2010

Adaptive user interface for vehicle swarm control

Paul Robinette

ADAPTIVE USER
INTERFACE FOR VEHICLE
SWARM CONTROL

by

PAUL M. ROBINETTE

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN COMPUTER ENGINEERING

2010

Approved by

Donald C. Wunsch, Advisor
Randy Hays Moss
G. Dan Waddill

ABSTRACT

An algorithm to automatically generate behaviors for robotic vehicles has been created and tested in a laboratory setting. This system is designed to be applied in situations where a large number of robotic vehicles must be controlled by a single operator. The system learns what behaviors the operator typically issues and offers these behaviors to the operator in future missions.

This algorithm uses the symbolic clustering method Gram-ART to generate these behaviors. Gram-ART has been shown to be successful at clustering such standard symbolic problems as the mushroom dataset and the Unix commands dataset.

The algorithm was tested by having users complete exploration and tracking missions. Users were brought in for two sessions of testing. In the first session, they familiarized themselves with the testing interface and generated training information for Gram-ART. In the second session, the users ran missions with and without the generated behaviors to determine what effect the generated behaviors had on the users' performance.

Through these human tests, missions with generated behaviors enabled are shown to have reduced operator workload over those without. Missions with generated behaviors required fewer button presses than those without while maintaining a similar or greater level of mission success. Users also responded positively in a survey after the second session. Most users' responses indicated that the generated behaviors increased their ability to complete the missions.

ACKNOWLEDGMENTS

First, I would like to thank my wife for supporting me through my thesis project.

I would also like to express gratitude to my advisor, Dr. Wunsch, and the rest of the Applied Computational Intelligence Laboratory for all of their help.

This work would not be possible without the help of the Vehicle Swarm Technology Lab at Boeing. Specifically, I would like to thank Dr. John Vian, Dr. Emad Saad, Dr. Steffan Bienawksi, Paul Pigg, and David Halaas.

Support from the Boeing Corporation, the Missouri University of Science and Technology Chancellor's Fellowship, the National Science Foundation, the Missouri University of Science & Technology Intelligent Systems Center, and the M.K. Finley Missouri Endowment, are gratefully acknowledged.

Finally, I would like to thank my committee members, Dr. Moss and Dr. Waddill, for their help and advice in this process.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

As the number of unmanned and autonomous vehicles in high-stress situations increases, the need for an adaptive interface to model and respond to a user's unique profile increases. Unmanned vehicles are being used at unprecedented levels in military environments. In laboratory tests, greater levels of vehicle autonomy allow the number of operators necessary per vehicle to drop to the point where multiple vehicles can be controlled with only one operator. These tests have involved static interfaces. To dynamically control more vehicles with only one operator, the interface must be able to change to meet the user's needs.

Robots have become more pervasive in many industries. Large numbers of unmanned robotic vehicles are most useful in tasks such as maintenance and reconnaissance. In maintenance tasks, a large number of vehicles work with an operator in the field to identify and repair problems in a large structure. In reconnaissance, the vehicles work together with high-level inputs provided by an operator in a control station. Several aspects of these tasks have already been explored, including the swarming of large numbers of vehicles, the safety of an operator in close proximity to vehicles, and the status of vehicle health. One area that has seen less progress so far is the user interface. While research is ongoing here, there is still more work to be done.

An adaptive user interface will help users to control more vehicles than existing interfaces by taking some of the workload off of the user and placing it on the interface [1]. Langley also reasons that modeling the user in the adaptive interface will produce better results. Machine learning can be used to assist the interface [2]. Such an interface must be designed according to modern human-computer interaction practices [3]. Several components are necessary for a functional adaptive interface. One major component is for the interface to be able to combine simple commands issued by the user into larger behaviors. The user would then be able to issue behaviors instead of commands. This effectively allows the user to issue large strings of commands with the touch of a single button.

Some adaptive user interfaces and components for adaptive user interfaces have been proposed; however, very few have been tested by having humans control robots. This research expands on previous adaptive user interfaces by introducing a new component based on a symbolic data clustering algorithm and then testing that component on human users controlling simulated robots.

2. RELATED WORK

2.1. USER MODELS

Parasuraman, et al. developed a model to describe the various levels of automation available in a system [4]. They define the categories of automation as:

- information acquisition
- information analysis
- decision and action selection
- action implementation.

Information acquisition is defined as the ability of the system to find and present information without requiring any action by the user. Information analysis consists of automatically extrapolating information to predict future events. Decision and action selection ranges from systems that recommend sequences of action to systems that actually execute some basic actions. Action implementation automatically executes whatever actions have been decided by previous levels. Naturally, these categories can overlap, but research on generating vehicle behaviors and presenting these behaviors to the user most closely fits the decision and action selection category.

The cognitive modeling architecture ACT-R has been used to model users for adaptive interface testing [5],[6]. ACT-R is an architecture that combines theories of cognition, visual attention and motor movement that has been successfully used to model humans as they accomplish tasks. ACT-R was used in place of a user to experiment with different components of the interface. This system was found to work for testing purposes using a variety of different user models.

2.2. ADAPTIVE USER INTERFACES

Several adaptive user interfaces have already been proposed. One area that could greatly benefit from automation is the software required to unite a database with an interface. To this end, Jayapandian and Jagadish have worked to automatically generate a form based on the content of a database [7]. This reduces the developer's workload. To generate a form, the target query must first be analyzed. The query is broken into elements relevant to the form such as selection, sort, and join. These elements then are compared to the elements of the other queries to determine the distance between the two queries. This distance metric is used to cluster the queries and determine what type of form should be generated.

Clustering has been used to improve the results of a search interface. In [8], documents were clustered based on their content. When a user's search returned a document it was assumed that other documents of the same cluster should be returned, also. This allowed for the ability to disambiguate similar terms used in different industries. For example, when the user's search string contained “java” the result returned would depend on whether the other search terms were related to software or coffee. This same technology was used in [9] to realize patterns of events. News reports were clustered based on content and used to predict larger events in progress.

One adaptive interface to control simulated robots has been accomplished by focusing on delegation of high-level user commands[10]. This allowed the user to issue high level commands such as “circle defense” or “patrol border” and then the system would automatically follow these commands. This system was shown to reduce operator workload through three experiments where users controlled simulated robots in a game of capture the flag.

An adaptive user interface has also been used to aid in mission planning [11]. In this work, the mission planner was integrated with a wizard to allow for easier creation of new missions. The wizard used previous successful mission information stored in a database to assist users as they created new missions. The wizard was shown to have reduced the total amount of time to create complex missions.

An intelligent file manipulator has been created using the Human Plausible Reasoning (HPR) Theory [12]. HPR describes how humans infer answers to questions by utilizing frequently used reasoning patterns. This work used HPR to predict the user's actions, goals, and possible errors. This system was tested on thirty users and was determined to generate plausible hypotheses about user errors.

An adaptive user interface has also been used to present relevant information to the user [13]. This research used a self organizing map to structure the information. It then determined which information was most relevant and gave it a measure of interest. This work was applied to the hotel industry to allow even users with a low level of computer skills to successfully complete their work.

3. BACKGROUND

3.1. ADAPTIVE RESONANCE THEORY

Adaptive Resonance Theory (ART) unifies top-down and bottom-up clustering methods into one algorithm [14-20].

The basic ART architecture is shown in Figure 3.1. The F1 layer represents the features, while the F2 layer represents the categories. These layers are connected by a series of weights between each node.

First, an input is presented to F1 and activates its corresponding features in F1. Using Equation 1, the degree of match is determined between the input and each node in F2. The match (T) is determined by taking the Fuzzy AND of the input (x) with the weights of the node in F2 (weights are defined as w, the node is defined as j) and normalizing it to the weights. The node with the highest degree of match is determined to be the winner. This winner is then verified using Equation 2.

The Fuzzy AND is again taken between the input and the winning node's weights, but the result is normalized by the input. This is then compared to the vigilance (ρ) parameter. If the result is greater than or equal to the vigilance value, then the matched node is determined to accurately represent the input. If not, then that node is marked as incorrect, and the process starts again. The next highest matching node is then the winner and must be compared to the vigilance parameter. If each matching node fails the vigilance test, then a new node is created using the input as its weights.

$$T(j)=\frac{|x \wedge w^j|}{|w^j|} \tag{1}$$

$$\frac{|x \wedge w^j|}{|x|} \geq \rho \tag{2}$$

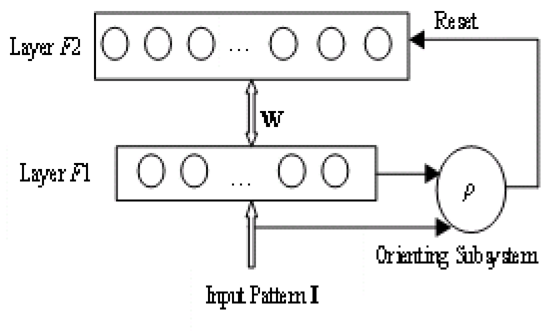

Figure 3.1: Adaptive Resonance Theory

3.2. GRAM-ART

Gram-ART is a variant of ART designed to cluster variable-length input patterns represented by trees or sequences. Typically, Gram-ART is implemented to cluster trees that represent information represented by Backus Naur Form; however, it can also be used to cluster any set of sequences composed of symbolic data [21],[22].

The magnitude function typically used in ART variants has no meaning in trees because the adjacency of symbols has no relevance to their values. For example, the numbers 1.1 and 1.2 can be considered close to each other; however, the letters A and B are not necessarily close to each other despite their adjacency in the alphabet. In Gram-ART, the magnitude is defined as the number of nodes in the tree.

Like the magnitude function, the Fuzzy AND operator does not apply to trees, so the trace of the input in the weight is used to define the intersection of the input and the category tree. The trace is the sum of the values stored in a given weight corresponding to the symbols in a given input (Equation 3).

$$|x \wedge w^j| = \sum_{m}^{i=0} w^j_{i,x_i} \tag{3}$$

Prototype trees, which can be any length, are formed based on the input data. Each node is a superposition of all matching nodes for that position of the tree. Figure 3.2

shows the formation of a prototype tree. When input A is matched with prototype P, P is initialized as a clone of A. When input B is then matched with P, the prototype changes to reflect the possibility of other symbols in the nodes.

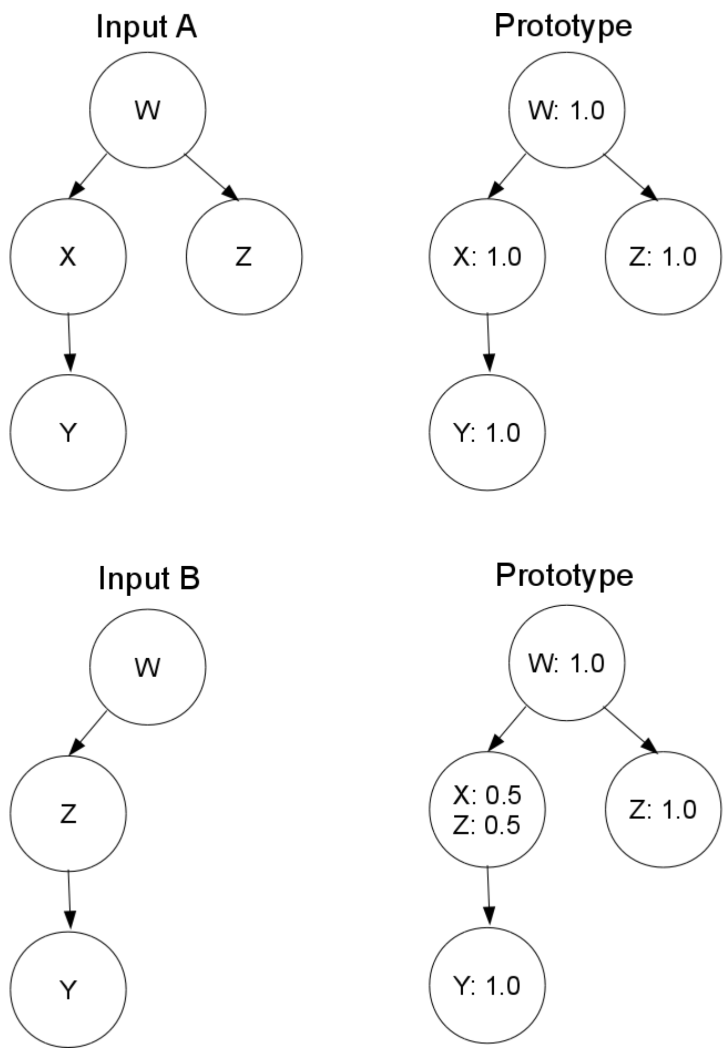

Figure 3.2: Gram-ART Creation of Prototype Tree
With Two Inputs

Initially, Gram-ART was tested against the Fisher Iris dataset to benchmark it against K-Means and Fuzzy-ART. The Iris dataset was converted into symbols for this process. Gram-ART performed better than Fuzzy-ART and as well as or better than K-Means for this test. Gram-ART could not beat K-Means in all cases because K-Means could be manually tuned for the exact number of clusters in the sample data.

