Software Requirements Specification for Software Engineering and Project

Group 2 *Coding Pharaohs*

Abdulaziz ALHULAYFI a1642362
Yu HONG a1616861
Jianqiu LI a1635717
Matthew NESTOR a1132338
Yifei PEI a1611648
Bowen TAO a1622211

November 4, 2013
Final Document

Contents

Revision History

Name	Date	Reason For Changes	Version
Yu hong,JianQiu Li	22/08/2013	Initial Version	0.1
Yu hong	24/08/2013	Minor Modification	0.2
Yifei Pei	25/08/2013	Minor Modification	0.3
Abdulaziz Alhulayfi	25/10/2013	Modification	0.4
Abdulaziz Alhulayfi	03/11/2013	Adding New System Features and Use Cases	1.0

1 Introduction

1.1 Purpose

This document covers the specification of software requirements for developing the SEP project in semester 2, 2013. The project objective is to develop a Robot that is capable of marking road closures and identifying dangerous areas such that people are alerted not to move into these areas. The project specifications and software requirements specified in this document are collected from the clients.

The scope of this document is to specify the requirements that serves both the host program operation and the robot. The robot should be able to detect dangerous areas and send the information back to the host, and the operator at the host side is able to control the movement of the robot.

1.2 Document Conventions

The specification of the software requirements consists of functional requirements, nonfunctional requirements, and security requirements. Each requirement will be classified into High, Medium, or Low priority, which will be tagged in its corresponding requirement statement. Priority classification indicates the importance of the requirement to the project success

1.3 Intended Audience and Reading Suggestions

This document is intended for all stakeholders of the SEP project in 2013, which include clients, who are the users of the system, and all members of group two in this class, who are contributing to the project.

The below sections contain the specification of software requirements which consist of functional requirements, nonfunctional requirements, and user requirements for both the robot and host systems, which are important information for the designers, developers and testers. Designers of the system need this information to design the architecture of the system, which is a key procedure before developers could start their work. Developers of the system also need these information to guide them implementing specific functions. Furthermore, testers need to test the specific functions of the system with accordance to these provided requirements.

Since this document contains all the requirements that describe the functionalities of the system, it is recommended that this document should be thoroughly read.

1.4 Project Scope

The project scopes are to produce a prototype of the robot required in SEP project, and to deliver the corresponding documents according to the schedule.

The robot should be able to identify and mark the road closures as well as dangerous areas in the map on both the robot and host side. To achieve this goal, a whiteboard marker would be mounted to the robot. Thus, whenever the robot marks on the map, the GUI (i.e., Graphical User Interface) on the host system would dynamically display the changes, with an XML (i.e., Extensible Markup Language) document being generated and updated. In addition, the robot should be able to automatically conduct the map by means of all possible features detected.

With regard to the documentation of this project, SRS, SDD, SPMP, and user manual will be created according to the schedule of the project and maintained through the process of developing phase.

1.5 References

References

[1] Project description

Sheng, M 2013 , ”Project Description”, <https://cs.adelaide.edu.au/users/third/sep/13s2-sep-Adelaide/project/ProjectDescription.pdf>

[2] Client's meeting minutes_12-08-13

Pei, Y 12/08/2013, https://version-control.adelaide.edu.au/svn/sep2013-2/documentation/minutes/client-meetings/1_client_meeting_minutes_12-08-13.pdf

[3] Client's meeting minutes_20-08-2013

https://version-control.adelaide.edu.au/svn/sep2013-2/documentation/minutes/client-meetings/2_client_meeting_minutes_20-08-2013.pdf

[4] Lego tutorial

http://lejos.sourceforge.net/nxt/nxj/tutorial/

[5] Blue-tooth features

http://www.bluetooth.com/Pages/Basics.aspx

[6] Sample XML file in the format specified in DTD

http://forums.cs.adelaide.edu.au/file.php?file=%2F523%2Fsep2013-map.xml

2 Overall Description

2.1 Product Perspective

This software product consists of major two components.

The first one is to run on the robot, a pre-existing LEGO Mindstorms NXT, for detecting and marking dangerous areas in the city. The second is the one that resides in the host system, which is capable of controlling the activities of the robot and displaying the mapping information.

Continuous communication is required between the robot and host system and is achieved by means of Bluetooth device.

