Workload and Situational Awareness management in UAV teams through interface modelling

António Sérgio Ferreira
Faculdade de Engenharia da Universidade do Porto, Underwater Systems and Technology Laboratory (USTL), Rua Dr. Roberto Frias s/n, 4200-465 Porto, Portugal {asbf}@fe.up.pt

Abstract. The practical use of Unnamed Aerial Vehicles (UAV) to extract the human element from dangerous situations is becoming a more feasible scenario with every new iteration of technological advancement. Cost and implementation time reductions have led to the emergence of various off-the-shelf solutions which allow a wide array of usage of these systems. However, on the most part, the development emphasis of these systems does not reside on usability quality or human factor effects. This hampers the viability of these systems in scenarios were human operators must be focused on other duties other then vehicle control and supervision, therefore a priority must be given to the control interface. Through the implementation of a Real-Time Strategy (RTS) game interface paradigm, in an already existing command and control framework, it was possible to develop a control and supervisory interface console which achieved a decrease of workload in preliminary testing.

Keywords: HCI, HRT, UAV, Workload, Situational Awareness, Real-time strategy games

1 Introduction

With every new chapter of technological advancement, either in the hardware and software sectors, it becomes easier or more cost-effective to develop and apply Human-Robot Teams (HRT) in operation scenarios of dangerous nature. A wide array of scenarios, either simple Search and Rescue missions [1] or more complex military missions [2], are steadily becoming routine theatres of operations for autonomous vehicles. Among the various types of automated vehicles UAVs, through their ability to quickly cover large tracts of ground from an aerial perspective, have demonstrated be a valuable asset when aiding the activities of human teams in a number of complex and demanding situations [3].

In order to do so these vehicles rely on complex control systems which allow the human team member to take advantage of their capabilities, however the rising complexity of said systems has led to an increasing need for more human operator per UAV platform. Nevertheless steps are being taken to allow future
unmanned vehicles systems to invert the operator-to-vehicle ratio so that one operator can control multiple vehicles connected through a decentralized network [4]. However this decentralization puts an increasingly high strain on the human operator’s workload [5] reducing the team’s overall performance [6] if steps aren’t taken to deal with these variables.

There are various ways to cope with the workload reduction challenge. It can be achieved by an emphasis on further developing and extending the control schema by creating frameworks which allow an adaptive level of automation [7] [8] and dealing with the inner-workings of control algorithms or by addressing interface usability issues, prioritizing the human user experience.

In order to achieve a low-entropy interface with the human operator, in a stress filled situation, a great amount of attention must be given to the saturation point of information the operator can handle at one given moment [9][10]. Conceiving the interface using a multi-modal approach is advantageous since it’s one way of avoiding the natural bottleneck that arises from using only one communication channel between the UAV and the operator [11].

Nevertheless, even with these approaches to reduce the operator workload there is another factor which must be taken into account when dealing with Human-Computer Interaction (HCI). Every interface has a learning curve which the user must overcome in order to fully take advantage of its capabilities, therefore it is safe to assume that the more familiar the interaction schema the lower the learning curve will be. It can than be assumed that video games could prove a valuable asset when looking for familiar interface layouts and techniques. On this premise there have been various successful applications of game-based interfaces in HRT control [12][13]. For this paper the RTS game genera will be used as the basis for interface analysis. This type of game paradigm has already shown promise in other research [14] surrounding autonomous vehicle control.

Fig. 1. Interface example of the Macbeth command and control system[21].
To deal with the emergent need to simplify the method of interaction with more complex and capable UAVs the Neptus framework, developed by the Underwater Systems and Technology Laboratory at FEUP, has been extended from its original role as a Command, Control, Communications and Intelligence (C4I) [19] for autonomous underwater vehicles (AUV) to also incorporate UAV control and supervision. It is upon this already established framework that the validity of an RTS paradigm will be tested. All development efforts were made so to minimize the use of additional peripherals than the standard single monitor setup to on the one hand, reduce logistical system costs and on the other hand, increase the number of operational scenarios where the console might be of viable use.

