Spring 2010

The effect of positive and negative messages on problem solving in computer programming tasks

Kristopher M. Thornburg
University of Iowa

THE EFFECT OF POSITIVE AND NEGATIVE MESSAGES ON PROBLEM SOLVING IN COMPUTER PROGRAMMING TASKS

by

Kristopher M. Thornburg

An Abstract

Of a thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy degree
in Industrial Engineering in
the Graduate College of
The University of Iowa

May 2010

Thesis Supervisor: Professor Geb W. Thomas

ABSTRACT

Many supervisory control systems require the operator to solve any problems that the system's automation cannot accommodate. Consequently, this class of systems would benefit from designs and methods which improve operator problem solving performance. Currently, human factors researchers develop designs and methods emphasizing the cognitive capacities and abilities of operators. For the most part, these approaches neglect the emotional state of the operator, although emotion has been shown to have an important impact on performance in many other domains.

This dissertation introduces the modified Multidimensional Problem Solving (m-MPS) Model, a theoretical model predicting how affect, one aspect of emotion, will influence problem solving performance. The model was tested in an experiment in which 32 participants attempted to correct a series of 5 bugs in a computer program. During their task, they received compiler messages with keywords specifically chosen to create a positive or negative affective state. The model predicted that the participants with messages designed to increase positive affect would seek solutions with a more divergent thought process, and this would be indicated with a more diverse set of problem-solving approaches, along with higher scores on a divergent thought measuring test administered throughout the experiment. Those with less positive affect would seek solutions in a smaller, less creative space and demonstrate less divergent thought. Unfortunately, the feedback messages did not appear to evoke an emotional response powerful enough to create a measurable change in emotional state. However, the messages did affect various aspects of the participants' performance in ways consistent with the model, including fewer repeated solutions with increasing divergent thought scores ($F(1,423) = 12.39$, $p < 0.01$) and the probability of continuing the problem solving process declines with each unsuccessful attempt ($Z = -2.98$, $p = 0.003$). The most compelling result was that participants receiving the negative messages were significantly less likely to successfully complete the problem-solving task (Wald $X^2 = 4.06$, $p = 0.044$). These results suggest that in human-computer interactions, messages are an important factor in creative problem solving performance. Further research is necessary to determine the source of these effects in supervisory control interfaces.

Abstract Approved: ________________________________

Thesis Supervisor

____

Title and Department

____

Date

THE EFFECT OF POSITIVE AND NEGATIVE MESSAGES ON PROBLEM SOLVING IN COMPUTER PROGRAMMING TASKS

by

Kristopher M. Thornburg

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy degree
in Industrial Engineering in
the Graduate College of
The University of Iowa

May 2010

Thesis Supervisor: Professor Geb W. Thomas

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Kristopher M. Thornburg

has been approved by the Examining Committee
for the thesis requirement for Doctor of Philosophy
degree in Industrial Engineering at the May 2010 graduation.

Thesis Committee: ___________________________________
Geb W. Thomas, Thesis Supervisor

____
John D. Lee

____
Linda Ng Boyle

____
David Watson

____
Joe Kearney

For Liza

What is a scientist after all? It is a curious man looking through
a keyhole, the keyhole of nature, trying to know what's going on.

Jacques-Yves Cousteau

ACKNOWLEDGEMENTS

I would like to thank my advisor, Professor Geb W. Thomas, for his continued support, encouragement, and questions. I would also like to thank the members of my committee for their thoughts and ideas. Members of the GROK Lab, HFSM Lab, and MIT HAL were very important and helpful during this journey. I am especially grateful for my friends, who would pick me up when I was down; my parents and their never-ending support and strength; and my wife for believing in me always. Thank you for encouraging me to look at the world differently.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I. INTRODUCTION

The human factors discipline seeks to infuse knowledge of how people move and think into design to help people work, travel, communicate, and even play, more safely, more reliably and more effectively. The sub-discipline of cognitive human factors specifically emphasizes how psychology influences design and human performance. Cognitive human factors researchers investigate how limitations in, for example, human memory, attention and vigilance influence task performance and how systems may be designed to overcome these limitations.

Human factors researchers have devoted particular attention to designing systems that help people supervise and interact with complex operations such as nuclear power plants, airplanes, military command and control systems, robots, and manufacturing plants. An important challenge in designing these systems is helping operators diagnose and respond to unexpected events. The traditional approach to the problem of responding to unexpected events has been to emphasize information: the information the operator has, the information that is required, and what decisions must be made. Once these factors are well understood, the human factors specialist then seeks to design a system that helps to provide the relevant information clearly without overwhelming the operator so that he or she can make an appropriate decision. Although such analyses carefully track the capabilities and limitations of the operator's cognitive process, they tend not to consider the emotional state of the operator in a problem solving situation.

