Software Requirements Specification

for

Autonomous Underwater Vehicle

RoboSub Mission Planning

Version 1 approved

Prepared by Brandon Pham

Heriberto Gonzalez

Ricardo Medina

Wilson Weng

California State University Department of Engineering

December 12, 2020

**Table of Contents**

Table of Contents................................................................................................................. pg 2

Revision History................................................................................................................... pg 3

1. Introduction................................................................................................................ pg 4

1.1. Purpose........................................................................................................... pg 4

1.2. Intended Audience and Reading Suggestions................................................ pg 4

1.3. Product Scope................................................................................................ pg 4

1.4. Definitions, Acronyms, and Abbreviations .................................................. pg 4

1.5. References...................................................................................................... pg 5

2. Overall Description.................................................................................................... pg 6

2.1. System Analysis…......................................................................................... pg 6

2.2. Product Perspective........................................................................................... pg 6

2.3. Product Functions........................................................................................... pg 6

2.4. User Classes and Characteristics.................................................................... pg 7

2.5. Operating Environment.................................................................................. pg 7

2.6. Design and Implementation Constraints........................................................ pg 7

2.7. User Documentation...................................................................................... pg 7

2.8. Assumptions and Dependencies.................................................................... pg 7

2.9. Apportioning of Requirements...................................................................... pg 7

3. External Interface Requirements............................................................................... pg 8

3.1. User Interfaces............................................................................................... pg 8

3.2. Hardware Interfaces....................................................................................... pg 8

3.3. Software Interfaces........................................................................................ pg 8

3.4. Communications Interfaces........................................................................... pg 8

4. Requirements Specification....................................................................................... pg 9

4.1. Functional Requirements............................................................................... pg 9

4.2. External Interface Requirements................................................................... pg 9

4.3. Logical Database Requirements.................................................................... pg 9

4.4. Design Constraints......................................................................................... pg 9

5. Other Nonfunctional Requirements........................................................................... pg 10

5.1. Performance Requirements............................................................................ pg 10

5.2. Safety Requirements...................................................................................... pg 10

5.3. Security Requirements................................................................................... pg 10

5.4. Software Quality Attributes........................................................................... pg 10

5.5. Business Rules............................................................................................... pg 10

6. Legal and Ethical Considerations.….......................................................................... pg 11

Appendix A: Glossary........................................................................................................ pg 12

Appendix B: Analysis Models........................................................................................... pg 13

Appendix C: To Be Determined List................................................................................. pg 14

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason For Changes | Version |
| Brandon Pham, Heriberto Gonzalez, Ricardo Medina, Wilson Weng | 12/11/2020 | Delivery of the Software Requirements Specification | 1 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**1. Introduction**

This software is intended to be implemented in the Autonomous Underwater Vehicle (AUV) under construction by the California State University Senior Design team under the Engineering Department. The goal of the AUV is to compete in an international competition hosted by Robonation to perform tasks in an underwater environment. These include movement, navigation, object detection and recognition, manipulation of objects, target elimination. This document focuses on the mission planning portion of the AUV.

**1.1 Purpose**

The purpose of this document is to explain the mission planning portion of the AUV being constructed by the Senior Design Engineering team to any future users and developers of the product so that they can design and program their own state machines

**1.2 Intended Audience and Reading Suggestions**

The types of readers that the document is intended for is for developers and users that intend to continue the work done by the 2020-2021 Senior Design Team. The reason for this is due to the inability of the entire team to test the AUV within unforeseen circumstances

**1.3 Product Scope**

The software of mission planning takes data obtained from the other portions of the AUV and transfers them around different states allowing it to perform the tasks required of it. The software when released will be used by future users of the AUV to add upon and improve its features

**1.4 Definitions, Acronyms, and Abbreviations**

AUV – Autonomous Underwater Vehicle

ROS – Robot Operating System

SMACH – State Machine

**1.5 References**

ROS - http://wiki.ros.org/

Smach - <http://wiki.ros.org/smach/Documentation>

Ubuntu 18.04.5 - <https://releases.ubuntu.com/18.04.5/>

VMware Workstation - https://my.vmware.com/en/web/vmware/downloads/details?downloadGroup=PLAYER1556&productId=800&rPId=47861

