PROJECT SKYNET

RAVEN NEST

CENG 490 Term Project

Mehmet Akif Akpınar
Kadir Ekmekçi
Rana Aygül
Mustafa Buğra Tamer

30.11.2014

Middle East Technical University Computer Engineering Department

Table of Contents

Table of Figures

1.INTRODUCTION

1.1.Problem Definition

With the great advancement of robotic field in recent years, usages and applications of UAVs (Unmanned Aerial Vehicles) and UGVs (Unmanned Ground Vehicles) is also increased with great momentum. Autonomous Robots are not only used in academic or military applications but also used in commercial usage as a hobby device or a delivery device.

Surveillance and aerial mapping are examples for the major fields which autonomous robots are used commonly. These robots -especially UAVs- are frequently used in surveillance and aerial mapping missions which are part of some search and rescue missions, military operations or homeland security operations. However since UAVs' aerial surveillance is much more effective than UGVs' ground surveillance, UAVs are mostly preferred for surveillance, imaging or mapping missions. Although small-scale remote controlled UAVs (Quadrocopters etc.) had many different advantages for these missions, their limited batteries and limited flight duration which is result of these limited batteries are the main handicaps for small-scale UAVs. These disadvantages not only causing inefficiency on missions that mentioned before but also becoming one of the main setbacks for increasing usage of UAVs.

Although there are studies about symbiotic multi-robot systems, number of these academic studies are very few and usage of these systems in the industry is much lower than that. With this project it is aimed that creating both theoretical and practical study for this new research area of robotic and Artificial Intelligence fields.

This project targets that bringing solutions to problems which caused by limited charge capacities and flight durations of small-scale UAVs or limited viewpoints of UGVs by solving them with symbiotic system in three main scenarios which they are also have some sub-scenarios.

1.2.Purpose

Software Requirement Specification document aims to specify the requirements for Raven Nest project and to give information about capabilities of the resulting application and results of the research which will be carried out during implementation of the project in details. This document will explain the scenarios of the desired project and necessary steps and describe optimized solution methods for given scenarios. It will also describe functionality of the system, system constraints, system interfaces and specific requirements.

The preparation of this SRS will specify all the requirements that will be necessary in next stages which are design, coding and testing.

1.3. Scope

Project RAVEN NEST is a project which will create a symbiotic multi-robot system for minimizing main disadvantages of UGVs and UAVs on mentioned missions and use this system in different scenarios. With this project, it is aimed that bringing efficient solutions to the problems which is caused by UGVs and UAVs' limited properties.

The project that will be implemented and that is specified in this document is Raven Nest. The goal of this project is to make contribution on research fields that include symbiotic multi-robot systems which work with Artificial Intelligence and Computer Vision Softwares, to emphasize the importance of these research areas for real life and to show that more interest is required to go further in this field of the science. We have two robots that will operate the system and one server that will be used in simulation part. The first robot is UAV (Unmanned Aerial Vehicle) which is AR Drone 2.0. The functionality of this robot is to get specified maze image fragments in certain time intervals and send them to ground vehicle and server. However, it can range to obtain data according to maze size, the hover height and external influences. The second robot is UGV (Unmanned Ground Vehicle) which is P3DX. The functionality of this robot is to solve data sets obtained by UAV with the help of Artificial Intelligence and Computer Vision algorithms and to move on the maze by following the most optimal path in accordance with its aim. The functionality of the server that will be used in this system is to get data set which contains obtained maze image fragments' result from UAV and to help simulating the operations of both UAV and UGV , the state of maze that UGV will move on and that shows the possible path.

When 'start' command is given to the system, UAV will take off from the top of UGV, hover above the maze to get the specific-brand maze image fragments and send obtained fragments' result
set in a certain time intervals. After all maze image fragments are sent to UGV and server, AI and CV algorithms will run for this data set, and optimal paths will be drawn for UGV to move on the maze and all those operations will be monitored on the separate screen with at most 1 second latency.

1.4. Definitions, Acronyms, and Abbreviations

UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
SRS	Software Requirements Specification
Quadrocopter	A multi-rotor helicopter that is lifted and propelled by four rotors
AR Drone	A radio controlled flying quadrocopter helicopter.
P3DX	Popular research mobile robot as unmanned ground vehicle
ROS	Robot Operating System
OpenCV	A library of programming functions mainly aimed at real-time computer vision
C++	A general purpose programming language
IEEE	Institute of Electrical and Electronics Engineering
State Diagram	A type of diagram used in computer science and related fields to describe the behavior of systems
Agile methodology	A group of software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams
Gazebo	Robot simulation tool

1.5 References

[1] IEEE Recommended Practice for Software Requirements Specifications :
IEEE Std 830-1998

[2] Guidelines for Preparing Software Requirements Specifications
https://cow.ceng.metu.edu.tr/Courses/index.php?course=ceng491&semester=20141

[2] http://wiki.ros.org

[3] http://wiki.ros.org/indigo

[4] http://wiki.ros.org/gazebo

[5] http://wiki.ros.org/hector_quadrotor

[6] http://wiki.ros.org/tum_ardrone

[7] http://ardrone2.parrot.com

[8]http://opencv.org

[10] http://www.mobilerobots.com/ResearchRobots/PioneerP3DX.aspx

1.6. Overview

This document contains a detailed description about the project Raven Nest. In the introduction part of this document, it mostly gives a general overview about the project including the definition of a real world problem that project intends to solve and the scope of this project that defines what the project is and that what the main process of the project is. Also in this part, the purpose of the SRS and the scope of the project are clearly explained. Second part of the document is the overall description of the project. In this part, the perspectives of the application, the functions included in the application and the constraints, assumptions and dependencies of the desired application are explained. The specific requirements of the project are identified in the next part of the document. This part includes interface, functional and nonfunctional requirements of the project. In the fourth part, data models and their descriptions are explained in detailed way with the relationships between them. The behavioral model and its descriptions are lastly included in the fifth part. In the seventh part, team structure and introduction of the team members of the project are explained. The final part is conclusion part. A brief summary of the whole document will be given in that part.