2.2 Product Features

The product consists of two components: the robot side and host side. The main features of the robot side includes:

- The robot should be able to explore a city area and identify dangerous areas autonomously. After the robot arrives at the initial starting position of the city area, it would start to explore and map the site without intervention from the operator.
- The operator is allowed to commence manual control when necessary.
- The robot should be able to mark the road closures, in the form of:

  A visual map constructed on paper/card;
  A visual representation in the map area of the GUI;
  textual representation XML file, which abide by the format specified by the DTD (i.e., Document Type Definition) file.
- The robot should be able to stop autonomously for the sake of safety in the case of low battery level or losing communication with the host side.

The main features of the host side includes:

- The GUI (i.e., Graphical User Interface) on the computer allows the operator to manually control and monitor the robot's movement, which is achieved by a set of buttons and a map panel, respectively.
- The operator is able to establish a communication with the robot by means of the GUI and Bluetooth device.
- The operator is able to stop the movement of the robot using the emergency stop button in the GUI whenever an emergency occurs.

2.3 User Classes and Characteristics

Since programming techniques and technical know-how are not required, people from different fields are able to use this product after some basic training.
In general, users can be classified into the following three groups:

- Trainer and Trainee
  For educational purposes, trainees can be motivated to create their own products for similar or distinguished critical tasks, after this product is demonstrated by the trainers.

- Operator of Exploration Projects
  Operators of exploration projects can use this product to explore and map the area where it might be dangerous for human to be exposed, which would be more efficient and accurate than manually mapping by humans.

2.4 Operating Environment

The application will be installed on a pre-built Lego Mindstorms NXT robot, which has 256 Kbytes Flash and 64 Kbytes Ram on the leJOS 0.9.1 pre-flashed firmware. Also, the application shall be compatible with most operating systems (i.e MacOSX, Windows, Linux)

2.5 Design and Implementation Constraints

The design and implementation constraints originate from the pre-built robot hardware and client requirements. These constraints are listed as follows:

- Hardware constraint. The Lego Robot has only 256 Kbytes to store the instructions, which is a restriction to development of the artificial intelligence of the robot.
- Programming language constraint. The system must be implemented in Java lejos firmware.
- The mapping data must be saved in the XML file abiding by the format defined in DTD.
- The tool 'make' must be used to build any version of the software.
- The software shall be tested in JUnit.
- The project shall be documented using latex documentation language.

2.6 User Documentation

- API Documentation. The source code of the software will be documented in Javadoc to hyperlink related documents together, which would generate an API (i.e., Application Programming Interface) documentation in HTML format. This API Documentation would facilitate the developers and maintainers to maintain the source code.
- User Manual. The user of the software will be provided with a User Manual, which contains the guide to the use of GUI and control of the robot.

2.7 Assumptions and Dependencies

- The communication between the robot and host computer is achieved by Bluetooth device, which has a maximum range of approximately 15 meters. We assume that the robot and the host computer would always be within this range during the mapping process.
- We assume that the operator is able to monitor the movement of the robot clearly while it is mapping.

3 User Requirements

3.1 Robot requirements

3.1.1 R01: Movement of the robot

Description:
The robot shall be able to move forward or backward and have the ability to rotate left or right, on a plain surface. The movement of the robot should be conducted in the following two conditions:

- The robot is moving under the operators control.
- The robot is moving autonomously by means of intelligence developed for it within the boundaries of the city.

Rationale:
The robot is used to explore the city area, which requires the robot the ability to go forward or backword, rotate left or right.

Acceptance Criteria: Run the program, connect the PC to the robot and press each movement direction button.

Source: Project Description 2 Some Requirments.

Priority: High

3.1.2 R02: Obtaining current location of the robot

Description:
The robot shall be able to calculate the current location in the map, known as city area. This location information would be sent back to the host continuously in order to obtain the path this is transversed by the robot.

Rationale:
Both the robot and the operator need to ensure that the robot is in the range of city area. Therefore, information of the current location would help to avoid crossing the boundaries. In addition, the robot is required to mark a road closure or avoid obstacle on both the map in the form of paper, visual representation on the GUI and XML file, which needs the information of current location.

Acceptance Criteria: Connect PC to the robot, load XML map file and move the robot to see robot changes its location accurately.

Source: Project Description 2.1 The Map.

