

Integrating the Sam Sensor with the Mars Rover Simulator

Dylan Dsouza, Joel Miranda, Jariel Gojar, Sheldon Karkada, Sanket Korgaonkar, Dakshayani G.

Abstract: A computer simulation of complex hardware and software amalgamated bot is often a better alternative to traditional scaled-up or scaled-down physical replications of the bots. In accordance, our project encompasses the simulation of the Mars Rover in a local machine. In its implementation, it consists of two parts, the Rover and the Mission Control and Command Center. The Rover and Mission Control communicate with each other through a common communications' protocol. The simulation also incorporates logging for status and analysis, making debugging possible. The simulation takes into account sensors such as the REMS, radio, spectrometer, Apxs, DAN and a radiation detector. In addition, there is a service layer that fetches relevant data for various rover sensors. The paper throws light on the implementation of the SAM (Sample Analysis on Mars) suite of instruments for the Mars Rover simulation.

Index Terms: Curiosity; N.A.S.A.; Sample Analysis on Mars; Java; Simulation

I. INTRODUCTION

This project is an attempt to simulate the Mars Rover, "Curiosity," an explorer built by NASA (National Aeronautics and Space Administration) to explore Mars. The project can be considered an investigative tool for the MSL Science Mission. The simulation incorporates the various sensors on the "Curiosity" rover, emulating their activity over the calibrated mission window. It also makes an attempt to emulate the working of the sensors, by actually obtaining data for most of the sensors to give real-time output from the data presented in NASA's online sensor data archives. This makes the project a fitting research tool. Since it has the potential to be modified to incorporate MSL engineering data, comprising of the physical health of the various sensors onboard the MSL, it could also serve as a diagnostic tool. Making use of the GUI in the simulator, novices too can get a feel of the rover. What makes the Mars Rover simulator portable is that it's written in Java and hence it can run on any computing platform that supports a JVM. Secondly, the simulation by itself can be used as a

collaborative tool to foster work between experts from interdisciplinary branches. The project also provides for good and efficient error logging.

II. LITERATURE REVIEW

A. Mars Rover

Mars rover is a motor vehicle that travels across the surface of the planet Mars upon arrival. A rover instead of a stationary lander would be able to cover more territory. It can be directed to interesting features, such as placing themselves in sunny positions to weather winter months, and they can advance the knowledge of how to perform remote robotic vehicle control.

B. Curiosity

Curiosity is a car-sized rover designed to explore the crater Gale on Mars as part of NASA's Mars Science Laboratory mission (MSL). The goals of the rover are the investigation of Martian climate and geology, assessment of whether the selected field site inside Gale has ever offered any environmental conditions favorable for microbial life, including the investigation of the role of water; and planetary habitability studies in preparation for human exploration. The rover is still operational, and as of February 17, 2019, Curiosity has been on Mars for 2323 sols (2386 total Earth days) since landing on August 6, 2012. Curiosity's design serves as the basis for the planned Mars Rover in 2020.

The general sample analysis strategy begins with high-resolution cameras to look for features of interest. If a particular surface is of interest, Curiosity can vaporize a small portion of it with an infrared laser and examine the resulting spectra signature to query the rock's elemental composition. If that signature is intriguing, the rover uses its long arm to swing over a microscope and an X-ray spectrometer to take a closer look. If the specimen warrants further analysis, Curiosity can drill into the boulder and deliver a powdered sample to either the Sample Analysis on Mars (SAM) or the Chemistry and Mineralogy (CheMin) analytical laboratories inside the rover.^{[2][3][4]} In total, the rover carries 17 cameras: Hazard Avoidance Camera (HazCams) (8), Navigation Camera (NavCams) (4), Mast Camera (MastCams) (2), Mars Hand Lens Imager (MAHLI) (1), Mars Descent Imager (MARDI) (1), and Chemistry and Camera (ChemCam) (1).^[1]

The other instruments/ instrument suites include the Rover Environmental Monitoring Station (REMS) which comprises of instruments to measure the Mars environment:

Revised Manuscript Received on April 12, 2019.

