

# Software Requirements Specifications

Oregon State University  
CS 461  
2017-2018

Prepared By:  
Kyle Prouty  
Hayden Anderson  
Lucien Tamno

October 30, 2017

## CONTENTS

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Purpose . . . . .	3
1.2	Scope . . . . .	3
1.3	Definitions, Acronyms, Abbreviations . . . . .	3
1.4	References . . . . .	3
1.5	Overview . . . . .	3
<b>2</b>	<b>Overall Description</b>	<b>4</b>
2.1	Product Perspective . . . . .	4
2.1.1	System Interfaces . . . . .	4
2.1.2	User Interfaces . . . . .	4
2.1.3	Hardware Interfaces . . . . .	4
2.1.4	Software Interfaces . . . . .	4
2.1.5	Communications Interfaces . . . . .	5
2.1.6	Memory constraints . . . . .	5
2.1.7	Operations . . . . .	5
2.1.8	Site Adaptation Requirements . . . . .	5
2.2	Product Functions . . . . .	5
2.3	User Characteristics . . . . .	6
2.4	Constraints . . . . .	6
2.5	Assumptions and Dependencies . . . . .	6
2.6	Apportioning of Requirements . . . . .	6
<b>3</b>	<b>Specific Requirements</b>	<b>7</b>
3.1	External Interfaces . . . . .	7
3.2	Functions . . . . .	7

		2
3.3	Performance Requirements . . . . .	7
3.4	Logical Database Requirements . . . . .	7
3.5	Design Constraints . . . . .	8
3.6	Standards Compliance . . . . .	8
<b>4</b>	<b>Software System Attributes</b>	<b>8</b>
4.1	Reliability . . . . .	8
4.2	Security . . . . .	8
4.3	Maintainability . . . . .	8
4.4	Portability . . . . .	8
4.5	Other Requirements . . . . .	8

# 1 INTRODUCTION

## 1.1 Purpose

The purpose of this document is to layout and define the objective and details of the framework to be developed. This document will specify an overall description, specific requirements, and software system attributes of this framework. The intended audience for this document includes any relevant stakeholders.

## 1.2 Scope

This main purpose of this framework will be to allow a user to input an objective into a collection of nodes, see the nodes organize on that object, and then see the result of that organization. Nodes will carryout an objective based on the services available. Three main goals will be achieved in order to complete the main objective. First, the framework will enable nodes to communicate with each other and have the ability to share processes. Next an objective will be able to be inputted into the framework. Lastly the nodes will self organize on the objective and output the results to the user.

## 1.3 Definitions, Acronyms, Abbreviations

Term	Definition
Stakeholder	User of the framework
GUI	Graphical User Interface
User	The person who would use this at its completion that doesnt know how it works
IoT	Internet of Things
Node	Our built device that acts as a layer between IoT enabled machines and other nodes
Social Device	An IoT enabled device that can connect and (talk) with other IoT devices.
MVC	Model View Controller; A software architecture for implementing an interface for the user
Docker	A container of isolated code
Unix	An open-source operating system
LAN	Local area network
MQTT	Message Queue Telemetry Transport: lightweight TCP/IP protocol
ARM architecture	Advanced RISC Machine(ARM): computer processor architecture
RASPBIAN	Debian based operating system based on the Unix operating system

## 1.4 References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society,1998.

## 1.5 Overview

In the next section of this document, Overall Description, will give an overview of software dependencies, the intended function, characteristics, constraints, and assumptions of the framework. This section is geared more towards giving

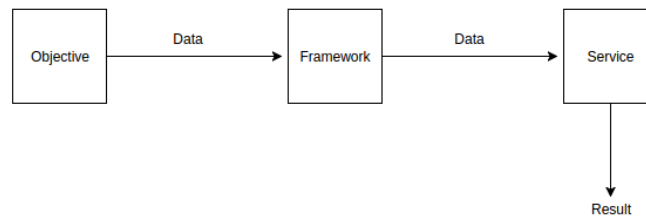
a more high level examination of the framework. The third and fourth sections, Specific Requirements and Software System Attributes, is intended for more of a technical audience who would be familiar with the technical aspects and relevant terminology of this framework.

## 2 OVERALL DESCRIPTION

### 2.1 Product Perspective

This framework is designed to offer services that will interact with outside systems. It will act as a middle layer between a user and the outside system. An example of an outside system is a printer.

Fig. 1. This figure shows an objective being inputted into the system, the framework receiving the objective and then organizing on that objective. Next the data is sent to the proper service, and finally the result.



#### 2.1.1 System Interfaces

The primary system interface will be a Unix based operating system. This framework will interact with the Unix operating system in order to utilize its services.

#### 2.1.2 User Interfaces

- A command line based interface used to connect to a specific node to change configuration setting.
- A web based gui that will interface between an administrator and the nodes. This graphical interface is used to show the nodes in the framework, the services each node offers, and the nodes connections with each other.
- A web based gui that will interface between a user and the nodes. This graphical interface is used to let a user upload a file into the collection of nodes and view the status of the result.

#### 2.1.3 Hardware Interfaces

This framework will use a combination of various ARM based micro-controllers. These micro-controllers must have the functionality to communicate with 802.11 b/g/n wireless LAN communication protocols.

#### 2.1.4 Software Interfaces

Nodes will only interact with the Unix based operating system. Users will interact with a web based browser.

### 2.1.5 *Communications Interfaces*

The network protocol that is required by this framework is Ethernet, with communication protocols TCP/IP and 802.11 b/g/n wireless LAN.

### 2.1.6 *Memory constraints*

The framework is only constrained to what the hardware that is used will allow.

### 2.1.7 *Operations*

There will be three modes of operation within the user organization.