Next, Gram-ART was used to cluster a symbolic dataset. Gram-ART successfully categorized every input in the Mushroom Dataset as poisonous or edible using only 24 clusters, compared to 913 for Fuzzy-ART.

The Unix User Dataset was then tested with Gram-ART. The entire dataset was inputted so that Gram-ART could learn which users typically issued which strings of commands. Gram-ART achieved a 96.5% success rate at this, compared to the previous record of 83.8%.

For more information on Gram-ART see Meuth, et al., see [21],[22].

4. METHODOLOGY

4.1. GRAPHICAL INTERFACE

SwarmSim (Figure 4.1) is the graphical interface used to control the simulated vehicles, which are displayed in the center of the application from an overhead view. Buttons to initiate commands are located on the toolbar above the vehicle display. The top toolbar is for built-in commands that are available in each mission. The toolbar next to that contains commands that will be available in half of the missions in the second session. These commands are combinations of built-in commands generated from the user's first session.

To send a command to a vehicle, the user must first select the vehicle and then press the button corresponding to the desired command.

Four basic commands are available to the user in all missions:

- Start Controller - Labeled "R1." This command must be issued to a vehicle before any other commands can be issued. It will turn the vehicle on and prepare it for the mission.
- Takeoff - A green arrow pointing up. This command will cause the vehicle to lift off the ground and hover in place.
- Waypoint - An orange circle with a targeting reticule in black. The vehicle must have taken off already before it can go to a waypoint. Once the button is pressed, SwarmSim will expect the user to click where the vehicle should go next.
- Land in place - A red arrow pointing down. This will cause the vehicle to land directly below its current location.

The vehicle can have four states visible to the user: off, standby, ready for commands and crashed. The off state is indicated by no status bar underneath the vehicle and the vehicle's name in red. The standby state is indicated by a status bar showing state of charge beneath the vehicle's icon and the vehicle's name in yellow. The ready state is indicated by the status bar underneath the vehicle's icon and the vehicle's name in green. A crash is indicated by a red 'X' over the vehicle's icon.

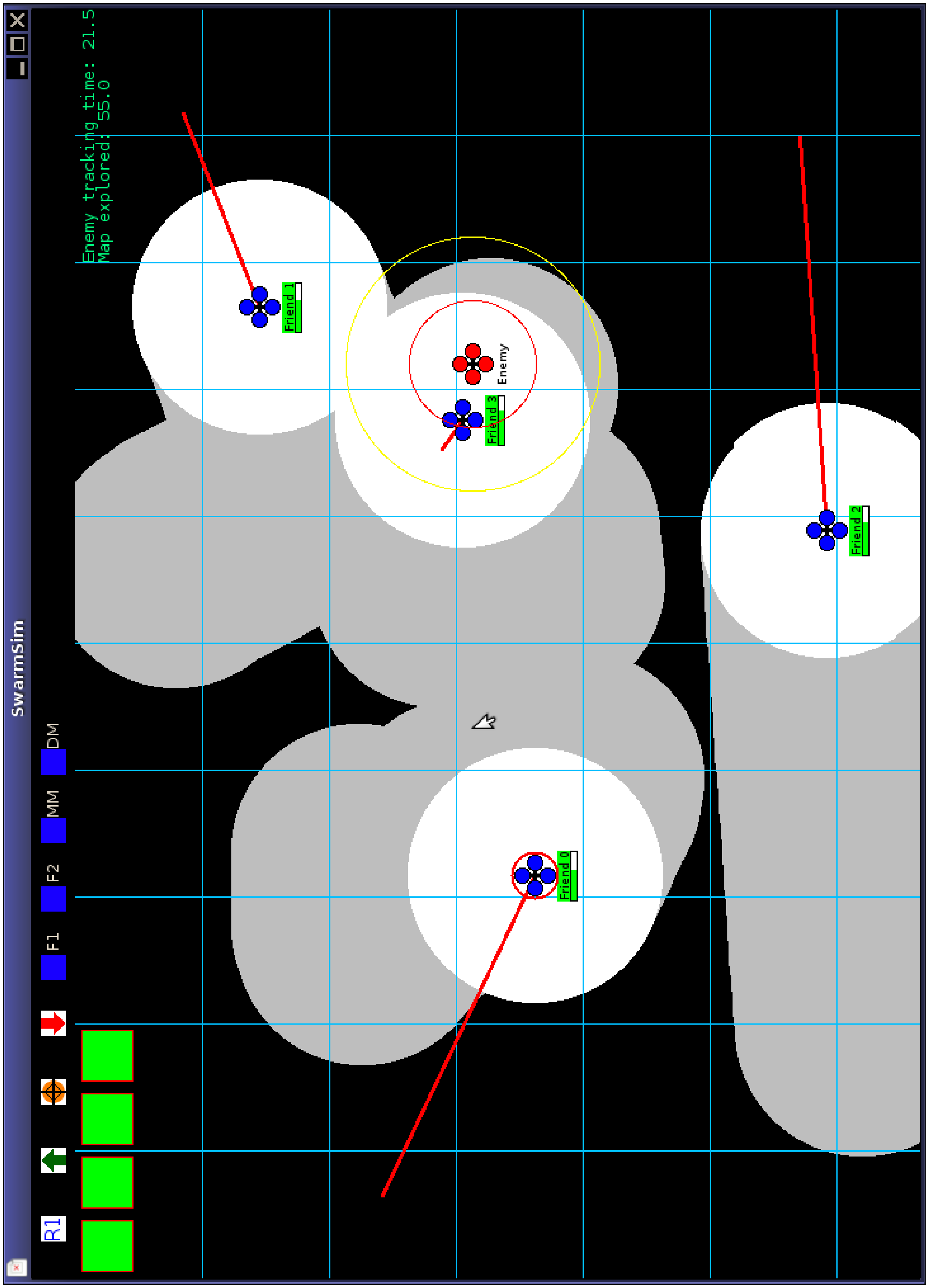

Figure 4.1: SwarmSim

The vehicle's state of charge (the amount of charge left in the battery, proportional to the amount of time the vehicle has left to fly) is shown in a bar beneath its image. When the battery is fully charged, the bar will extend the entire width of the vehicle's icon and be green in color. The bar will shrink to the left as charge is used. When the vehicle is running low on battery, the bar will turn yellow, and when the battery is critical, the bar will turn red. When the battery level is critical, the vehicle will automatically return to base to refuel. Once the vehicle is finished refueling, it will enter the standby state again. The battery holds 200 seconds of charge and takes twenty seconds to recharge fully. While this ratio of flight to charge is not typical of actual robotic vehicles, it was necessary to allow the vehicles to run down their charge, refuel and return to action in the same mission.

Areas that the vehicles are currently able to see are shown to the user as white circles surrounding the vehicles. Areas that recently have been seen by a vehicle are shown in gray. A track of the last twenty seconds is considered recent. All unexplored or not recently seen areas of the map are shown in black. The percentage of the map marked as recently seen is presented to the user in the top right corner.

The enemy vehicle initially is hidden. When the user flies a vehicle within 100 units of distance (these units are mapped as pixels on the screen but are otherwise arbitrary) from the enemy, the enemy becomes visible. Two circles appear around the enemy, a red one 50 units from the center of the enemy and a yellow one 100 units from the enemy. When all vehicles leave the yellow circle, the enemy disappears until a vehicle enters the circle again. The simulator keeps track of the time spent inside the inner and outer circles; however, the user is presented only with the time spent inside the outer circle in the top right corner of the interface.

4.2. BEHAVIOR GENERATION

Behavior generation is performed in three steps. First, the waypoints are clustered to reduce the raw number of symbols sent to Gram-ART. Second, the issued commands are turned into symbols and strung into sequences to be fed to Gram-ART. Finally, the sequences are presented to Gram-ART. Figure 4.2 shows this architecture.

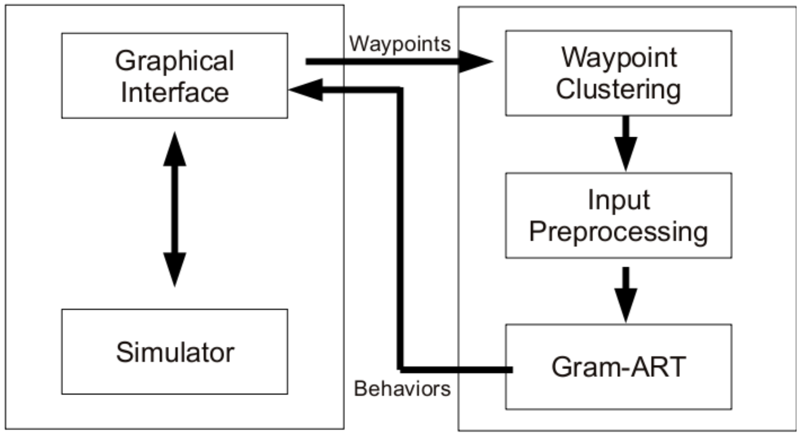

Figure 4.2: Architecture of Behavior Generation

4.2.1. Waypoint Clustering Waypoints given to friendly vehicles need to be translated into symbols in order to work as Gram-ART input. To accomplish this, waypoints are clustered using ART. Initially, only the starting positions of the vehicles are used as clusters. Waypoints are presented to the clustering program and matched with their nearest centroid. If the point is more than 100 units from the closest centroid, then a new centroid is formed with this point at its center. Centroids are revised each time a point is added so that they represent the average of the points that belong in the centroid. The process repeats until points stop switching clusters.

Another system is used to capture points relative to the enemy vehicle. It is assumed that if the user placed a waypoint inside of the outer ring around the enemy, then the user meant to have the vehicle move in relation to the enemy. False positives found at this stage are eliminated as outliers when Gram-ART runs. Waypoints within the outer ring of the enemy are classified into four symbols: front, left, right and back. These directions are determined based on the heading of the enemy at the time the waypoint was generated.

4.2.2. Gram-ART Input Sequences A set of symbols (Table 4.1) is defined for each user. These sets are identical except for the waypoint coordinates, which are outputs from the ART clustering program. The logs from the first testing session are converted into these symbols so that Gram-ART can find common sequences. Sequence lengths can vary depending on their purpose. Each sequence is limited to commands issued to a single vehicle.

Table 4.1: Input Symbols

Symbol	Description
off	Vehicle has transitioned to off state
standby	Vehicle is powered on but still on the ground
ready	Vehicle is airborne, waiting for commands
crashed	Vehicle has crashed
return	Vehicle is on reserve fuel and returning to charging station
discovered	Enemy is in visual range
lost	Enemy is now out of visual range
(x,y)	Waypoint coordinate (where x and y are variables)
e0	Enemy front
e1	Enemy right
e2	Enemy back
e3	Enemy left

The first set of sequences, intended to catch common starting procedures, is triggered by the standby symbol (indicating that a vehicle is entering the standby state) and continues for the next four commands given to that vehicle. This is intended to catch sequences that start with standby, move to ready and finally issue three waypoint commands. This produces sequences used at the beginning of the mission as well as sequences used after the vehicle refuels.

The second set of sequences captures the waypoints given in relation to the enemy. These are restricted to three commands in length. The enemy-relative sequence is intended to allow complex following behavior, where a user may wish to orbit around the enemy in some way. The length of this sequence was determined after testing several lengths.

The final set of sequences is intended to find search patterns. The waypoint commands for each vehicle are broken into sequences in three different passes. On the first pass, the sequences are three commands long, on the second the length is four and on the third the length is five. This gives many opportunities for Gram-ART to find common sequences.

4.2.3. Gram-ART Gram-ART is run on the input sequences using a vigilance value of 0.6, a value determined after several tests. It allowed for some flexibility in command sequences, such as an option to have an alternate value for a spot in a sequence.

The output from Gram-ART includes all templates, including those that only matched one input value. A post-processing check is run to eliminate all templates that did not match at least four inputs. This allows the user to choose between popular vehicle behaviors.

4.3. EXPERIMENTAL SETUP

The missions flown by the user all had the same basic structure. First, the user was asked to start and takeoff as many vehicles as he felt were required. A target vehicle was hidden in the unexplored area.

