5-2008

Ecological Interface Design in Variable Workload Multitasking

James Rubinstein
Clemson University, jrubins@clemson.edu

ECOLOGICAL INTERFACE DESIGN IN VARIABLE WORKLOAD MULTITASKING

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Applied Psychology

---

by
James S Rubinstein
May 2008

Accepted by:
Dr. Lee Gugerty, Committee Chair
Dr. Rich Pak
Dr. Fred Switzer

ABSTRACT

This study was undertaken to examine the question of how well Ecological Interface Design (EID) would support operators of a multitasking work domains. Previous research has shown that EID can support better operator performance while controlling a simulated process. Recently, there has been some interest in applying EID to automobiles, planes, and other multitasking domains. This research aimed to answer a more basic question: whether or not people could detect errors using EID while trying to do well on a visual psychomotor task.

The experiment used two tasks. The first task involved monitoring errors in a simulated process control plant, using an EID interface or a non-EID interface. The second task was the ball task. The ball task had participants try to catch virtual balls on screen by moving a block on the screen. The ball task had two levels, fast and slow.

It was predicted that the participants in the EID condition would perform better at error monitoring than participants in the non-EID interface condition. It was further predicted that error monitoring in the EID condition would be less negatively affected by the increase in workload than in the non-EID condition. The results did not support the predicted superiority for EID. Although these findings are inconclusive, they suggest potential problems in using EID in multitasking environments.

DEDICATION

This work is dedicated to my wife and daughter, who only know me as a student, and to my parents, who have always pushed me to be better.

ACKNOWLEDGMENTS

This work acknowledges the work of many people who have done research in the field of Ecological Interface Design. I am grateful that they are willing to share their knowledge, tools, and expertise with the world. This work could not have been done without their help. This research would never be completed without the help of Anne McLaughlin, who provided the ball task; David Clark, my intrepid programmer; Steve Morgan, who showed me around Lee Steam; my committee members, for their guidance, Fred Switzer and Rich Pak. Finally, without the guidance, advice, and patience of my advisor, Lee Gugerty, none of this would have been possible. Thank you all.

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

Imagine you are a pilot flying a Boeing 737 passenger jet from Hong Kong to Melbourne, Australia. Imagine that the flight is going smoothly, but midway through the flight, over the Pacific Ocean, a cabin crewmember notifies you that there is a smell of fuel in the first class cabin. A fuel smell in the fuselage is a terrible sign. You check your instruments, but everything appears normal. You step into the first class cabin, and you can clearly smell jet fuel. Looking out the window you can see a thin stream of fuel leaking out of the number three engine. You rush back to the cockpit and re-check the instruments. There, engine number three is consuming more fuel than the other engines, but why? There is no indication of any malfunction. The plane is flying smoothly, thanks to automated flight controls, so there would be no cause for concern except for the smell of jet fuel slowly permeating the cabin. There is no problem now, but there could have been a disaster. How could you have missed such a possible catastrophe, you wonder. If not for the keen nose of a cabin crewmember, that problem may have gone completely unnoticed. Thankfully, the problem was caught in time and corrective measures were taken to get you, your crew, and your passengers home safely.

This might sound like an unlikely story, but it happened to a Qantas Airlines pilot in August 2006 (ATSB, 2006). There are many other stories from aviation where automation hid a problem or the pilot just did not have the information necessary to detect a problem. Sometimes these problems are caught in time, but sometimes they are not. It makes sense to provide pilots with better information to support problem detection and decision making. Therefore, it is easy to see why there is interest in using Ecological

Interface Design (EID) in aircraft (Dinadis & Vicente, 1999; Lintern, Waite, & Talleur, 1999; Rasmussen, 1999).

Not only has there been interest in applying EID to manned aircraft, but there has also been some movement to applying EID to automobiles and Unmanned Aerial Vehicles (UAVs) (Kruit, Mulder, Amelink, & van Paassen, 2005; Wang, Shen, Hou, & Yi, 2002; Still & Temme). While it makes sense to use EID in these new environments, it makes sense to first test whether the benefits of EID are applicable to those types of environments. In flying, driving, piloting a UAV and many other tasks, a system's operator must contend with a multitasking environment. An aircraft pilot cannot ignore flying a plane to diagnose an equipment problem; similarly, he or she cannot ignore an equipment problem because doing so might be very dangerous. Diagnosing equipment problems is often made more difficult by two issues. First, many equipment problems are novel to system operators, so operators have not received training in how to diagnose them. Second, equipment interfaces often present only low-level information not directly relevant to operators' task goals, thus requiring operators to infer higher-order information that *is* more relevant to their goals. EID interfaces attempts to deal with these two issues by making information presented to operators as relevant as possible and giving them the ability to make decisions based on higher-order information, thus freeing them of the need to infer higher-order information. EID interfaces are thought to support better understanding of a system and how to deal with faults, a benefit that should be especially apparent for novel problems.

It seems reasonable to expect that interfaces with these characteristics would be particularly helpful when there are many tasks and distractions are frequent. This study tested that hypothesis. EID interfaces have not often been tested in multitasking environments like the one described above. One goal of this research was to determine whether the benefits of EID would apply in multitasking environments. In addition, there is some research suggesting that EID interfaces require more use of spatial cognitive abilities, which might mitigate the benefits of EID in work domains that demand attention be divided between a task using EID and another task that uses spatial cognitive abilities.(Bowen, 2004; Pawlak & Vicente, 1996). Thus, the overall goal of this research is to determine whether the benefits of EID will apply in multitasking environments where there is a spatial task and another task implemented with EID. In this study, the EID task involves error monitoring and fault detection. This research will help determine whether it makes sense to apply the EID design framework to work domains such as driving, flying, or any other multitasking work domain with similar tasks. In the following section, I describe the EID approach in more detail and review empirical research on EID.

EID is an interface design technique that uses a combination of information content and display formatting to provide system operators the information they need to adapt to novel decision making situations. EID is a design method that attempts to display higher-order task information, thereby allowing operators to grasp higher-order concepts of how a system works, such as mass and temperature combining to give the energy contained in a system. (Vicente & Rasmussen, 1990) These higher-order concepts are shown in a way that increases a system operator's knowledge of a system, thus supporting knowledge-based behavior (Vicente, 2002). The success of EID lies in its ability to give operators information that is relevant to the underlying goal-structure of the task, meaning that operators do not have to use lower-order variables to infer a higher-order variable. For instance, an important high-level variable for an operator of a process control plant is the energy contained in a reservoir, but most non-EID displays would show only separate displays of water mass and temperature in the reservoir. EID combines the mass of the water with temperature to give energy, thus freeing the operator from determining that variable himself or herself.

Another key characteristic of EID is that it uses perceptual cues such as symmetry in determining how to present information visually. EID uses emergent features of a visual cue to make sure that the relevant information is perceptually salient. Emergent features displays use perceptual cues like symmetry, line, and angles to reveal information about a system to the operator. The first research to using the term EID was performed by Vicente and Rasmussen in 1989 (Vicente, 2002). That research was based on Rasmussen's abstraction hierarchy and his Skills, Rules Knowledge taxonomy. The abstraction hierarchy is the framework for defining the goals, the physical functions, and the constraints of a system. The Skills, Rules, Knowledge taxonomy refers to the behavior of a system's operator; it details three levels of cognitive involvement in a task.

Skill based behavior is the lowest level of cognitive involvement, where sensory-motor performance is automatic so there is a direct coupling of the individual and the environment. The inputs and outputs (responses) of the sensory-motor system are linked in real time (Rasmussen, 1983). An example of skill based behavior is lane keeping while driving (Bowen, 2004). Lane keeping is a tracking task where inputs and outputs are coupled in real time.