- Node Operation Mode
- User Mode
- Administrator Mode

The main mode of operation is the node operation mode. This mode will operate exclusively in the background. Next is the user mode which will be input and output operations. Third there will be an administer mode which will allow the sending of commands and the viewing of state and status operations.

### 2.1.8 *Site Adaptation Requirements*

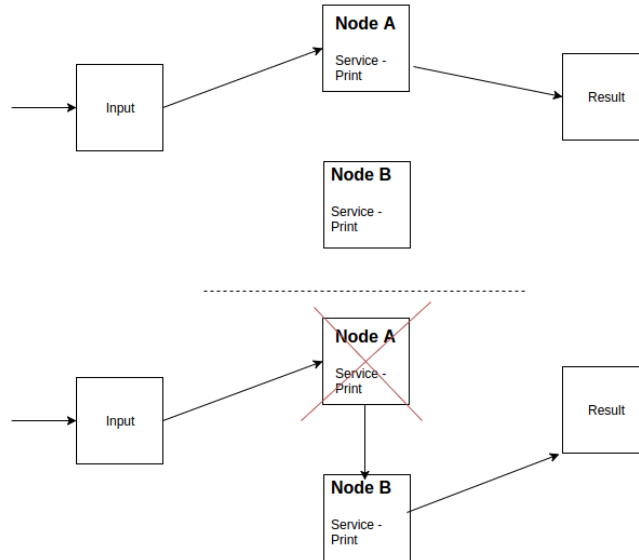
No site adaptations required.

## 2.2 **Product Functions**

- A user interface to send an objective to the framework.
- A user interface to visual see the results of sending an objective to the framework.
- An administrator interface to send commands to the framework.
- An administrator interface that will show available services offered by each node.
- An administrator interface that will show the state of the framework.
- A command that will block services on a specific node.
- A command that will shutdown services.
- A node will get information about other nodes on the network
- A node will offer a print service.
- A node will share processes with other nodes.
- A node will keep a memory of tasks sent from user
- A node will be able to share tasks will other nodes

The figure below shows an objective being inputted into the framework and then a result. Next it shows the same objective going to the same node in the framework but this time the nodes have to self organize and reroute the objective to a different node which achieves the same result.

Fig. 2. Basic description of how this framework will function.



### 2.3 User Characteristics

The users of this framework will be our relevant stakeholders. These users will have had some level of formal technical education.

### 2.4 Constraints

This framework will not communicate to outside sources for any of its configuration settings. Each individual node must be able to lead the framework on decisions required. Nodes cannot utilize an external database for its inner information structure. Nodes cannot exchange information by using an external database.

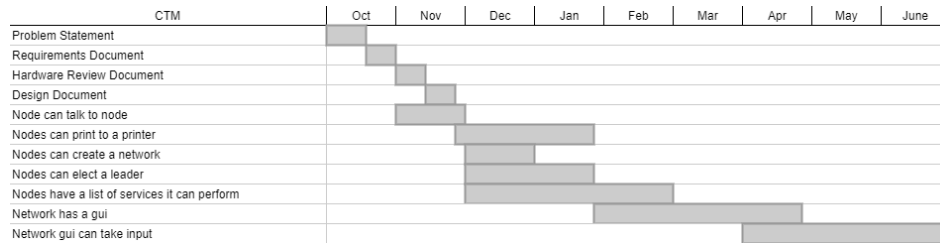
### 2.5 Assumptions and Dependencies

This document and current requirements assume that each hardware device we use can handle concurrency and has wifi capabilities. Each device has the memory and storage large enough to fit our docker image and run. Each physical device is able to handle a small simple gui.

### 2.6 Apportioning of Requirements

Refer to Gantt chart in figure below for this information.

Fig. 3. Gantt Chart



### 3 SPECIFIC REQUIREMENTS

#### 3.1 External Interfaces

- Input:
  - User objectives
  - Administrator commands
  - Node operation configuration commands
- Output:
  - Results of inputted objectives
  - Services
  - State and configuration data

#### 3.2 Functions

The system shall ...

- Enable nodes to self organize on an inputted objective
- Allow an objective to be inputted
- Output readable configuration data
- Output readable service and state data
- Automatically validate and connect to authorized nodes
- Allow nodes to share tasks and processes with each other
- Allow services to be disabled
- Allow services to be shutdown
- Offer a print service

#### 3.3 Performance Requirements

This framework will support at a minimum two nodes. Each objective received by the framework from the user will start and execute on 90% of the services required by the objective given. Only one user will be able to input an objective at a time. Only one objective can be handled by the network at a time. Objectives can be queued and executed consecutively. Each node will support, at a minimum, a print service. A user will have to wait for an objective to complete before inputting a new objective.

#### 3.4 Logical Database Requirements

There are no database requirements.



### **3.5 Design Constraints**

There is not an external database to which the nodes should be writing to or retrieving from to be able to be a part of the nodal network or carry out objectives.

### **3.6 Standards Compliance**

This framework will need to stay compliant with IEEE WLAN and MQTT standards.

## **4 SOFTWARE SYSTEM ATTRIBUTES**

### **4.1 Reliability**

The reliability of this framework will be measured by the ability to connect, send commands, and view the results.

### **4.2 Security**

The framework will only accept authorized nodes. Nodes will be password protected and will only accept configuration settings and commands from verified users.

### **4.3 Maintainability**

Code will be well commented and readable to 8 out of 10 software developers. A MVC design scheme will be implemented. File structure will be well organized and self explanatory to navigate.

### **4.4 Portability**

This framework will be fully portable to any system that can run a Docker container.

### **4.5 Other Requirements**

No additional requirements for this framework.