Priority: High

3.1.3 R03: Identifying Road Closures and Obstacles

Description:
The robot shall be able to identify the road closures and obstacles with accordance to the data provided in the pre-defined or half-defined XML file, and mark them on the map, represented as paper.

Rationale:
The robot shall avoid to enter the dangerous areas in order to ensure its safety, which is of top priority.

Acceptance Criteria: Connect PC to robot, run the robot on manual or automatic mode to expore and mark road closures by lowering down its marker to mark the closure and avoid obstacles.

Source: Project Description 2.2 Map Representation for Prototype

Priority: High

3.1.4 R04: Identifying boundaries of the map

Description:
The robot shall be able to identify the boundaries of the city area and avoid wandering out of it, under both manual and automatic mode.

Rationale:
The robot is responsible for exploring the given city area. Crossing the boundaries could cause a potential hazard for the safety of the robot and make the mapping process less efficient.

Acceptance Criteria: Run the robot to explore the map and check whether it stays withing the boundaries. The robot shall not cross outside boundaries.

Source: Project Description 2.1 The Map.

Priority: High

3.1.5 R05: Detecting obstacles

Description:
Using ultrasonic sensor and bump sensor, the robot shall be able to detect and locate obstacles around it.

Rationale:
The city area to be explored may have many obstacles on the road. The robot equipped with ultrasonic sensor and bump sensor is able to detect these obstacles and have them marked on the map.

Acceptance Criteria: Place an obstacle and run the robot into it to see the robot detecting the obstacle.

Source: Project Description 2.1 The Map.

Priority: High

3.1.6 R06: Detecting roads

Description:
Within the city area, there are a number of roads, in which the robot is supposed to move. As roads are represented black lines, the robot can distinguish the roads by the changes of colors using light sensor.

Rationale:
In order to explore the city, the robot needs to transverse from point A to point B, which is achieved by moving along the road. Therefore, the robot should be capable of detecting roads using light sensor in order to accomplish its task.

Acceptance Criteria: Place the robot on road represented by a black line and run the robot. The robot shall not move out the road.

Source: Client Meeting Minutes

Priority: High

3.1.7 R07: Detecting intersections

Description:

Within the city area, there are a number of intersections which the robot needs to distinguish. These intersections are represented by black circles. The robot can distinguish the intersections by the changes of colors using light sensor.

Rationale:

In order to explore all paths in the city, the robot needs to know all intersections that it has traveled through. The robot can accomplish this task by identifying the black circle spots.

Acceptance Criteria: run the robot on an intersection to see the robot choses a certain path

Source: Client Meeting Minutes

Priority: High

3.1.8 R08: Marking road closures

Description:

Road closures shall be marked under the following two circumstances:

- Road closures shall be displayed on the map if they are already pre-defined in the XML file.
- The robot shall mark the road closure whenever it detects one.

When the robot marks the road closure, it needs to mark in the form of:

- A visual map constructed on paper/card;
- A visual representation in the map area of the GUI
- textual representation XML file, which abide by the format specified by the DTD (i.e., Document Type Definition) file.

Rationale:

The city struck by natural disaster leaves several area unsafe. Marking road closures could help to prevent people from moving into these areas.

Acceptance Criteria: Run the robot into a pre-defined road closure for the robot to mark road closure on the map

Source: Project Description 2.1 The Map.

Priority: High

3.1.9 R09: Accepting commands

Description:

The robot shall be able to perform commands from the operator at all times.

- Under manual control mode, the robot should be able to accept and perform the command from the operator, such as forward, backward and rotate.
- Under automatic control mode, the operator normally do not intervene in the process of mapping. However, the operator can interrupt the automatic mode by giving manual commands to the robot when necessary.

Rationale:
The communication between the host and the robot is represented as giving and accepting commands.

Acceptance Criteria: Press on any buttons in the GUI. The robot shall respond to the pressing by performing the right action.

Source: Project Description 2.3 Operation

Priority: High

3.1.10 R10: Automated mapping

Description:
After manually directed to the initial starting position, the robot should start mapping automatically. The process of mapping will be terminated or interrupted under the following circumstances, respectively:

- The mapping is completed
- Unexpected event occurs, such as loss of communication with the host or wandering out of the city boundaries

Rationale:
Automated mapping could help to reduce time, cost and effort required to accomplish the task than manual control mode.

