P7 USV TRUSTWORTHY INTERFACE

Thanh Pham
tpham22@islander.tamucc.edu

Chia-Cheng
jjjj222@tamu.edu

Vinay Datta
vpinnaka@islander.tamucc.edu

Abstract-User interfaces are important in human-robot interaction. The goal of user interfaces is to make it easy for user to interact with robot, either issuing commands or receiving responses. A good interface will not only increase the efficiency of working with robots, but also build the confidence and trust about the whole system. In this paper, we present the procedure used for analyzing, designing, and implementing step of EMILY trustworthy interface

I. INTRODUCTION

Until now there is no optimal user interface for EMILY, only the mission planner (Figure 1) available for windows PC serves as an interface to monitor the status of EMILY onboard hardware. Therefore, users have hard time keeping track of the status of EMILY. It happens very often that EMILY stop working in the middle of mission and the user have no idea about what goes wrong. And it highly decreases the trust of EMILY.

In this work, we designed an USV interface that encourages trust by displaying robot state, wireless strength, battery strength, and other options such as GPS signal strength, Compass and speed. Also, the interface provides appropriate alerting and reminding. Options available on the interface are decided after deep analysis of three most popular different UAV interfaces and partial problem set derived from EMILY ethnographers interview.

Figure 1. Interface of Mission Planner.

The rest of the paper is organized as follow. We first discuss some of the related work. After that we describe our approach and implementation in section III and IV. The experimental results are in section V. Finally we conclude our work in section VI.

II. RELATED WORK

There are many quadcopters today come with operating interfaces on tablet, such as DJI phantom and Aeryon Scout. Most of them include a map, a camera view, and other status. They provide an illusion as if the operator is on the quadcopter, and it facilitates the control. These interfaces are close to what we want except they are designed for UAV.

Other works, such as DroneDeploy, provide an universal interface for many UAVs. The purpose of these interface is to monitor the operation of UAV. It focuses on automatic control instead of manual operation.

Joshua designed an interface for Micro Unmanned Aerial Systems (mUAV), but the target of it is mission specialist instead of pilot. This interface includes only camera view and the functionality to control the camera and take pictures. Other information for pilot are removed for clarity.

Figure 2. Interface of DJI Phantom.

Figure 3. Interface of Aeryon Scout.

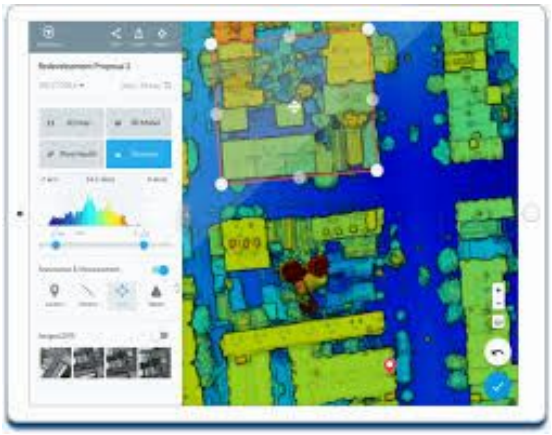

Figure 4. Interface of DroneDeploy.

Figure 5. Interface of Mission Specialist

III. APPROACH

In order to create a trustworthy interface for EMILY, we have evaluated different status information that increases the trust. We generated a problem list from the interview with ethnographers also we analyzed the popular drone displays such as DJI pilot, Aeryon Scout, and DroneDeploy. We came up with a list of feature that will increase the trust of EMILY and incorporated them into our interface.

3.1 Interview with Ethnographers:

The main objective of USV trustworthy interface is how to choose the right features that can encourage the trust of EMILY to the users. The starting point of the project is going through the ethnographers' observation about EMILY's operation. We started our requirement analysis for our project with ethnographer's interview. The interview is a question and answer session which lasted for 30 minutes and we have two participants. The questions which we asked the ethnographers are as follows:

1. What are problems that you see when EMILY is functioning?
2. Which environment condition cause problems to EMILY to function?
3. How frequently do you lose communication with EMILY?
4. What is the major problem you usually face when EMILY is out of sight?
5. When EMILY reaches the victims, what problem do you see there?
6. How long do you see that EMILY can continuously work without recharging?
7. What is the maximum distance between EMILY and user that you see It can work well?
8. What is the probability that EMILY can finish the mission?