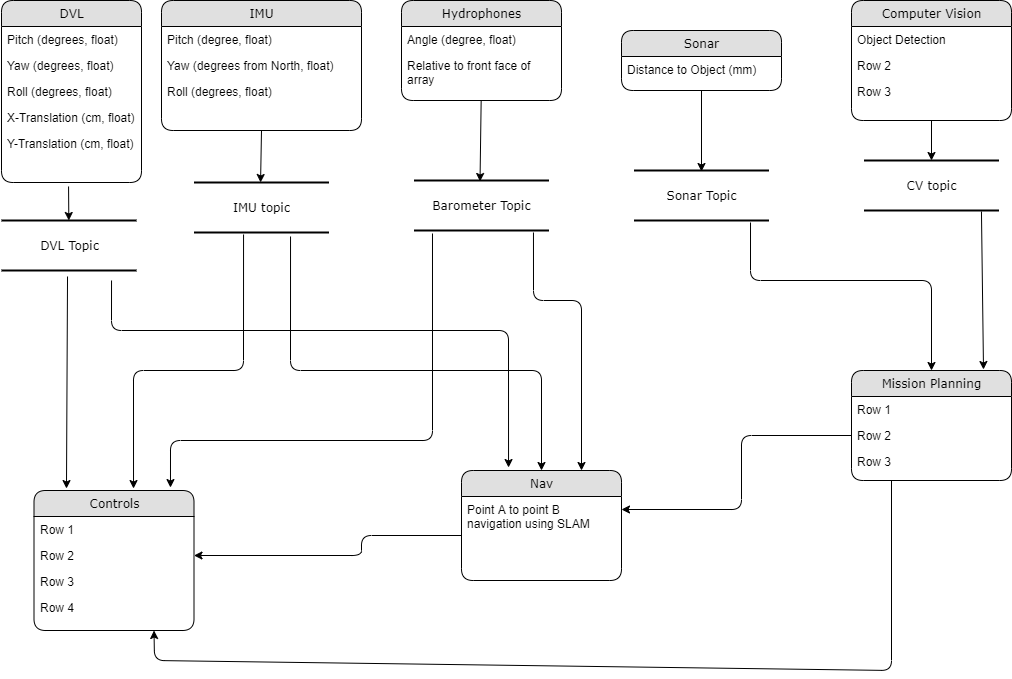
**2. Overall Description**

**2.1 System Analysis**

1. The goal of the project is to develop an AI that controls the functions of the AUV to clear tasks
2. Major technical hurdles involve the difficulty of testing the AUV due to the Covid-19 pandemic
3. Solutions to solve the technical hurdles

**2.2 Product Perspective**

The software in this SRS is a component of the AUV under construction by the Senior Design Engineering team. The AUV is also built for a competition that is typically run every year, so it will have similarities to other AUVs built by competing teams. The motivation for building this product is to see if the finished AUV performs better than other AUVs in a competitive setting.



**2.3 Product Functions**

The functions of the software involve:

* Defining the current state that the AUV is in
* Taking data obtained from other components of the AUV
* Defining when the AUV transitions to another state from its current state
* Passing data between different states

**2.4 User Classes and Characteristics**

The user classes that are anticipated to use this product are software developers that seek to utilize ROS and Smach to program a future AUV and engineers that wish to understand how an AUV functions for testing purposes

**2.5 Operating Environment**

The environment in which the software will live in is Ubuntu 18.04.5 utilizing ROS Melodic and Smach with Python 2.7 programming language.

**2.6 Design and Implementation Constraints**

Constraints of the AUV are:

* Communication between the people designing software to the AUV and the people designing the hardware of the AUV.
* Ability to test the AUV within the restrictions of the 2020-2021 year

**2.7 User Documentation**

In this current version, no additional documentation will be provided for this systems software.

**2.8 Assumptions and Dependencies**

Factors that may affect the requirements in the document are:

* The use of other Ubuntu versions
* Another version of ROS
* A programming language other than Python

**2.9 Apportioning of Requirements**

Delayed requirements involve the inability of the current Senior Design team to test the AUV due to recent lockdowns and quarantines.

**3. External Interface Requirements**

**3.1 User Interfaces**

Mission planning does not require any user interface since the AUV will be running autonomously.

**3.2 Hardware Interfaces**

Mission Planning’s product does not directly interface with any hardware from the AUV, only through interfacing with other systems that directly interact with hardware.

**3.3 Software Interfaces**

Other software products that the AUV will utilize is Python 2.7 with APIs from ROS Melodic and Smach within the Ubuntu 18.04.5 operating system. Users may have to download VM ware Workstation player to use Ubuntu.