In order to evaluate both workload and situational awareness metrics specific tests will be employed. For workload evaluation the method used was the NASA Task Load Index (NASA-TLX) questionnaire, already widely used in military and commercial aviation pilot workload tests [15][16]. For situational awareness the method used was the Situation Awareness Global Assessment Technique (SAGAT) also widely used, not only in the aviation field but also HRT testing[17][18]. Although time constraints and availability of certified human operators limited the testing phase to only one account, the test proved promising results and will be augmented in future developments.

In Section 2 a quick overview of the framework used as basis for this paper is given followed by a conceptual description of the developed prototype in Section 3. All the tests performed on the prototype are detailed in Section 4 and the subsequent conclusions are presented in Section 5. In Section 6 an analysis of the envisioned future work is made.

2 Related Work

There already exist command and control frameworks which demonstrate an emphasis on the quality of the human interface, as seen in Fig. 1, with the intent of reducing workload in situations where one sole human operator must command various autonomous vehicles. However the recurring standard interface paradigm is one of high complexity and high logistical and maintenance cost, even for the control of single autonomous platform, as demonstrated in Fig. 2.

One which attempts to reverse the aforementioned standard is the Neptus C4I framework which is a distributed framework for operations with networked vehicles, systems, and human operators. Neptus supports all the phases of a mission life cycle: planning, simulation, execution, and post-mission analysis. Moreover, it allows operators to plan and supervise missions concurrently [19].

The Neptus framework communicates with all autonomous vehicles thought the IMC [20] communication protocol. This protocol, based on XML, augments the frameworks compatibility with various kinds of autonomous vehicles by allowing a level of abstraction which promotes decoupled software development.

Furthermore, in order to minimize undesirable interdependencies throughout the framework’s architecture, the Neptus framework consists of a series of independent plugins surrounding a core section of functionalities.
Nevertheles, Neptus encompasses a console building application which facilitates the rapid creation of new operation consoles for new vehicles with new sensor suites, as well as the remodelling of old consoles for current vehicles (Fig. 3). It is one such console that ultimately will be conceived as a result of the work developed.

3 Prototype Development

Using the Neptus framework, development began towards the creation of a console that would incorporate conceptual ideas implemented in modern RTS interfaces in order to further expand its usability by the human operator.
3.1 Conceptual Design Details

One of the major difficulties in this kind of endeavour is the simplification of the large amount of data deemed necessary for the safe operation and supervision of UAV systems. The usually overwhelming quantities of numeric and textual data tend to lead to a multi-monitor solution which quickly raises the complexity of the interaction experience. In order to avoid this necessity a lot of information fusion is needed.

The standard RTS game relies heavily on representative icons to convey information to the player. Distinct icons can, on their own, quickly transmit large amounts of information, that would otherwise require a large amount of textual representation, just by small changes in size, orientation and draw rate. With simple icon modifications one can summarize a vast amount of the UAV’s relevant data. Coupled with this the RTS genera also employs color as a way to clearly indicate unit status and different levels of urgency. Both these elements combined on their own can account for a great amount of information simplification. However there are other aspects that can be extracted from the RTS example which can prove invaluable to the overall console’s efficiency.

![Fig. 4. Example of similarities in layouts between different RTS games interfaces.](image)

Modern RTS games, although having different themes at their core, follow a seemingly standardized formula when laying out their different interface components (Fig. 4). One can reason that this behaviour derived from the genera’s vast years of experiments with different approaches which slowly converged into a model that is now amply used by the majority of developers. Moreover, even the various interface components converged to form a pre-set of core interface elements that are rehashed from game to game (mini-map, unit status, etc...).

These unofficial standards, combined with the ample size of the game market, led to an unintentional spread of these tendencies to a wide audience thus making it a useful basis of development when aiming at a reduced interface learning curve.

3.2 Architectural Details

Following the conceptual design, a layout for the console’s components was idealized. As shown in Fig. 5 the focus is to emulate the default RTS layout by
having a central large map region (section 1), a mini-map to aid the human operator to not lose context of other UAVs operating outside his view scope (section 2), a series of smaller additional panels which will house simplified numerical data (sections 3 through 5) and two final panels which will be used to house contextual information about other autonomous vehicles operating in the area.