Acceptance Criteria: Run the robot on the automatic mode. The robot shall start exploring independentlly.

Source: Project Description 2.1 The Map.

Priority: High

3.2 Mapping requirements

3.2.1 M01: Displaying transversed path

Description:
During the process of mapping, the transversed path of the robot shall be displayed on the GUI of the host side.

Rationale:
The transversed path displayed on the GUI could facilitate the operator to trace the history movement record of the robot.

Acceptance Criteria: Run the robot on the map to follow certain paths. The transversed path of the robot shall be displayed on the GUI of the host side

Source: Project Description 2.1 The Map.

Priority: High

3.2.2 M02: Map Saving

Description:
During the process of mapping, the obtained mapping information, such as road closures and obstacles, as well as the transversed path information would be sent to the host side and stored in the form of XML file abiding by the format specified in the DTD.

Rationale:
The result of mapping should be stored in a document format such that the map could be reused.

Acceptance Criteria: Chose the save option from the file options to save a currenntly explored map.

Source: Project Description 2.1 The Map.

Priority: Medium

3.2.3 M03: Map Loading

Description:
The host system should be able to open a saved map in the form of XML file in the format defined in the DTD. The opened map would be displayed on the GUI of the host side.

Rationale:
The host system should be able to reuse the map that is previously saved. In addition, the map can be updated during the mapping process, if the map was partially completed before.

Acceptance Criteria: Chose the load option from the file options to load XML file map from the PC

Source: Project Description 2.1 The Map.

Priority: Medium

3.2.4 M04: Marked road closures and obstacles in XML

Description:
Information in relation to road closures and obstacles would be recorded in the form of textual representation XML file, which abides by the format specified by the DTD.

Rationale:
Road closures and obstacles are the main objects the robot is required to detect if they are not already defined in XML file. Therefore, these information should be stored in an appropriate document.

Acceptance Criteria: Chose the save option from the file options to save a currenntly explored map. The map shall be saved in XML file format in the PC

Source: Project Description 2.1 The Map.

Priority: High

3.3 Host requirements

3.3.1 H01: GUI

Description:
The program shall provide a graphical user interface which allows the operator to control and monitor the movement of the robot. The GUI should contain a number of buttons to perform the communication with the robot, such as connect to robot, movement directions control, mark road closure, stop. In addition, the GUI also contains a map panel that would display the mapping process under both the manual control mode and automatic mode. Finally, the GUI provides a list menu that allows the operator to save and load map in the form of XML file.

Rationale:

GUI is an important interface for communication between the operator and the robot. In addition, the map displayed on the GUI could also help the operator to monitor the movement of the robot.

Acceptance Criteria: Run the program for the user interface to be generated with all buttons needed

Source: Client Meeting Minutes

Priority: High

3.3.2 H02: Manual Control

Description:

The operator shall be able to control the movement of the robot under the manual control mode by means of GUI, which contains a number of buttons that could control the robot movement and other features, such as mark road closure, return to base, and stop.

Rationale:

- Under manual control mode, remote control allows the operator to control the movement of the robot in order to accomplish some certain task, such as arriving at the initial start position.

Acceptance Criteria: Run the robot and press any manual control button. The robot shall respond to the pressing by performing the right action.

Source: Project Description 2.3 Operation

Priority: High

3.3.3 H03: Displaying the robot current position on GUI

Description:

The GUI on the host side would provide a map panel that displays a visual representation of the map with the robot current position shown on it.

Rationale:

The operator should monitor the robot position at all times in case that the robot crosses the boundaries or enters a dangerous area.

Acceptance Criteria: Run the robot on the physical map with XML map loaded on GUI. The robot current position shall be displayed accurately.

Source: Client Meeting Minutes

Priority: High

3.3.4 H04: Emergency Stop

Description:

Under automatic mode, the operator can stop the process of mapping when observing an emergency.

Rationale:

When unexpected event occurs, such as bumping in a obstacle, the operator can choose to stop the process of mapping in order to ensure the safety of the robot.

Acceptance Criteria: Run the robot and press the button ”STOP”. The robot shall respond to the pressing by stoping any action.

Source: Project Description 2.4 Safety

Priority: High

3.3.5 H05: Bluetooth Communication

Description:
The host system should be programmed in such a way that the Bluetooth communication is enabled.