Dylan Dsouza, Department of Computer Engineering, Fr. C. Rodrigues Institute of Technology, Navi Mumbai, India (Email: dsouzadyn@gmail.com)

Joel Miranda, Department of Computer Engineering, Fr. C. Rodrigues Institute of Technology, Navi Mumbai, India (Email: joelmiranda99@gmail.com)

Jariel Gojar, Department of Computer Engineering, Fr. C. Rodrigues Institute of Technology, Navi Mumbai, India (Email: jariel.gojar@gmail.com)

Sheldon Karkada, Department of Computer Engineering, Fr. C. Rodrigues Institute of Technology, Navi Mumbai, India (Email: ash997.sk@gmail.com)

Sanket Korgaonkar, Department of Computer Engineering, Fr. C. Rodrigues Institute of Technology, Navi Mumbai, India (Email: sanket.korgaonkar@gmail.com)

Dakshayani G., Department of Computer Engineering, Fr. C. Rodrigues Institute of Technology, Navi Mumbai, India (Email: dakshayani@fcr.it.ac.in)

humidity, pressure, temperatures, wind speeds, and ultraviolet radiation^{[5][6]}, the Alpha particle X-ray spectrometer (APXS) instrument that irradiates samples with alpha particles and maps the spectra of X-rays that are re-emitted for determining the elemental composition of samples^[7], the CheMin which is the Chemistry and Mineralogy X-ray powder diffraction and fluorescence instrument^[8], The SAM instrument suite analyzes organics and gases from both atmospheric and solid samples^{[3][9]}, the Radiation Assessment Detector (RAD) whose role is to characterize the broad spectrum of radiation environment found inside the spacecraft during the cruise phase and while on Mars^[10] and the Dynamic Albedo of Neutrons (DAN) instrument that employs a neutron source and detector for measuring hydrogen or ice and water at or near the Martian surface^[11].

C. Sample Analysis on Mars (SAM)

The Sample Analysis at Mars (SAM) is a powerful set of three instruments onboard the Mars Science Laboratory (MSL) Curiosity rover that works together to investigate the chemistry of the Martian surface and atmosphere within Gale Crater. SAM's measurements will help scientists better understand environmental conditions over time and assess whether Mars could support and preserve evidence of microbial life, either now or at some time in its past. Though SAM's instruments would fill a laboratory here on Earth, they have been miniaturized to roughly the size of a microwave oven in order to fit inside the Curiosity rover. SAM's instrument suite consists of a Gas Chromatograph (GC), a Quadrupole Mass Spectrometer (QMS), and a Tunable Laser Spectrometer (TLS), as well as systems that manipulate and process samples. SAM will analyze gases, either drawn directly from the atmosphere or extracted from regolith or powdered rock samples by heating or chemically treating the samples. SAM will search for and characterize organic and inorganic molecules important to life on Earth, as well as information about the chemistry of past and present Martian environments^{[3][15]}.

III. EXISTING SYSTEM

A. Kernel

The kernel of the Mars rover is the core of the project, it handles the basic and advanced functionality of the rover. The kernel forms the state machine, thus handling all the operations of the rover as it goes from one state to another and back again. It is thus responsible for orchestrating the functioning of the Mars rover. It binds all the component's functionality together. The kernel goes through the process of initialization also known as bootstrapping in our case, it then runs through each sensor checking their health along with the health of the system's battery which is responsible for keeping the rover alive. Our kernel is a simple state machine which relies on the spacecraft clock and the position sensor.

B. Spacecraft Clock

The spacecraft clock is a critical part of the Mars rover which is responsible for determining the lifespan of the rover. With each usage, the lifespan of the rover decreases. This is kept track by the spacecraft clock. It also gives

valuable metrics like the sol, current UTC time, how much time has passed for the mission as well as the time scale factor since the actual communication has a delay of 20 minutes in real life, we can change this by adjusting the timescale factor. Based on the space clock it is possible to retrieve data logs of the rover. The calibrations of each of the sensors are triggered by the spacecraft clock.