Rule based behavior has more cognitive involvement. An individual engaged in rule based behavior uses pre-learned if-then rules to solve problems. An example of rule based behavior is obeying traffic signs (Bowen, 2004). A driver can see a sign, interpret its meaning, and follow its direction. A driver can see a red octagon, understand the rule, then begin to slow down, which uses the skill based lane keeping and applying the brakes to slow the car.

Knowledge based behavior is the most cognitively involved of the three states. When engaged in knowledge based behavior, the individual perceives the world using symbols, and these symbols make up a mental model of how the world works in a particular situation. Using these symbols enables the individual engaged in knowledge based behavior to form a goal and work towards that goal (Rasmussen, 1983). To further our driving example, the driver of the car would use a mental model of how to get to his location (goal), and would use the stop sign as a landmark (symbol) for knowing where to turn to stay on course for achieving that goal.

One of the fundamental goals of EID is to support knowledge based behavior by allowing the use of skill and rule based behavior. In other words, the operator of a system is able to more frequently engage in skill and rule based behavior while operating the system, he or she has more cognitive resources available to solve problems using knowledge based behavior (Vicente, 2002). EID is also designed to explicitly support the operator's mental model. EID externalizes the mental model by representing it in the display, thereby decreasing error within the operator's model (Rasmussen & Vicente, 1989).

In EID, an abstraction hierarchy is used to determine what variables are relevant to operating a system. The abstraction hierarchy is a 'structural means-ends hierarchy,' which describes the links between physical components and entities (structures) as they relate to accomplishing a goal. On the top level of the abstraction hierarchy, the goal or *functional purpose* of a system is present. For a car, the functional purpose of the system is to move the occupants safely from one place to another. The next level in the hierarchy consists of *abstract functions* like energy and torque created by the engine, and the mass of the car. Below that are the *generalized functions* such as airflow and engine pressure. Next are the myriad *physical functions* of the car like engine displacement and size of wheels. At the bottom of the Abstraction Hierarchy is the *physical form*, such as size of the car or number of passengers (Burns & Hajdukiewicz, 2004).

Connecting the levels in the hierarchy are the means-ends links. Briefly, each level of the hierarchy affects the levels above it and below it. For instance the energy created by the engine is an abstract function, but it is constrained by the displacement of the pistons, the mass of air flowing into the engine, and a host of other things lower down the hierarchy. So if the end (goal) is moving people from one place to another, the means is power created by the engine, which has its own means of turning fuel and air into force, which relies on the air and fuel systems. These can, in turn, be described in physical function and physical form (Burns & Hajdukiewicz, 2004).

Creating an abstraction hierarchy allows an interface designer to break down a complex system into its many parts. Doing so reveals the constraints of the system. In the above example, a car's engine is constrained in the force it can create by how much air and fuel it can burn. It is also constrained by how much power is contained in a specified amount of fuel. Just as importantly, the abstraction hierarchy reveals, in its higher levels, the higher-order information that is essential to understanding and controlling the system. Janzen and Vicente (1998) used a modified version of the DURESS interface; separating different levels of the abstraction hierarchy on to different windows in the interface. They found that operators who spent more time in the higher-level windows performed better at controlling the simulation (Janzen & Vicente, 1998), thus demonstrating the importance of higher-order information when controlling a system. The purpose of the EID is to make these physical constraints and affordances of a system, and this higher-order information, visible to the operator of that system.

Making the constraints visible involves the actual design of the display. The advantage of EID is in the information it presents, but it is also in the manner that information is presented. EID uses emergent feature graphics (sometimes called configural displays). Emergent features graphics use easily perceptible cues like shapes and lines to demonstrate the state of a system or subsystem. These graphical cues are salient, meaning they capture attention, appearing to stand out from other features on a display. Bennett and Flach (1992, p. 514) suggest that the success of these types of displays comes from exploiting our "exquisite pattern recognition capabilities." They go on to say, "Mapping multiple process variables into a single geometric form provides high-level visual properties such as closure and symmetry. These properties can provide important information about the domain" (Bennett & Flach, 1992, p. 514).

One example of an emergent features display is from the DURESS II microworld, which consists of displays and controls to operate a 'thermal hydraulic process control simulation' (Pawlak & Vicente,1996, p. 654). In DURESS, the physical and functional (P+F) interface demonstrates the equilibrium of a reservoir's water level by showing the mass-in bar graph connected by a line to the mass-out bar graph. If the reservoir's water level is in equilibrium, the line between these two, horizontal bar graphs will be a straight vertical line. This is an emergent feature; it is perceptually salient, capturing the operator's attention. It also serves the important function of allowing the operator to glance at the display to gather that the water level is in equilibrium. The converse is also true, if the system is not in equilibrium, it will be obvious to the system operator that it is not. If the mass in is greater than the mass out then the bar graph for mass in is farther out than the bar graph for mass in, thus the line becomes slanted towards the side with the greatest mass-flow. Figure 1.1 shows this effect of the change in mass-in compared to mass-out in a DURESS reservoir.

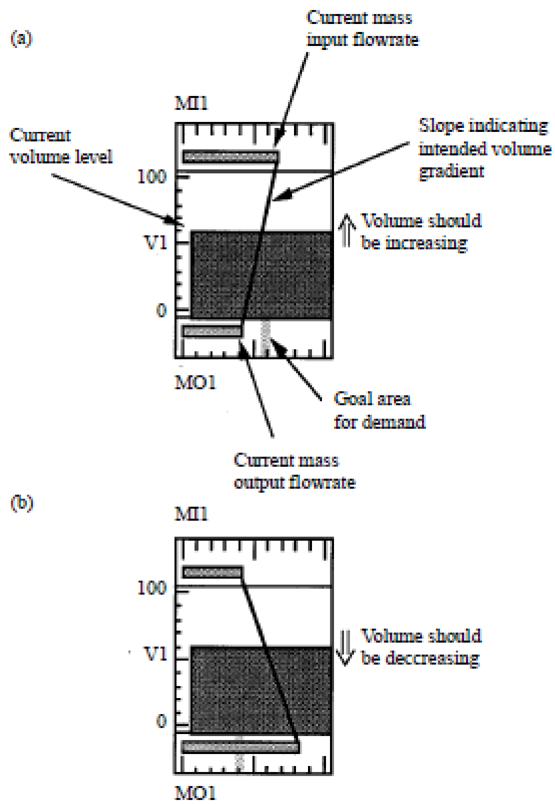

Figure 1.1 Close up view of the mass balance for a reservoir in DURESS.

From Pawlak and Vicente (1996)

It should be noted, however that the advantages of EID are not purely due to visual formatting, either. Xu, Dainoff, and Mark (1999) applied the abstraction hierarchy to a hypertext database and found that organization of the database based on the abstraction hierarchy was greatly preferred to the normal hierarchical organization system. Ham and Yoon (2001) performed an experiment using the physical (lowest) level of the abstraction hierarchy for a power plant cooling system; comparing that to the physical plus generalized function level (mid-level) or the physical plus abstract function level (higher level). All three displays used similar bar graphs, but the physical plus generalized function level was superior to the other two levels under fault conditions (Vicente, 2002). Although the generalized function level performed better than the abstract function level, it should be noted that both outperformed the physical only level. These benefits were most apparent during complex trials (Vicente 2002). It can be inferred the generalized function level and the abstract function level were better than the physical only level because of the difference in information, not form, since all levels of the display used similar form.

Hajdukiewicz and Vicente (2002) tried to separate out what effects of EID were due to display formatting and which were due to the information content presented. They used high level system changes and low level system changes for the P (non-EID) and P+F (EID) interfaces of DURESS II. They found that both display formatting and information content were important to EID’s success. They suggest that displaying functional information using emergent features graphics allows operators to react to information at a basic perceptual level (Hajdukiewicz & Vicente, 2002).