Responding to a safety-related warning light in an aircraft, a distress signal from a military unit, or an unexpected, but time-sensitive message from a robotic spacecraft can be a stressful experience. Research from other disciplines suggests that a person's emotional state can have a large impact (positive or negative) on his or her ability to make decisions and solve problems. However, the traditional approach to complex system design has generally overlooked the opportunity to design the system specifically to evoke, or at least mitigate emotions, to augment a positive effect or avoid any debilitating effects that a predictable emotional state might have on an operator's problem-solving capabilities. Researchers in other fields have already begun the path towards such innovative design techniques.

Psychologists have long explored the underlying theory and effects of mood, personality and mental disorders. Meanwhile, marketing professionals have spent decades refining techniques to induce particular emotional responses. At the core of these investigations is affect. Affect is the basis of emotion. It has two dimensions: positive and negative. A person's position in these two dimensions leads to their emotional state. For example, a person with a large negative affect might feel frustrated or annoyed. At the same time, he or she might have increased positive affect, resulting in conflicting feelings that encourage or strengthen their resolve.

Within the past 30 years, great strides have been made in measuring affect and emotional reactions to different stimuli (Ashby et al., 1999). Researchers have developed sensitive tools to measure changes in affect and emotion (Isen, 2001; Warr et al., 1983; Diener and Emmons, 1985; McAdams and Constantian, 1983; Stone et al., 1985; Watson et al., 1988). Important findings have also been made regarding inducing or promoting affective or emotional reactions. For example, Isen (2001) considers how affect can be changed with marketing materials. Berridge and Winkielman (2003) used other psychological manipulations to change affect. Both research teams found that positive affect can have a positive impact on a variety of human behaviors, including openness to new products and therapies (Isen, 2001; Berridge and Winkielman, 2003; Winkielman and Berridge, 2004), creative problem solving (Isen et al., 1987), and general performance (Isen, 2001; Lee, 2007). Given that positive affect can benefit problem solving performance, it is surprising that the role of emotion and affect has not received more attention within the human factors field, particularly regarding human-computer interaction.

The human-computer interaction (HCI) literature includes investigations of emotion. HCI studies of emotion emphasize detecting and mitigating negative emotional reactions like frustration, rather than using emotion to optimize cognitive and physical performance (Aboulafia and Bannon, 2004; Lazar at al., 2006; Ceaparu et al. 2004; Picard, 1997). Frustration with human-computer systems is a detriment to human performance (Lazar at al., 2006; Ceaparu et al. 2004). Designing system interfaces that reduce or eliminate negative affect could improve performance. Although positive affect could improve task performance (Isen et al., 1987; Isen, 2001; Lee, 2007), this relationship has not yet been investigated in the HCI domain.

Positive affect has been shown to increase human performance in other domains (Isen, 2001; Lee, 2007), such as solving problems creatively (Isen et al., 1987). Creative problem solving can be important in many different contexts, including complex human-computer interaction tasks. Because many complex systems place the operator in a supervisory position, the operator is expected to solve problems that the complex systems cannot accommodate (Lee, 2007). As system complexities continue to grow, greater demands will be placed on operators' creative problem solving capabilities. Although research continues to determine how information can be better presented to assist the operator in this task, the opportunities to enhance performance by manipulating the operator's affect have not yet been fully explored.

This research connects HCI human performance research with psychological affect research to show that promoting positive affect can improve creative problem solving performance in domains of interest to human factors. This dissertation begins by examining some of the theory behind affect, the benefits of positive and negative affect, and how affect influences people's abilities to diagnose and solve real-world problems. Next, a discussion of current design practices is presented, followed by a modified version of an existing model that is designed to explain how affect influences problem solving. Then this dissertation explains a novel design direction in human-computer interaction to support problem solvers: using positively worded messages to help problem solvers maintain or promote positive affect. The positive messages are intended to promote or maintain positive affect, increase divergent thinking and ultimately improve problem solving performance. A pilot study and an experiment explore this new direction and show that it provides benefits that the current design directions do not. Each study offers evidence in support of this direction and the model's description of affect and problem solving. Finally, this dissertation explains how these results fit within and extend the existing body of knowledge.