C. Daemon Processes

Daemon processes are used in the Mars Rover for running processes in the background without affecting the other processes that would require specific resources or operational authority. Daemon processes are important for the Mars Rover since they specify the processes which require regular updates to be given to the kernel.

The different daemon processes for Mars rover are :

- Pacemaker
- Garbage Collector
- Battery Monitor
- Sleep Monitor

1. Pacemaker

The pacemaker is responsible for managing the "Heartbeat" of the Mars rover. It sets all the vital information such as battery level, rover configuration, module reporting, Sol. no. and location. Pacemaker enables vital functionalities and periodically keeps examining them to maintain the pulse.

2. Garbage Collector

Garbage Collector is used for maintaining the instruction queue so that the rover is not overloaded with commands especially when it is in the sleep and hibernate state.

The queue monitor of Garbage Collector is implemented using multithreading which implements the Runnable interface. If the instruction queue has reached the maximum limit and the rover is in listening state then the commands are cleared and the number of commands removed is recorded. If the rover is in hibernating state and the queue is not empty then the process functions as expected. There is also a provision for distress signals whereby the shutdown can be initiated when the signal is received and the instruction queue will be cleared.

3. Battery Monitor

The battery monitor is used for checking the battery level and indicating the status for safe process execution to occur. The battery monitor is required to check whether the rover can be used further in the monitored battery level or further charging is required for it to continue. It also indicates the level below which the rover's battery crosses the minimum level. The battery monitor in mars simulator makes use of multithreading in order to efficiently execute the task. It first goes into Listening State in order to check the primary and auxiliary power unit and determines whether there is enough power to execute the commands. If the power is not sufficient, it enters into the Recharge-State and notifies the kernel about it. The commands which are sent during this

period are stored in the instruction queue. If the battery level crosses the alert threshold, an interrupt is sent for the rover to shut down and after recharge, the bootup begins and commands stored in the instruction queue are executed. If the alert threshold is not crossed but the power is still low, the rover enters into hibernating state instead of a complete shutdown. The battery monitor is scheduled to perform regular monitoring at a fixed schedule. At each schedule, the results of the check and current state of the rover are updated

4. Sleep Monitor

Sleep monitor shows the state of the rover. It is utilized to indicate whether the rover is in sleep mode or it can be used for other processes. The different processes check the sleep state beforehand in order to ensure the commands are in accordance with the state specified.

The sleep monitor handles the task of activating the Sleep state and Awaken state. It also makes use of multithreading using runnable interface to handle the sleep states. It receives a value to set the time after which the rover should enter the sleep mode. The monitor then determines the current time and the time at which the message was received. If the difference is zero, it puts the rover in the sleep state. The sleep time specifies the time for which the rover should be in a sleep state. The monitor then determines whether sleep time has been achieved and then awakens the rover from the sleep state. A schedule is set to regulate the sleep monitor.

IV. SAM IMPLEMENTATION & RESULTS

The graphical user interface was developed with a view for bringing access to MSL data to a broader scientific community and doing it with a more intuitive easy interface. It consists of all the sensors, represented as buttons as well as a text box for input.

A. Animation Engine

The animation engine used for the simulation is based upon Java's swing user interface framework. Each sensor has an animation engine associated with it. The animation engine for each sensor is triggered once the kernel is in the corresponding sensor's state. The current animation is 2 dimensional and a blip is used to denote the current position of the rover. Then the animation engine consists of a factory based pattern. It implements the animation factory. This is based on the swing engine. The animation consists of a blip which blinks as the data exchange takes place, which adds visual feedback for the SAM sensor.

B. Protocol Buffers

The protocol buffer developed by Google is a platform independent, language independent, extensible mechanism for serializing structured data. It is similar to JSON (JavaScript Object Notation) and XML (eXtensible Markup Language) but faster and simpler. A .proto file is written for each sensor which is used to structure the data from the NASA data application program interface (API)^[12]. When compiled the buffered messages create access classes which can be used by both Mars Rover and Mission Control modules.

