
Software Requirements Specification

for

Lunar Rover Mapping Robot

Version 2.0

Group: UG12



THE UNIVERSITY
of ADELAIDE

Contents

1	Introduction	1
1.1	Purpose	1
1.2	Intended Audience	1
1.3	Project Scope	1
2	Overall Description	2
2.1	Product Perspective	2
2.2	Product Features	2
2.3	User Classes and Characteristics	2
2.4	Operating Environment	3
2.5	Design and Implementation Constraints	3
2.6	User Documentation	3
2.7	Assumption and Dependencies	3
3	User Requirements	4
3.1	UR 01: Survey Map	4
3.2	UR 02: Map Manipulation	4
3.3	UR 03: Operating the Robot	4
3.4	UR 04: Safety	4
4	System Features	5
4.1	SF 01: Manual Control	5
4.2	SF 02: Autonomous Control	5
4.3	SF 03: Mapping	5
5	External Interface Requirements	7
5.1	User Interfaces	7
5.2	Hardware Interfaces	7
5.3	Software Interfaces	7
5.4	Communications Interfaces	7
6	Other Non-functional Requirements	8
6.1	Performance Requirements	8
6.1.1	NF-R01: Survey Speed	8
6.1.2	NF-R02: Real-Time Communication	8
6.2	Safety Requirements	8
6.2.1	NF-R03: Robot Safety	8
6.3	Security Requirements	8
6.4	Software Quality Attributes	8
6.4.1	NF-R04: Modular Design	8
	Appendix A: Glossary	9
	References	9

Revision History

Version	Date	Reason for changes
1.0	22/8/2017	Initial draft
2.0	28/10/2017	Final version

List of Figures

1	Operator and subsystem interaction	2
2	GUI	7

1 Introduction

1.1 Purpose

This document details the software requirements for the Lunar Rover Mapping Robot Project. It presents the detailed description of the purpose for the system and will explain system features, external interfaces, and non-functional requirements. This document is a guide for developing the initial version of the project and will be proposed to the customer for its approval.

1.2 Intended Audience

This document is intended to be read by both the project client and the project developers. For the client, it specifies what the client should expect of the finished product. For developers, it specifies what features need to be completed in order for the project to be considered complete.

1.3 Project Scope

There are two main objectives of the product specified in this document. The primary objective of the product is to demonstrate a solution, that is within the available budget and capabilities of the organization, in accomplishing the remaining two challenges outlined in the **Google Lunar XPRIZE**[1], (1) robot must travel 500 meters on the surface of the Moon and (2) transmit high-definition video and images back to earth, accomplishing this objective will give the organization the edge in winning the prize money of US\$20 million. The secondary objective is to survey a particular area and locate the specific location of the remnants of the Apollo 17 landing site from December 1972 [2] and provide precise measurements of the tracks that lead to/from it, accomplishing this objective will satisfy the organization's interest.

2 Overall Description

2.1 Product Perspective

This SRS document presents the proposed software requirements for a Lunar Rover Mapping Robot prototype. The product consists of two major components, the controller and the mapping robot. These components interact with each other as shown in figure 1. The controller, which is operated remotely by a human operator through use of a GUI, sends commands to the robot for it to execute. The mapping robot will send sensor data from the Moon to the controller to be processed.

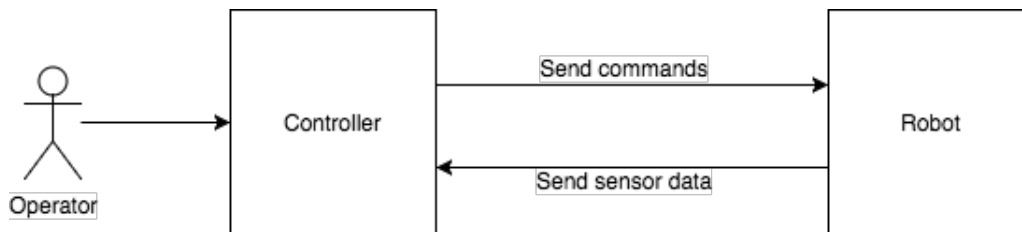


Figure 1: Operator and subsystem interaction

2.2 Product Features

Major features of the Lunar Rover Mapping Robot specify that

- The system will display the entire survey area in real-time, including with the current location of the vehicle, the mapped and unmapped regions, debris and radiation area;
- The system will store and read the mapping data in **Extensible Markup Language (XML)** format;
- The system will enable the operator to zoom in and out on a particular area on the map;
- The system will enable the operator to move the robot to a specific point on the map;
- The system will enable the operator to mark **No-Go-Zones (NGZ)** on the map;
- The system will provide manual or autonomous control of the mapping robot to the operator;
- The system will enable the operator to stop the vehicle immediately;
- The system will be capable of detecting objects to avoid collision or minimize the collision force.