The primary goal of each mission was to find and follow the target vehicle. When the target was visible, two circles were displayed around it. Ideally, the user placed a vehicle inside the inner circle. If this was not possible, the user was instructed to keep the vehicle within the outer circle. The target vehicle moved continuously, so the user had to follow it effectively and find it after the controllable vehicles returned from refueling. The target vehicle changed directions randomly at random intervals. A clock in the upper right hand corner displayed the total amount of time the user was able to keep a controllable vehicle inside the outer circle.

The secondary goal of each mission was to explore as much of the area as possible. The user was instructed to attempt this objective only if the primary goal was already in progress.

In the first session, the user familiarized himself with the interface and ran simple missions to train the system. There were four basic types of missions in this session:

Basic - The user was given four vehicles and instructed to follow the primary and secondary objectives stated above to the best of his abilities. The mission ended at five minutes. A single, random vehicle in ready state was crashed at a random time in the middle four minutes of the mission.

Explore - The user was given four vehicles and instructed to focus on finding the target vehicle. The mission ended as soon as the target vehicle was discovered. No vehicles were allowed to crash.

Crash - The user was given four vehicles and instructed to proceed as with a normal mission. A random vehicle in ready state was crashed at thirty seconds into the mission. The mission continued for another thirty seconds to record the user's response to this event.

Track - The user was given four vehicles and instructed to follow the target vehicle as closely as possible. The entire map was marked as explored, so the target vehicle was visible for the entire mission. A random vehicle in ready state was randomly crashed at a random time in the mission. The mission ended after two minutes.

The first session started with two basic missions to familiarize the user with the interface. Logs were generated for these missions. The user then completed five explore missions to determine what commands the user typically issued at the start of a basic mission. Next, the user was presented with five crash missions. These were intended to discover typical reactions to a vehicle's crash; however, they also acted as short basic missions. This gave the system considerably more data about user actions during critical parts of the mission, such as takeoff, target discovery and the start of secondary exploration. After the crash session, the user was presented with the track mission three times. This allowed the system to learn typical commands that the user would issue to follow the target vehicle. Finally, the user was presented with the basic mission again in an attempt to learn any new command sequences that the user had devised.

The second session took place two weeks after the first session. Each user was presented with six total missions, all variations on the basic mission. At the beginning of the session, the user was presented with the command sequences discovered by Gram-ART and was asked to identify which of these he would like to use. He then was asked to name each command in order to make it more meaningful to him than if the name had been created by the investigator. After naming the sequence, the user indicated if he would like this sequence to loop. The investigator explained to the user that there was no harm in looping a command because he could break the loop by issuing another command at any time.

The first two missions were basic missions with the addition of adaptive commands generated by Gram-ART. This was the user's first exposure to the adaptive commands, so the log files were not used. The next four missions were presented in random order. Their variations are listed below. These four missions allowed data to be collected about two variables: number friendly of vehicles and generated sequences.

A: Single Vehicle, Non-Adaptive

B: Four Vehicles, Non-Adaptive

C: Single Vehicle, Adaptive

D: Four Vehicles, Adaptive

After completing the missions, the user was asked to take a survey to report his observations about the system. The survey had fourteen questions:

1. Rank the following missions in order of difficulty to complete where 1 is easiest and 4 is hardest. [Missions A,B,C,D listed]
2. For each mission, what was the difficulty in accomplishing the primary goal?
3. For each mission, what was the difficulty in accomplishing the secondary goal?
4. How many vehicles do you feel would be optimal for the missions where only built-in commands are provided? The optimal number should be the most that you feel you can control effectively.
5. How many vehicles do you feel would be optimal for the missions where generated commands are provided? The optimal number should be the most that you feel you can control effectively.
6. How many of the generated commands were useful?
7. How often do you play video games where timing and/or hand eye coordination are useful?

8. How much do you enjoy playing video games?
9. How much did you enjoy using this simulator?
10. How old are you?
11. What generated commands did you find the most useful and why?
12. What generated commands did you find the least useful and why?
13. What commands should have been generated but weren't?
14. Please take the remaining space to list any comments you would like the experimenter to know.

Users were solicited from the university Robotics Team and the researcher's lab. Seven users volunteered for the testing. All users had their first session on the same day and their second session two weeks later. All users used the same computer for both testing sessions.

5. RESULTS

5.1. EARLY WORK

Several components of this project were tried before the main thrust of the work began. Initially, the behaviors were generated using a Markov Model. Also, an information filter was attempted using a Bayesian classifier.

5.1.1. Macro Generator A prototype of the behavior generator was written using a Markov Model instead of Gram-ART to find patterns. This version took in real data from a demonstration of several robotic vehicles and outputted strings of commands based on the command statistics. Table 5.1 shows the statistics determined by the macro generator. The percentage is the probability that the command in that column will follow the command in that row. This produced command strings such as activate->standby->takeoff->waypoint. While this approach did work moderately well, Gram-ART was found to be more applicable to this problem.

Table 5.1: Initial Macro Generator Results

Command	activate	standby	takeoff	waypoint	land
activate	0.0%	100.0%	0.0%	0.0%	0.0%
standby	0.0%	0.0%	100.0%	0.0%	0.0%
takeoff	0.0%	0.0%	0.0%	100.0%	0.0%
waypoint	1.4%	0.0%	0.0%	96.8%	1.4%
land	0.0%	0.0%	0.0%	0.0%	0.0%

5.1.2. Information Filter A typical robotic swarm sends many messages each second from vehicles in the swarm to other vehicles and the user. If the user were to view all of the messages for a several-hundred-vehicle swarm he would quickly become overwhelmed. The information must be filtered such that the users see critical messages but ignore information that will not affect their next command decision. To solve this, a prototype interface was developed that initially displayed all messages that units send, and then allowed the user to rate the message as useful or not. The interface remembers the ratings and displays relevant information for the current state of the swarm. The decision to display the information is made by using a Bayesian classifier trained on the rated responses. An example of the message filter is shown in Figure 5.1. This is only a prototype interface, so the commands are still denoted by their numerical identification instead of a more readable string format. The "Mod" column represents the users response to this information. A mod of 1 indicates that the user is interested in similar information while -1 indicates he is not interested. The interface takes this information and then interpolates which messages from which vehicles should be displayed. This interface was not tested along with the behavior generation, but will be added to the next version.

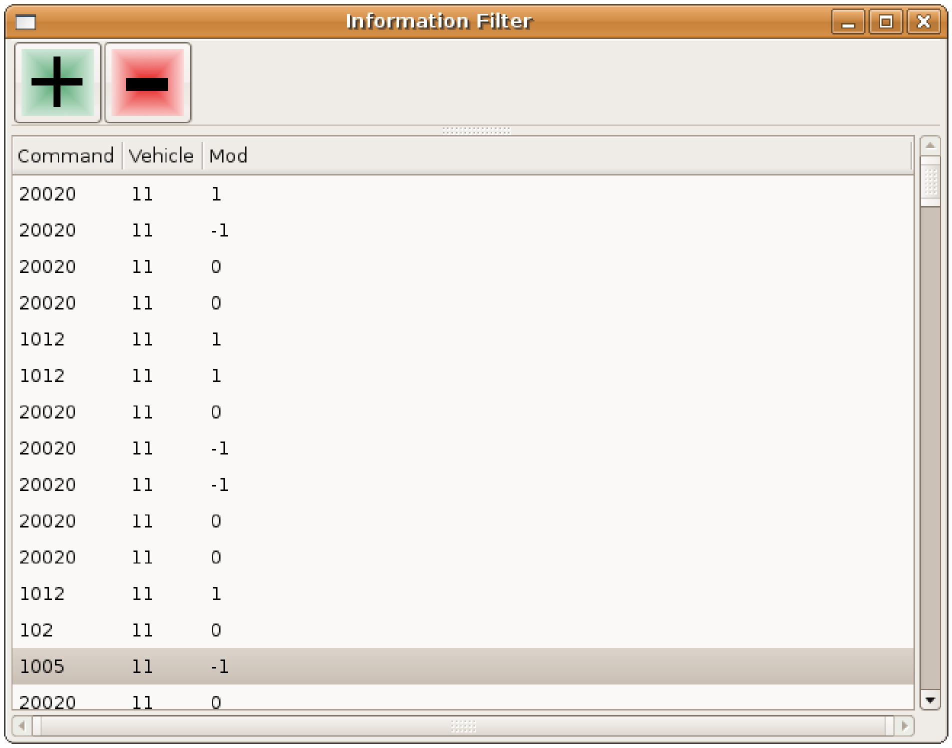

Figure 5.1: Information Filter

5.2. FIRST SESSION

The beginning of the first session was intended to familiarize the user with the interface. Measuring the secondary objective can provide a good measure of the user's comfort. Good performance in the secondary objective implies that the user is able to accomplish the primary objective as well. Figure 5.2 and Figure 5.3 show the differences in the secondary objective (exploration) for one user over the course of the first and second missions, respectively. This is given as a typical example to illustrate how a user improves at the beginning of the session. Note that the initial time before the user deploys the vehicles in the beginning is halved by the second mission. Note also that the trough that happens after the first peak is much more shallow, and recovery time after dips is lower.

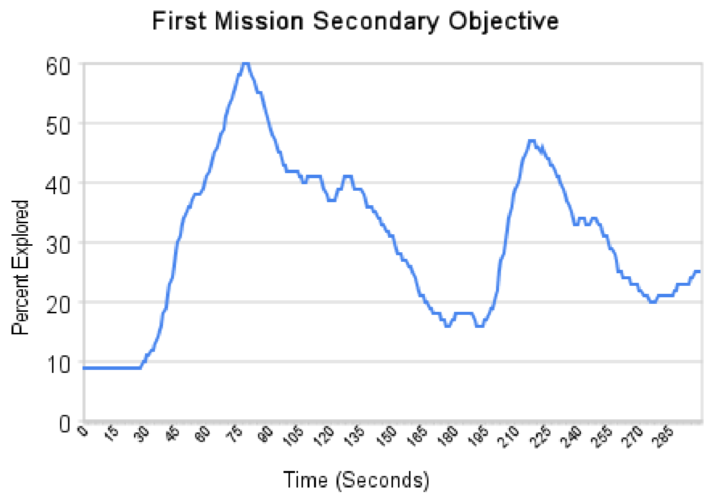

Figure 5.2: First Session, First Mission, Secondary Objective

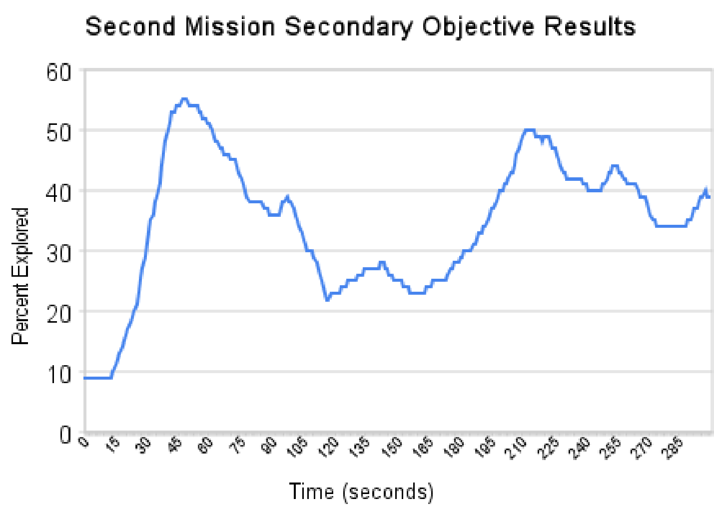

Figure 5.3: First Session, Second Mission, Secondary Objective

Users employed many different strategies during the first session. Some users intentionally tried to follow similar search patterns at the beginning of each mission. Other users assigned multiple vehicles to follow the enemy when possible. This did not increase their score but did inadvertently reinforce the follow behavior when Gram-ART was run.

After the first session, Gram-ART generated behaviors for each user. One example of these behaviors is shown in Table 5.2. The behaviors are between three and five waypoints long. The coordinate pairs are in reference to the area the vehicles fly in, where (0,0) is the top left and (1000,700) is the bottom right. In some cases, Gram-ART identified multiple waypoints with equal likelihood for that position in the sequence. This is noted by giving all waypoints separated by the word “OR” The words “Enemy Front” refer to the position directly in front of the enemy's direction of travel. See Figure 5.4 for a diagram of the coordinates on the graphical interface.

