System Requirements Specification

for

Modeling Unmanned Aerial Swarms Using Unreal Engine and AirSim Simulator

Version 2.1.0 approved

Prepared by Dillon Mead

Embry-Riddle Aeronautical University EECS Department

04 FEB 2022

Table of Contents

Table of Contents ... 1
Revision History ... 1
1. Introduction ... 2
1.1 Purpose ... 2
1.2 Document Conventions ... 2
1.3 Intended Audience and Reading Suggestions ... 2
1.4 Product Scope ... 2
1.5 References ... 2
2. Overall Description ... 3
2.1 Product Perspective ... 3
2.2 Product Functions ... 3
2.3 User Classes and Characteristics ... 3
2.4 Operating Environment ... 3
2.5 Design and Implementation Constraints ... 3
2.6 User Documentation ... 4
2.7 Assumptions and Dependencies ... 4
3. External Interface Requirements ... 4
3.1 User Interfaces ... 4
3.2 Hardware Interfaces ... 4
3.3 Software Interfaces ... 4
3.4 Communications Interfaces ... 5
4. System Features ... 5
4.1 System Feature 1 ... **Error! Bookmark not defined.**
4.2 System Feature 2 (and so on) ... **Error! Bookmark not defined.**
5. Other Nonfunctional Requirements ... 6
5.1 Performance Requirements ... 7
5.2 Safety Requirements ... 7
5.3 Security Requirements ... 7
5.4 Software Quality Attributes ... 7
5.5 Business Rules ... 7
6. Other Requirements ... 7
Appendix A: Glossary ... 8
Appendix B: Analysis Models ... 8

Revision History

Name	Date	Reason For Changes	Version
Dillon Mead	13/09/21	Initial document creation.	1.0.0
Dillon Mead	27/09/21	First Semester first version edit.	1.1.0
Dillon Mead	26/10/21	Second version edit.	1.2.0
John Mueller	26/10/21	Added Stimulus/Response in section 4	1.2.1
Dillon Mead	29/11/21	Third version edit.	1.3.0
Dillon Mead	30/11/21	Edits sections 2 and 4. Adds highlight convention.	1.3.0

Dillon Mead	04/02/22	Second Semester first version.	2.1.0

1. Introduction

1.1 Purpose

The purpose of this SRS is to define the requirements for a simulated unmanned aerial swarm in Microsoft's AirSim simulator. This SRS encompasses all requirements for the simulated unmanned aerial swarm system.

1.2 Document Conventions

Text highlighted in yellow are portions of the product not currently developed. These features will be explored during future development. Features completed during the 2nd Semester will be identified by this label and by separate paragraphs.

1.3 Intended Audience and Reading Suggestions

This document is intended for developers, project managers, testers, and document writers.

1.4 Product Scope

This system simulates an unmanned aerial swarm. The aerial swarm will behave in a unified, cohesive manner. The general mission for the aerial swarm will be to collect data from a three-dimensional environment.
Aerial Swarm autonomy will be explored during future work.

2nd Semester:
This system will employ a collision avoidance algorithm that integrates with the aerial swarm pathing method and includes object detection capabilities. The collision avoidance will activate for both stationary and moving objects.

1.5 References

[1] Microsoft Corporation. "AirSim." Version 1.6.0. August 2021. https://github.com/Microsoft/AirSim.
[2] Microsoft Corporation. "AirSim FAQ." https://microsoft.github.io/AirSim/faq/#what-computer-do-you-need.
[3] Epic Games, Inc. "Unreal Engine." Version 4.27. August 2021. https://www.unrealengine.com/en-US/.
[4] Aerial Swarm Simulators. "Modeling Unmanned Aerial Swarms Using Unreal Engine and AirSim Simulator System Design Document." Version 3.0. November 30, 2021. https://github.com/mead-d/Modeling-Unmanned-Aerial-Swarms-Using-Unreal-Game-Engine-and-AirSim-Simulator/tree/main/Deliverables.

2. Overall Description

2.1 Product Perspective

This system is a new product inspired by earlier research from Professor Ilhan Akbas. The original idea was to use aerial wireless sensor and actor networks (aerial swarm) to perform observation tasks while mitigating risk to the user. For example, the user could be a geologist and the observation task is the smoke plume from a volcano. It is easier to gather data with an aerial swarm and minimize potential injury or loss of life.

2.2 Product Functions

- Multiple Unmanned Aerial Vehicles operate as unified, cohesive swarm.
- Each UAV maintains communication with the swarm.
- The aerial swarm maintains communication with the ground station.
- The aerial swarm can measure the volume of a given object.

