of point features corresponding to the scout rover in camera images, allowing for the estimation of the relative position of the scout rover. A database matching rotation and scale invariant Speeded Up Robust Feature (SURF) [11] descriptors to 3D locations on the scout rover chassis is created offline. In each image believed to be containing the scout rover, SURF descriptors of significant point features are extracted, and matched to database using a RANSAC based procedure to generate the most consistent matches. This information is then used to estimate the camera position via back projection (calculation of the camera pose given 3D location of point features in the image), allowing for relative localization. Markered tracking follows standard approaches in computer vision and augmented reality applications. Implemented using the ARToolKit library [12], one or more markers placed at defined locations relative to the scout chassis are detected in camera images, resulting in the calculation of the relative scout location. Two potential marker designs (Fig. 5) are being tested: the first comprising a single marker raised over the scout chassis, and the second a cube placed on the scout with distinct markers on its five visible sides.

Each approach is expected to have its benefits – marker based tracking is less computationally demanding but might not work well in case of partial occlusions. Markerless tracking, as it is based on point features, does not have the same drop in performance due to occlusions, but is more computationally intensive and SURF based features might be affected by drastic changes in illumination, especially shadows. Both approaches will be benchmarked, with the best method included in the system.

5. VALIDTION ENVIRONMENT

A simulation environment has been developed to validate the concepts the software system before a full range of field validation trials are conducted.

The Gazebo simulator [13] is chosen because it allows for multi-robot simulation in outdoors environment that has been chosen as the simulation environment for the latest DARPA Grand Challenge. It simulates robots, sensors and environment in a three-dimensional world, with rigid body physics simulation using the popular ODE physics engine [14]. In Gazebo, each model (robot, sensor or environment) has a plugin that is compatible with ROS, allowing easy interfacing from the software subsystem.

5.1. Environment Model

Data collected by the High Resolution Imaging Science Experiment on board the Mars Reconnaissance Orbiter (published by the University of Arizona [2]) was used to create representations of the Martian surface. Available in the Planetary Data System Format used by NASA. these files are composed of a header holding metada and binary data in an array. Following the description in the metadata and viewing the data with NASAVIEW, the binary data can be extracted and used to produce a digital elevation map to be imported in the simulator.

Figure 6. Martian dune as an elevation map with height as grayscale values, ready to be imported

5.2. Robot Model

The description of robots in Gazebo uses the XML based Simulation Description Format (SDF). Two types of elements are used to describe a robot: links (rigid bodies) and joints.

Each link describes a rigid body with its mass and inertia, collision and visual geometries (either a simple geometric form or a mesh), and position. Each joint defines one child link and one parent link, a relative pose for the joint and the axis and whether the joint is a prismatic or a revolute one.

Finally the world (environment) is described with the same formalism, including robots, light sources and plugins used in the simulation (especially for communication).

Figure 7. Scout rover on the Martian surface in Gazebo, built from HiRISE data.

6. REFERENCES

1. Sonsalla, R.U., Fritsche, M., Vögele, T., & Kirchner, F. (2013). Concept Study for the FASTER Micro Scout Rover. In *Proc. 12th Symposium on Advanced SpaceTechnologies in Robotics and Automation,* ESA Publications Division, Euopean Space Agency, Noordwijk, The Netherlands (to appear).

2. Zielinski, S. (2007). Mars imagery now available on the Web. *Eos, Transactions American Geophysical Union,* 88(26), 270.

3. Lewinger, W.A., Comin, F., Ransom, S., Richter, L., Al-Milli, S., Spiteri, C., Gao, Y., Matthews, M. & Saaj, C. (2013). Multi-Level Soil Sensing Systems to identify Safe Trafficability Areas for Extra-Planetary Rovers. In *Proc. 12th Symposium on Advanced SpaceTechnologies in Robotics and Automation,* ESA Publications Division, Euopean Space Agency, Noordwijk, The Netherlands (to appear).

4. Volpe, R., Estlin, T., Laubach, S., Olson, C., & Balaram, J. (2000). Enhanced mars rover navigation techniques. In *Proc. IEEE International Conference on Robotics and Automation, 2000* (ICRA'00), IEEE, Vol. 1, pp. 926-931.

5. European Space Agency, ECSS-E-70-11: Ground Systems and Operations – Telemetry and Telecommand Packet Utilisation, Jan 2003**.**

6. Mallet, A., Pasteur, C., Herrb, M., Lemaignan, S., & Ingrand, F. (2010). GenoM3: Building middleware-independent robotic components. In *Proc. IEEE International Conference on Robotics and Automation (ICRA), 2010*, IEEE, pp. 4627-4632.

7. Ceballos, A., De Silva, L., Herrb, M., Ingrand, F., Mallet, A., Medina, A., & Prieto, M. (2011). GenoM as a Robotics Framework for Planetary Rover Surface Operations. In *Proc. 11th Symposium on Advanced SpaceTechnologies in Robotics and Automation,* ESA Publications Division, Euopean Space Agency, Noordwijk, The Netherlands, pp12-14.

8. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R. & Ng, A.Y. (2009). ROS: an open-source Robot Operating System. In *Proc. IEEE International Conference on Robotics and Automation (ICRA), 2009: Workshop on open source software,* IEEE, Vol. 3, No. 3.2.

9. Allen, J. (1983). Maintaining knowledge about temporal intervals. *Comm. ACM.* **26**(1):832–843.

10. Nau, D., Au, T., Ilghami, O., Kuter, U., Murdock, J.W., Wu, D., & Yaman, F. (2003). SHOP2: An HTN Planning System. *Journal of Artificial Intelligence Research,* **20**:379–404.

11. Bay, H., Tuytelaars, T., & Van Gool, L. (2006). Surf: Speeded up robust features. In *Computer Vision–ECCV 2006,* Springer, Berlin/Heidelberg, Germany, pp. 404-417.

12. Kato, H. (2002). ARToolKit: library for Vision-Based augmented reality. *Institute of Electronics, Information and Communication Engineers Technical Report,* Vol. 101, No. 652, pp 79-86.

13. Koenig, N., & Howard, A. (2004). Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *Proc. IEEE/RSSJ International*

Conference on Intelligent Robots and Systems, *2004*, IEEE, Vol. 3, pp. 2149-2154.

14. Smith, R. (2005). Open dynamics engine. Online at *http://sourceforge.net/p/opende* (as of 1 May 2013).

15. Kandiyil, R., & Gao, Y. (2012). A Generic Domain Configurable Planner using HTN for Autonomous Multi-Agent Space System. In *Proc. 11th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, ESA Publications Division, Euopean Space Agency, Noordwijk, The Netherlands.