Table 5.2: Example Behaviors Generated by Gram-ART

Waypoints				
First	Second	Third	Fourth	Fifth
(128,524) OR (325,45)	(332,598)	(773,278)	(903,256)	(773,278)
(332,598) OR (194,582)	(326,468)	(440,340)	(542,240)	(644,196) OR (429,223)
(921,147)	(877,390)	(707,354)		
Enemy Front	Enemy Front	Enemy Front		
(542,240)	(303,238)	(542,240)		
(332,598)	(707,354)	(573,75)	(542,240)	(760,48)
(128,524)	(710,550)	(573,75)	(542,240)	(760,48)
(326,468)	(544,568)	(869,534)	(812,162)	(707,354)
(627,468) OR (544,568)	(627,468)	(707,354)	(644,196)	(573,75)

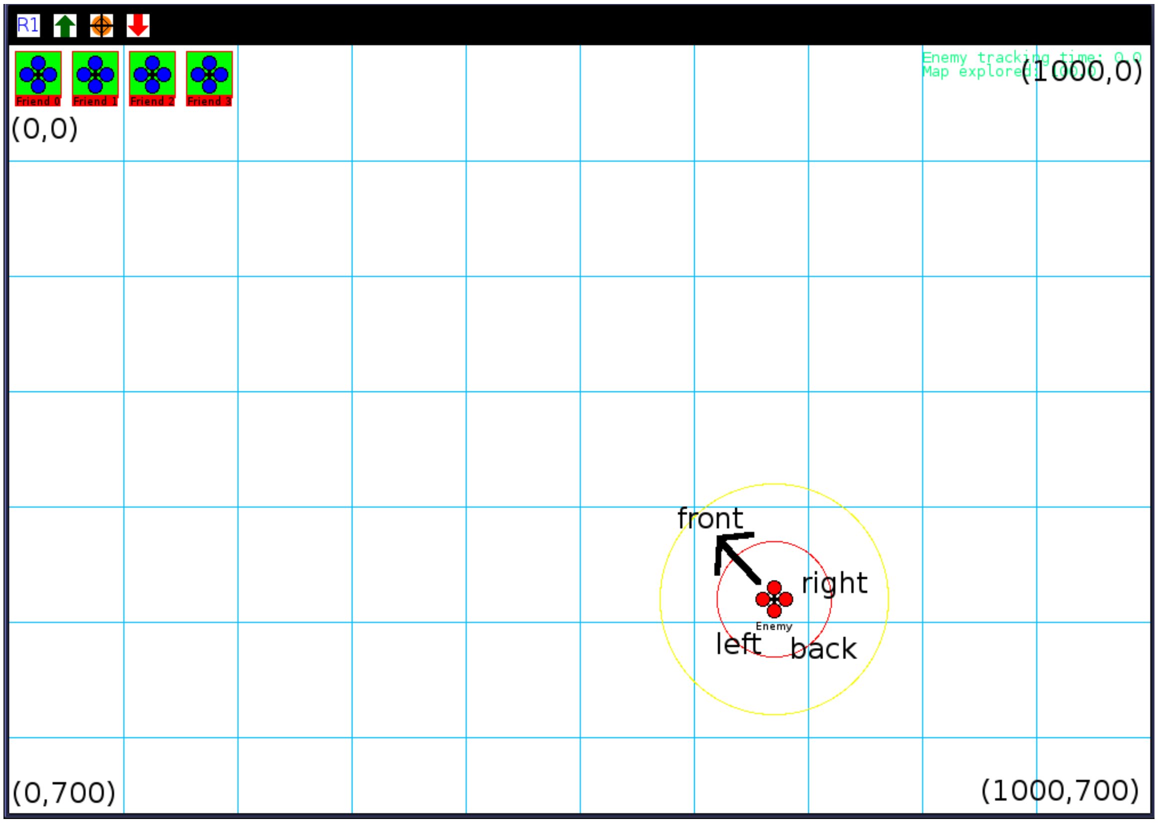

Figure 5.4: Coordinates on Graphical Interface

5.3. SECOND SESSION

The results from the second session can be broken into four categories: primary objective accomplishments, secondary objective accomplishments, number of button presses required and survey responses.

5.3.1. Primary Objective Users were given the primary objective of tracking the enemy. Ideally, the user would track the enemy within the inner circle. Figure 5.5 shows the results of tracking the enemy inside the inner circle during the four test missions in the second session of testing. Figure 5.6 shows the results of tracking the enemy inside the outer circle during the same missions. Performance gains were seen when the user had generated behaviors available with four vehicles; however, this hindered the progress of most users when only one vehicle was available. In most cases, the users required considerably more time to find the enemy during Mission C, and they verbally complained that the generated behaviors were not useful when applied to just one vehicle. This is most likely because the generated behaviors were trained using four vehicles. Mission A had an average inner track time of 109.8 seconds, Mission B had 151.8 seconds, Mission C had 51.1 seconds and Mission D had 209.0 seconds.

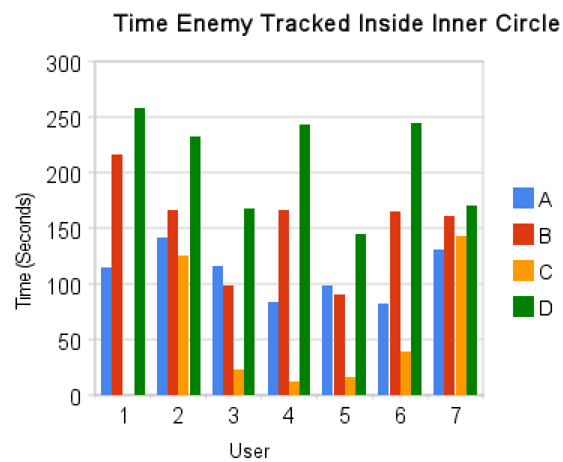

Figure 5.5: Time Enemy Tracked Inside Inner Circle

Missions B and D fared approximately equally in time tracking the enemy inside the outer circle. This shows that the user was able to track the enemy with the same degree of precision when using the generated commands as when manually controlling all vehicles. For the outer circle, Mission A had an average track time of 150.8 seconds, Mission B had 219.8 seconds, Mission C had 81.5 seconds and Mission D had 239.1 seconds.

Users developed some interesting strategies to track the enemy. Most users developed a simple strategy in which they allocated one vehicle to track the enemy and tasked the remaining vehicles to explore. Some users landed one of the exploration vehicles to conserve fuel so that a vehicle was available for tracking when the other three returned to refuel. One user even landed a vehicle inside the enemy's inner circle and had it takeoff again whenever the enemy moved. During a real mission this would actually take more fuel to accomplish, but the simulation did not account for the extra fuel required to takeoff, so this user managed to gain several seconds more track time.

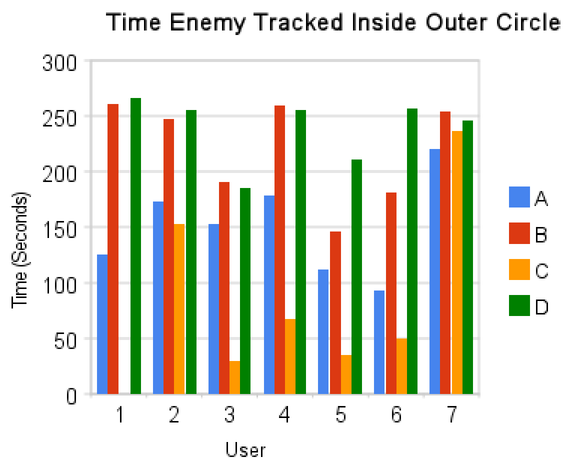

Figure 5.6: Time Enemy Tracked Inside Outer Circle

5.3.2. Secondary Objective Figure 5.7 shows the average explored area for each user and each mission. The average explored area for Mission A was 12.1%, B was 13.0%, C was 41.9% and D was 44.8%. This shows improvement, though slight, for missions with generated behaviors over missions without. Naturally, missions that allowed the use of four vehicles had considerably better results than those with just one.

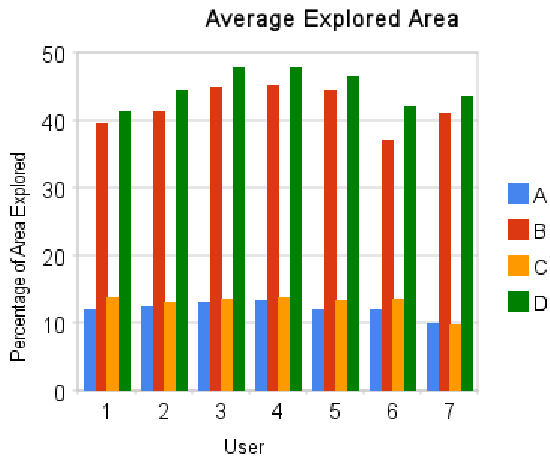

Figure 5.7: Average Explored Area

5.3.3. Button Press Frequency The best empirical evidence of workload is shown through the number of buttons a user is required to press to accomplish a mission. Figure 5.8 shows the number of buttons each user pressed during a mission. It can be seen that, among all users, Mission B required considerably more button presses than Mission D. All but one user required more button presses for Mission A than for Mission C. This user micro-managed his vehicle in Mission C and did not make extensive use of the generated behaviors for this mission. The average number of button presses required for Mission A was 48.1, B was 72.7, C was 26.9, and D was 43.4.

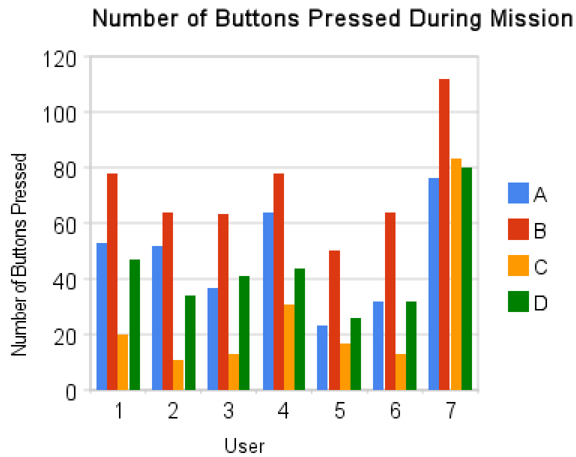

Figure 5.8: Number of Buttons Pressed During Missions

5.3.4. Survey Results After completing the second session, the users were asked to take a survey. The first question asked the user to rank the missions in order of difficulty. These results are shown in Figure 5.9. Most users ranked Mission A as the hardest (average of 3.4) and Mission D as the easiest (average of 1.7). Averages were calculated by setting the most difficult ranking as 4 and the least as 1. This was the expected result because Mission A allowed only one vehicle and did not make use of any generated behaviors, while Mission D allowed more vehicles with generated behaviors, allowing the two objectives to be more easily be accomplished. Mission B received an average ranking of 2.7, while Mission C received an average of 2.3. One user ranked Mission B as the most difficult because he felt it more tasking to control more vehicles. Additionally, one user ranked Mission D as the most difficult. He later ranked all missions as fairly easy, but he gave no indication as to why Mission D would be the hardest.

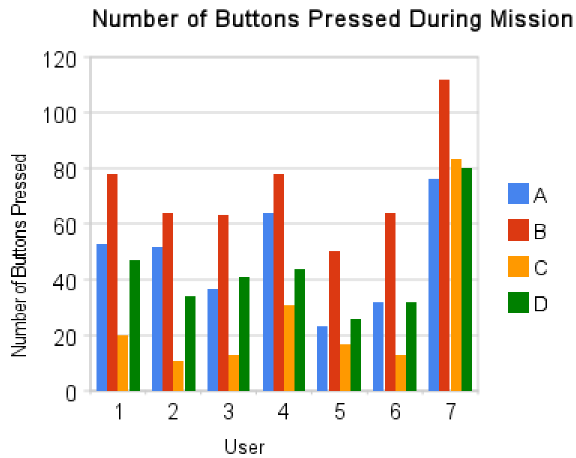

Figure 5.9: Difficulty Rankings of Missions

The second question asked the user to give each mission a difficulty rating for the primary objective (enemy tracking). Users were given the difficulty options of easy (1), moderately easy (2), moderate (3), moderately hard (4), and hard (5). All but one user rated Mission D as easy. Most rated the missions with only one vehicle available (A and C) as the most difficult; however, they disagreed about how difficult these missions were. Mission A received an average rating of 3.0, Mission B received an average of 2.3, Mission C received an average of 2.4 and Mission D received an average of 1.1. These results can be seen in Figure 5.10.