Answers we collected from ethnographers facilitated us in generating a problem list which are faced by EMILY operators. Table 2.1 shows a partial problem list obtained from the summary of ethnographers interview. The battery status information of EMILY is unknown so it is unpredictable about the duration that EMILY can continue to work. The wireless signal strength is unknown so it is uncertain about EMILY status if it is still has communication to the controller or not. These are the two major concerns expressed by the ethnographers in the interview. Also ethnographers expressed that speed of the EMILY and distance from home will increase the trust of the robot. In addition victim status and warning messages with sound are some suggestions we obtained.

Problem List obtained from Ethnographers interview
• Battery status information • Wireless signal strength • GPS strength • Speed of EMILY • Distance from home • Victim status • No warnings about EMILY status

Table 3.1: Problem list generated from Ethnographers interview

3.2 Analysis of popular drone displays:

After summarising the ethnographers interview we moved on to drone display analysis. Our primary motivation is, if any of the popular drone displays solve the problems which we listed and how these displays increase the trust of UAV. Our analysis resulted some important features of the display which increase the trust of EMILY.

DJI Phantom Pilot:

1. DJI Pilot display status of UAV in the middle of Action toolbar, this feature increases the trust so we thought of incorporating it in our display
2. Pilot app also have a switch between Map View and Camera View: This function offers the user an extra information about the condition of area and plan when needed

DroneDeploy:

1. DroneDeploy has a sidebar interaction with status indication. This type of interaction is easy to interpret and use for the user
2. Tracking the UAV on map is another important feature obtained from DroneDeploy
3. In dronedeploy interface voice message is used to alert the transition from one behavior to another

From the summary of our analysis we have listed the trustworthy features that can be incorporated in our interface. Table 3.2 show the list of features selected from popular drone displays.

Features selected from popular drone displays
• Tracking the location on Map • Status on sidebar • Voice alerting • Status message • Camera and map view

Table 3.2 : List of features selected from popular drone displays

3.3 Choice of Platform

For implementing trustworthy interface on mobile device, a comparison between 2 most popular mobile platform Android and iOS were analysed [1]. Among them Android is more appropriate because it is open source, widely used and highly supported. Whereas iOS restricts the usage to specific devices.

3.4 Interface color selection and design

A meticulous analysis is conducted while choosing the color for our interface. Most of the time EMILY is operated in sun, therefore we have designed the interface such that it is visible even in direct sunlight. Figure 3.1show our trustworthy interface. According to Aries Arditi et al [2] we should exaggerate lightness differences between foreground and background colours, and avoid using colours of similar lightness adjacent to one another, even if they differ in saturation or hue. So, we have selected yellow and white background with black text.

Figure 3.1 EMILY Interface with trustworthy features

Features on display	Justification
Battery status	Operator need to know how long that EMILY can last for work, so they can prepare the plan or modify it if needed
Wireless strength	Wireless strength will provide the status of communication between operator and EMILY. It is important for operator to know that they can control EMILY and receive information from it
GPS state	EMILY's movement is based on the LatLong coordination, operator always want to make sure that EMILY is in GPS signal coverage
EMILY speed	Operator need this information to control speed of EMILY in manual condition
Distance from home	EMILY needed to be in fixed range from operator to guarantee that it is in good communication with operator and it can be rescue in special condition
Elapsed time	Operator don't want to use

	EMILY continuously for a long time. They want to keep EMILY in good state, so it can last long
Tracking position	Previous positions of EMILY are required to ensure that EMILY has been following the order correctly.

Table 3.3 Justification for each feature on emily interface

Feature	Icon	Justification
Battery status		This icon of battery status is used in almost android smartphone or tablet. So, it is easy for user to recognize its meaning
Wireless strength		This bar icon representing the wireless status has been using in android smartphone or tablet for a long time
GPS state		This icon represents the ground station in satellite communication. We want to focus on the receiving side of satellite data
EMILY speed		This icon represents the speedometer used in car. User can easily understand what it is used for
Distance from home		This home icon stresses the distance from EMILY to home.
Elapsed time		Clock icon means to emphasize the time duration of EMILY. This icon used in iPhone or Android smartphone
Tracking position		This cursor used to show the current position and direction of EMILY. It's like the cursor used in all computer nowadays.
Configuration function		The gear icon used for configuration function because of it's common use in Window setting icon.
Connect button		The plug icon shows the connection status of our interface with EMILY. This icon has been used in many android application.