2.3 User Classes and Characteristics

There is only one user class for this system. The user must have knowledge of computers and programming. The user must know how to operate in the AirSim environment. This product is developed with python 3.5 – 3.8, therefore the user must also need to understand the python language and how to operate in the python environment. Developers and Testers have the same user rights and capabilities.

2.4 Operating Environment

This system must use Microsoft's AirSim Simulator [1]. Microsoft's AirSim may be used on any computer using either Windows, MacOS, or Linux. There is no explicit hardware requirement when using the basic "blocks" environment [2]. A minimum of 4GB VRAM (GPU RAM) is required when using a more detailed environment. The current minimum software necessary for Microsoft's AirSim is Visual Studio 2019 and Unreal Engine 4.26.

2.5 Design and Implementation Constraints

The supporting software, Unreal Engine 4.27 [3], Microsoft's AirSim, Visual Studio 2019 (VS19), and additional tools or extensions must be installed and built in a specific manner. Microsoft provides a list of instructions for installation and building at the following location: https://microsoft.github.io/AirSim/build_windows/. Other difficulties may occur during installation that are not within Microsoft's directions. Chiefly, administrative rights are necessary throughout the installation and build process on the installation computer. C++ tools are required in VS19 and it is important to include the python tools when necessary. This system uses those python libraries available through AirSim to implement the product.

2.6 User Documentation

The most recent version of the System Design Document, System Requirements Specification, and Test Plan may be found within the Aerial Swarm Simulator team's GitHub repository within the deliverables folder: https://github.com/mead-d/Modeling-Unmanned-Aerial-Swarms-Using-Unreal-Game-Engine-and-AirSim-Simulator/tree/main/Deliverables.
The only exception is the AirSim Simulator documentation that may be found at: https://microsoft.github.io/AirSim/.

2.7 Assumptions and Dependencies

Assumptions:
We assume that Unreal Engine 4 and Microsoft's AirSim Simulator will remain open source.
The aerial swarm must have at least three UAV to properly conduct a given mission.
The operating environment is assumed to be clear of an overwhelming amount of obstacles.
There are no adversarial actors searching for the aerial swarm.

Dependencies:
This system is currently only usable within Microsoft's AirSim environment. AirSim is itself dependent on Unreal Engine 4 and Visual Studio 2019. Any changes to Unreal Engine 4, Visual Studio 2019, or Microsoft's AirSim have the potential to cause errors and irregularities within the Aerial Swarm Simulator system.

3. External Interface Requirements

3.1 User Interfaces

[Req 1] The Aerial Swarm Simulator system shall be modelled and simulated in Microsoft's AirSim Simulator.

[Req 2] Visual Studio 2019 shall be used to edit files and environment variables.

[Req 3] The User shall implement mission scenarios by executing the appropriate script through Visual Studio 2019 within the AirSim environment.

[Req 4] System data shall be display to the user through the python environment terminal.

3.2 Hardware Interfaces

This system has no hardware interface requirements.

3.3 Software Interfaces

Microsoft's AirSim Simulator interfaces with Unreal Engine 4 by declaring which graphical environment to generate. The Unreal Engine 4 physics system is also utilized for AirSim object attributes and interactions. AirSim also uses OpenCV for object detection/ image recognition in the Unreal Engine 4 environment.

3.4 Communications Interfaces

The system does not currently use any communications interfaces.

4. System Features

4.1 Ground Station/ Control

4.1.1 Description and Priority

Ground Station is the user/ computer that is used. Ground Station is what transmits the mission task input and what receives the sensor data and swarm status.

4.1.2 Stimulus/Response Sequences

Action: User issues mission task for aerial swarm.
Response: Aerial swarm begins mission and reports status of aerial swarm.
Action: Aerial swarm transmits sensor data.
Response: Ground station records sensor data.

4.1.3 Functional Requirements

[Req 5] Ground station shall have the ability to assign missions to the aerial swarm.
[Req 6] Ground station shall receive reports on the aerial swarm status including status of all individual UAV.
[Req 7] Ground station shall receive sensor data from aerial swarm.
[Req 8] Ground station shall record sensor data.
[Req 9] Ground station shall display aerial swarm status including status of all individual UAV.

4.2 UAS Swarm

4.2.1 Description and Priority

The aerial swarm is a unified, cohesive grouping of multiple individual UAV. There is a designated lead UAV that leads the swarm and organizes the swarm while operating, collects all the individual UAV statuses, and reports the overall swarm status and individual UAV statuses to the ground station.

4.2.2 Stimulus/Response Sequences

Action: User spawns in an aerial swarm.
Response: First UAV spawned is designated as the lead UAV.
Action: User assigns mission task to aerial swarm through script.
Response: Lead UAV organizes and coordinates with other UAVs to complete mission.
Action: Object identified for analysis.
Response: Aerial swarm maneuvers to object. Aerial swarm measures physical dimensions of object and calculates the volume.