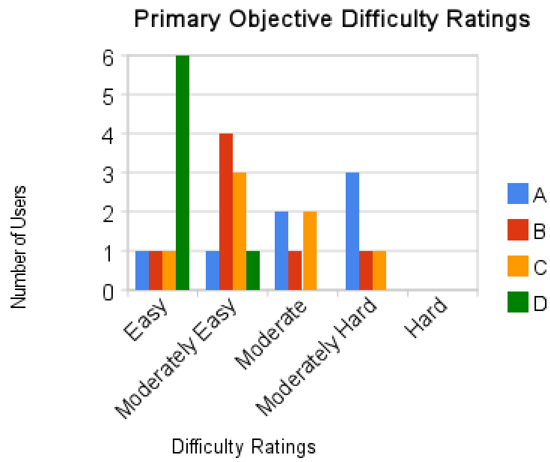

Figure 5.10: Primary Objective Difficulty Ratings

Figure 5.11 shows how the users rated each mission based on the difficulty of the secondary objective (exploration). The users were given the same difficulty scale as in question two. Ratings were much more scattered for this objective, which was expected because some users will have more difficulty accomplishing two objectives at once than other users. Mission A received an average rating of 3.0, Mission B received an average of 3.1, Mission C received an average of 2.1 and Mission D received an average of 1.7.

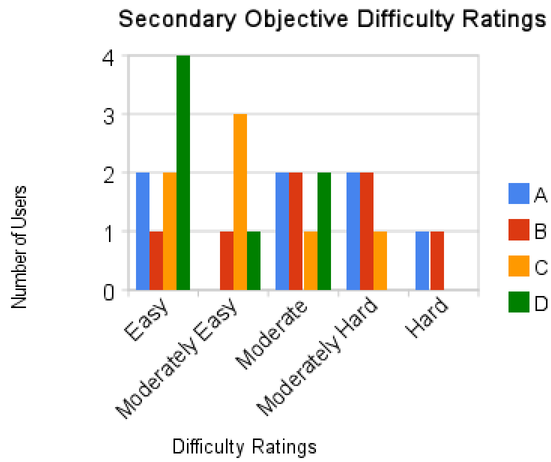

Figure 5.11: Secondary Objective Difficulty Ratings

Next, the users were asked how many vehicles they felt were ideal to accomplish the basic mission with and without the generated commands (Figure 5.12). The average number of vehicles chosen for the built-in commands was 4.3, while the average for using the generated commands was 6.4.

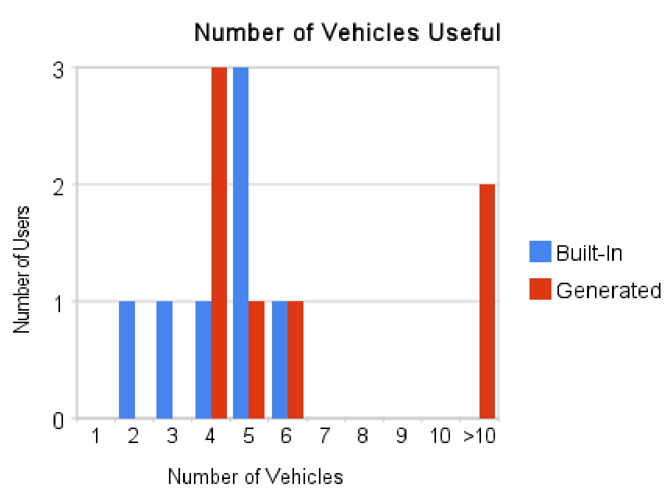

Figure 5.12: Number of Vehicles Useful

Users were also asked how useful they found the commands (Figure 5.13). The options given to the user were: none of the generated commands were useful (denoted as 1), a few of the generated commands were useful (2), some of the generated commands were useful (3), most of the generated commands were useful (4), and all of the generated commands were useful (5). The average response was 3.3, which would place it between the some and most categories.

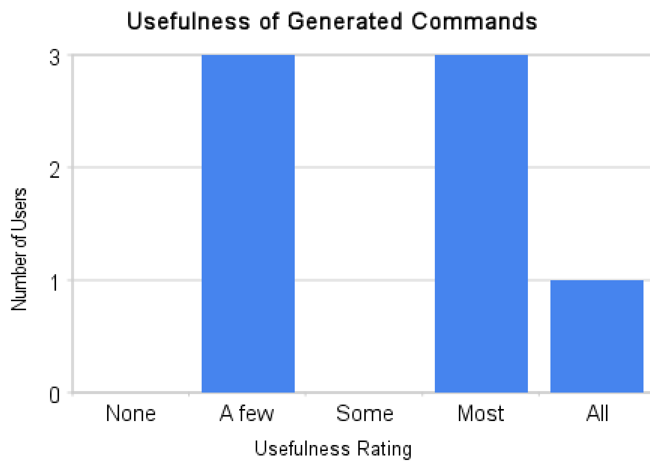

Figure 5.13: User Ratings of Usefuless of Generated Commands

Users were asked how often they played video games in order to judge their experience with programs like the simulator (Figure 5.14). Four users claimed to play video games multiple times per week but less than daily. The other users were split between daily, once weekly and multiple times per month. All users stated that they played video games regularly.

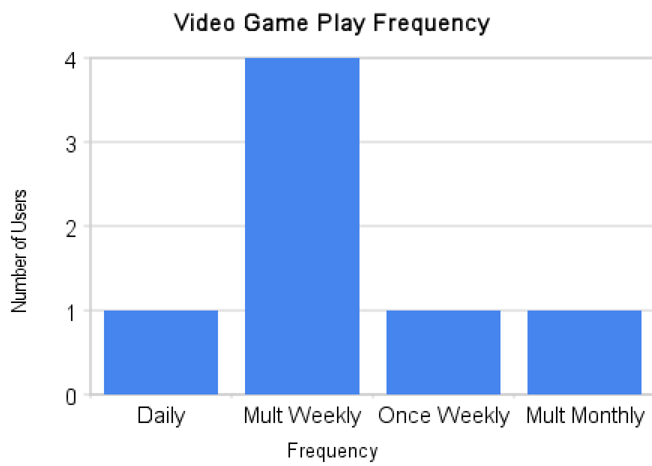

Figure 5.14: Video Game Play Frequency

Users' responses to the questions about video game enjoyment and simulator enjoyment are shown in Figure 5.15. Given the age group, it is not surprising that most users claim to enjoy video games very much and the rest enjoy them sometimes. Also not surprisingly, more users claimed to enjoy professionally made video games over the simulator created in a research lab. All users still claimed to at least somewhat enjoy the simulator experience.

Figure 5.15: Video Game and Simulator Enjoyment

Users fell into two age categories (Figure 5.16). Five listed themselves in the 18-24 group and two in the 25-30 group. Given that the users were all undergraduate and graduate students, these results were expected.

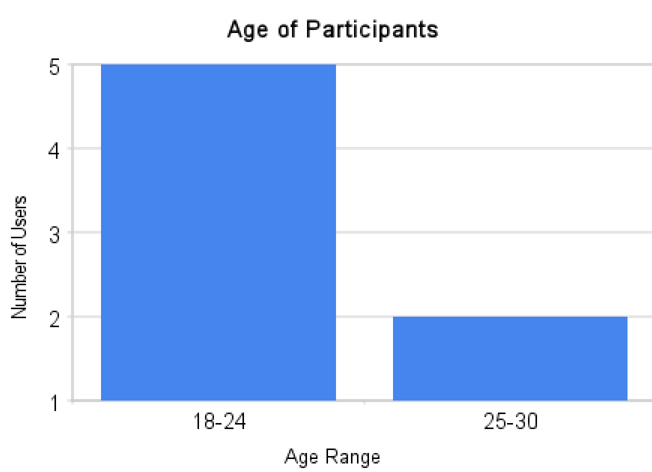

Figure 5.16: Age of Participants

The final section of the survey asked users to describe their opinion of certain aspects of the mission. The first of these questions asked the users which generated behaviors they found most useful. All users responded that enemy tracking behaviors were useful. Several commented that this generated behavior allowed them to switch focus to more easily accomplish the secondary task. One user commented that this generated behavior allowed him to press fewer buttons. Another user commented that this command made it easier for him to follow his strategy of landing the tracking vehicle at strategic points to conserve fuel.

The next short-answer question asked the users which generated behaviors were the least useful. Two users responded that some of the commands were not useful because they covered a very small area. One user noted that two commands generated for him were identical when looped. One was that the enemy was tracked in front, then to its left, then front. The other was that the enemy was tracked in front, then in front again, then to its left. One user responded that he only found the tracking commands useful. Another user stated that the generated commands were not useful initially but that they became useful once he spread the vehicles out. One user responded that most of the generated commands were not useful. Two users claimed that all of the generated commands were useful.

The third short-answer question asked the users which behaviors should have been generated. Five users suggested improvements, including a circle the perimeter behavior, a keep distance from neighboring friendly vehicles behavior, an automated land-wait-continue behavior, a takeoff sequence and a takeoff all vehicles behavior. The other two users indicated that all necessary behaviors were generated.

The final question asked the users for any additional comments about the experiment. Users made several good suggestions for future versions of the simulator and for the adaptive interface. Several users suggested adding more information about the generated behaviors and vehicle trajectories to the interface. One even suggested that hovering over a button should display the waypoints for that behavior in the main mission area. Users also suggested that hotkeys be added to allow future users an easier interface. Hotkeys were intentionally left out of this version of the interface to force the users to go through the same process (clicking a button on a toolbar) for every command. Hotkeys also would have increased the time the users needed to learn the interface. Additionally, it was suggested that users be allowed to queue waypoints. One user gave some comments on the survey questions themselves. He pointed to ambiguity in the first few questions that ask the user to chose whether to judge difficulty as accomplishing objectives or performing to maximum potential. Two users commented here that using the generated behaviors reduced their workload.

6. DISCUSSION

Statistical analysis confirms that the four vehicle missions for which users had generated commands available had performance levels as good as or better than those without. For the primary objective, the mean time to track in the inner circle was 151.8 seconds for Mission B and 209.0 seconds for Mission D. After performing a t-test between these two datasets, Mission D was shown to perform statistically better than Mission B with a 98% significance level. This significance level was reduced to 81% for tracking inside the outer circle.

These tests showed exactly the opposite for the primary objective of the one-vehicle missions. Mission A (average of 109.8 seconds) performed better than Mission C (average of 51.1 seconds) at a 99% significance level for tracking inside the inner circle. This level was only reduced to 96% when tracking inside the outer circle. These results are most likely because the generated behaviors were trained for four vehicles instead of for one.

For the secondary objective, Mission D (average 44.8% explored) performed better than Mission B (average 41.9% explored) in a t-test with a significance level of 96%. Mission C (average 13.0% explored) only performed better than Mission A (average 12.1% explored) in a t-test with a significance level of 88%.

The most important statistic recorded for each mission was the number of buttons a user pressed. For this, Mission D (average of 43.4 buttons pressed) was statistically less than Mission B (average of 72.7 buttons pressed) in a t-test with a significance level of 99%. Likewise, Mission C (average of 26.9 buttons pressed) performed better than Mission A (average of 48.1 buttons pressed) in a t-test with a significance of 95%.

There are some important assumptions to note with using t-tests for this data. One is that each data set is normal and has approximately the same variance. Another is that the samples are randomly selected. The first assumption is fairly accurate with this data; however, the samples were not randomly selected due to the very small number of total samples.

7. CONCLUSIONS

With few exceptions, users were able to better meet mission objectives with generated behaviors than without. While in some cases these improvements were small, the users were able to accomplish this level of fitness with considerably less interaction with the graphical interface. Most users also responded positively to these generated behaviors in the survey.

There is certainly more work that can be done on this topic. The graphical interface can be modified to add indications for which button performs which generated behaviors. Before this interface is deployed in any real-world operations, hotkeys should be added to allow the user to run behaviors at the touch of a single key.

The preprocessing could also be modified to allow for generation of behaviors that affect multiple vehicles. This would allow more of a swarming effect, and could even develop emergent behavior that would perform complex operations such as detecting when a vehicle is incapacitated and tasking another vehicle to finish the first vehicle's task.

APPENDIX A.

RAW SURVEY RESPONSES

(transcribed by author)

User ID: 1

Adaptive User Interface Survey Questions

Please answer each of these questions about your experience with the adaptive tools. Choose the best answer for each multiple choice question.