CHAPTER II. BACKGROUND

Affect

Affect, in psychological terms, refers to the general valence, or disposition, of a person's emotional state which psychologists consider to be the basis of human emotion and mood. Emotion is generally differentiated from mood by being fairly short-lived and a result of a definite cause (Forgas, 1994). Mood, on the other hand, is described as "relatively enduring" and lower intensity than emotion (Forgas, 1994). Affect is a combination of positive and negative valences but is not an emotion or mood itself. Affect is a pre-cognitive process that happens as an automatic reaction to stimuli whereas emotion and mood require a certain degree of cognitive processing (Dijksterhuis and Smith, 2002). This chapter will introduce several prominent theories of affect exist, ranging from a bipolar continuum model to a model of affective spacial systems (Lewis et al., 2007; Larsen et al., 2003; Berridge, 2003; Fredrickson, 2004; Tellegen et al., 1999; Watson et al., 1999).

Models of Affect

Affect Models

The most prominent models of affect can be grouped into two distinct categories: bipolar models and affective space models. This section will examine these two broad categories. Bipolar models of affect describe affect on a single continuum, usually resulting in a description of affect as being either negative or positive. Naturally, positive affect is negatively correlated with negative affect in this model, so that a person may not be in both a positive and negative mood at the same time. Figure 1a indicates a typical bipolar continuum of affect in which an individual's affect at any particular time can be characterized.

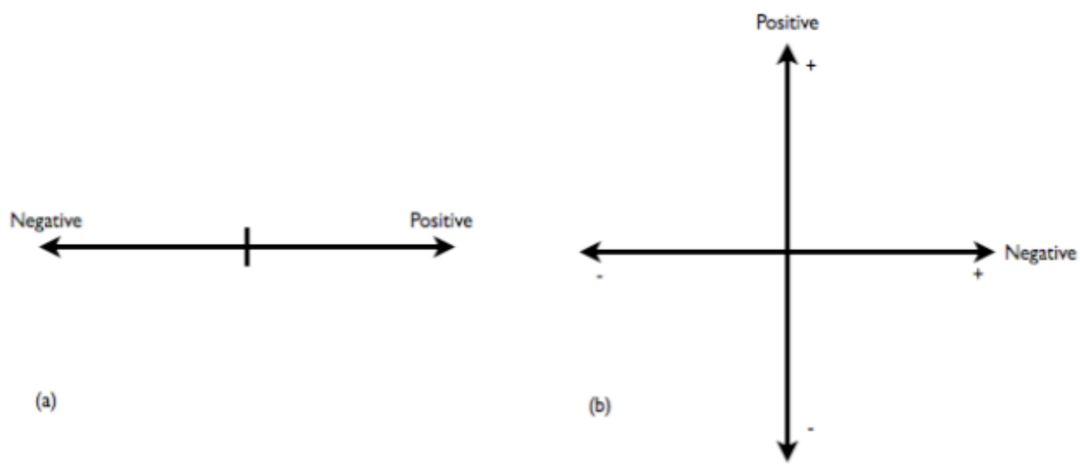

Figure 1: Models of affect, including a) a bipolar continuum model of affect along a single axis and b) an affective space model created with two axes representing positive or negative affect.

Affective space models consider positive and negative affect separately. Unlike the bipolar continuum model, affective space models describe a relationship between positive and negative affect as not necessarily correlated in any particular way. In this model, positive and negative affect are mediated by different processes which operate independently. Commonly, affective space models are visualized as a two axis system, as shown in Figure 1b. Affective space models are more commonly cited in the literature as a better description of affective processes. Evidence for these claims can be drawn directly from neurological experiments based on affect.

<u>Neurological Evidence</u>

The emergence of PET and fMRI scanning has provided an unprecedented opportunity for neuroscientists and psychologists to see exactly what parts of the brain are involved with emotion and affect. In fMRI studies, the posterior cingulate is the area of the brain that shows particular activation when the person is in the presence of affective stimuli (Figure 2). Positive stimuli and negative stimuli will induce activation in different areas, supporting different mechanisms for positive and negative affect and emotion (Maddock et al., 2003; Fossati et al. 2003). The ventral pallidum has been shown to be a necessary system for a normal positive affective reaction in humans (Berridge, 2003; Berridge and Winkielman, 2003; Winkielman and Berridge, 2004; Dalgleish, 2004). This normal positive affective reaction can be either conscious or unconscious, depending upon which brain networks are activated (Berridge, 2003; Berridge and Winkielman, 2003; Winkielman and Berridge, 2004). The parabrachial nucleus, part of the brainstem nuclei, along with the nucleus accumbens shell, play a part in enhanced positive affective reactions which have been demonstrated through neural stimulation in rats and humans (Berridge, 2003). Though these systems have been identified as important, the positive affect system or systems are still not fully identified (Berridge, 2003).