To summarize, the EID framework uses the abstraction hierarchy to determine what information is important in the skills, rules, knowledge taxonomy, and uses emergent features graphics to make the higher-order information salient. This raises a question. Does relying on the visual-spatial nature of emergent features graphics mean that the visual-spatial cognitive resources of an operator must be available for the benefits of EID to be seen?

Pawlak and Vicente (1996) used a secondary task that was either verbal or spatial to determine which cognitive resources EID loaded more. Their experiment used a loading task methodology. Participants were given a spatial loading task or a verbal loading task while controlling the P or P+F DURESS II interface. They found performance with the P interface was more adversely affected by the verbal loading task, while the performance of the P+F interface was more adversely affected by the spatial task. This suggests that the P+F (EID) interface uses the spatial resources of the operator, while the P (non-EID) interface uses the verbal resources more (Pawlak & Vicente, 1996).

Bowen (2004) performed an experiment using DURESS II that measured the verbal and spatial abilities of participants, a priori. He also measured holist cognitive scores using the Spy Ring History test, to control for Holist cognitive style, which has been shown to predict performance using EID (Torenvliet, Jamieson, & Vicente, 2000). Bowen's results demonstrate that EID is superior during normal trials, which had not been observed previously, and during fault trials. Under normal and fault conditions, those with higher spatial ability performed better using the P+F interface, but there was no correlation between spatial ability and performance for the P interface.

The findings of Pawlak and Vicente (1996) and Bowen (2004) are troubling to those who seek to apply EID to domains where there are two or more tasks to accomplish simultaneously, and doubly troubling when some of those multiple tasks are visual-spatial in nature. However, the benefits of EID in fault detection and diagnosis may still be evident, even in these multitasking domains. EID can claim several different reasons for the benefits it can provide. These reasons boil down to information content and display formatting. The information content provided by EID is determined by the abstraction hierarchy. That information is designed to be ‘transparent’, revealing the inner working of the system. Often these inner workings, for example, physical laws governing energy transfer, cannot be seen by the human eye. EID makes these constraints on the workings of a system visible. EID also supports the building of a correct mental model of system layout and function. Therefore, EID supports knowledge based behavior. Resources required to solve problems are freed when using EID, which is why EID consistently demonstrates superior performance during fault trials.

The other advantage of EID is display formatting; EID makes use of emergent features graphics to make important system information salient to the operator. Salience is important because it allows an operator to see relevant information at a glance. It also promotes understanding of the system’s state.

For these reasons, the main hypothesis of this experiment is that EID will outperform normal interfaces, even in multitasking domains. In fact, the greatest benefit may be seen in multitasking work domains where system operators do not have time to integrate much information before making a decision. Stress is known to decrease performance, and the multitasking work domain can be especially stressful (Wickens & Hollands, 2000). Often the chief source of stress is the work domain itself (Hancock & Szalma, 2003). One of the ways stress reduces performance is through the narrowing of information resources used to make decisions (Hancock & Szalma, 2003). Many displays for multiple task work domains (again, like driving and flying) could, at best, be considered single sensor single instrument displays, which are the type of display that are least relevant and useful to human operators (Vicente & Rasmussen, 1990). Often, the multitasking work domain has a very great risk associated with incorrect judgments, heightening the stress and the need for well-supported decision making and performance.

Research Design and Hypotheses

The chief research question of this research project was whether the benefits of EID would generalize to these multitasking environments where decisions must be made quickly, without sacrificing the quality of performance for either task. This question was examined by having participants simultaneously perform both a visual-motor task and a task involving error monitoring and fault detection. The visual-motor task was the ‘falling ball’ task where participants must try to catch falling balls on a computer screen. There were two levels of ball task difficulty, determined by the rate of balls falling on the screen. The error monitoring task used the DURESS II system, where participants monitored the reservoir system and reported any faults.

The DURESS II system is a process control simulation used to simulate a dual reservoir system where there are different mass and temperature demands for each reservoir. The goal of the system is output the specified amount of water from each reservoir at the specified temperature. This is done through controlling a system of valves, pumps, and heaters. There are two main interfaces for DURESS II. The P interface shows only the physical level of the abstraction hierarchy. The P+F interface shows the physical and functional levels of the abstraction hierarchy. The P+F interface is an example of EID. The DURESS II program used here has been modified by the addition of an error detection button. This allows participants to stop the simulator when they detect a fault in order to report the fault.

Supervisory control of a complex system like DURESS can be seen as comprised of three main components: monitoring and fault detection, fault diagnosis, and correction. Fault diagnosis and correction have been intentionally avoided in this experiment to reduce the complexity of the task. The monitoring and fault detection part of the supervisory control task is the main focus of this research. Participants' performance at monitoring and fault detection was measured by the speed and the accuracy with which they detect faults in DURESS.

The first two hypotheses are not theoretical in nature; they are mostly used as manipulation checks. The first hypothesis was that the percentage of balls caught would decrease as ball speed increased, because faster ball speeds result in fewer caught balls under single task circumstances. This effect would demonstrate that increasing the ball task speed was a good manipulation of workload.

In order to ensure that participants did not ignore the DURESS task to attend to the ball task at the higher ball speed, participants in both display conditions were instructed to give approximately equal attention and effort to the ball and the DURESS task at all times, and they were promised a monetary reward if they did so. Therefore, the second hypothesis was that the percentage of balls caught in the ball task would be unaffected by the display condition.

Given the theoretical arguments and empirical evidence for EID advantages presented above, the third hypothesis was that the participants in the P+F display condition would be faster and more accurate at detecting errors in DURESS than their counterparts in the P display condition. The fourth hypothesis was that increasing the speed of the ball task would decrease the speed and accuracy of detecting DURESS errors. This hypothesis follows the logic that increases in ball task speed are salient to the participants and will therefore capture their attention. Given the argument above that EID may be especially advantageous in multitasking situations, the fifth hypothesis was that the speed and accuracy of detecting DURESS errors would be less negatively affected by increasing the speed of the ball task in the P+F condition than in the P condition.

METHODS

Participants

Twenty Clemson University Undergraduate students were be used as participants in this experiment, 9 females 11 males. Ages ranged from 18 to 22 years old. Mean age was 18.9 years of age. They were all considered novices at the task.

Design

This experiment used a mixed factorial design. The display independent variable, which had two conditions, P and P+F, was between subjects. The speed of the ball task independent variable, which had two levels, slow and fast (15 pixels per second and 45 pixels per second) was within subjects. There were 9 (6 male, 3 female) participants in the P condition and 11(5 male, 6 female) participants in the P+F condition. Participants were assigned alternately to the P or P+F group based on their scheduled times.

Tasks

DURESS II

The purpose of DURESS is to simulate a dual-reservoir system where the goal is to output a specified volume of water from one reservoir at a specified temperature, and a different volume of water from a second reservoir at a different temperature (Pawlak & Vicente, 1996). Both reservoirs have a single feed, which is split into an upper and lower string. Each string has its own pump (PA and PB in Figure 2.1). After each pump is a valve (VA and VB), after which the strings are split again. The upper valve in each string feeds the upper reservoir (VA1 and VB1), while the lower valve in each string feeds the lower reservoir (VA2 and VB2). Each reservoir has its own temperature controls and output valve.