2. OVERALL DESCRIPTION

This section will give information about product perspective, product functions, constraints, assumptions and dependencies.

2.1. Product Perspective

Our project is going to provide three facilities. The first one is to get related maze image parts with the help of UAV. UAV will take off from top of UGV and start to get parts of maze images in a certain time intervals. In every this time intervals, UAV will send the obtained image parts to UGV and continue to get remained maze image parts until all maze images are revealed. The second facility is to collect all obtained image parts from UAV, to integrate those image parts and to process integrated image. The third facility is real time simulation of UAV and UGV movement behaviors, which means that when UGV and UAV work synchronously; their behaviors are simulated in the separate screen to be able to monitor their interaction and instantaneous movement.

2.1.1. System Interfaces

Solving Maze Interface has only two inputs from User. One of the inputs is from starting the mission and the other one is for choosing the scenario. In this project there are three subsystems which interacts each other which are Server, UAV and UGV. Server sends start and configuration command to UAV. UAV system will send the extracted data from the images taken to both Server and UGV. Server will use this information for creating real time simulation and UGV will use it to create the path and it moves on this path for finishing the mission.

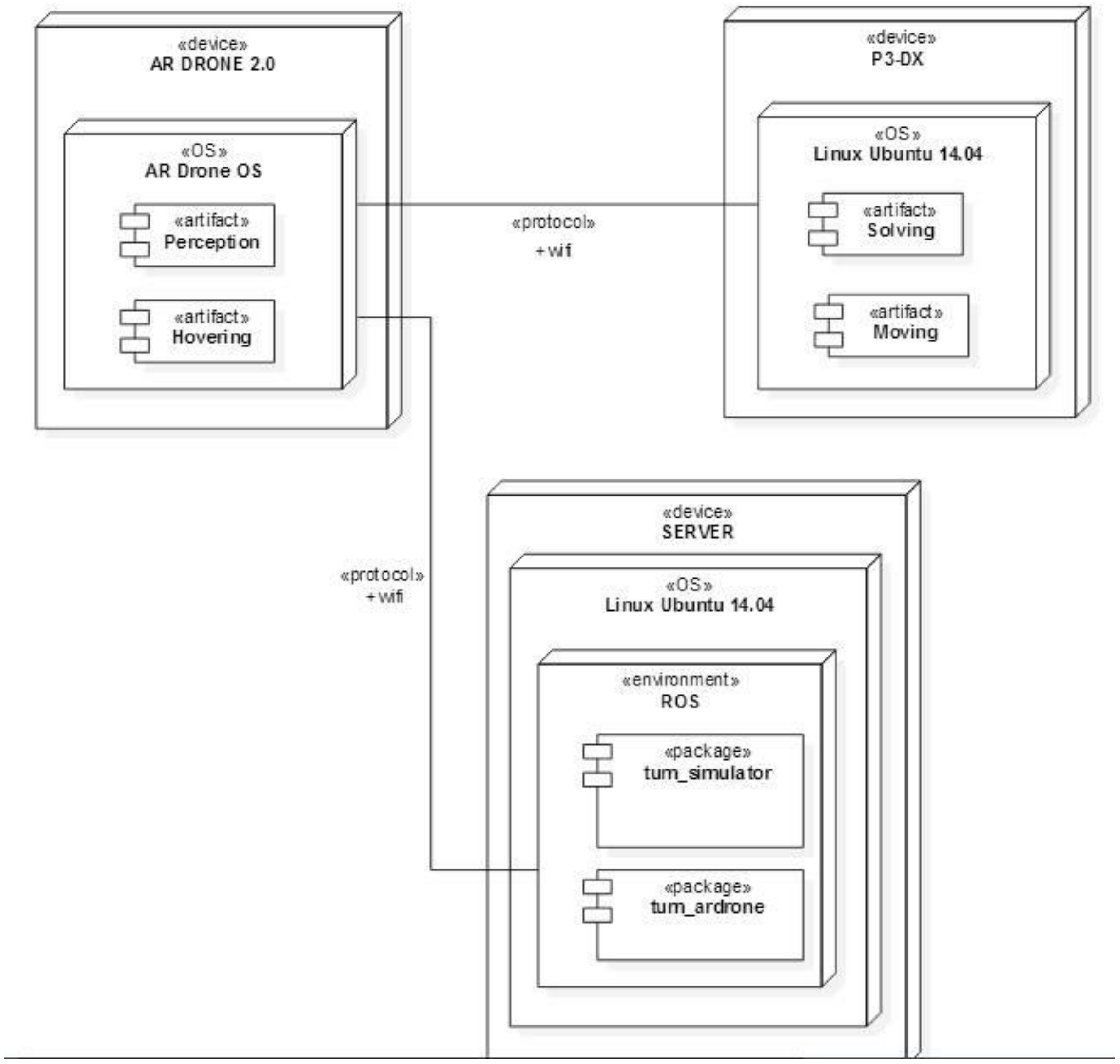

Figure 1 Deployment Diagram of Project

2.1.2. User Interfaces

The system is not going to interact with humans directly when it is working, sending commands will be done in the terminal of Server. However the system provides a real time simulation which is running on the Gazebo for monitoring the robot movements.

2.1.3. Hardware Interfaces

The system is planned to work on the AR Drone 2.0, P3-DX and a Server Computer which runs a Linux. Both AR Drone 2.0 and P3-Dx will be provided by Selim Temizer. AR Drone 2.0 is going to take images and sends to P3-DX which is going to use for creating a path to follow. UAV will also use P3-DX for taking off and landing on.

2.1.4 Software Interfaces

2.1.4.1. ROS Indigo

The Robot Operating System (ROS) is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms.