C. Process Communication

For the communication among the modules, Java and Apache Kafka are used. The Mission Control communicates with the Mars Rover module using the Kafka services. The data from the NASA data application program interface (API)^[12] are obtained by the Mars Rover sensors individually based on the currently set space clock (spaceclk) and sol-id. This data is serialized and structured into protocol buffer messages. Every protocol buffer message is a small logical record of information which contains a sequence of name-value pairs. These buffer messages generate data access classes which provide simple accessors for each field as well as methods to serialize/parse the whole structure to or from raw bytes. The Mission Control uses these classes to populate, serialize and retrieve protocol buffer messages.

D. Building the Sensor

The sensor implements the State interface since each sensor is actually a state machine. The rover has a state machine based kernel and thus using a state-based sensor makes it easier to integrate with the entire project. The State machine is just one essential part of the sensor. The main tasks are fetching data from the repositories then getting them stored locally. Now an issue stands due to the volume of data, in order to tackle this, we crawl the data a priori. As new data comes we update our local copy. This reduces network overhead. The sensor has an initial state in which it syncs with the space clock, this is the calibration state. In the calibration state, every parameter is tested if it is working according to spec. Once we implement the sensor, we need to integrate it with the kernel, in order to do this, we need to register it as an equipment of the rover.

E. SAM Results

Fig. 5.1 shows the Graphical User Interface for the Mars Rover simulator. The S.A.M. suite of instruments can have their test performed once an appropriate Ephemeris Time along with the sol id is fixed. The test is initiated as the user selects the SAM sensor from the GUI. The simulator transitions from the 'sleeping' state to the 'listening' to actually illustrating the test performance with the animation engine. Fig. 5.2 illustrates a sample output of the SAM suite of instruments for the test conducted on the Martian day as denoted by sol id="1225". It reflects the chemical composition of the sample picked at the site on the Gale Crater where Curiosity was present on the Martian day with sol id="1225".

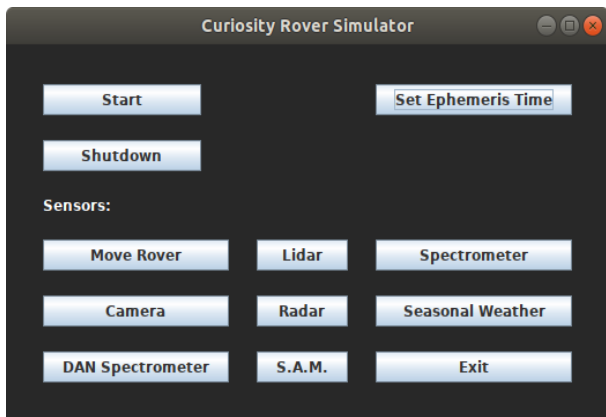


Figure 5.1 depicts the Graphical User Interface created in order to easily control the Mars Rover Simulator.

```
sol = 1225 ExperimentId = e1d25327 URLs = sn25327f1225dr2_spyr_qns_egacomp_1.csv sn25327f1225dr2_spyr_qns_egaxxx_1.csv
"SPECIES", "ABUNDANCE"
=====
1,3-bis(1,1-dimethylethyl)-1,1,3,3-tetramethyldisiloxane",8.00E-10
=====
1,3-bis(1,1-dimethylethyl)-1,1,3,3-tetramethyldisiloxane
8.0E-10
=====
tert-butylidimethylsilanol",4.00E-10
=====
tert-butylidimethylsilanol
4.0E-10
=====
tert-butylidimethylfluorosilane",8.00E-10
=====
tert-butylidimethylfluorosilane
8.0E-10
=====
2,2,2-trifluoro-N-methyl-acetamide",6.60E-09
=====
2,2,2-trifluoro-N-methyl-acetamide
6.6E-9
=====
CH3Cl",1.16E-08
=====
CH3Cl
1.16E-8
=====
CH2Cl2",1.80E-09
=====
CH2Cl2
1.8E-9
=====
HCl",7.90E-07
=====
HCl
7.9E-7
=====
H2O",2.85E-05
=====
H2O
2.85E-5
=====
CO2",7.90E-06
=====
CO2
7.9E-6
=====
SO2",6.60E-06
=====
SO2
6.6E-6
=====
O2",2.10E-06
=====
O2
2.1E-6
=====
NO",2.21E-07
=====
NO
2.21E-7
=====
H2S",5.88E-08
=====
H2S
5.88E-8
=====
```