1. Rank the following missions in order of difficulty to complete where 1 is easiest and 4 is hardest.

4 Single vehicle, built-in commands
2 Multiple vehicles, built-in commands
3 Single vehicle, generated sequences of commands
1 Multiple vehicles, generated sequences of commands

2. For each mission, what was the difficulty in accomplishing the **primary goal**?

Mission	Easy	Moderately easy	Moderate	Moderately Hard	Hard
Single vehicle, built-in commands				X	
Multiple vehicles, built-in commands		X			
Single vehicle, generated sequences of commands				X	
Multiple vehicles, generated sequences of commands	X				

3. For each mission, what was the difficulty in accomplishing the **secondary goal**?

Mission	Easy	Moderately easy	Moderate	Moderately Hard	Hard
Single vehicle, built-in commands	X				
Multiple vehicles, built-in commands			X		
Single vehicle, generated sequences of commands	X				
Multiple vehicles, generated sequences of commands	X				

4. How many vehicles do you feel would be optimal for the missions where only built-in commands are provided? The optimal number should be the most that you feel you can control effectively.

a. 1
b. 2
c. 3 <-
d. 4
e. 5
f. 6
g. 7
h. 8
i. 9
j. 10
k. >10

5. How many vehicles do you feel would be optimal for the missions where generated commands are provided? The optimal number should be the most that you feel you can control effectively.

a. 1
b. 2
c. 3
d. 4
e. 5
f. 6
g. 7
h. 8
i. 9
j. 10
k. >10 <-

6. How many of the generated commands were useful?

a. None of the generated commands were useful

b. A few of the generated commands were useful <-

c. Some of the generated commands were useful

d. Most of the generated commands were useful

e. All of the generated commands were useful

7. How often do you play video games where timing and/or hand eye coordination are useful?

a. Daily

b. A few times a week <-

c. Once a week

d. A few times a month

e. Once a month

f. Very rarely

8. How much do you enjoy playing video games?

a. Very much

b. Somewhat <-

c. Not at all

9. How much did you enjoy using this simulator?

a. Very much

b. Somewhat <-

c. Not at all

10. How old are you?

a. 18-24 <-
b. 25-30
c 31-40
d. 41-50
e. 51-60
f. >60

11. What generated commands did you find the most useful and why?

The command to permanently stay in front of the enemy was quite useful since after finding the enemy I could completely focus on the secondary objective.

12. What generated commands did you find the least useful and why?

The move commands felt a little weak because I could not find one I liked

13. What commands should have been generated but weren't?

I would have liked a ready all button and/or a combination ready lift-off

14. Please take the remaining space to list any comments you would like the experimenter to know.

User ID: 2

Adaptive User Interface Survey Questions

Please answer each of these questions about your experience with the adaptive tools. Choose the best answer for each multiple choice question.

1. Rank the following missions in order of difficulty to complete where 1 is easiest and 4 is hardest.

4 Single vehicle, built-in commands
3 Multiple vehicles, built-in commands
2 Single vehicle, generated sequences of commands
1 Multiple vehicles, generated sequences of commands

2. For each mission, what was the difficulty in accomplishing the **primary goal**?

Mission	Easy	Moderately easy	Moderate	Moderately Hard	Hard
Single vehicle, built-in commands			X		
Multiple vehicles, built-in commands		X			
Single vehicle, generated sequences of commands		X			
Multiple vehicles, generated sequences of commands	X				

3. For each mission, what was the difficulty in accomplishing the **secondary goal**?

Mission	Easy	Moderately easy	Moderate	Moderately Hard	Hard
Single vehicle, built-in commands			X		
Multiple vehicles, built-in commands				X	
Single vehicle, generated sequences of commands		X			
Multiple vehicles, generated sequences of commands	X				

4. How many vehicles do you feel would be optimal for the missions where only built-in commands are provided? The optimal number should be the most that you feel you can control effectively.

a. 1

b. 2

c. 3

d. 4 <-

e. 5

f. 6

g. 7

h. 8

i. 9

j. 10

k. >10

5. How many vehicles do you feel would be optimal for the missions where generated commands are provided? The optimal number should be the most that you feel you can control effectively.

a. 1

b. 2

c. 3

d. 4

e. 5

f. 6 <-

g. 7

h. 8

i. 9

j. 10

k. >10

6. How many of the generated commands were useful?

a. None of the generated commands were useful
b. A few of the generated commands were useful
c. Some of the generated commands were useful
d. Most of the generated commands were useful <-
e. All of the generated commands were useful

7. How often do you play video games where timing and/or hand eye coordination are useful?

a. Daily
b. A few times a week
c. Once a week
d. A few times a month <-
e. Once a month
f. Very rarely

8. How much do you enjoy playing video games?

a. Very much
b. Somewhat <-
c. Not at all

9. How much did you enjoy using this simulator?

a. Very much
b. Somewhat <-
c. Not at all

10. How old are you?

a. 18-24 <-
b. 25-30
c 31-40
d. 41-50
e. 51-60
f. >60

11. What generated commands did you find the most useful and why?

front->front->front (looped)
completely automates enemy tracking so that repetitive clicking isn't needed and more time can be focused on secondary tasks

12. What generated commands did you find the least useful and why?

Two of the generated commands were identical when looped: front->front->left and front->left->front

Neither were extremely useful since front->front->front was more effective.

13. What commands should have been generated but weren't?

A takeoff sequence that includes R1, takeoff and waypoint but leaves actual selection of the target position to the user.

14. Please take the remaining space to list any comments you would like the experimenter to know.

Survey questions on ease of accomplishing objectives do not differentiate between difficulty in performing up to maximum potential for the mission and ability to accomplish objectives. (i.e. one vehicle requires less attention but cannot perform as well)

User ID: 3

Adaptive User Interface Survey Questions

Please answer each of these questions about your experience with the adaptive tools. Choose the best answer for each multiple choice question.

1. Rank the following missions in order of difficulty to complete where 1 is easiest and 4 is hardest.

3 Single vehicle, built-in commands
4 Multiple vehicles, built-in commands
1 Single vehicle, generated sequences of commands
3 Multiple vehicles, generated sequences of commands

2. For each mission, what was the difficulty in accomplishing the **primary goal**?

Mission	Easy	Moderately easy	Moderate	Moderately Hard	Hard
Single vehicle, built-in commands	X				
Multiple vehicles, built-in commands				X	
Single vehicle, generated sequences of commands	X				
Multiple vehicles, generated sequences of commands	X				

3. For each mission, what was the difficulty in accomplishing the **secondary goal**?

Mission	Easy	Moderately easy	Moderate	Moderately Hard	Hard
Single vehicle, built-in commands	X				
Multiple vehicles, built-in commands					X
Single vehicle, generated sequences of commands	X				
Multiple vehicles, generated sequences of commands			X		

4. How many vehicles do you feel would be optimal for the missions where only built-in commands are provided? The optimal number should be the most that you feel you can control effectively.

a. 1

b. 2 <-

c. 3

d. 4

e. 5

f. 6

g. 7

h. 8

i. 9

j. 10

k. >10

5. How many vehicles do you feel would be optimal for the missions where generated commands are provided? The optimal number should be the most that you feel you can control effectively.

a. 1

b. 2

c. 3

d. 4 <-

e. 5

f. 6

g. 7

h. 8

i. 9

j. 10

k. >10

6. How many of the generated commands were useful?

a. None of the generated commands were useful
b. A few of the generated commands were useful <-
c. Some of the generated commands were useful
d. Most of the generated commands were useful
e. All of the generated commands were useful

7. How often do you play video games where timing and/or hand eye coordination are useful?

a. Daily
b. A few times a week <-
c. Once a week
d. A few times a month
e. Once a month
f. Very rarely

8. How much do you enjoy playing video games?

a. Very much <-
b. Somewhat
c. Not at all

9. How much did you enjoy using this simulator?

a. Very much
b. Somewhat <-
c. Not at all

10. How old are you?

a. 18-24 <-

b. 25-30

c 31-40

d. 41-50

e. 51-60

f. >60

11. What generated commands did you find the most useful and why?

Following the enemy. Simple task, but requires constant attention if done manually.

12. What generated commands did you find the least useful and why?

a1 [(839,291)->(471,206)->(58,31)], small area covered, accomplishes little

13. What commands should have been generated but weren't?

circling perimeter

keeping distance from neighboring drones

moving large unexplored spaces

14. Please take the remaining space to list any comments you would like the experimenter to know.

UI needs more information, like macro course plotted out

hot keys are less distracting than buttons

ability to manually queue waypoints could prove informative

User ID: 4

Adaptive User Interface Survey Questions

Please answer each of these questions about your experience with the adaptive tools. Choose the best answer for each multiple choice question.

1. Rank the following missions in order of difficulty to complete where 1 is easiest and 4 is hardest.

4 Single vehicle, built-in commands
2 Multiple vehicles, built-in commands
3 Single vehicle, generated sequences of commands
1 Multiple vehicles, generated sequences of commands

2. For each mission, what was the difficulty in accomplishing the **primary goal**?

Mission	Easy	Moderately easy	Moderate	Moderately Hard	Hard
Single vehicle, built-in commands				X	
Multiple vehicles, built-in commands		X			
Single vehicle, generated sequences of commands			X		
Multiple vehicles, generated sequences of commands		X			

3. For each mission, what was the difficulty in accomplishing the **secondary goal**?

Mission	Easy	Moderately easy	Moderate	Moderately Hard	Hard
Single vehicle, built-in commands			X		
Multiple vehicles, built-in commands		X			
Single vehicle, generated sequences of commands		X			
Multiple vehicles, generated sequences of commands	X				

4. How many vehicles do you feel would be optimal for the missions where only built-in commands are provided? The optimal number should be the most that you feel you can control effectively.

a. 1

b. 2

c. 3

d. 4

e. 5 <-

f. 6

g. 7

h. 8

i. 9

j. 10

k. >10

5. How many vehicles do you feel would be optimal for the missions where generated commands are provided? The optimal number should be the most that you feel you can control effectively.

a. 1

b. 2

c. 3

d. 4 <-

e. 5

f. 6

g. 7

h. 8

i. 9

j. 10

k. >10

6. How many of the generated commands were useful?

a. None of the generated commands were useful

b. A few of the generated commands were useful

c. Some of the generated commands were useful

d. Most of the generated commands were useful <-

e. All of the generated commands were useful

7. How often do you play video games where timing and/or hand eye coordination are useful?

a. Daily

b. A few times a week <-

c. Once a week

d. A few times a month

e. Once a month

f. Very rarely

8. How much do you enjoy playing video games?

a. Very much

b. Somewhat <-

c. Not at all

9. How much did you enjoy using this simulator?

a. Very much

b. Somewhat <-

c. Not at all

10. How old are you?

a. 18-24 <-

b. 25-30

c 31-40

d. 41-50

e. 51-60

f. >60

11. What generated commands did you find the most useful and why?

the enemy tracking command - takes most attention

the explore bottom command - allowed focus to shift

the explore center command - allowed focus to shift

the explore top command - allowed focus to shift

12. What generated commands did you find the least useful and why?

commands that patrolled very small distances

13. What commands should have been generated but weren't?

none

14. Please take the remaining space to list any comments you would like the experimenter to know.

well made and the generated commands are pretty impressive

User ID: 5

Adaptive User Interface Survey Questions

Please answer each of these questions about your experience with the adaptive tools. Choose the best answer for each multiple choice question.

1. Rank the following missions in order of difficulty to complete where 1 is easiest and 4 is hardest.

4 Single vehicle, built-in commands
2 Multiple vehicles, built-in commands
3 Single vehicle, generated sequences of commands
1 Multiple vehicles, generated sequences of commands

2. For each mission, what was the difficulty in accomplishing the **primary goal**?

Mission	Easy	Moderately easy	Moderate	Moderately Hard	Hard
Single vehicle, built-in commands			X		
Multiple vehicles, built-in commands		X			
Single vehicle, generated sequences of commands			X		
Multiple vehicles, generated sequences of commands	X				

3. For each mission, what was the difficulty in accomplishing the **secondary goal**?

Mission	Easy	Moderately easy	Moderate	Moderately Hard	Hard
Single vehicle, built-in commands				X	
Multiple vehicles, built-in commands			X		
Single vehicle, generated sequences of commands				X	
Multiple vehicles, generated sequences of commands			X		

4. How many vehicles do you feel would be optimal for the missions where only built-in commands are provided? The optimal number should be the most that you feel you can control effectively.

a. 1

b. 2

c. 3

d. 4

e. 5

f. 6 <-

g. 7

h. 8

i. 9

j. 10

k. >10

5. How many vehicles do you feel would be optimal for the missions where generated commands are provided? The optimal number should be the most that you feel you can control effectively.

a. 1

b. 2

c. 3

d. 4 <-

e. 5

f. 6

g. 7

h. 8

i. 9

j. 10

k. >10

6. How many of the generated commands were useful?

a. None of the generated commands were useful
b. A few of the generated commands were useful <-
c. Some of the generated commands were useful
d. Most of the generated commands were useful
e. All of the generated commands were useful

7. How often do you play video games where timing and/or hand eye coordination are useful?

a. Daily
b. A few times a week <-
c. Once a week
d. A few times a month
e. Once a month
f. Very rarely

8. How much do you enjoy playing video games?

a. Very much <-
b. Somewhat
c. Not at all

9. How much did you enjoy using this simulator?

a. Very much
b. Somewhat <-
c. Not at all

10. How old are you?

a. 18-24

b. 25-30 <-

c 31-40

d. 41-50

e. 51-60

f. >60

11. What generated commands did you find the most useful and why?

Follow the enemy - much of the time is spent entering commands to follow the enemy once it is found. Having the automated command makes it easier to accomplish the secondary goal.