ROS Indigo will be used for main tool of our project. It will help for sending commands and running algorithms on the robots.

2.14.2. tum_simulator

tum_simulator is a ROS package contains the implementation of a gazebo simulator for the AR drone 2.0 and has been written by Hongrong Huang and Juergen Sturm of the Computer Vision Group at the Technical University of Munich.

tum_simulator will be used in this project for creating our real time simulation and sending data to AR Drone API.

2.1.4.3. tum_ardrone

tum_ardrone is a ROS package contains the implementation of a system that enables a low-cost quadrocopter coupled with a ground-based laptop to navigate autonomously in previously unknown and GPS- denied environments.

tum_ardrone will be used in this project for creating our path and mission planning algorithms and sending data to AR Drone API.

2.1.4.4. AR Drone API

AR Drone API is the reference project for developing augmented reality games based on the AR Drone. And it will be used for sending data to directly AR Drone.

2.1.5. Communication Interface

UGV, UAV and Server components of the project communicates over Wi-Fi.

2.1.6. Memory

There are not any required limitations for memory. Memory capabilities of any system would be enough to run this software.

2.1.7. Operations

There are only two user initiated operation which are sending start command and scenario selection. Sending start command operation starts the mission and scenario selection determines which scenario will be started. Other basic operation which user can do is the monitoring mission via real-time simulation. This part requires no interaction with the user. The details are also explained in the following sections.

2.1.8. Site Adaptation Requirements

There are three scenarios in this project and each one of them needs specific environmental requirements to accomplish. In first scenario which is solving a fixed size maze, size of the maze and the coordinates of the UGV must be pre-known. In the second and third scenarios which are solving an unknown size maze and mapping a region, there is no site adaptation requirement.