**3.4 Communications Interfaces**

The product in terms of mission planning does not have requirements that involve the use of communication functions like a network server or electronic forms. The reason for this is that all the functions that the product will perform is created in a way that allows the AUV to perform without any dependency to the internet. However, these functions are placed in the AUV by use of websites designed to store code integral to the AUV’s function.

**4. Requirements Specification**

**4.1 Functional Requirements**

|  |  |
| --- | --- |
| Requirement No. | Requirement Description |
| 4.1.1 | The system shall transition to different states based on the current task the AUV is faced with. |
| 4.1.2 | The system shall receive data from other systems that is required to perform the current state. |
| 4.1.3 | The system shall send required inputs for other systems to perform their operations. |
| 4.1.4 | The system shall follow a predetermined plan of the competition tasks to perform. |
| 4.1.5 | The system shall consist of a series of state machines for each individual task in one larger overall state machine. |

**4.2 External Interface Requirements**

|  |  |
| --- | --- |
| Requirement No. | Requirement Description |
| 4.2.1 | The system shall handle data using ROS’ publishers and subscribers. |
| 4.2.2 | The system shall interface the AUV’s subsystems together to operate the AUV. |
| 4.2.3 | The system shall output various movement commands to the controls system for the AUV’s mobility. |

**4.3 Logical Database Requirements**

The AUV will be using data for its various systems in real time during the competition, so the Mission planning software currently does not have any database requirements since there is no need for storing data.

**4.4 Design Constraints**

There are no other additional design constraints at the moment that require a technical description that are already stated in section 2.5.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

* The number of terminals that the AUV has is one
* The number of simultaneous users that the AUV supports is two
* The amount of information that is to be handled by mission planning is dependent on the data that is received from other parts of the AUV like computer vision

**5.2 Safety Requirements**

The possible loss and damage that could be a concern with the AUV is that the machine could harm itself damaging significant and expensive equipment. Another possible source of damage is the AUV colliding and harming a person residing within the immediate vicinity of the AUV. Other damages that are noteworthy but unlikely to happen is damage to environmental areas like the wall of a pool.

**5.3 Security Requirements**

There is little to no security or privacy issues with the use of the product considering that anything the AUV is tasked to do is created locally. One concern is if an accident in uploading code to a repository would cause important code to be replaced or erased all together.

**5.4 Software Quality Attributes**

Characteristics that the product that will be important to future users is the adaptability of learning how to develop the product with existing concepts rather than starting over from scratch.

**5.5 Business Rules**

Any individual can perform the functions of the product assuming they know how to use the Python programming language and have familiarity with how ROS and Smach works.

**6. Legal and Ethical Considerations**

There are legal or ethical issues with the project aside from usual common sense when operating any kind of machinery. These involve whether the code for the AUV will be made open to the public, and if it is open source under what license the code will be under.

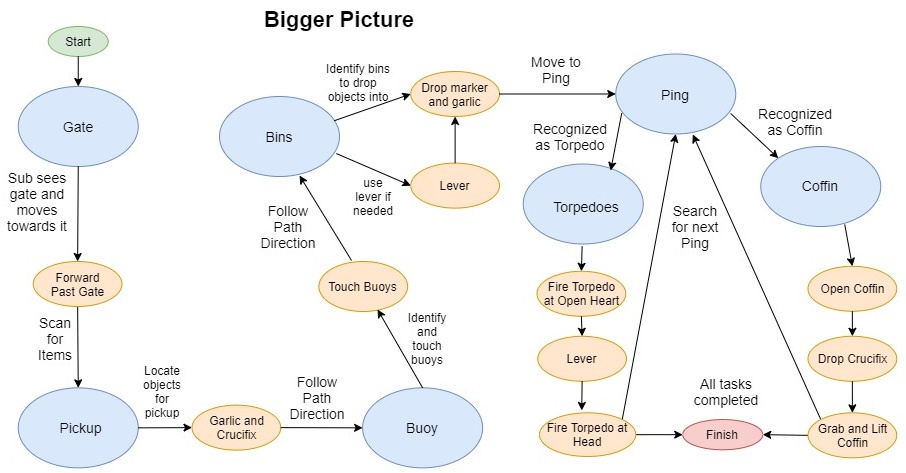
**Appendix A: Glossary**

AUV – Autonomous Underwater Vehicle

ROS – Robot Operating System

SMACH – State Machine

**Appendix B: Analysis Models**



**Appendix C: To Be Determined List**