4.2.3 Functional Requirements

[Req 10] The aerial swarm shall designate a lead UAV for swarm organization and communication.
[Req 11] The aerial swarm shall reassign the lead UAV when the current lead becomes non-functional.
[Req 12] The aerial swarm shall have at least three UAV for any given mission.
[Req 13] The lead UAV shall receive status data from all individual UAV every 0.1 seconds.
[Req 14] The lead UAV shall transmit status data of the aerial swarm and all individual UAV.

[Req 15] The aerial swarm shall measure the volume of an identified object.
[Req 16] The aerial swarm shall transmit sensor data to a repository in the ground station.
[Req 17] The aerial swarm shall determine the positioning of individual UAV and transmit the data to individual UAV.
[Req 18] The aerial swarm shall adjust and continue the mission task when any individual UAV becomes inactive.
[Req 19] The aerial swarm shall return to "home" location when mission task is complete.

4.3 Individual UAV

4.3.1 Description and Priority

An atomic vehicle of the aerial swarm. Individual UAV will have a position and status that is communicated with the swarm. Each UAV will also have some payload that includes sensors such as a camera.

4.3.2 Stimulus/Response Sequences

Action: Lead UAV receives mission task.
Response: Individual UAV follow lead UAV along path.
Action: Swarm formation changed based on task.
Response: Lead UAV sends position coordinates to individual UAV for specified formation.
Action: Status of UAV requested.
Response: UAV sends status to lead UAV.

4.3.3 Functional Requirements

[Req 20] Individual UAV shall communicate position with the aerial swarm using North, East, and Down coordinates.
[Req 21] Individual UAV shall communicate active status with the aerial swarm.
[Req 22] Individual UAV shall avoid collisions with objects including other UAV.
[Req 23] Individual UAV shall carry a payload that will house sensors.
[Req 24] Sensor data shall be routed through the aerial swarm via the lead UAV.

4.4 Collision Avoidance (2nd Semester)

4.3.1 Description and Priority

High priority. The aerial swarm, or even a single UAV, must be able to travel within an environment without colliding with another object.

4.3.2 Stimulus/Response Sequences

Action: L
Response: I

4.3.3 Functional Requirements

[Req 25] A UAV shall avoid collisions with all other objects while moving.
[Req 26] The aerial swarm shall avoid collisions with all other objects.
[Req 27] The aerial swarm shall have the ability to sense objects in the environment.
[Req 28] The Collision Avoidance system shall continuously execute.
[Req 29] The Collision Avoidance system shall identify an impending collision.
[Req 30] The Collision Avoidance system shall implement the avoidance algorithm when an impending collision is detected.
[Req 31] The Collision Avoidance system shall not implement the avoidance algorithm if an impending collision is not detected.

[Req 32] The UAV or aerial swarm shall continue on the original mission path after successfully avoiding a collision.
[Req 33] The avoidance algorithm shall choose an avoidance path without additional impending collisions.
[Req 34] The Collision Avoidance system shall record each impending collision.
[Req 35] The Collision Avoidance system shall record every execution of the avoidance algorithm.
[Req 36] The Collision Avoidance system shall record any collision between the UAV and another object.

5. Other Requirements

5.1 Performance Requirements

[Req 37] The aerial swarm shall continue task and attempt to finish the mission queue when any individual UAV becomes inactive.
[Req 38] The aerial swarm shall report to ground control when a UAV becomes inactive.

5.2 Safety Requirements

We do not have any safety requirements.

5.3 Security Requirements

We do not have any security requirements.

5.4 Software Quality Attributes

Adaptability: Modularity facilitates high cohesion and low coupling attributes for our system.
Availability: Currently our system is located in the private Github. All foundation software is open source.
Flexibility: Aerial swarm may be able to continue mission with multiple UAV for given tasks. The aerial swarm must successfully avoid collisions with other objects.
Maintainability: Code commenting, and naming conventions allow for readability. System modules are modular classes/ functions.
Reusability: Each modular class or file may be reused separately.
Testability: Unreal Engine 4.27, Microsoft's AirSim 1.6.0, Visual Studio 2019, and additional tools are necessary for executing our system. Modular classes and functions allow for testing small portions of the system individually.

5.5 Business Rules

There are no business rules for this system.

6. Other Requirements

There are no other requirements.

Appendix A: Glossary

Active: UAV is operational and in flight.
Inactive: UAV is not operational or is not in flight.
UAV: Unmanned Aerial Vehicle.
UAS: Unmanned Aerial System – "aerial swarm".

Appendix B: Analysis Models

Analysis Models may be found in the System Design Document [4].