Contents

1 Problem description

The problem description is taken from the project offer on blackboard[1]

The market for unmanned surface vessels (sailing drones) is growing, and there is a need for an open architecture auto-pilot that is usable on different platforms.

This project is about designing and implementing the electronics and embedded software for an autopilot which is able to receive steering commands from a navigation system (distance to go left/right) and from thoes instructions, steer a small boat (et. control left/right thrusters).

Project details:

- PCB design of autopilot electronics (if something existing cannot be used)
- Implement PID control loop to get back on course based on received left/right commands
- Control of thrusters in case of two-thruster catamaran
- Control of wheel in case of outboard motor on boat
- Test trials at sea
- EIVA will provide necessary hardware, include test boats / catamarans, including PCB production cost
- EIVA will provide work-space (desk, lunch etc) at EIVA's facilities
- IF the project is successful, EIVA will offer to finalise the product and make it available for sale with a commision to the student

2 Requirements specification draft

The requirements for the product are prioritized using the MoSCoW method. Using this, the requirements for the product are divided into four sections, where the most important elements are given the highest priority. **Must** are the requirements that are an absolute necessity for the product. **Should** are the requirements that are also of high priority, but not quite mandatory. **Could** are requirements which may be met, if the time and other constraints of the project allow for it. **Won't** are the requirements the product will not meet, but could be developed at a later point in its lifetime.

The following priorities have been chosen for this project:

Must
- Navigate waypoints from user input
- Be compatible with NMEA protocol GPS input
- Use GPS for localization
- Implement a PID control loop

Should
- Control thrusters in two-thruster catamaran
- Use a graphical user interface for user interaction
- Be able to change the PID parameters

Could
- Control wheel in outboard motor on boat
- Be generic enough to use with other engine types

Won't
- Use pylogon-coverage for a specified area
- Avoid obstacles

The following two diagrams specify the actor context to the system, and the use cases for the system.