2.3 User Classes and Characteristics

Those interacting with the system will be knowledgeable about the system, how it operates, and its capabilities. They will be familiar with the functionality of the system, such as sensors and actuators, and how they interact with one another. These users will have unrestricted control over the robot. They will control the robot remotely from Earth.

2.4 Operating Environment

Operators will run the controller application on their own computers and connect to the robot wirelessly. As the controller works on any platform with the Java running environment, the operator's computer is required to have Java installed.

2.5 Design and Implementation Constraints

The system will be coded as an event-driven program. A Handler module will be used for handling event command and sensor data transfers between the controller and the robot. The low-level functionality will be managed by a separate Robot module. Queues are used to communicate messages between modules. As an emergency stop button is needed, the priority of received messages must be considered.

The LeJOS operating system for the Lego Mindstorms EV3 will be used to interact with the robot.

2.6 User Documentation

Demonstration will be provided to operators by developers as a tutorial for the correct usage of the system. The demonstration will include how the controller connect to the robot side, and how to control the robot manually or autonomously through the controllers.

2.7 Assumption and Dependencies

We assume that there will be minimal communication delay between the computer/controller and the robot, which will be on the surface of the Moon. We also assume that objects found on the **A1** paper map will be 3D objects with sizes that are capable of being accurately detected by the sensors. In addition, we assume that the organization would provide us with a **document type definition (DTD)** for the mapping data and that there are no restrictions on how many wheels the robot should have.

3 User Requirements

3.1 UR 01: Survey Map

Description: The system map shall be able to show found features, current location of the vehicle, and distinguish between mapped and unmapped regions, all in real time.

Rationale: It is vital to the system to be able to see the map and interact with the robot.

Acceptance Criteria: Real time: Without significant delay. Map details, including features, vehicle location and mapped/unmapped areas, shall be clearly visible.

Priority: High.

Source: Client Project Specification

3.2 UR 02: Map Manipulation

Description: The map needs to be able to be imported and exported.

Rationale: Exporting means that the map data acquired by the robot can be processed and used externally. Importing means that the robot can take advantage of previous work.

Acceptance Criteria: The map export must conform to the DTD. The map import must be able to import any XML file conforming to the DTD and successfully show it on screen.

Priority: Medium.

Source: Client Project Specification.

3.3 UR 03: Operating the Robot

Description: The operator must be able to instantly stop the robot. The operator shall be able to designate a location for the vehicle to immediately move to.

Rationale: The operator needs to be able to stop the robot instantly for the safety of the robot. The operator needs to control where the robot goes.

Acceptance Criteria: "Stop" means no robot motion. "Immediately" means as soon as communication allows. It shall be able to stop at any time.

Priority: High

Source: Client Project Specification.

3.4 UR 04: Safety

Description: The robot must not collide with anything with any significant force.

Rationale: This may damage the robot and surroundings.

Acceptance Criteria: The object being collided with shall not move more than 2 cm.

Priority: Medium

Source: Client Project Specification.

4 System Features

4.1 SF 01: Manual Control

Description: The system shall enable manual control mode. In manual mode, the user has control over the robots movement through the UI. Both modes includes being able to stop the vehicle at any time.

Rationale: Manual mode is necessary in case something goes wrong with the autonomous mode, or the users judgment is necessary.

Acceptance Criteria: This requirement can be verified in part by entering manual mode and verifying that the robot is being controlled through user input.

Priority: High

Source: UR 03, UR 04

Functional Requirements:

1. GUI Design.
2. Robot movement control.
3. Communication between robot and program.

4.2 SF 02: Autonomous Control

Description: The system shall enable autonomous control mode. In autonomous mode, the user will set a path and the robot will determine the best path to use and direct itself there. While in autonomous mode, the robot will avoid hard collisions with objects and avoid craters, debris and radiation areas. In manual mode, the user has control over the robots movement through the UI. Both modes includes being able to stop the vehicle at any time.

Rationale: Autonomous mode is necessary for the robot to move without the user specifying every step. This is useful because the robot can make decisions quickly and efficiently.

Acceptance Criteria: This requirement can be verified by entering autonomous mode and verifying that the robot can navigate its own way through the terrain.

Priority: Medium.

Source: UR 03.

Functional Requirements:

1. AI path finding
2. GUI destination selection.

4.3 SF 03: Mapping

Description: The system shall provide support for mapping functions. The user can designate No Go Zones (NGZ) and points that the robot should navigate to. The user shall also be able to interact with the map, including zooming. The map shall be in XML format and the system shall both read and write to the map. A partial map needs to be able to be uploaded and used. The robot shall also be able to sense obstacles and features and have them displayed on the map.