Table 3.4 Justification of icon used in the application

IV. IMPLEMENTATION

Implementation of Trustworthy interface involves two aspects, communicating data from mission planner and displaying the information on our interface. Mission planner runs on a windows computer whereas our interface is an android application running on a 10 inch tablet. We use an ad-hoc network for communication between Mission planner and our interface. We have implemented a server program in python which continuously read data from mission planner and send the information over the network. Our python server works for both autonomous and manual control of EMILY. Once our application is connected to the server, the status information is continuously updated on our interface and also alerts the pilot in critical situations.

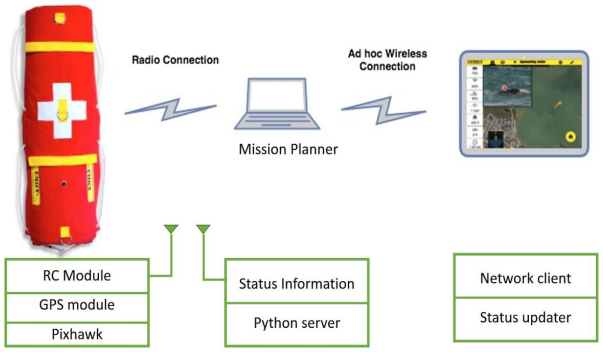

Figure 4.1: Block diagram for Trustworthy interface implementation

4.1 Implementing a server program

The first stage of implementation is developing a python script that reads data from Mission planner and send the data over network. Dedicated python server first creates a socket and continuously listens until a client is connected. If any client is connected the program reads a set of status informations from Mission planner. The list includes battery status, wireless signal strength, GPS state, speed, distance from home, latitude and longitude of EMILY's home, present lat-lng location of EMILY and also the armed status of EMILY. Collection of only specific information is done because the feature which we have incorporated in our interface some or the other way requires this shorthand information list. The collected data is converted into JSON format and sent continuously to the client until the connection is dropped. Pseudo code for server program is illustrated in the table 3.1

Server Pseudo code

```
Create TCP Socket
Loop until keystroke
   Listen for connection
   If connected
        Read data from client
        Loop until connection is closed
            Collect status info from Mission Planner
            Send status data to client in JSON format
Close Socket
```

Table 4.1 Pseudo code for python server

4.2 Ad-hoc network

Create an ad-hoc network in windows laptop in which Mission planner is running. Connect the android tablet in which EMILY trustworthy interface is running. Having a secure network will avoid network interference and provide a reliable connection between the server and the android client. In general the wifi range of a laptop is 10 meters so, it

is recommended to have the operator using android tablet in this range.

4.3 Android interface implementation

The trustworthy interface which the operator uses is a Android application designed specifically for 10 inch tablet. The implementation is multithreaded with four threads running concurrently. Thread 1 is responsible for network interface. In android terms thread 1 is a async task which runs continuously until disconnected by the user. Thread 2 is responsible for updating the EMILY location on map. Thread 3 is responsible for continuously updating the icons and values of the side bar which display the status information of the EMILY. If any of the status such as battery, wireless or GPS is below the threshold level this thread is responsible for visually differentiating the status information. Another thread is also implemented to play the warning sounds and updating the status message. The reason behind of having two different threads for visual alerts and voice alert is because the media play rate is much slower than the updation rate of the sidebar. Overall, the quad threaded application performs appropriately with a delay time of less than a second. Most of the delay between the Mission planner updation and EMILY trustworthy interface arises in network transfer. Though we are using basic socket program which does not have any overhead apart from the data, the network transfer is a major concern. Table 4.2 illustrates the pseudo code for EMILY trustworthy interface.

Pseudo code for Android interface
Thread 1 - Network task
Create a socket Connect to the server with IP and Port Send an ack message Loop until disconnected by user: Receive data from server Convert string to JSON Insert values of JSON object into Datamodel After disconnection close the socket
Thread 2- Updating EMILY on map
Get EMILY current location Add EMILY location to list of coordinates For every second Draw the lines with all the coordinate in the list Populate EMILY's current location with marker If disconnected from server Clear all the lines on map
Thread 3 - Updating status information
Initially populate the status info. with dummy values Run the thread handler in loop with one sec. delay Get the data from the Data model Update the icons and values dynamically
Thread 4 - Warning alert and status message
Continuously compare the status values(battery, wireless, GPS) with threshold values in settings If less than threshold Play an alert voice Update the status message

Table 4.2: Android application Pseudo code

V. EXPERIMENT

In this step, we had to prepared a detailed plan including devices and scenario to test the function of our interface.