Figure 5.2 reflects the readings of the SAM test run with sol id="1225"

V. CONCLUSION

Thus, the SAM sensor was implemented by building the sensor module, creating an animation engine, designing a protocol buffer and facilitating process communication. Thereby, the simulator was successfully built in all its entirety forming a software replica of the Mars Rover Curiosity.

We see the future scope of improvement for the SAM suite of instruments and the Rover Simulator in terms of the visualization it provides to the user. We also see it as a diagnostic tool and a testbed for Machine Learning algorithms. We also recognize ample opportunities for analysis of the sensor data to be plausible to extract inferences before manual analysis of data.

REFERENCES

1. Mann, Adam (August 7, 2012). "The Photo-Geek's Guide to Curiosity Rover's 17 Cameras". Wired. Retrieved January 16, 2015.
2. Amos, Jonathan (August 3, 2012). "Gale Crater: Geological 'sweet shop' awaits Mars rover". BBC News. Retrieved August 6, 2012.
3. "MSL Science Corner: Sample Analysis at Mars (SAM)". NASA/JPL. Retrieved September 9, 2009.
4. "Overview of the SAM instrument suite". NASA. Archived from the original on February 22, 2007.
5. "MSL Science Corner: Rover Environmental Monitoring Station (REMS)". NASA/JPL. Retrieved September 9, 2009
6. Gómez-Elvira, J.; Haberle, B.; Harri, A.; Martínez-Frías, J.; Renno, N.; Ramos, M.; Richardson, M.; de la Torre, M.; Alves, J.; Armiens, C.; Gómez, F.; Lepinette, A.; Mora, L.; Martín, J.; Martín-Torres, J.; Navarro, S.; Peinado, V.; Rodríguez-Manfredi, J. A.; Romeral, J.; Sebastián, E.; Torres, J.; Zorzano, M. P.; Urquí, R.; Moreno, J.; Serrano, J.; Castañer, L.; Jiménez, V.; Genzer, M.; Polko, J. (February 2011). "Rover Environmental Monitoring Station for MSL mission"(PDF). 4th International Workshop on the Mars Atmosphere: Modelling and Observations: 473. Bibcode:2011mamo.conf.473G. Retrieved August 6, 2012
7. "MSL Science Corner: Alpha Particle X-ray Spectrometer (APXS)". NASA/JPL. Retrieved September 9, 2009.
8. "MSL Chemistry & Mineralogy X-ray diffraction(CheMin)". NASA/JPL. Retrieved November 25, 2011
9. "Sample Analysis at Mars (SAM) Instrument Suite". NASA. October 2008. Archived from the original on October 7, 2008. Retrieved October 9, 2009.
10. "SwRI Radiation Assessment Detector (RAD) Homepage". Southwest Research Institute. Retrieved January 19, 2011.
11. "Laboratory for Space Gamma Spectroscopy - DAN". Laboratory for Space Gamma Spectroscopy. Archived from the original on May 21, 2013. Retrieved September 20, 2012.
12. "Open Data | NASA Open Data Portal." NASA. Accessed February 19, 2019. <https://data.nasa.gov/>.
13. "PDS Geosciences Node Data and Services." Geosciences. Accessed February 19, 2019. <http://pds-geosciences.wustl.edu/dataserv/default.html>