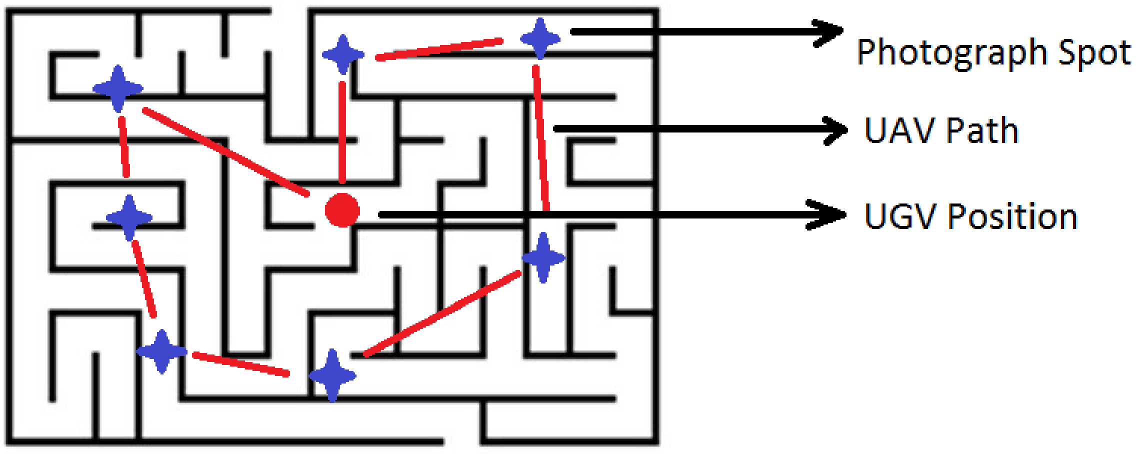

Figure 2 Fixed Maze Figure

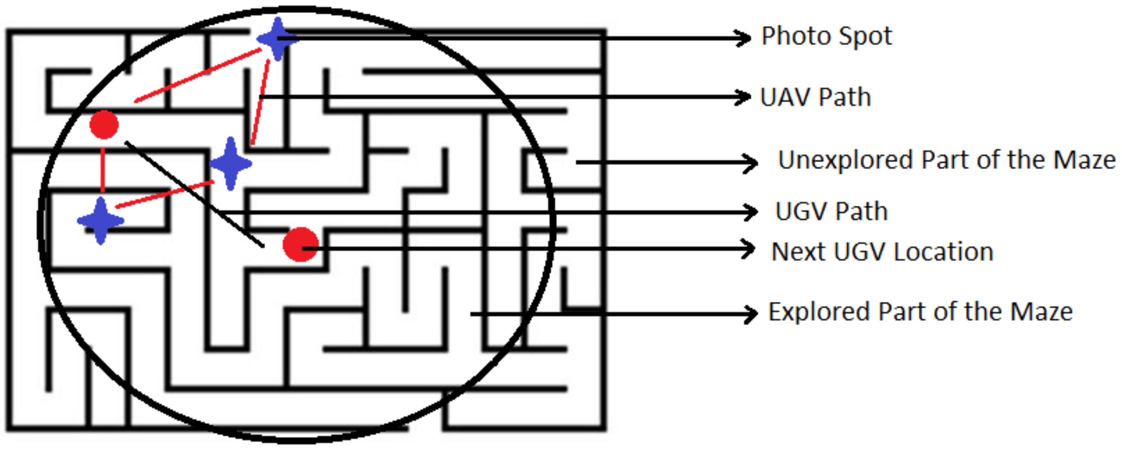

Figure 3 Unknown Size Maze Figure

2.2. Product Functions

2.2.1. Scenario 1: Solving a Fixed Size Maze

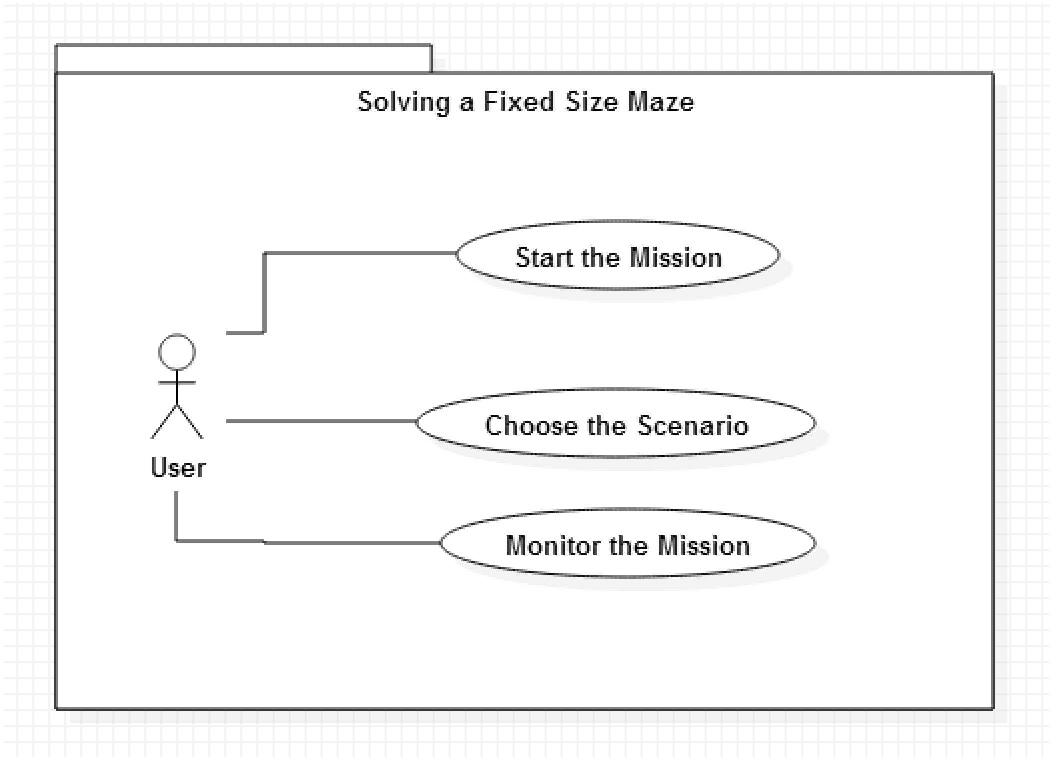

Figure 4 Use Case Diagram of Fixed Size Maze

- User should be able to start the mission
- User should be able to choose the scenario
- User should be able to monitor the mission via the Real-Time Simulation screen.
- UAV will be on the UGV and ready to take off in the initial condition.
- UAV will hover on the maze and take photos of it from certain locations for sending them to UGV.
- After sending images to UGV, UAV will land on to UGV.
- UAV will also send these images to server for simulating the mission.
- UGV will process the raw images and it will construct the maze from these images for solving it.

- UGV will create a path throughout the maze by solving the virtually created maze and will follow this path for exiting from the maze.
- Server will process the raw images of the maze for creating a simulation of the mission.

2.2.2. Scenario 2: Exploring an Unknown Size Maze

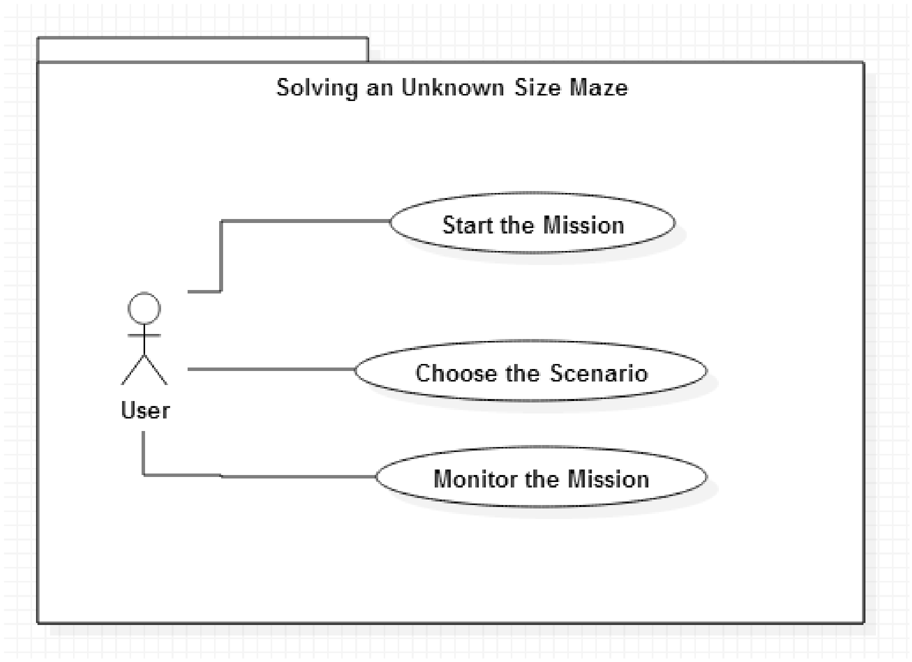

Figure 5 Use Case Diagram of Unknown Size Maze

- User should be able to start the mission
- User should be able to choose the scenario
- User should be able to monitor the mission via the Real-Time Simulation screen.
- UAV will be on the UGV and ready to take off in the initial condition.
- UAV will hover on the maze and take photos of it from certain locations for sending them to UGV.
- After sending images to UGV, UAV will land on to UGV.
- UAV will also send these images to server for simulating the mission.

- Server will process the raw images of the maze for creating a simulation of the mission.
- UGV will process the raw images and it will construct a virtual maze from these images for solving it.
- UGV will guess a location for getting close the exit by using a heuristic algorithm on the virtually created maze.
- UGV will move to next location and the UAV will take off from the UGV on this location again for exploring unknown parts of the maze and finding a next location to move on.
- Until UGV find an exit from the maze, UGV and UAV will follow same steps over and over.