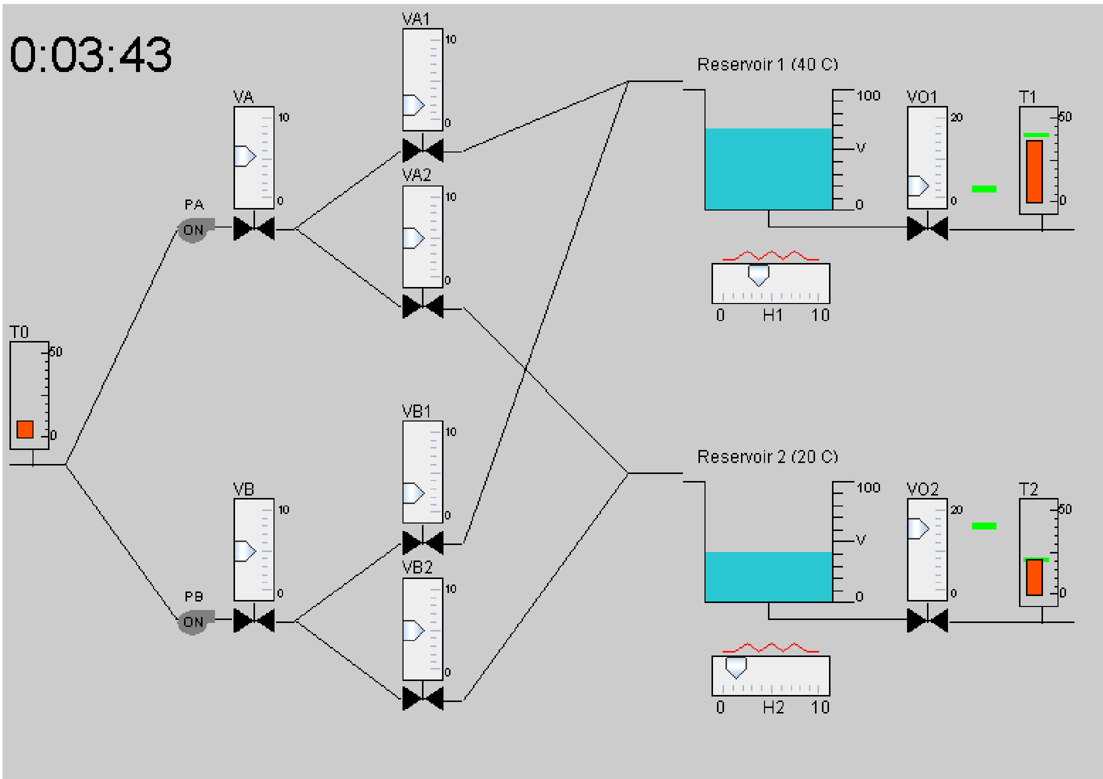

Figure 2.1. The P interface of DURESS II

There are two main interfaces in DURESS II. The one shown in Figure 2.1 is the P interface. This shows the physical function level of the abstraction hierarchy. Figure 2.2 shows the P+F interface which shows the physical and functional levels of the abstraction hierarchy. The two interfaces differ in the information content presented and the manner or form of presentation. The P+F interface has noticeably more information about flow rates of all valves. It also has information about the energy contained in a reservoir outside of the temperature. Other P + F features to note are the mass and energy balances for each reservoir. These provide information about how much water is in the reservoir, and how much is flowing in and out. This makes for easy comparison, since there is a vertical line connecting the flow in and flow out bar graphs, if the mass is in equilibrium then the line is vertical (an emergent feature). The same can be said for the energy balance. The triangular balance in between the mass and energy balances is the temperature balance, demonstrating the connection between mass and temperature: the greater the mass, the greater the energy for a given temperature. Therefore if mass increases, the angle of the line segment increases, showing a greater level of energy if temperature remains constant. This information – mass, energy and temperature balance – is the higher order information that the abstraction hierarchy defines as relevant to operators' overall goals.

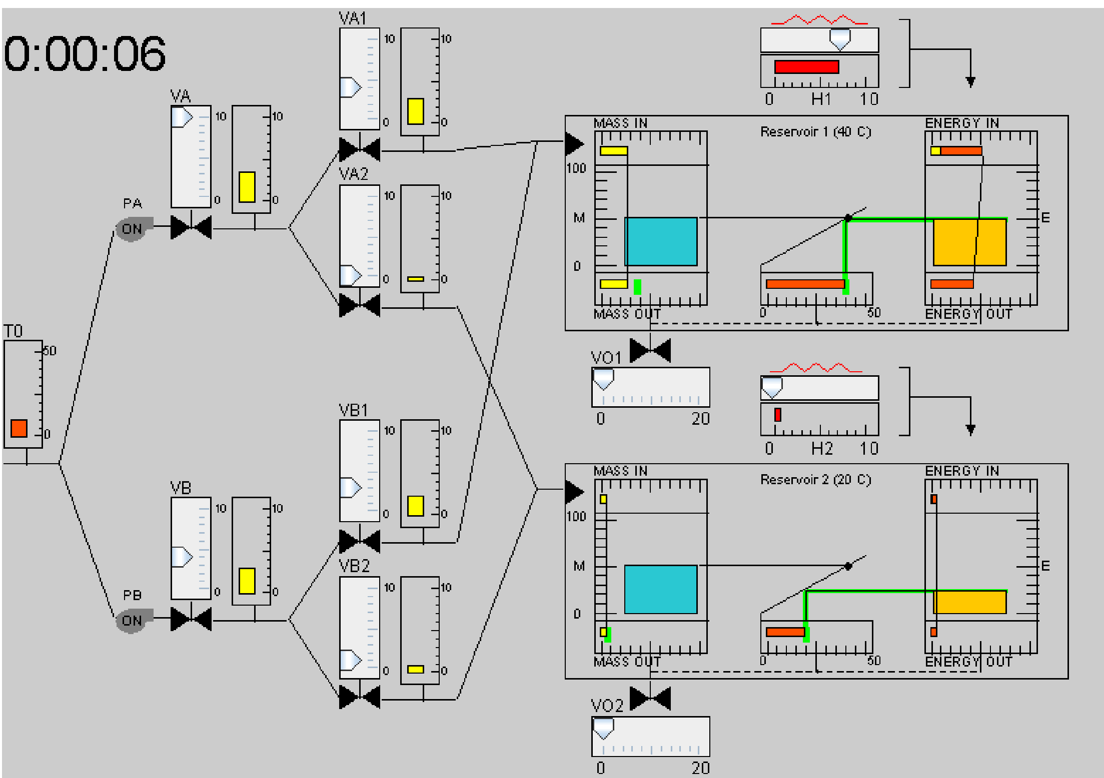

Figure 2.2. The P+F DURESS II interface

In contrast to the P + F interface, the P interface does not show any information about mass, energy or temperature balance. The P interface only shows the physical state of the system. For instance the P interface shows the valve settings, but does not show the flow rate through the valves. The P interface shows the mass of water contained in the reservoir, but it does not show the balance of mass moving in or out of the reservoir.

This experiment uses the Java version of DURESS II supplied by the University of Toronto's Cognitive Engineering Laboratory. This experiment uses standard DURESS II configuration files to control the sequence of events during a scenario (i.e., changes in valve settings and flow rates). The participants did not control the DURESS II interface in this study. Instead, during a scenario, participants monitored the interface for system errors such as a blocked valve. Participants were able to detect errors through information relayed by the interface (e.g. a decrease in flow into a reservoir). When they detected an error, participants pressed the spacebar. The participants then pressed an on screen button to move on to the next scenario. If the participant did not detect an error, the scenario stopped (after 3 minutes) and the participant moved on to the next scenario.

Each DURESS scenario began with the system going to a state of equilibrium. The faults in the DURESS task occurred at predetermined times within the program. Some scenarios did not have any errors; in those cases, the scenario ended after three minutes, if the participant did not report an error. Each participant did 15 scenarios, 12 of which had faults. Each scenario lasted a maximum of 3 minutes. Faults occurred from 0.5 to 2.27 minutes into a scenario.

Data recorded from the DURESS task were the participants' reporting of a fault (by space bar press) and the time to detect faults from fault onset time. Data were sampled ten times per second.

Ball Task.