The central concept is to concentrate all the information with the highest priority in the main panel (section 1) through use of icon disposition and color, relegating textual information to one of the support panels.

Before the development of this new operation console started there was already a pre-existent standard command console for UAV operations which served as a benchmark in the later testing stage, in order to determine the effectiveness of the new design implementation.

### 3.3 Conceived Prototype

In Fig. 6 we can see the finalized operational version of the developed console. As expected, from the RTS paradigm influence, we have a large main map, aided by a smaller mini-map, which provides a top-down view of the operation area. However the typical isometric point-of-view was not adopted due to rendering limitations when attempting to model 3D environments. On a normal operation scenario it is complicated to acquire an up-to-date and accurate 3D model of the area so it was deemed that, as this stage a simple map-based representation would be preferable.

Hovering over the map are icon indicators of the various stages of the UAV’s flight plan, from the pre-launch stage to the final runway taxi quickly giving the operator a grasp of the mission’s current stage.

The mini-maps viewport feature allows the operator to quickly know what part of the operation theatre is being shown on the main panel, as well as what are the current visual limits.
Aiding the main viewing panel are a simple set of numerical data regarding velocity and GPS indications, however the UAV’s actuator status is complied and provided by the miniature segmented UAV model. Each section informs the human operator of its status by alternating color, providing the operator with a faster means of understanding the UAV’s overall condition.

At the far right of the console are two auxiliary panels one of which allows a quick visual reference of the all active autonomous vehicles’ altitudes and another which lists all detected autonomous vehicles in the operation area. Through this list the operator can also switch between operating vehicles.

4 Tests and Results

In order to test the efficiency of the new proposed interface alterations, on reducing the operator’s perceived workload and augmenting his situational awareness, a simulation was held at the USTL with one of its senior UAV operator.

The simulation asked for a team of 3 UAVs to accomplish a simple set of surveillance tasks over a pre-designated and already known area. One of the operators was target of the evaluation while the other two were only considered participants. Firstly the operator used the already existing Neptus UAV interface to execute his mission, afterwards the new developed interface was used.

4.1 Workload Analysis

At the end of the simulation the operator under evaluation was given a standard NASA-TLX questionnaire to determine his perceived workload. The results obtained are shown in Fig. 7.
4.2 Situational Awareness Analysis

During the simulation the operator would, from time to time, quickly be negated access to the operation interface in order to answer a set of pre-determined queries with the intent of ascertaining how aware of his surrounding environment he was. The queries are as follows:

1. Point-out the position of each UAV currently active;
2. Single-out the currently selected UAV;
3. Determine the altitude of all active UAVs;
4. Determine the altitude of the selected UAV;
5. Determine the speed of the selected UAV;
6. Determine the current UAVs waypoint;
7. Determine the current UAVs ID.

A total of ten interruptions were made and the results obtained are shown in Fig. 8.

These preliminary results show promise as it seems accurate to extrapolate that the modifications made on the command interface, using the RTS paradigm, have lowered the operators workload and, at the same time, risen his situational awareness during the mission’s execution.

5 Conclusions

Throughout this paper references were made to the growing importance of UAV systems, paying special attention to their valuable application in real world scenarios. It was then presented the concepts behind a possible solution to be implemented into a pre-existent framework with the ultimate goal of managing UAV workload a mission scenario and the details around this solution were presented and discussed.
Fig. 8. Overall situational awareness percentage comparison, throughout a series of pre-determined queries.

The C4I operation console, ultimately created based on the presented solution, enabled the reduction of the workload felt by the operator while at the same time increasing his situational awareness.

6 Future Work

Although promising these type of results are, by their nature, subjective to the human operator’s sense of workload and own experience of operation procedure. This means that further testing must be performed in order to ascertain if the overall improvements detected prevail in a wide variety of human operators. Furthermore, a closer look into the level of situational awareness gained must be taken by providing more challenging and complex scenarios to work with, while expanding the set of tasks to be evaluated.

References