2.2.3. Scenario 3: Mapping a Region around UGV

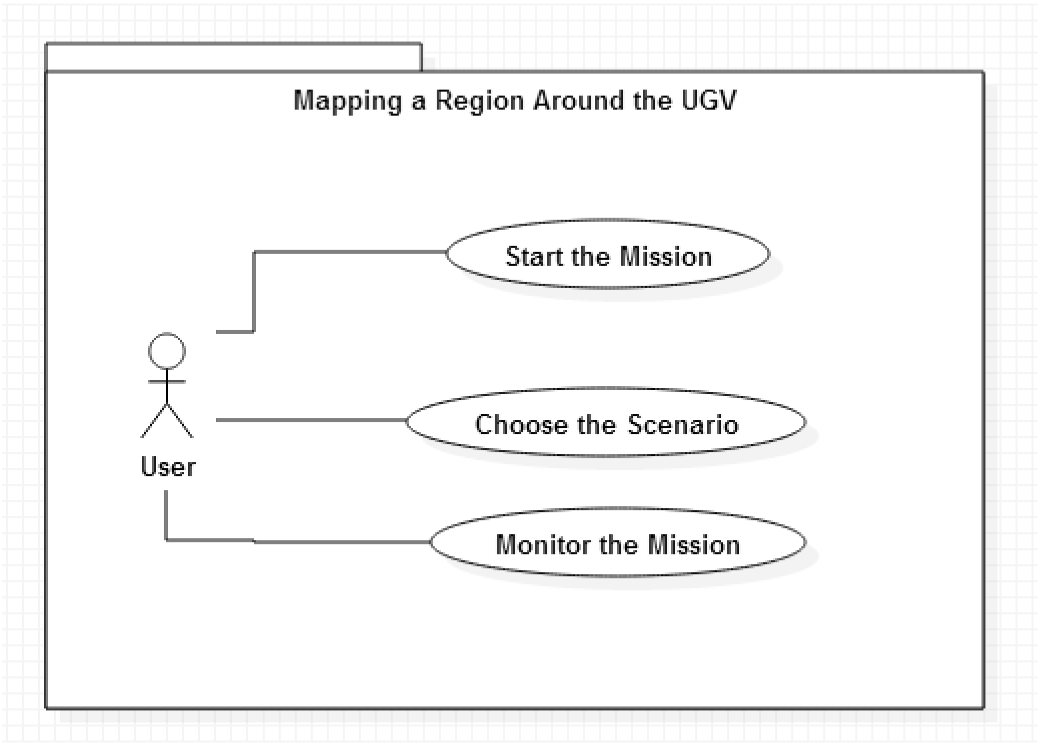

Figure 6 Use Case Diagram of Mapping a Region

- User should be able to start the mission
- User should be able to choose the scenario
- User should be able to monitor the mission via the Real-Time Simulation screen.
- UAV will be on the UGV and ready to take off in the initial condition.
- UAV will hover on the maze and take photos of it from certain locations for sending them to UGV.
- After sending images to UGV, UAV will land on to UGV.
- UAV will also send these images to server for simulating the mission.
- Server will process the raw images of the maze for creating a simulation of the mission.
- UGV will process the raw images and it will map the region coordinate the assets around the UGV.
- UGV will continue to move after its environment explored.

2.3. Constraints

- Robot Operating System is currently only supported by Ubuntu. Therefore, all required installations were made on Ubuntu 14.04.
- Because of installed ROS version, AR Drone 2.0 is used as UAV and P3DX is used as UGV.
- UAV will be hover from different height for indoor and outdoor cases so that the connection between UAV and UGV cannot be interrupted. For the clarity of the maze image, UAV also should not be away from ground more than 10 meters for outdoor case.
- The system will be started up by only one user since there will not be any user interaction among system operation.
- Since UAV has limited battery, it can only hover a certain time in the specified range. When its battery closes to finish, UAV turns to back and

lands to the top of UGV.

- Server to be used for simulation section and UGV are dependent to UAV in terms of getting maze image fragments. Therefore, any deficiency to be occurred on UAV will affect both UGV and server operations.
- The goal of this project is to get specified maze image fragments with the help of UAV, make progress the UGV by processing images and simulate both UGV and UAV motion on the screen, but the reliability of result is not guaranteed dependent to robots' working situations.

2.4. Assumptions and Dependencies

- The functionality of UAV is to take the image fragments of specified maze and send them to server and UGV, so we use different color for brands than ground color so that UAV can be able to distinguish brands and cannot go outside the brands.
- In this project, we will use P3DX as UGV and AR Drone 2.0 as UAV, therefore in every stage of this project we will depend on those robots and operations between them. Also, since packages such as tum_simulator, tum_ardrone, etc. that we will use along this project are open source, they will facilitate our work on those robots and system operation.
- Since the overall system operations are platform dependent, we will currently work on Ubuntu operating system.
- We assume that system operation will be started up by one user and nothing will be made on process as long as system works well.
- We assume that UAV has enough battery to hover above maze until it takes all maze image fragments.
- UAV will take off from above UGV and after UAV finishes taking image fragments, it should again land above the UGV. In order to land the same position, it should distinguish the position it took off. Therefore, there will

be a different colored area on the top of UGV so that UAV can land easily.

- We assume that the outside influences are minimized in order not to affect UAV and UGV functionalities through system operations.

3. SPECIFIC REQUIREMENTS

Specific Requirements can be divided into 3 parts as Interface Requirements, Functional Requirements and Non-Functional Requirements.

3.1. Interface Requirements

Since this is an Auto Pilot Project and all of the paths will be created automatically, only interaction with the system will be done by terminal in a very restricted way.

When user sends the start command, mission plan will be created and robots will move according to this plan.

After the mission created, user can be monitors the mission via Real-Time simulation. In this case output will be the simulation of the maze and robots movements.

No other input will be accepted until mission ends.