- Equipment required for experimentation:
  - + Mission planner software
  - + A windows PC or laptop to run Mission planner
  - + Emulator of android device
  - + EMILY

- Preparation steps:
  - + Running EMILY's Mission Planner in a laptop to control EMILY
  - + Connecting EMILY's Mission planner software to our interface through ad-hoc network
  - + The interface can receive information from EMILY and show status on the screen

- Metrics for our interface that we care are as following:
  - + The confidence and trust that operator feel about EMILY when using our interface.
  - + Is there any confusion about status information presented on the interface?
  - + Are the warning, icon changing is reasonable for special condition?

- Data collected in the experiment including speed of EMILY, wireless signal strength, GPS signal, distance to Home, distance to waypoint, elapsed time, current position in latlong.

- Scenarios for testing:
  - + Case 1: Create the simulation with normal condition to test the response time of application on updating the status information of EMILY
  - + Case 2: Create in the simulation the circumstance of low battery, low wireless signal to test the warning message of application (suitable warning and on time)
  - + Case 3: Create the circumstance of lost connection to check the warning and updated information (suitable warning and on time)

We used 2 types of experiment to test our interface:

- Type 1: Simulation type:
    + We ran a simulation python script in the Mission Planner and transfer information to our interface through ad-hoc network

In this way, we did not need real EMILY running and we can create all possible scenarios such as low battery or low wireless signal to test the warning function of our interface (figure 5.1).

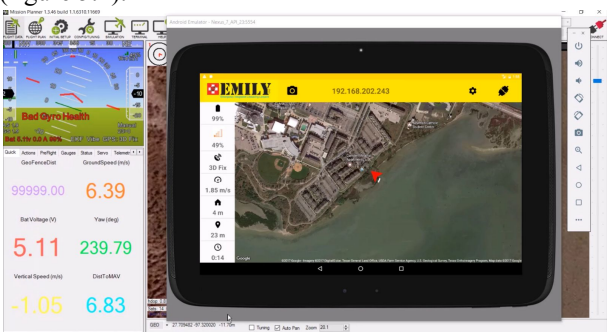

Figure 5.1 Simulation testing type

- Type 2: Running EMILY in real condition:
    + We ran EMILY in real condition with manual and autonomous ways
    + EMILY exchanged data to Mission Planner in real time, our interface received data from Mission Planner and show that on the screen.

In this type of experiment, we tested in 2 phases:

+ Phase 1: We put EMILY into car and run around the campus to test the position tracking function (figure 5.2).

Figure 5.2 Testing EMILY on land

+ Phase 2: We ran EMILY on the lake. Our application will show all information received from Mission Planner on its interface (figure 5.3).

Figure 5.3 Testing EMILY on the lake

Our application retrieve data from Mission Planner every second. We compared value shown on the Mission Planner with the value shown on our interface to check that if our interface demonstrated exact data and on time. In some condition, the icon representing each type of data will change shape when data go beyond the predefined threshold. At that time, there will be some voice warning about current status of EMILY.

In two type of experiment, we saw that our interface can continuously update right status information of EMILY. The icon and warning could reflex to any circumstance of EMILY on time and precisely but finally we still need objective feedback from experienced operators of EMILY about our interface.

In two phase of real condition testing, we took live video of procedure from beginning to the end. Then, we send them to operators of EMILY in order to have their opinions about our application's functioning and most importantly, whether or not our interface can increase the trustworthy of EMILY.

After watching our videos, EMILY operators commented that our interface is good but they need more functions allow them to control EMILY, or swap between map view and camera view.

VI. CONCLUSION

After this project, we have finished the ethnographers interviewing step, designing step, and implementation step of the interface with some functions that meet part of final requirements. In the designing step, we have gone through multiple mockups to satisfy EMILY operators. In the implementation step, there are some functions that we have not finished such as camera view or manual switching functions. In the future work, missing functions can be added to round off the interface application and enhance the trustworthy of EMILY.

VII. REFERENCE

[1]http://www.diffen.com/difference/Android_vs_iOS [Access 21 March 2017]
[2]https://www.nzao.co.nz/sites/default/files/EffectiveColorContrastBooklet.pdf