The ball task is a simple visual psycho-motor task (see Figure 2.3). This task had two different rates of balls falling past the screen: slow and fast; 15 pixels per second and 45 pixels per second, respectively. Seven balls at a time were moving from the top to the bottom of the screen. The participant's goal was to catch as many of the balls as possible with a block cursor at the bottom of the screen. To catch a ball, the participant had to move the block beneath a ball just before it moved off the bottom of the screen. The screen was divided into ten columns. The block cursor was controlled using the arrow keys on the keyboard. For instance, when the left arrow key was pressed the block moved one column to the left. When a ball was caught the block changed color from black to green for approximately half a second. When a ball was caught or missed, another ball replaced it at the top of the screen in a randomly selected column. The number of balls caught and missed was recorded. All key presses were recorded, as were the block and ball locations. The data were sampled at ten times per second. A ball-task trial ended when the spacebar was pressed to detect a fault in the DURESS scenario, or when a DURESS scenario timed out. The ball task restarted when a new DURESS scenario began. No feedback regarding the ball task was given between scenarios.

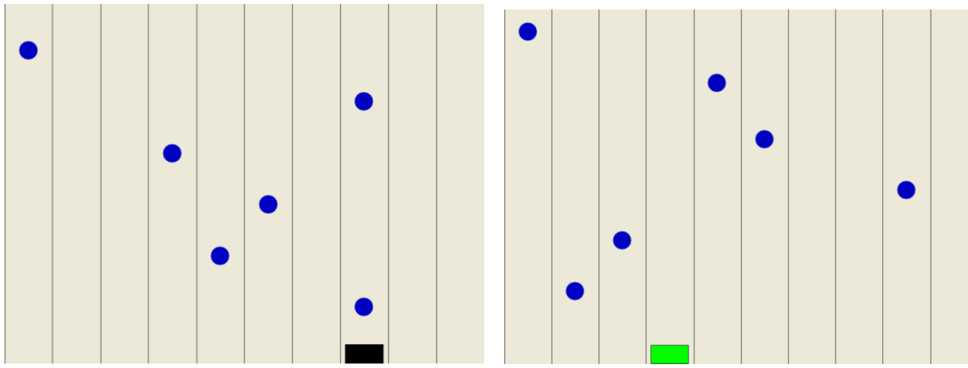

Figure 2.3. The ball task showing the cursor in normal state (left) and caught state (right)

<u>Materials and Dual-Task Configuration</u>

The tasks were administered on an IBM personal computer. The computer was equipped with 2 3GHZ Intel Pentium 4 processors, 1 gigabyte of RAM, and 80 gigabytes of storage memory. The monitors were 15-inch monitors running at 1024x 768 pixels resolution. Two monitors were used for the computer. The input devices were a standard mouse and keyboard. The ball task was presented on the left screen and DURESS was presented on the right. See Figure 2.4 for workstation layout.

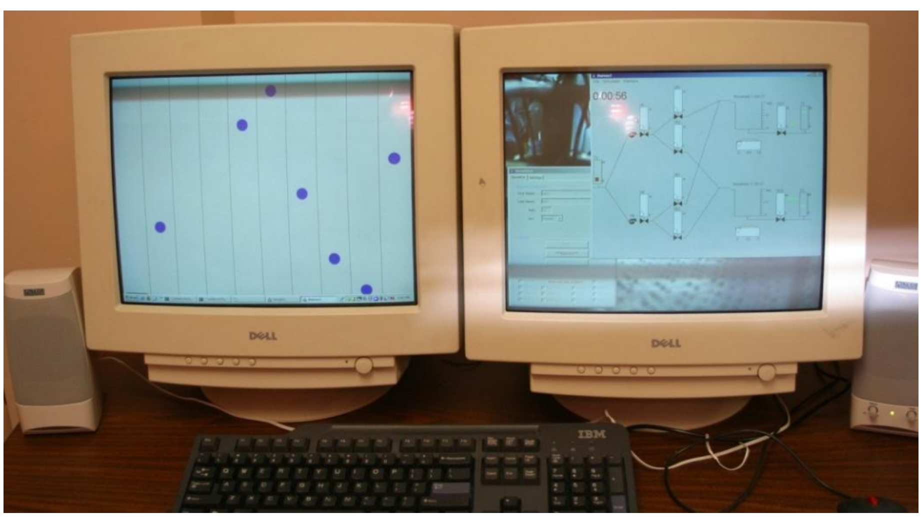

Figure 2.4. The workstation used by participants

Participants sat in between the two monitors with a keyboard in front of them. The two tasks were administered at the same time. When the participants hit the space bar after detecting a fault in DURESS, the ball task stopped. Participants then hit the 'next scenario' button on screen using the mouse and a new scenario started in DURESS. At the same time, a new trial in the ball task began.

The 15 scenarios controlling the DURESS interface were assigned a ball task speed and ordered so there was no confound between task difficulty, timing of faults, or order of scenarios. Each scenario was always presented with the same ball task speed for all participants. All participants saw the scenarios in the same order. All scenarios used in the trials had been pilot tested. In the pilot testing, 5 participants completed the 15 DURESS scenarios (along with other scenarios) without doing the ball task. The pilot data were also used to determine the perceived 'size' of the fault in each scenario. Size was determined subjectively through the experience of the experimenter and the pilot participants. A 'big' fault was more immediately noticeable than a 'small' one because of the rapid shift in display elements. The scenarios also varied in the timing of the fault (early vs. late). Early faults were probably easier than late faults because the effects of early faults had more time to become larger and more salient. The difficulty and the timing of the fault were balanced to ensure that difficulty and timing were distributed equally across the two ball speeds. For the 6 fault trials assigned to the slow ball speed, there were 2 small and 4 big faults, and 4 early and 2 late faults. For the 6 fault trials assigned to the fast ball speed, there were 2 small and 4 big faults, and 4 early and 2 late faults.

The pilot data were also used to determine the difficulty of each DURESS scenario as measured by average fault detection time. For the trials assigned to the slow ball speed, there were 4 incorrect trials (out of 27 trials); for the trials assigned to the fast ball speed, there were 7 incorrect trials (out of 30 trials). An incorrect trial was defined as a false alarm or a miss. Thus, the DURESS scenarios paired with the fast ball speed may have been more difficult than those paired with the slow ball speed.

Procedure

Participants were run one at a time. The participants received an informed consent form to sign, and then were given a demographic questionnaire. After completing the questionnaire, the participants were informed of the potential to receive a monetary bonus for good performance. They then began PowerPoint based training, tailored to either the P interface or the P+F interface, depending on which condition they were assigned. After completing the PowerPoint training slides, the participants did five practice scenarios using only the DURESS interface. They were informed they should monitor the interface for faults and report them by pressing the spacebar. The first trial was a non-fault trial. Participants were informed of this and instructed to watch the scenario for the full three minutes to become familiar with DURESS as it fluctuated. Participants were told the remaining four trials may or may not have faults. They were given feedback regarding whether they were correct in detecting a fault or not.

Next, the participants did three practice sessions with the ball task (as a single task); two sessions with the fast speed and one with the slow. Each session lasted three minutes. The participants were given feedback on their percentage of balls caught. The final group of practice runs was five combined DURESS and ball task scenarios. After each scenario the participants were informed of their DURESS and their ball task performance.

After the practice period participants were given final instructions from the experimenter. They were reminded to perform each task to the best of their ability, and to divide their attention and effort equally between the two tasks. They were told if they were able to do so, they would be given a monetary bonus. In the experimental session, participants did 15 scenarios in which they performed the DURESS and ball tasks together.

Each experimental trial lasted a maximum of three minutes. After completing the experiment, the participants were be thanked for their time and debriefed. All participants were given the performance-based monetary bonus, and all received class credit for participating in the experiment.

Dependent Variables

For the ball task, the percentage of balls caught in each scenario was measured.