Rationale: The user needs to be able to navigate the lunar rover around the Moon in manual mode, as well as set, monitor and correct progress in autonomous mode. The map allows these to happen.

Acceptance Criteria This requirement can be verified by uploading a partial map and updating the map with new information from the robot or the user.

Priority: High.

Source: UR 01, UR 02.

Functional Requirements:

1. Internal map storage.
2. XML importing and exporting.
3. GUI Map display.
4. Robot sensors

5 External Interface Requirements

5.1 User Interfaces

Users will interface with the software via an Java graphical user interface. The interface is composed by three sections, which are controller section, map display section and switch section. User can manually control the robot to send forward, backward, turn clockwise, turn anticlockwise and emergency stop in controller section. The map will be shown to the user in map section and user can adjust the size of the map by zoom in/out. The map can be exported via "save" button. User can also switch the robot into automatic mode or manual mode. 2.

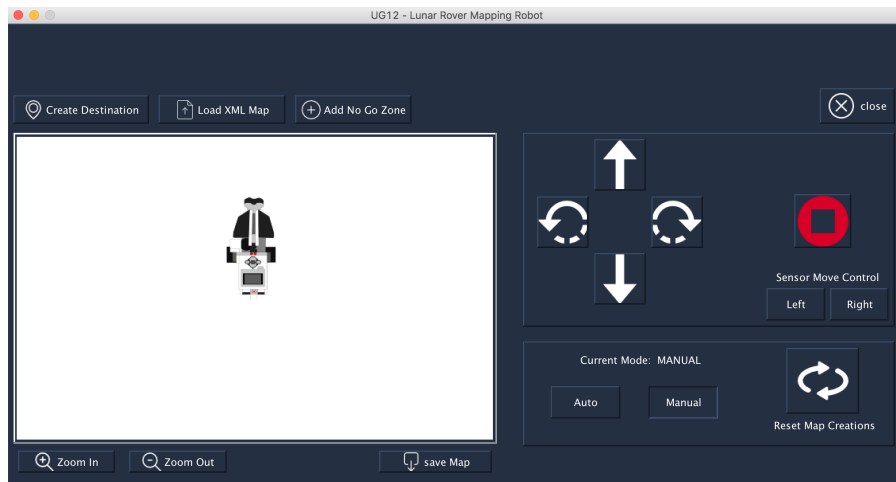


Figure 2: GUI

5.2 Hardware Interfaces

The robot with which the software interfaces will have two motor-driven wheels to provide control over the motion of the robot. The robot will communicate sensor data back to the software.

5.3 Software Interfaces

The software will connect to a Lego Mindstorms EV3 unit running the LeJOS operating system. The connection will be made using the Java RMI mechanism supported by LeJOS.

5.4 Communications Interfaces

The software will communicate with the robot over a wireless network connection. The connection may be maintained over Bluetooth or WiFi standards.

6 Other Non-functional Requirements

6.1 Performance Requirements

6.1.1 NF-R01: Survey Speed

Description The robot must be able to complete a survey of the demo in less than 20 minutes.

Rationale The final demonstration of the robot will be no longer than 20 minutes, so must be able to complete an autonomous survey with in that time, and ideally in much less.

6.1.2 NF-R02: Real-Time Communication

Description The rover must be able to react to commands and send feed back in real time.

Rationale When under manual control, the rover needs to react to updates in commands to provide smooth motion. When under autonomous mode, the rover must respond to any emergency commands. The rover must also relay feedback promptly to aid in pathfinding decisions.

6.2 Safety Requirements

6.2.1 NF-R03: Robot Safety

Description The Rover must suffer no damage to its equipment, whether from falling into a crater, hitting a physical obstacle, or by any other means.

Rationale The Lunar Rover will be operating on the surface of the moon, controlled remotely from Earth. If damage occurs, there will be no way to repair or recover the Rover.

6.3 Security Requirements

Description The robot must be appropriately secured.

Rationale As this is a prototype, security protocols will not be implemented.

6.4 Software Quality Attributes

6.4.1 NF-R04: Modular Design

Description The software will be made as separate modules that communicate between each other.

Rationale Using a modular design will allow each module to be modified or replaced without significantly affecting other modules, should changes need to be made to the robot or the mapping system.

Appendix A: Glossary

Acronyms

DTD	Document Type Definition
NGZ	No-Go-Zones
SRS	Software Requirements Specification
UI	User Interface
XML	Extended Markup Language

References

- [1] Google Lunar XPRIZE. 2017. *Google Lunar XPRIZE*. [ONLINE] Available at: <https://lunar.xprize.org>. [Accessed 21 August 2017].
- [2] Wikipedia. 2017. *Apollo 17*. [ONLINE] Available at: https://en.wikipedia.org/wiki/Apollo_17. [Accessed 21 August 2017].