12. What generated commands did you find the least useful and why?

Most of the generated commands weren't very useful in general

13. What commands should have been generated but weren't?

The generated commands didn't follow the search method I was trying to use. A more even distribution of flight paths would have been beneficial.

14. Please take the remaining space to list any comments you would like the experimenter to know.

A secondary line indicating where the unit will go when using the generated commands or an icon showing which path a unit is on. Otherwise the units are too indistinguishable.

User ID: 6

Adaptive User Interface Survey Questions

Please answer each of these questions about your experience with the adaptive tools. Choose the best answer for each multiple choice question.

1. Rank the following missions in order of difficulty to complete where 1 is easiest and 4 is hardest.

4 Single vehicle, built-in commands
3 Multiple vehicles, built-in commands
2 Single vehicle, generated sequences of commands
1 Multiple vehicles, generated sequences of commands

2. For each mission, what was the difficulty in accomplishing the **primary goal**?

Mission	Easy	Moderately easy	Moderate	Moderately Hard	Hard
Single vehicle, built-in commands				X	
Multiple vehicles, built-in commands			X		
Single vehicle, generated sequences of commands		X			
Multiple vehicles, generated sequences of commands	X				

3. For each mission, what was the difficulty in accomplishing the **secondary goal**?

Mission	Easy	Moderately easy	Moderate	Moderately Hard	Hard
Single vehicle, built-in commands					X
Multiple vehicles, built-in commands				X	
Single vehicle, generated sequences of commands			X		
Multiple vehicles, generated sequences of commands	X				

4. How many vehicles do you feel would be optimal for the missions where only built-in commands are provided? The optimal number should be the most that you feel you can control effectively.

a. 1

b. 2

c. 3

d. 4

e. 5 <-

f. 6

g. 7

h. 8

i. 9

j. 10

k. >10

5. How many vehicles do you feel would be optimal for the missions where generated commands are provided? The optimal number should be the most that you feel you can control effectively.

a. 1

b. 2

c. 3

d. 4

e. 5

f. 6

g. 7

h. 8

i. 9

j. 10

k. >10 <-

6. How many of the generated commands were useful?

a. None of the generated commands were useful
b. A few of the generated commands were useful
c. Some of the generated commands were useful
d. Most of the generated commands were useful
e. All of the generated commands were useful <-

7. How often do you play video games where timing and/or hand eye coordination are useful?

a. Daily <-
b. A few times a week
c. Once a week
d. A few times a month
e. Once a month
f. Very rarely

8. How much do you enjoy playing video games?

a. Very much <-
b. Somewhat
c. Not at all

9. How much did you enjoy using this simulator?

a. Very much <-
b. Somewhat
c. Not at all

10. How old are you?

a. 18-24

b. 25-30 <-

c 31-40

d. 41-50

e. 51-60

f. >60

11. What generated commands did you find the most useful and why?

Staying in front of the enemy - allowed me to switch my focus to secondary objective

12. What generated commands did you find the least useful and why?

They all came in very handy

13. What commands should have been generated but weren't?

14. Please take the remaining space to list any comments you would like the experimenter to know.

Using the generated commands was great. I was able to pay more attention to my strategy, rather than constantly having to switch vehicles and select commands.

User ID: 7

Adaptive User Interface Survey Questions

Please answer each of these questions about your experience with the adaptive tools. Choose the best answer for each multiple choice question.

1. Rank the following missions in order of difficulty to complete where 1 is easiest and 4 is hardest.

1 Single vehicle, built-in commands
3 Multiple vehicles, built-in commands
2 Single vehicle, generated sequences of commands
4 Multiple vehicles, generated sequences of commands

2. For each mission, what was the difficulty in accomplishing the **primary goal**?

Mission	Easy	Moderately easy	Moderate	Moderately Hard	Hard
Single vehicle, built-in commands		X			
Multiple vehicles, built-in commands	X				
Single vehicle, generated sequences of commands		X			
Multiple vehicles, generated sequences of commands		X			

3. For each mission, what was the difficulty in accomplishing the **secondary goal**?

Mission	Easy	Moderately easy	Moderate	Moderately Hard	Hard
Single vehicle, built-in commands				X	
Multiple vehicles, built-in commands	X				
Single vehicle, generated sequences of commands		X			
Multiple vehicles, generated sequences of commands		X			

4. How many vehicles do you feel would be optimal for the missions where only built-in commands are provided? The optimal number should be the most that you feel you can control effectively.

a. 1

b. 2

c. 3

d. 4

e. 5 <-

f. 6

g. 7

h. 8

i. 9

j. 10

k. >10

5. How many vehicles do you feel would be optimal for the missions where generated commands are provided? The optimal number should be the most that you feel you can control effectively.

a. 1

b. 2

c. 3

d. 4

e. 5 <-

f. 6

g. 7

h. 8

i. 9

j. 10

k. >10

6. How many of the generated commands were useful?

a. None of the generated commands were useful
b. A few of the generated commands were useful
c. Some of the generated commands were useful
d. Most of the generated commands were useful <-
e. All of the generated commands were useful

7. How often do you play video games where timing and/or hand eye coordination are useful?

a. Daily
b. A few times a week
c. Once a week <-
d. A few times a month
e. Once a month
f. Very rarely

8. How much do you enjoy playing video games?

a. Very much <-
b. Somewhat
c. Not at all

9. How much did you enjoy using this simulator?

a. Very much
b. Somewhat <-
c. Not at all

10. How old are you?

a. 18-24 <-

b. 25-30

c 31-40

d. 41-50

e. 51-60

f. >60

11. What generated commands did you find the most useful and why?

Tracking. I could hover nearby and quickly wait to move again.

12. What generated commands did you find the least useful and why?

For the start of each simulation, I usually just fan out. Then I like to assign the bot to a loop to cover an area, so the loop commands weren't as useful at first.

13. What commands should have been generated but weren't?

Some automated landing, wait and continue

14. Please take the remaining space to list any comments you would like the experimenter to know.

Finding the coords wasn't completely obvious. The generated macros should be more descriptive on the menu bar.

APPENDIX B.
IRB FORMS

APPLICATION TO THE UNIVERSITY OF MISSOURI-ROLLA
CAMPUS INSTITUTIONAL REVIEW BOARD
FOR THE PROTECTION OF HUMAN SUBJECTS IN RESEARCH (UMRIRB-1)

Review Requested: ☐ **Exemption** ☒ **Expedited** ☐ **Full Board**

1a. Primary Investigator: Paul Robinette

Daytime Phone Number: 314-740-3859

Mailing Address: 301 w 16TH St, G11 ECE

City/State/Zip: Rolla, MO 65409

E-Mail Address: pmrmq3@mst.edu

Department: **Electrical and Computer Engineering**

1b.Additional Applicant(s):

1c.Advisor: Dr. Don Wunsch

Daytime Phone Number: 573-341-4521

Advisor's E-Mail Address: dwunsch@mst.edu

Department: Electrical and Computer Engineering

Campus Mailing Address: 131 ECE

2. Project Period: From February 15, 2010 **to** April 30, 2010

3. Funding Source(s): MK Finley Endowment

4. Site of Work: G11 ECE

5a. Title of Project:

Adaptive User Interface Applied to Vehicle Swarm Control

5b. Brief description of its general purpose:

I would like to test my adaptive user interface tools by having several students run a simulation using Boeing's swarm simulator. The users would be controlling several robotic vehicles to accomplish a simple search mission. When they have found the target they will be instructed to track it (follow with a vehicle) to the best of their ability. During the mission, several events will happen to disrupt their work, such as a vehicle crashing and vehicles running out of fuel. Each mission will be designed to take about 5 minutes. There will be four different variations of the mission that each user will run. Several days before their first run they will be asked to come in for training, where they will run some test missions.

6. Give details of the procedures that relate to the subjects' participation, including at a minimum the following information (append additional page(s) if necessary):

a) How will the subjects be selected and recruited? (Append copy of letter, ad, or transcript of verbal announcement.)

Subjects will most likely be recruited from the Robotics Competition Team at S&T. An email will be sent out to ask for volunteers.

b) What inducement is offered?

No inducement other than the chance to help in a research project will be offered.

c) Number and salient characteristics of subject, i.e., age range, sex, institutional affiliation, other pertinent characterizations.

Up to 10 undergraduate students (expected to be 18-23 years old).

d) If a cooperating institution (school, hospital, prison, etc.) is involved, has written permission been obtained? (Append letters). N/A

e) Number of times observations will be made? 2 sessions, 4 missions each

f) What do the subjects do, or what is done to them, in the study? (Append copy of questionnaires or test instruments, description of procedure to be conducted on the subject.

Subjects will control several simulated vehicles. The time they are able to track a target vehicle will be recorded. The area that they have searched will be recorded. The number of vehicles they are able to use will be recorded. The time to respond to changes in the mission will be recorded. They will be given a survey afterwards.

g) Is it clear to the subject that their participation is voluntary, that they may withdraw at any time, and that that they may refuse to answer any specific question that may be asked them? yes

h) Number of subjects to be used in the project: Up to 10

i) Please indicate below if any of your proposed subjects might fit into the following categories:

	Yes	No	Age
Minors?		X	
Pregnant Women?	X		
Women of Child-Bearing Age?	X		
Institutionalized Persons?		X	
Incompetent Persons?		X	
Students?	X		
Low-Income Persons?	X		
Minorities?	X		

j) Cite your experience with this type of research.

I have never performed human factors testing on subjects before.

7. **reason that oral, rather than written, consent is being used. Also, explain how you will ascertain that the subjects understand what they are agreeing to.**

Written consent

8. In your view, what benefits may result from the study that would justify asking the subjects to participate?

The study will show how effective the adaptive user interface tools are when controlling many vehicles in a time critical mission.

9a. Do you see any chance that subjects might be harmed in any way? Do you deceive them in any way? Are there any physical risks? Psychological? (Might a subject feel demeaned or embarrassed or worried or upset? Social? (Possible loss of status, privacy, reputation?)

No, this will be like playing a video game.

9b. How do you ensure confidentiality of information collected? (Consider 9a and 9b from the point of view of the subject.)

Information will be correlated to a randomly generated number in all data sets. This number will be associated with a name in a document that will only be viewed by the researcher. The number will be associated with the name in case further testing is deemed necessary. The document will be deleted once final testing is complete. All results published will only list the random number.

Paul Robinette	Don Wunsch	2/20/2010
Applicant's Name (Please Print)	**Faculty Advisor's Name (Please Print)**	**Date**

Paul Robinette		
Applicant's Signature	**Faculty Advisor's Signature**	**Date**

INSTRUCTIONS FOR COMPLETION OF THE "APPLICATION TO THE UMR CAMPUS IRB" (UMRIRB-1)

CITI Collaborative Institutional Training Initiative (CITI)

Campus SBR Curriculum Completion Report
Printed on

Learner: Paul Robinette (username: pmrmq3)
Institution: University of Missouri-Columbia
Contact Information Department: Computer Engineering
Email: pmrmq3@mst.edu

Campus SBR:

Stage 1. Basic Course Passed on 09/28/09 (Ref # 3580656)

Required Modules	Date Completed	
University of Missouri-Columbia	09/28/09	no quiz
Introduction	09/28/09	no quiz
Belmont Report and CITI Course Introduction	09/28/09	2/3 (67%)
Students in Research - SBR	09/28/09	7/10 (70%)
History and Ethical Principles - SBR	09/28/09	4/4 (100%)
Defining Research with Human Subjects - SBR	09/28/09	5/5 (100%)
The Regulations and The Social and Behavioral Sciences - SBR	09/28/09	4/5 (80%)
Assessing Risk in Social and Behavioral Sciences - SBR	09/28/09	4/5 (80%)
Informed Consent - SBR	09/28/09	3/4 (75%)
Privacy and Confidentiality - SBR	09/28/09	3/3 (100%)
Research with Prisoners - SBR	09/28/09	4/4 (100%)
Research with Children - SBR	09/28/09	3/4 (75%)
Research in Public Elementary and Secondary Schools - SBR	09/28/09	4/4 (100%)
International Research - SBR	09/28/09	3/3 (100%)
Internet Research - SBR	09/28/09	5/5 (100%)
HIPAA and Human Subjects Research	09/28/09	2/2 (100%)
Workers as Research Subjects-A Vulnerable Population	09/28/09	4/4 (100%)
Conflicts of Interest in Research Involving Human Subjects	09/28/09	1/2 (50%)

For this Completion Report to be valid, the learner listed above must be affiliated with a CITI participating institution. Falsified information and

unauthorized use of the CITI course site is unethical, and may be considered scientific misconduct by your institution.