DURESS error detection performance was measured two ways: by accuracy, P(A), and Speeded Sensitivity. P(A) is a measure of signal detection accuracy that averages the proportion of hits with the proportion of correct rejections. A hit was counted when a participant reported a fault on a signal trial (a trial where a fault occurred) after the fault occurred. Because a participant could have responded before a fault had occurred, all 15 trials (both fault and catch trials) had the potential to be false alarm trials. All 15 trials could also have been correct rejections. However, only 12 of the trials (the ones containing faults) could have been hits or misses. A trial was only considered a fault trial if the participant actually saw the fault occur. In other words, fault trials where the participant reported an error before the fault occurred (a false alarm) were not counted a fault trials.

There are two components to the Speeded Sensitivity variable; 1. the amount of time elapsed before signals occurred or a catch trial ended, and 2. the amount of time signals were present. Component 1, the elapsed time before signals or the end of catch trials, was calculated for all trials. For signal (fault) trials, if there was no false positive before the signal occurred, then the time the signal occurred was added to component total. If a signal trial was a false positive, then the response time was added to the component total. For catch trials, if there was no false positive, the trial duration was added to the component total, but if there was a false positive, the response time was added to the component total. Finally, the component total was divided by the number of trials (15) to give the average the elapsed time before signals or the end of catch trials. A high average elapsed time before signals or end of catch trials would occur when false positives were few and when any false positive that did occur happened later rather than sooner.

Component 2, the elapsed time that signals were present, was calculated only for trials where signals (faults) actually occurred (i.e., catch trials and false positives before signals occurred were excluded). If the fault trial was a hit, then the fault detection time, the difference between the response time and the signal onset time, was added to the component total. If the trial was a miss, then the difference between the end of the trial and the signal onset time was added to the component total. The component total was then divided by the number of trials where signal was present (maximum of 12), to give the average time signals were present. A lower average time signals were present indicated a faster response to signals.

The mean elapsed time before signals or the end of catch trials (Component 1) and the mean time signals were present (Component 2) were then normalize into Z scores for each participant. The Z score for time signals were present (Component 2) was multiplied by -1 so that a greater Z score reflected better performance (as was already the case for Component 1). For each participant, the Z scores for Components 1 and 2 were then averaged to create Speeded Sensitivity. A high speeded sensitivity score indicated a fast, accurate response to faults and a slow or no response when faults were not present.

RESULTS AND DISCUSSION

An alpha level of .05 was used to determine statistical significance. Effect size was quantified in terms of semi-partial η^2, which estimates the proportion of variance in a dependent variable accounted for by an independent variable. According to Cohen (1977), percentage of variance accounted for of .01, .06, and .138 represent small, medium, and large effect sizes, respectively.

The first hypothesis was that the percentage of balls caught would decrease as ball speed increased. In the slow condition the mean percentage caught was 93.5% (SD = 6.6). In the fast condition the mean percentage caught was 48.5% (SD = 7.5). This supports the hypothesis, as an ANOVA showed a significant effect of ball speed on percentage caught and a very large effect size, $F(1,18) = 2937$, $p < .05$, semi-partial η^2 =.99. The decrease in percent balls caught suggests that increasing the speed of the ball task did increase overall workload, as intended.

The second hypothesis was that the percentage of balls caught in the ball task would be unaffected by the interface type. The mean percent caught was 72.5% for the P interface (SD = 24.3), and 69.5% (SD = 23.8) for the P+F interface. An ANOVA revealed no significant of interface type on percentage of balls caught, but a small effect size, $F(1,18) = 1.00$, $p = .33$, semi-partial η^2 =.05. As Figure 3.1 shows, there was no interaction of interface and ball speed on percentage of balls caught, $F(1,18) = 1.52$, p = .23., semi-partial η^2 = .00. These null effects suggest that participants did not place different emphasis on the ball task in the P versus the P+F condition. Therefore, any effects of interface type found when testing later hypotheses should be due to differences in the interfaces and not due to variation in dual-task attention allocation across interface conditions.

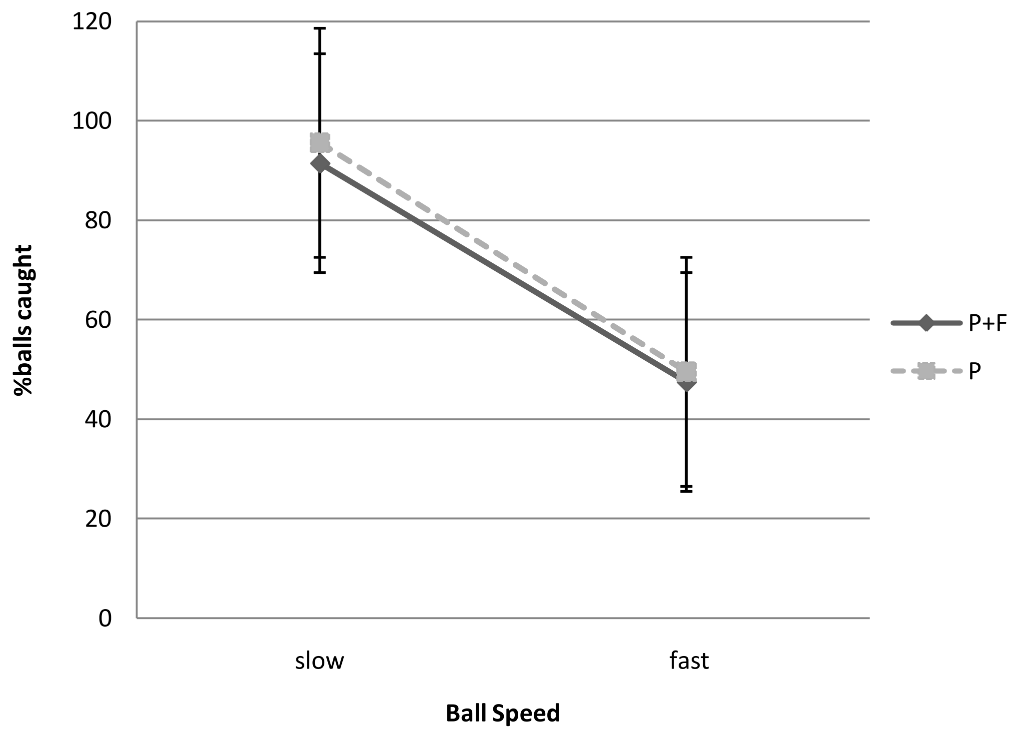

Figure 3.1. Percent balls caught by ball speed for each interface type, with standard error bars

The third hypothesis was that the participants in the P+F display condition would be faster and more accurate at detecting errors in DURESS than their counterparts in the P display condition, irrespective of ball speed. The mean P(A) for the P interface was 0.96 (*SD* = 0.05), while for the P+F interface it was 0.93 (*SD* = 0.09). This does not support the hypothesis, as shown by an ANOVA that revealed no significant effect of interface on P(A), $F(1,18) = 1.39$, $p = .25$, semi-partial $\eta^2 = .07$. Though non-significant, these data suggest a medium sized effect of interface that is the opposite of the predicted P+F advantage.

The mean Speeded Sensitivity was 0.16 ($SD = 0.43$) for the P interface and 0.13 ($SD = 0.65$) for the P+F interface, which does not support the hypothesis. An ANOVA revealed no significant effect of interface on Speeded Sensitivity, $F(1,18) = 2.804$, $p = .11$, semi-partial $\eta^2 = .13$. As with P(A), these data suggest a medium sized though non-significant effect of interface, opposite of the hypothesis.

The fourth hypothesis was that increasing the speed of the ball task would decrease the speed and accuracy of detecting DURESS errors, irrespective of interface type. The mean P(A) across interface types for the slow ball speed was 0.93 ($SD = 0.07$), while for the fast ball speed it was 0.96 (SD = 0.07). This does not support the hypothesis. An ANOVA revealed an effect of ball speed on accuracy that approached significance, $F(1,18) = 3.64$, $p = .07$, semi-partial $\eta^2 = .14$. This is a large effect size in the opposite direction of the predicted effect of ball speed and accuracy.