3.2. Functional Requirements

3.2.1. Starting the Mission and Choosing the Scenario

When user sends a “Start” command with “Scenario 1” UAV will take off start Scenario 1 immediately. UAV will move according to path created for Scenario 1.

3.2.2. Starting Simulation

When the mission starts, simulation screen which shows the movement of both UGV and UAV will be open. Simulation will work on Gazebo environment which is a Simulator for ROS.

3.2.3. Hovering Over the Maze and Taking Photos

UAV will hover over the maze according to predefined path which is determined according to chosen scenario. UAV will also take photos of certain locations which are pre-determined.

3.2.4. Sending the Raw Images to UGV

UAV will send images to UGV for virtual maze creation process via Wi-Fi.

3.2.5. Sending the Raw Images to Server

UAV will send images to Server for virtual maze creation in the simulation process via Wi-Fi. Simulation is expected to show the every movement of the robots and every detail of the maze.

3.2.6. Creating Virtual Maze and Exit Path

UGV will create exact copy of the maze and exit path of maze virtually for solving and getting out from it. Maze and path will also be visible in the simulation. This requirement is valid for only Scenario 1.

3.2.7. Determining Next Flight Position

UGV will create explored part of the maze and next take off position of the maze virtually for getting close to exit point of maze. This requirement is valid for only Scenario 2.

3.2.8. Mapping the Region

UGV will create explored part of the region around UGV and determine coordinate of the valuable assets with respect to UGV’s coordinate. This requirement is valid for only Scenario 3.

3.3. Nonfunctional Requirements

The purpose of this section is to specify performance requirements of the system and constraints that will be met during stages of the project.

3.3.1. Performance Requirements

- There will be only one user that will start up the process by sending start command to quadrocopter.
- The software for UAV (Unmanned Aerial Vehicle) shall run the algorithm at most 1 second since the overall system is based on real-time operation.
- UAV shall hover from the height of 2 meters at most 5 minutes, take and send photos of maze fragments at most every 10 seconds for indoor case, and shall hover from height of 10 meters at most 10 minutes, take and send photos at most every 30 seconds for outdoor case.
- After all images are sent from UAV, the software shall process and compute all collected images at most 1 minute.
- The software for UGV (Unmanned Ground Vehicle) shall recognize the processed maze image at most 1 second since the software is going to run on the robotic environment which requires the synchronous working principle.

- Server shall be used so that result of image fragments can be simulated with system running at the same time.
- The software that will simulate the motion of UAV and UGV and possible paths among running system shall respond at most 1 second with 1 second latency due to simultaneous screening.

3.3.2. Design Constraints

3.3.2.1. Hardware Constraints

- The system shall run on a robotic environment with camera system.
- AR Drone 2.0 was chosen as UAV and P3-DX as UGV according to installed ROS version and packages.
- UAV shall hover in a certain time intervals because of battery limitation.

3.3.2.2. Software System Attributes

3.3.2.2.1. Reliability

- Algorithms and software to be used in this project shall give the correct result to the corresponding cases in specified time limits stated in previous sections.

3.3.2.2.2. Security

- The security shall be handled by secure sockets and encrypted data transmission.
- UAV shall not be able to get command from external world except one specified user.

3.3.2.2.3. Scalability

- Scalability is measured by size of maze to be solved. Although maze size changes in different rate, system quality shall stay close to average performance.

3.3.2.2.4. Languages and Tools

- The project will be coded by using C++ and used OpenCV library for image processing part.
- A server will be used for simulation part of the project.
- System will be platform dependent and it will work only on Linux.

3.3.2.2.5. Usability

- Since the main goal of this project is to make contribution on symbiotic multi-robot systems research and it will be used only for a small range like academic studies and defense industry, usability is not our primary concern, which means it will be not open to public usage.

3.3.2.2.6. Availability

- As long as any physical defects on robots are not met, the system will be available until overall system finishes the specified tasks.

4. DATA MODEL AND DESCRIPTION

In this project data management is not a primary concern since this project needs only few data manipulation to accomplish. Only data will be used in this projects are sets of 2D images. These images will be taken by UAV and sent as a binary file to UGV. Then, UGV process this information for solving a maze or mapping region according the scenario.

5. BEHAVIORAL MODEL AND DESCRIPTION

This section presents a description of the behavior of the software.

5.1 Description for Behavior

System will start with UAV phase, and then it will continue with UGV and server phases. UGV and server phases will happen simultaneously. And according to the scenario user will have selected, system will decide whether these phases will happen in loop or not.

System will be started with the input from the user. After the user selects the scenario and gives the start command, system will be active. First, UAV phase will begin, in this phase UAV will take off , and it will start hovering. It is going to take photos from certain points while hovering. It will send those photos to the server and UGV, for processing and running the algorithms, as it go. After scanning the certain area, it will land on the UGV, then UGV and server phases will start. And according to the scenario user will have selected, the system will decide whether this will happen in loop or not. And it will continue until UAV finishes the scanning the whole area, and UGV reaches the target or exit. In the UGV and server phases, UGV and server will fulfill their duties. UGV will receive the images and process them by running the CV algorithms, then it will run the AI algorithms, and it will start following the calculated path until it reaches the target or exit. Simultaneously server will receive the images and run the simulator, until UGV reaches the target or exit.