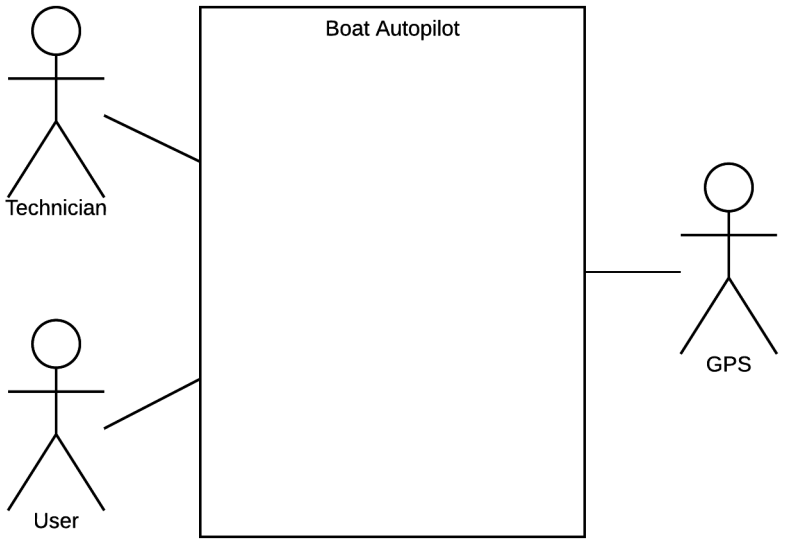

Figure 1: Actor context diagram

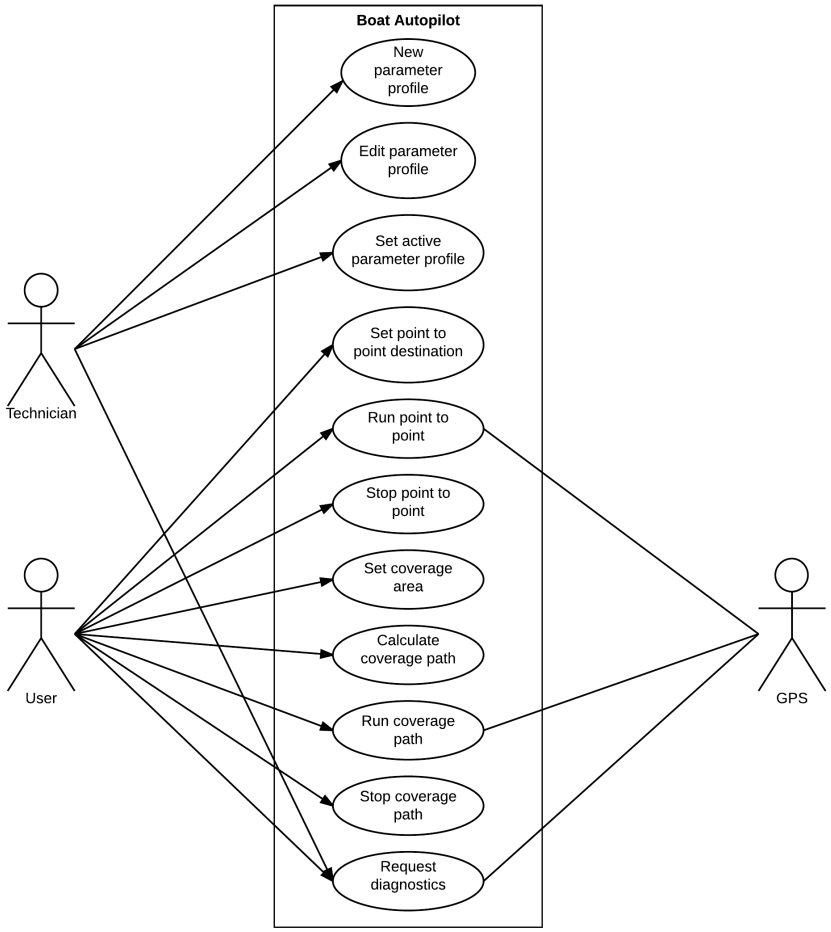

Figure 2: Use case diagram

2.1 Actor description

The following section describes the system's actors. Every actor has a type and a short description of its function and impact on the system.

2.1.1 Actor: Technician

Type:

Primary

Description:

A technician with knowledge of the system. This actor triggers the use case "Specify parameters".

2.1.2 Actor: User

Type:

Primary

Description:

The user, or customer. This actor triggers the use case "Specify waypoints".

2.1.3 Actor: GPS

Type:

Secondary

Description:

The global positioning system. This actor assists in carrying out the use case "Navigate waypoints".

2.2 Use cases

2.2.1 Use case 1 - Enter parameters

Goal:

To enter the parameters for the boat being used.

Initialization:

Technician

Actors:

Technician (primary)

References:

None

Simultaneous occurances:

One

Prerequisite:

None

Result:

The autopilot parameters have been updated.

Main scenario:

1. The technician investigates the boat which will be used, specifically its drive system.
2. The boat parameters are set using the graphical user interface.

2.2.2 Use case 2 - Specify waypoints

Goal:

To enter waypoints, outlining a desired path through an area.

Initialization:

User

Actors:

User (primary)

References:

None

Simultaneous occurances:

One

Prerequisite:

Use case 1 - Enter parameters has been completed.

Result:

Waypoints for the path have been set.

Main scenario:

1. The user accesses the graphical user interface and sets a series of waypoints, detailing a path through the area starting at Waypoint 1.

2.2.3 Use case 3 - Navigate waypoints

Goal:

To navigate the set waypoints, outlining a desired path through an area.

Initialization:

User

Actors:

User (primary),

References:

None

Simultaneous occurances:

One

Prerequisite:

Use case 2 - Specify waypoints has been completed.

Result:

The boat has completed its path through each waypoint.

Main scenario:

1. The user accesses the graphical user interface and selects "Start".
2. The Boat Autopilot updates its current location using GPS signals.
3. The Boat Autopilot begins navigating to the first waypoint using a feedback control loop, and continues this until each waypoint has been visited. The Boat's location is updated throughout using GPS.

Bibliography

[1] EIVA. *ELECTRONICS/SOFTWARE - Boat autopilot*. 2017.