Rationale:
The communication between the robot and the host is achieved by Bluetooth device. Therefore, it is necessary to implement Bluetooth communication in order to establish communication between them.

Acceptance Criteria: Turn on the robot and press connect button in the GUI. Wait for few seconds for the connection to be established

Source: Client Meeting Minutes

Priority: High

3.3.6 H06: Alert Message

Description:
Once the robot is enforced to stop due to the event of connection interruption or low battery power, one alert message is going to be sent to the operator telling the inferior circumstances of the robot. On the host side, the message will be shown on the GUI so the operator can reflect accordingly.

Rationale:
The robot should have a safe way to deal with the occurrence of inferior situations such as low battery power and connection interruption. After the emergency stop to make the robot remain safe, the operator should have a way to know the information so actions can be taken to make the robot work again.

Acceptance Criteria: Plug out the bluetooth device for alert message to be displayed in GUI.

Source: Client Meeting Minutes

Priority: Medium

4 System Features

4.1 Manual Control

4.1.1 Description

Priority: High

The system allows the operator to manually control the movement of the robot to perform the basic movements, such as moving forward or backward, rotating left or right, stop, etc. The manual control can be conducted under manual control mode, which are interchangeable by means of the buttons on the GUI.

4.1.2 Stimulus/Response Sequences

After the Bluetooth connection between the robot and the host side is established, the operator can control the movement of the robot by means of the buttons on the GUI. During the process of automatic mapping, the automated behaviour of the robot can be halted by any operation of the operator.

4.1.3 Requirements

The details of Functional requirements of this feature are described in Section 3.3 as in number H02.

4.2 Automatic Mapping

4.2.1 Description

Priority: High

Once the automatic mode is selected, the robot is able to perform the process of mapping automatically without the intervention of the operator.

4.2.2 Stimulus/Response Sequences

To perform automatic mapping, the operator should press the corresponding button to activate the robots automatic mode. Then the robot would commence the mapping procedures with accordance to algorithms developed for it. During the process of mapping, the robot would mark the obstacles and road closures in the means specified in 3.1 as in number R08. After the mapping process is completed, the operator is able to save the result of mapping in the means specified in 3.2 as in number M02.

4.2.3 Requirements

The details of Functional requirements of this feature are described in Section 3.1 as in number R10.

5 External Interface Requirements

5.1 User Interfaces

A user interface is provided for the operator to manually control and monitor the movement of the robot. The user interface should consist of four parts: command buttons, robot information area, map area and list menu. The details of these parts are listed as follows:

- Command buttons. A number of buttons that allows the operator to control the movement of the robot. The command buttons include: forward, backward, left, right, stop, connection (i.e., to establish the connection with the robot), automatic mapping. The robot would accept and perform the corresponding commands when the operator presses these buttons.
- Robot information area. This area contains the information in relation to the current battery level of the robot, the distance between the robot and objects in the front of it, and connection status of the robot.
- Map area. The current map would be displayed in this area along with the path that the robot transversed. In addition, the current location of the robot would also be shown within the map.
- List menu. List menu contains the functions that allow the operator to save and reload the map in the form of XML file in the format specified by the DTD.

5.2 Hardware Interfaces

As specified in Section 2.4, a pre-built Lego Mindstorms NXT robot would be provided to serve as the environment for running the software application. The robot has 256 Kbytes Flash and 64 Kbytes Ram on the leJOS 0.9.1 pre-flashed firmware, which also contains a Bluetooth module to allow the communication between the robot and the host.

5.3 Software Interfaces

The software application consists of two systems: robot control system and host system. The robot control system will be running on Lego Mindstorms NXT robot with the firmware of version leJOS 0.9.1, while the host system would require Java 6 or higher version platform.

5.4 Communications interfaces

The communication between the robot and the host side is achieved by means of the Bluetooth device without any encryption.

6 Use Cases

6.1 UC001: Robot Movement Control

Precondition: The robot must be turned on, connected to the PC and placed on the map.