5.2 State Transition Diagrams

5.2.1. UAV Phase

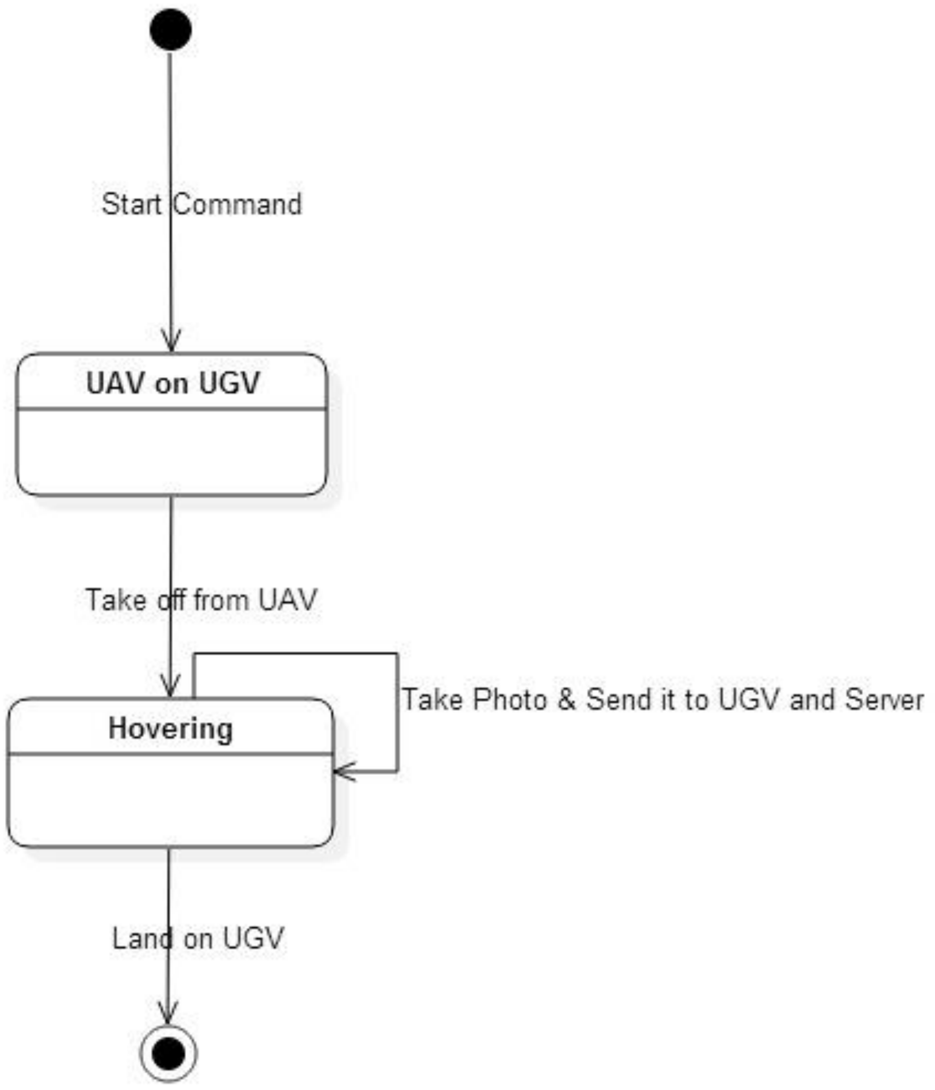

Figure 7 State Transition Diagram of UAV Phase

The Above diagram explains the phase where UAV is active.

5.2.2. UGV Phase

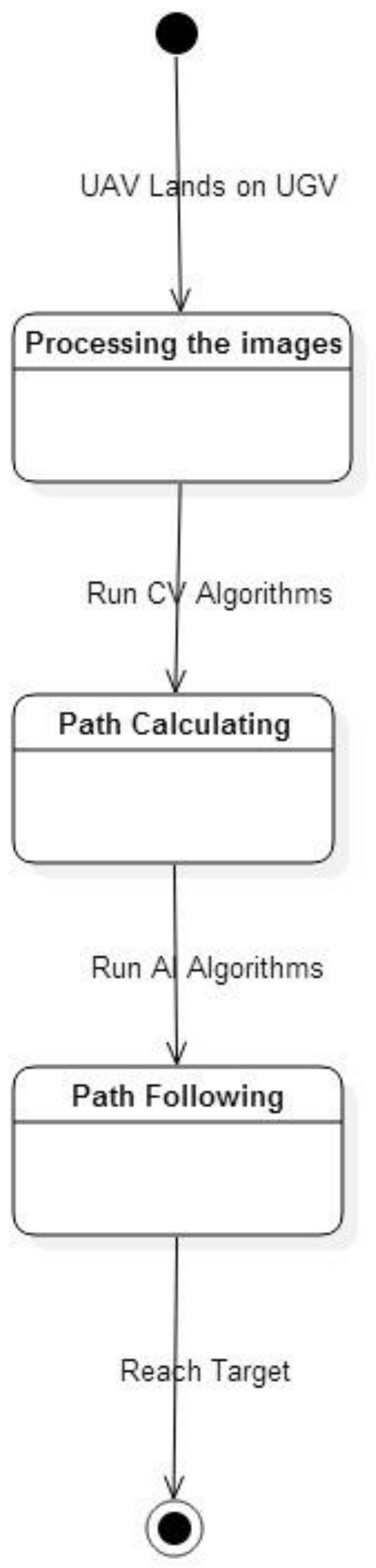

Figure 8 State Transition Diagram of UGV Phase

The Above diagram explains the phase where UGV is active.