Paul Braunschweiger Ph.D.
Professor, University of Miami
Director Office of Research Education
CITI Course Coordinator

CITI Collaborative Institutional Training Initiative (CITI)

Campus SBR Curriculum Completion Report
Printed on 2/10/2010

Learner: Donald Wunsch (username: dwunsch)
Institution: University of Missouri-Columbia
Contact Information Department: Electrical & Computer Engineering
Email: dwunsch@mst.edu

Campus SBR:

Stage 1. Basic Course Passed on 10/29/09 (Ref # 3691053)

Required Modules	**Date Completed**	
University of Missouri-Columbia	10/28/09	no quiz
Introduction	10/28/09	no quiz
Belmont Report and CITI Course Introduction	10/28/09	3/3 (100%)
Students in Research - SBR	10/28/09	8/10 (80%)
History and Ethical Principles - SBR	10/28/09	3/4 (75%)
Defining Research with Human Subjects - SBR	10/28/09	5/5 (100%)
The Regulations and The Social and Behavioral Sciences - SBR	10/28/09	5/5 (100%)
Assessing Risk in Social and Behavioral Sciences - SBR	10/29/09	5/5 (100%)
Informed Consent - SBR	10/29/09	4/4 (100%)
Privacy and Confidentiality - SBR	10/29/09	3/3 (100%)
Research with Prisoners - SBR	10/29/09	4/4 (100%)
Research with Children - SBR	10/29/09	3/4 (75%)
Research in Public Elementary and Secondary Schools - SBR	10/29/09	3/4 (75%)
International Research - SBR	10/29/09	3/3 (100%)
Internet Research - SBR	10/29/09	5/5 (100%)
HIPAA and Human Subjects Research	10/29/09	2/2 (100%)
Workers as Research Subjects-A Vulnerable Population	10/29/09	3/4 (75%)
Conflicts of Interest in Research Involving Human Subjects	10/29/09	0/2 (0%)

For this Completion Report to be valid, the learner listed above must be affiliated with a CITI participating institution. Falsified information and unauthorized use of the CITI course site is unethical, and may be considered scientific misconduct by your institution.

Paul Braunschweiger Ph.D.
Professor, University of Miami
Director Office of Research Education
CITI Course Coordinator

Email Soliciting Users

Folks,

I need some volunteers to help me test my thesis project. For the last year I have been working on improving Boeing's user interface to help humans control autonomous aerial vehicles more effectively. I will need some of you to help me test the tools I have developed to determine how effective they are. I anticipate that the users will be asked to spend 2 sessions, each about 1 hour long, in order to test everything. The test will be very similar to a video game.

This is strictly voluntary, but it will help me complete my work for my thesis.

If you have any questions please contact me.

Thanks,
Paul

RESEARCH SUBJECT INFORMATION AND CONSENT FORM

Title: Adaptive User Interface Test

Sponsor: Boeing

Investigator: Paul Robinette

G11 ECE

301 W 16th Street

Rolla, MO 65409

314-740-3859

pmrmq3@mst.edu

Site(s): Missouri S&T Campus, G11 ECE

This consent form may contain words that you do not understand. Please ask the researcher or the study staff to explain any words or information that you do not clearly understand. You may take home an unsigned copy of this consent form to think about or discuss with family or friends before making your decision.

SUMMARY

- Your decision to be in this study is voluntary.
- If you decide to be in this study and then change your mind, you can leave the study at any time.
- You will be in this study for 2 sessions, each approximately 1 hour long.
- If you agree to be in this study, your research records will become part of this study. They may be looked at or copied by the sponsor of this study or government agencies or other groups associated with the study.

More detailed information about this study is in this consent form. Please read it carefully.

PURPOSE OF THE STUDY:

The purposes of this study are:

- To determine if there is a difference between a normal command and control interface for vehicles and an adaptive command and control interface
- To create an intelligent command and control module that can replace a human user in some missions

You will be in this study for 2 approximately 1 hour sessions. Approximately 5-10 subjects will participate in this study. The study is scheduled to take place between February 21, 2010 and April 30, 2010, and will be done April 30, 2010.

PROCEDURES

If you decide to participate, you will:

- Control simulated robotic vehicles to accomplish a mission
- Primary goal: follow a target vehicle as it moves randomly
- Secondary goal: explore as much of the surrounding area as possible
- Commands issued, state of the system and progress towards goals will all be logged throughout the mission

RISKS AND DISCOMFORTS

There are no anticipated risks in this study that are greater than you will encounter while playing a video game. If you experience any discomfort, you should inform the researcher immediately and stop your participation.

BENEFITS

You are not expected to benefit directly from participation in the study. The results from the study may contribute to the fields of human computer interaction and computational intelligence.

COSTS

There is no cost to you for participating in this study.

PAYMENT FOR PARTICIPATION

You will not receive any additional payment for participating in this study.

ALTERNATIVE TREATMENT

This is not a treatment study. Your alternative is to not participate in this study.

CONFIDENTIALITY

Information from this study will be given to the sponsor. Research records, including logs, and the consent form signed by you may be looked at and/or copied for research and regulatory purposes by:

- The sponsor
- The Boeing Company

Absolute confidentiality cannot be guaranteed because of the need to give information to these parties. The results of this research study may be presented at meetings or in publications. Your identity will not be disclosed in those presentations. Your identity will not be released to the general public without your consent, unless specifically required by law.

VOLUNTARY PARTICIPATION AND WITHDRAWAL

Your participation in this study is voluntary. You may decide not to participate or you may leave the study at any time. Your decision will not result in any penalty or loss of benefits to which you are entitled. If significant new findings develop during the course of this study that may relate to your decision to continue participation, you will be informed.

Your participation in this study may be stopped at any time by the researcher or the sponsor without your consent because:

- you have not followed study instructions;
- the sponsor has stopped the study; or
- administrative reasons require your withdrawal.

SOURCE OF FUNDING FOR THE STUDY

This study is being funded by Boeing.

QUESTIONS

If you have any questions about this study or your participation in this study, contact:

Paul Robinette at (314) 740-3859

If you have questions about your rights as a research subject, you may contact:

Greg Lim

Human Subjects Protection Program Administrator

The Boeing Company

(425) 865 1068

E-mail: Gregorio.Lim@Boeing.com

Do not sign this consent form unless you have had a chance to ask questions and have received satisfactory answers to all of your questions.

If you agree to be in this study, you will receive a signed and dated copy of this consent form for your records.

CONSENT

I have read the information in this consent form. All my questions about the study and my participation in it have been answered. I freely consent to be in this research study.

I affirm that I am over 18 years of age.

I authorize the use and disclosure of my information to the parties listed in the confidentiality section of this consent for the purposes described above.

By signing this consent form, I have not given up any of my legal rights.

____
Subject Name

CONSENT SIGNATURE:

Signature of Subject	Date
Signature of Person Conducting Informed Consent Discussion	Date

BIBLIOGRAPHY

[1] P. Langley, “User Modeling in Adaptive Interfaces,” *Proceedings of the Seventh International Conference on User Modeling*, Springer, 1999, pp. 357–370.

[2] P. Langley, “Machine Learning for Adaptive User Interfaces,” *Proceedings of the 21st German Annual Conference on Artificial Intelligence*, Springer, 1997, pp. 53–62.

[3] S. Hakiel, “Delivering ease of use [software development],” *Computing Control Engineering Journal*, vol. 8, Apr. 1997, pp. 81 -87.

[4] R. Parasuraman, T.B. Sheridan, and C.D. Wickens, “A model for types and levels of human interaction with automation,” *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 30, May 2000, pp. 286 -297.

[5] J.R. Anderson, M. Matessa, and C. Lebiere, “ACT-R: A Theory of Higher Level Cognition and Its Relation to Visual Attention,” *Human Computer Interaction*, vol. 12, 1997, pp. 439-462.

[6] F.E. Ritter, D. Van Rooy, R.S. Amant, and K. Simpson, “Providing user models direct access to interfaces: an exploratory study of a simple interface with implications for HRI and HCI,” *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 36, May 2006, pp. 592 -601.

[7] M. Jayapandian and H.V. Jagadish, “Automating the Design and Construction of Query Forms,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, Oct. 2009, pp. 1389 -1402.

[8] C. Chang and C. Hsu, “Enabling concept-based relevance feedback for information retrieval on the WWW,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 11, Aug. 1999, pp. 595 -609.

[9] C. Wei and Y. Chang, “Discovering Event Evolution Patterns From Document Sequences,” *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 37, Mar. 2007, pp. 273 -283.

[10] R. Parasuraman, S. Galster, P. Squire, H. Furukawa, and C. Miller, “A flexible delegation-type interface enhances system performance in human supervision of multiple robots: empirical studies with RoboFlag,” *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 35, Jul. 2005, pp. 481 - 493.

[11] Y. Endo, D. MacKenzie, and R. Arkin, “Usability evaluation of high-level user assistance for robot mission specification,” *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 34, May 2004, pp. 168 -180.

[12] M. Virvou and K. Kabassi, “Adapting the human plausible reasoning theory to a graphical user interface,” *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 34, Jul. 2004, pp. 546 - 563.

[13] J. Ontrup, H. Ritter, S.W. Scholz, and R. Wagner, “Detecting, Assessing and Monitoring Relevant Topics in Virtual Information Environments,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, Mar. 2009, pp. 415 -427.

[14] S. Grossberg, “Adaptive pattern classification and universal recoding, II: Feedback, expectation, olfaction, and illusions,” *Biological Cybernetics*, vol. 23, 1976, pp. 187-202.

[15] G.A. Carpenter, S. Grossberg, and D.B. Rosen, “ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition,” *Neural Networks*, vol. 4, pp. 493-504.

[16] S. Grossberg and G.A. Carpenter, “A massively parallel architecture for a self-organizing neural pattern recognition machine,” *Computer Vision, Graphics, and Image Processing*, vol. 37, 1987, pp. 54-115.

[17] G.A. Carpenter and N. Markuzon, “ARTMAP-IC and medical diagnosis: Instance counting and inconsistent cases,” *Neural Networks*, vol. 11, 1998, pp. 323-336.

[18] G.A. Carpenter and S. Grossberg, “Fuzzy ART: Fast Stable Learning and Categorization of analog patters by an adaptive resonance system,” *Neural Networks*, vol. 4, 1991, pp. 759-771.

[19] R. Xu and D. Wunsch, “Survey of clustering algorithms,” *Neural Networks, IEEE Transactions on*, vol. 16, May 2005, pp. 645 -678.

[20] R. Xu and D. Wunsch, *Clustering*, Wiley-IEEE Press, 2008.

[21] R.J. Meuth, “Meta-Learning Computational Intelligence Architectures,” Ph.D. Dissertation, Electrical and Computer Engineering Department, Missouri University of Science and Technology, 2009.

[22] R.J. Meuth, J. Seiffertt, P. Robinette, and D.C. Wunsch II, “Gram-ART: Variable Dimensionality Representation with Non-Parametric Templates,” *Neural Networks (submitted)*, 2010.

VITA

Paul Michael Robinette was born on August 7, 1984 in St. Louis, Missouri. After graduating from Parkway North High School in June 2003, Paul attended the Missouri University of Science and Technology. He joined several student organizations, including the Amateur Radio Club, the Society of Physics Students and the Robotics Competition Team. Paul was elected to numerous leadership positions during this time, including two terms as president of the Robotics Team. He graduated with a B.S. in Computer Engineering, a B.S. in Physics and a minor in Russian in May 2008.

While an undergraduate, Paul formed the company Rolla Engineered Solutions, LLC with a fellow student. They fulfilled several software and hardware projects for numerous clients before winning the University of Missouri Student Entrepreneur of the Year award in 2008. Most recently, they have designed a miniature robot to assist university level engineering education.

In 2007, while still a senior, Paul started his M.S. In Computer Engineering. He became a graduate research assistant in the Applied Computational Intelligence Laboratory. While a GRA, Paul worked on projects for the Boeing Corporation, 21st Century Systems International and the U.S. Army. His first publication as a primary author was "An Agent-Based Computational Model of a Self-Organizing Project Management Paradigm for Research Teams" at the 2009 International Joint Conference on Neural Networks. He then published "LabRat™: Miniature Robot for Students, Researchers, and Hobbyists" at the 2009 International Conference on Robotics and Intelligent Systems. So far, Paul has nine publications, including several symposiums and conferences.