Basic Flow:

1. The robot is placed on the starting point of the map and is ready to recieve commands from the operator.
2. The operator presses on moving forwards button for the robot to move forward.
3. The robot moves forwards.
4. The operator releases the move forwards button.
5. The robot stops moving forwards.
6. The operator presses the rotating right button.
7. The robot rotates 90 degrees clockwise and stops.
8. The operator presses the rotating left button.
9. The robot rotates back 90 degrees anticlockwise and stops.
10. The operator presses on moving backwards button for the robot to move backwards.
11. The robot moves backwards.
12. The operator releases the move backwards button.
13. The robot stops moving backwards.

6.2 UC002: Map Loading and Saving

Precondition: The robot must be turned on, connected to the PC and placed on the map.

Basic Flow:

1. The operator choses the ”Load” option from file menu options.
2. The operator browse for XML format map in PC.
3. The operator choses the XML file.
4. The operator open the XML file.
5. Map is loaded in the graphical user interface.
6. The robot is placed on the starting point of the map and is ready to recieve commands from the operator.
7. The operator presses on ”Start Auto Mapping” button for the robot to start exploring.
8. The robot starts moving and exploring the map.
9. The robot marks road closures and obstacles.
10. The robot finishes exploration of the map and stops.
11. The operator choses the ”Save” option from file menu options.
12. The operator locate where the map shall be saved in PC.
13. The operator saves the XML file map.

7 Other Non-functional Requirements

7.1 Performance Requirements

7.1.1 P01: Real Time Mapping

Description:

The system is required to construct the map in real-time during the process of mapping. The maximum delay can be accepted will be less than 0.5 seconds, as it is a safety-critical project.

Rationale:

With real time mapping, the operator can immediately observe any changes on the map in order to ensure the safety of the robot and completeness of mapping.

Acceptance Criteria: Load XML map and run the robot. Mapping shall be parallel with the robot movement

Source: 1_client_meeting_minutes_12-08-13

Priority: Medium

7.2 Safety Requirements

7.2.1 SA01: Moving Speed

Description:

The default speed of the robot shall be 5cm/s that is to ensure speed is safe for the robot

Rationale:

The robot shall not move fast, otherwise the robot could be damaged due to too much force by bumping in any obstacles. The speed option shall be between 0cm/s and 15cm/s.

Acceptance Criteria: Run the robot and change the speed from the speed slider in GUI. The robot shall increase or decrease its speed accordinglly.

Source: 1_client_meeting_minutes_12-08-13

Priority: High

7.2.2 SA02: Autonomously halt

Description:

The robot should be able to halt autonomously under the following two circumstances:

- Communication interruption
- Low battery power

After the autonomous halt, the robot should be able to send alert to the operator so the operator will notice the interrupted circumstance.

Rationale:

The city area to be explored could be very dangerous. When the above events occur, the best solution would have the robot autonomously halt.

Acceptance Criteria: Plug out the bluetooth device for alert message to be displayed in GUI. Run the robot with a low battery level.

Source: 1_client_meeting_minutes_12-08-13

Priority: High

7.3 Security Requirements

7.3.1 SE01: Bluetooth Communication Security

Description:
The connection between the robot and the host should be established after the operator inputs a 4 digits pin number.

Rationale:
The LEGO Mindstorms NXT robot provides a Bluetooth pairing with 4 digits pin to avoid any unintentional connection.

Acceptance Criteria: Before establishing a connection the robot shall ask the operator to enter 4 digits pin number for the connection to be established.

Source:

Priority: Low

7.4 Software Quality Attributes

7.4.1 SO01: Maintainability

Description:
After being delivered, the product shall be easy to use and maintain for users.

Rationale:
The user of the product may have new requirements that would need some modification on the system. Therefore, using the coding convention and appropriate documentation, the system would be easy to maintain.

Acceptance Criteria: Ensure all deliverables are clearly explained and well understood

Source: 1_client_meeting_minutes_12-08-13

Priority: Medium

8 Other Requirements

8.1 Quality Requirements

8.1.1 O01: Delivery Requirement

Description:

The allocated tasks should be finished and submitted on time with good quality.

Rationale:

Each member in this group are endeavoring to deliver the best product for the client.

Acceptance Criteria: NIL

Source: Group Meeting Minutes

Priority: High

A Glossary

API Application Programming Interface

DTD Document Type Definition

GUI Graphical User Interface

PIN Personal Identification Number

SDD Software Design Document

SEP Software Engineering and Project

SPMP Software Project Management Plan

SRS Software Requirements Specification

XML Extensible Markup Language