5.2.3. Server Phase

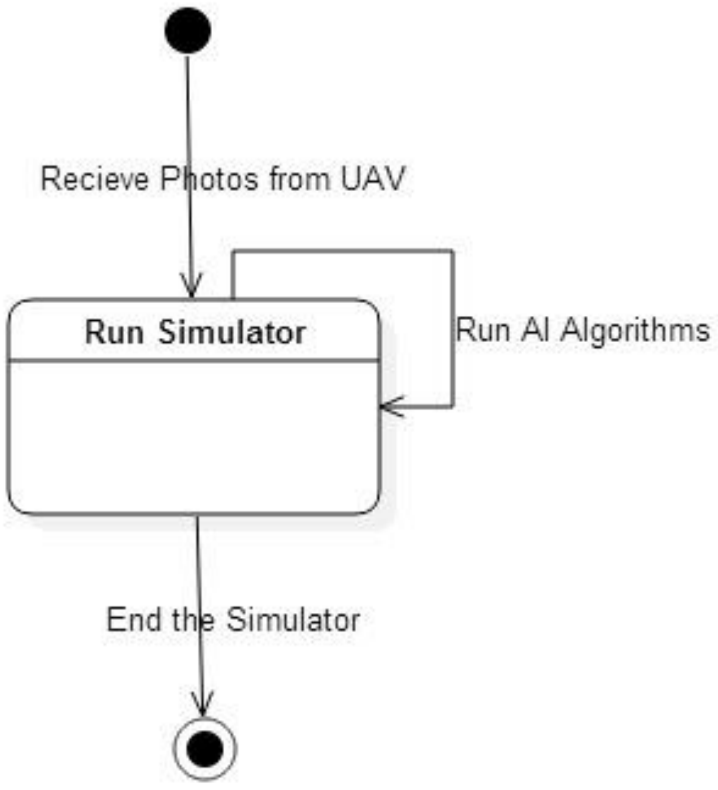

Figure 9 State Transition Diagram of Server Phase

The Above diagram explains the phase where Server is active.

6. PLANNING

In this part of the document, the structure of the team responsible from the project, the basic schedule, and the process model will be presented.

6.1 Team Structure

Our team consists of Mehmet Akif AKPINAR, Kadir EKMEKÇİ, Rana AYGÜL and Mustafa Buğra TAMER. In our team, we discuss and decide on important issues or topics all together but Mehmet Akif is team leader. These assignments will be planned to divide the workload equally. We specify some general topics and distribute them among us which has shown in table at the end of this part. After research and implementation of any topic, member shares his research and shows implementations with other members. Besides of it, some of minor subjects have learned by all members. We will have weekly meetings with our assistant Burak Kerim AKKUŞ. We will also have regular meetings with our consultant teacher who is Asst. Prof. Selim TEMİZER to take some advice in order to handle if any problem occurs and talk about direction of project.

Member	General(Major) Subject(s) in Project
Mehmet Akif AKPINAR	AI and CV algorithms
Kadir EKMEKÇİ	AI and CV algorithms
Rana AYGÜL	Simulation of the Missions
Mustafa Buğra TAMER	Interconnection between the Robots and Server

6.2 Estimation

Table shows after groups and projects took shape and project documents prepared.

Table start week date is 3.11.2014.

Estimation Weeks	**Deadline**	**Task Explanation**
Weeks 1,2,3	24.11.2014	Setup ROS & Gazebo on personal computer. Creating a basic Quadrocopter simulation.
Weeks 4,5,6	15.12.2014	Making Test Flights with actual Quadrocopter
Weeks 7,8	29.12.2014	Creating a basic Autopilot for Quadrocopter simulation.
Weeks 9,10,11,12	26.01.2015	Creating and deploying a stabilization algorithm for Quadrocopter
Weeks 13,14,15	16.02.2015	Creating a more advanced Autopilot for Quadrocopter simulation
Weeks 16,17,18,19	16.03.2015	Creating final Autopilot which is capable of following every route
Weeks 20,21,22,23	13.04.2015	Work on Scerio 1

Weeks 24,25,26,27	11.05.2015	Work on Scerio 2
Weeks 28,29,30,31	05.06.2015	Work on Scerio 3

6.3 Process Model

We will apply agile model for our symbiotic multi-robot system so that system can respond quickly to changing requirements after testing system. Agile method is based on an iterative approach, each iteration involves planning, requirements analysis, design, implementation, testing. Once we will generate the initial version of symbiotic multi-robot system, then our system will be developed according to accuracy of our system and testing results.

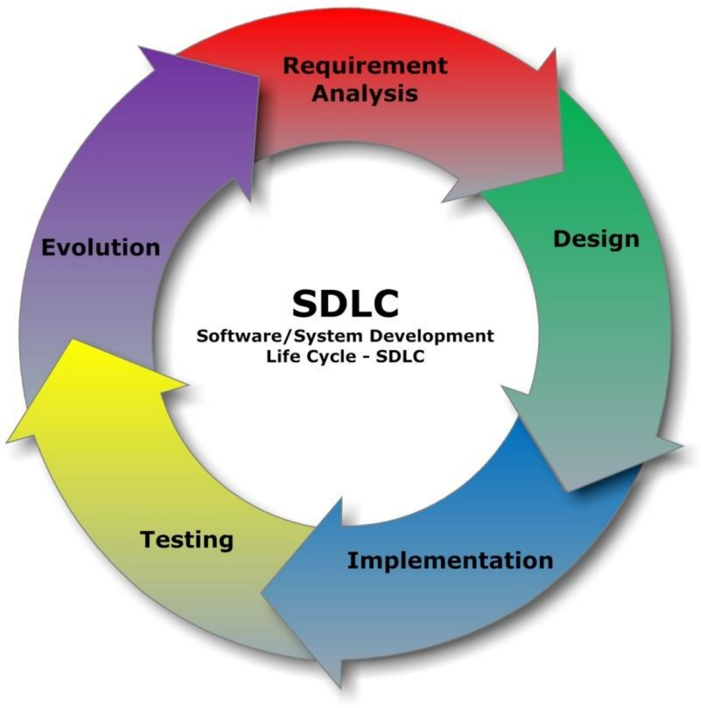

Figure 10 Process Model of This Project

7. CONCLUSION

This document gives information about the project “Raven Nest” which aims to bringing efficient solutions in different scenarios where symbiotic multi-robots (UGV and UAV) have limited properties. This is advised by Asst. Prof. Selim TEMİZER. This document prepared for showing a complete description of the symbiotic multi-robot system behavior and the necessary requirements for developing the project. Diagrams are added to explain or describe the idea of the process and concept in order to have a clear understanding. The project will be designed throughout the project is explained clearly and the related problems that might be experienced in the process are stated.