The mean Speeded Sensitivity for the slow ball speed was -0.001 ($SD = 0.46$), and the mean for the fast ball speed was 0.03 ($SD = 0.68$). Again, this effect is the opposite of the hypothesis. An ANOVA revealed that the effect of ball speed on Speeded Sensitivity was not significant, $F(1,18) = 0.03$, $p=.87$, semi-partial $\eta^2 = 0.001$. Thus, ball speed had almost no effect on Speeded Sensitivity.

The fifth hypothesis was that the speed and accuracy of detecting DURESS errors would be less negatively affected by increasing the speed of the ball task in the P+F condition than in the P condition. As Figure 3.2 shows, P(A) increased slightly with ball speed for the P interface, and P(A) did not change much with ball speed for the P+F interface. This is the opposite of the hypothesis. An ANOVA revealed that this interaction approached significance and showed a large effect size, $F(1,18) = 3.96$, $p=.06$, semi-partial $\eta^2 = .16$.

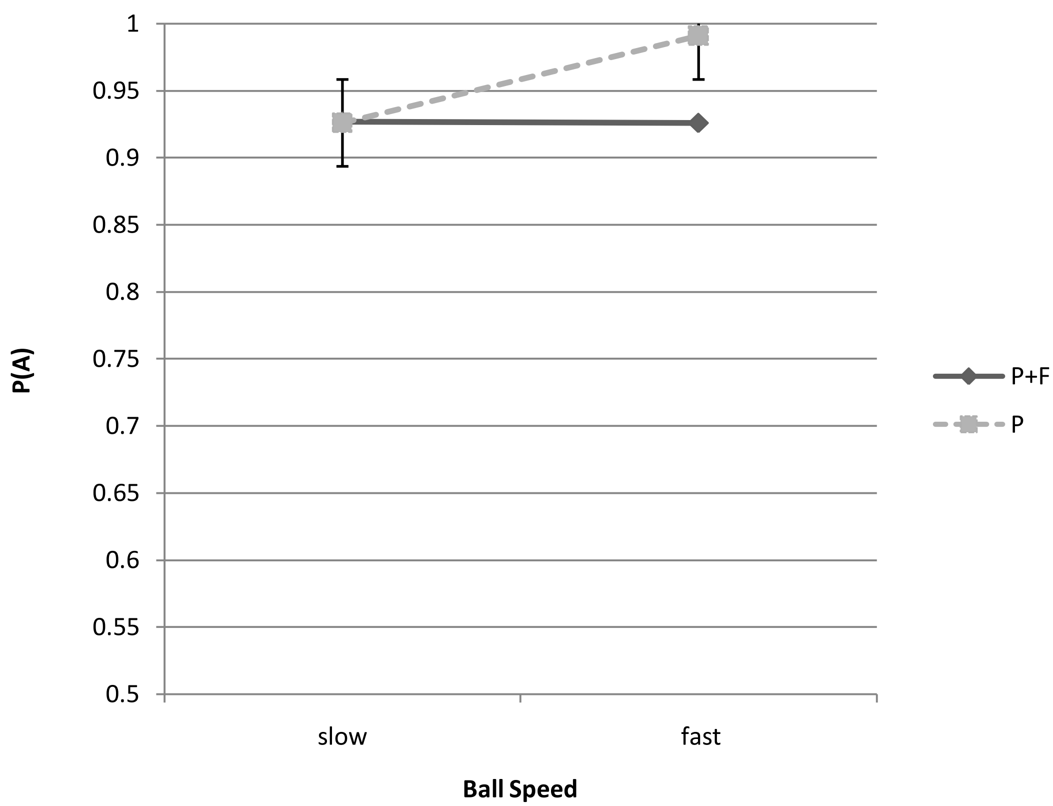

Figure 3.2. Accuracy by ball speed for each interface type, with standard error bars.

As Figure 3.3 shows, Speed Sensitivity increased with ball speed for the P interface, and decreased with ball speed for the P+F interface. Both of these trends are counter to the hypothesis. An ANOVA revealed no significant interaction between ball speed and interface for Speeded Sensitivity, but a large effect size, $F(1,18) = 2.94$, $p = .10$, $\eta^2 = .14$.

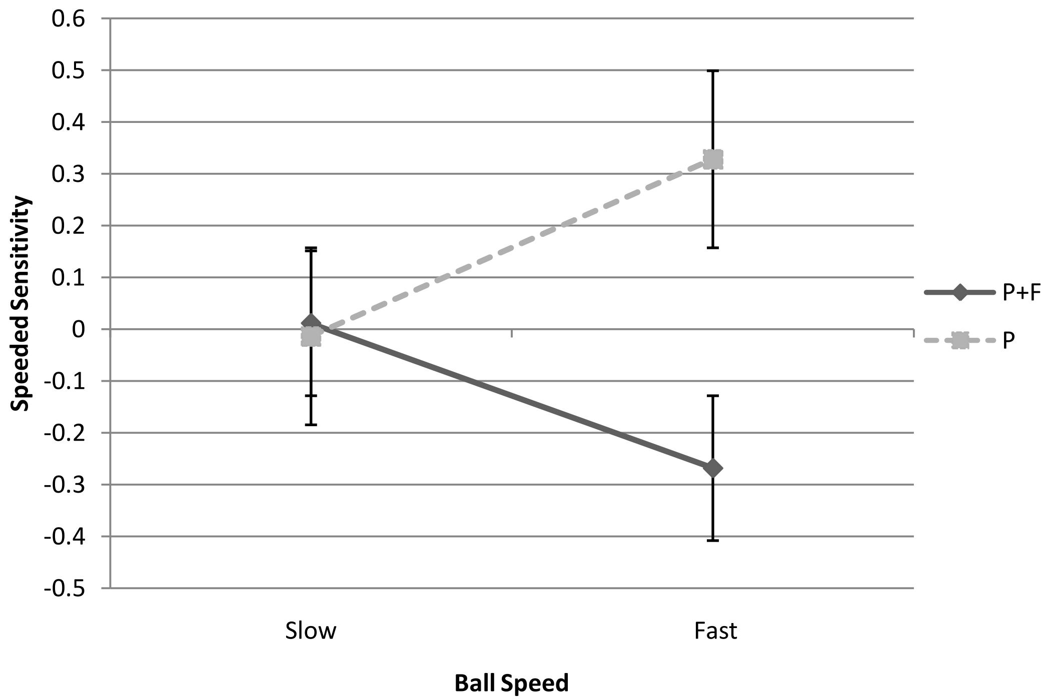

Figure 3.3. Speed by ball speed for each interface type, with standard error bars.

CONCLUSIONS

To summarize the results, contrary to predictions, participants using an EID interface were not faster or more accurate at detecting faults when compared to participants using a non-EID interface. In fact, the non-EID interface group was faster and more accurate than the EID interface group, but not significantly so. Also contrary to predictions, increasing in ball-task workload did not reduce the speed and accuracy at which participants detected faults in the DURESS system. Finally, the results did not support the predicted interaction in which the EID interface led to a smaller decline in DURESS performance with increasing workload than for the non-EID interface

While this experiment had few conclusive results, there are some potential inferences that could be drawn from it. First, I will consider reasons why the expected advantage of the EID over the non-EID interface was not found. The multitasking environment is one with special needs and constraints. It requires operators to be able to make decisions both quickly and accurately. Supporting those decisions may take special information. One possible reason that the P+F interface did not do well here was the lack of domain knowledge or training. In all previous EID literature, the participants were either well educated in the domain or given substantial training. This experiment intentionally avoided expert participants or a long training process. In order for EID to be useful in applications such as passenger cars, it must be useful with minimal training. However, using EID in areas where operators are well trained (such as pilots) could still make sense.

Training for the P+F group was also much longer than for the P group since the P+F interface had more features that required introduction. The greater complexity of the P+F interface, coupled with the longer training for this interface, may have left participants in the P+F condition confused, relative to participants who used the simpler P interface. The possible greater confusion for the P+F interface after the brief training used here could explain, at least partly, why the predicted P+F advantage was not found. Again, if EID must rely on training then it would be unfeasible to use for the general population.

Another possible reason that EID did not perform as well was that the greater complexity of the P+F interface may have led to greater cognitive narrowing to a few cues in this condition than in the P condition. Greater workload is known to reduce the number of information sources that a system operator focuses on for information regarding the work environment. (Hancock & Szalma, 2003). Since the complexity of the P+F interface could be overwhelming to operators, they could have compensated by focusing on a few of the many sources of information available. Anecdotal reports from the participants suggest that some degree of cognitive narrowing occurred with the P+F interface in this study. For example, when asked where he/she found a fault many participants answered the same part of the interface each time (e.g. the energy balance). If cognitive narrowing were the reason that the P+F interface fared poorly, then EID would never be successful in multitasking domains, because it requires greater complexity, which can overwhelm operators.

A third possible reason for the poor performance of EID has to do with the fact that EID interfaces, more than non-EID interfaces, require good spatial ability of operators. (Bowen, 2004; Pawlak & Vicente, 1996). .If most of the participants in this study had low spatial ability, the P+F group would be more negatively affected than the P group. This scenario could be prevented by testing the spatial ability of operators to control for performance. However, if performance with EID interfaces is markedly inferior for operators with low spatial ability, then EID would not be a viable option for the general public.

A final explanation for the poor EID performance in this study could be the gender imbalance between the conditions, in which there was a predominance of males (6 of 9) in the P condition but not in the P+F condition (5 males, 6 females). Overall males performed better than females; males were faster and more accurate in the P and P+F conditions, but not significantly so. The P interface condition had a greater percentage of males, which could explain why the P interface did better than expected. However, the males in the P condition were faster and more accurate than the males in P+F condition, and the females in the P condition were faster and more accurate than females in the P+F condition (again, not significant). So, the results are unlikely to be attributable to gender. Further experiments should use better gender balancing to remove this confound.

Another surprising finding of this study was the lack of an expected decrease in DURESS performance as dual-task workload increased. As the ball-task speed increased, ball task performance (as measured by percent correct) declined as expected, but DURESS performance did not decline. Perhaps DURESS performance did not decline because participants perceived the DURESS task to be more important and therefore allocated less attention to the ball-task when its workload increased in order to maintain adequate performance on the DURESS task. In other words, participants may not have been able to follow the instructions to allocated equal attention to the ball and DURESS tasks at all times. Evidence for this explanation comes from research showing that in dual-task situations, people may de-emphasize a lower-priority task in order to maintain adequate performance on a higher-priority task (Wickens, 1980).

More research needs to be completed, as EID could provide great benefit, or it could pose a great risk. Many multitasking work domains are complex, and have high-stakes outcomes, so more research is absolutely necessary before declaring EID to be 'the answer' or not. EID remains a promising solution to increasing the awareness of human operators in complex multitasking environments, but this experiment shows it may not be 'the' answer.

REFERENCES

Australian Transport Safety Bureau (2006). ATSB TRANSPORT SAFETY INVESTIGATION REPORT, Aviation Occurrence Report – 200604807

Bennett, K. B., & Flach, J. M. (1992). Graphical Displays: Implications for Divided Attention, Focused Attention, and Problem Solving. *Human Factors, 34* (5), 513-533.

Bowen, S.A.M. (2004), The Effects of Spatial Ability on Performance with Ecological Interfaces: Mental Models and Knowledge-Based Behaviors. *Dissertation Abstracts International*, 65/06, 3204. (UMI No. AAT 3134674).

Burns, C. M., & Hajdukiewicz, J. R. (2004). *Ecological Interface Design.* Boca Raton, Florida: CRC Press.

Christoffersen, K., Hunter, C. N., & Vicente, K. J. (1998). A longitudinal study of the effects of ecological interface design on deep knowledge. *International Journal of Human-Computer Studies, 48*, 729-762.

Cohen, J. (1977). Statistical power analysis for the behavioral sciences (Rev. ed.). New York: Academic Press.

Dinadis, N., & Vicente, K. J. (1999). Designing Functional Visualizations for Aircraft Systems Status Displays. *International Journal of Aviation Psychology, 9* (3), 241-269.

Hajdukiewicz, J. R., & Vicente, K. J. (2002). Designing for Adaptation to Novelty and Change: Functional Information, Emergent Feature Graphics, and Higher-Level Control. *Human Factors, 44* (4), 592-610.

Hancock, P. A., & Szalma, J. L. (2003, January). Operator Stress and Display Design. *Ergonomics in Design*, 13-18.

Janzen, M. E., & Vicente, K. J. (1998). Attention allocation within the abstraction hierarchy. *International Journal of Human-Computer Studies, 48*, 521-545.

Kruit, J. D., Mulder, M., Amelink, M., & van Paassen, M. M. (2005). Design of a Rally Driver Support System using Ecological Interface Design Principles. *2005 IEEE International Conference on Systems, Man and Cybernetics. 2*, pp. 1235- 1239. IEEE.

Lintern, G., Waite, T., & Talleur, D. A. (1999). Functional Interface Design for the Modern Aircraft Cockpit. *The International Journal of Aviation Psychology, 9* (3), 225-240.

Pawlak, W. S., & Vicente, K. J. (1996). Inducing Effective Operator Control Through Ecological Interface Design. *International Journal of Human-ComputerStudies, 44*, 653-688.

Rasmussen, J. (1999). Ecological Interface Design for Reliable Human-Machine Systems. *International Journal of Aviation Psychology, 9* (3), 203-223.

Rasmussen, J. (1983). Skills, Rules, and Knowledge; Signs, Signals, and Symbols, and Other Distinctions in Human Performance Models. *IEEE Transactions on Systems, Man, and Cybernetics, SMC 13* (3), 257-267.

Rasmussen, J., & Vicente, K. J. (1989). Coping with human errors through system design: implications for ecological interface design. *International Journal of Man-Machine Studies, 31* (5), 517-534.

Still, D. L., & Temme, L. A. (n.d.). *OZ: A Human-Centered Cockpit Display*. Retrieved October 5, 2006, from Institute for Human and Machine Cognition (ihmc.us): http://www.ihmc.us/research/projects/OZ_UCAV/downloads/OzHumanCenteredComputing.pdf

Torenvliet, G. L., Jamieson, G. A., & Vicente, K. J. (2000). Making the Most of Ecological Interface Design: The Role of Indivdual Differences. *Applied Ergonomics*, *31*, 395-408.

Vicente, K. J. (2002). Ecological Interface Design: Progress and Challenges. *Human Factors*, *44*(1), 62-78.

Vicente, K. J., & Rasmussen, J. (1990). The Ecology of Human-Machine Systems II: Mediating "Direct Perception" in Complex Work Domains. *Ecological Psychology*, *2*(3), 207-249

Wang, W., Shen, Z., Hou, F., & Yi, B. (2002). Ecological driver-vehicle interface design based on situation awareness. *Intelligent Vehicle Symposium, 2002. IEEE, 2*, 352 - 357.

Wickens, C. D., (1980). The structure of attentional resources. In R. S. Nickerson (Ed.), Attention and Performance VIII (pp. 239-257). Hillsdale, NJ: Lawrence Erlbaum

Wickens, C. D., & Hollands, J. G. (2000). *Engineering Psychology and Human Performance* (3rd, pp. 481-485). Up Saddle River, New Jersey: Prentice Hall.

Xu, W., Dainoff, M. J., & Mark, L. S. (1999). Facilitate Complex Search Tasks in Hypertext by Externalizing Functional Properties of a Work Domain. *International Journal of Human-Computer Interaction, 11*(3), 201-229.