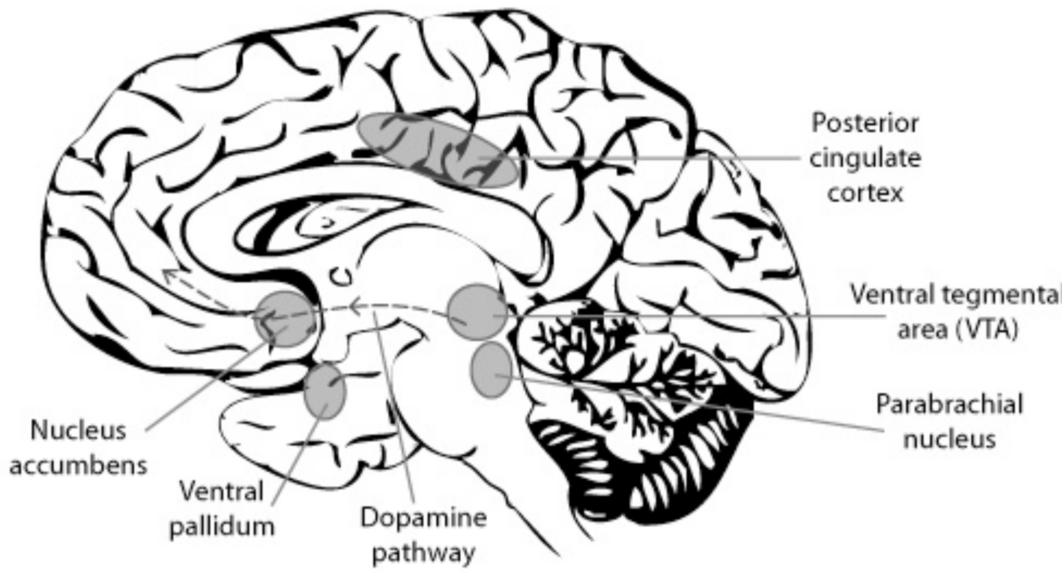

Figure 2: Locations of the various structures in the brain shown to be important in affect and emotion.

Positive and negative affect have been shown to be mediated by separate neural pathways, giving support to the theory of an affective space, rather than a bipolar affective continuum (Ashby et al., 1999; Berridge, 2003; Lewis et al., 2007). Based on this evidence, Ashby, Isen, and Turken (1999) propose a neurophysiological theory of affect. Their theory suggests that during periods of positive affect, there is a corresponding dopamine release in the mesocorticolimbic system of the brain and possibly in the nigrostriatal system. The dopamine producing cells in the mesocorticolimbic system of interest are referred to as the ventral tegmental area (VTA in Figure 2), which project into the nucleus accumbens. As Berridge (2003) points out, the nucleus accumbens shell is an area that plays a part in enhanced positive affective reactions. The increased dopamine levels in these areas of the brain are assumed to influence performance in a variety of cognitive and behavioral tasks (Ashby et al., 1999). The nigrostriatal system is associated with increased dopamine release and the resulting increase in motor activity, which is also associated with positive affect. (This system is thought to be responsible for the actions associated with the phrase "jumping for joy".) It is unclear, however, how the dopamine release in these brain systems impacts conscious emotion or if the system operates subconsciously, requiring another system to make the transition to consciousness.

Based on this neurological evidence, it is apparent that a model of affective space more accurately represents the affective processes in the brain. The areas of the brain that are activated as an affective response to positive stimuli and negative stimuli operate more or less independently in an automatic fashion. Besides medical scanning technology, affect can be measured using several different methods, which are outlined in the following sections.

Affect Measurement

Affect can be measured using physiological or subjective measures. Each method has specific benefits and limitations. The following sections describe physiological and subjective measures that have been used in research and compares the two general methods.

Physiological Measurements

Changes in peoples' physiology can indicate changes in affect. For example, changes in facial muscle activation can reveal changes in affect. So can posture, hand tension and activity, gestural activity, vocal expression, language (choice of wording), and galvanic skin response (Picard and Daily, 2005). Measuring the activation of the zygomaticus major and the corrugator supercilii muscles (smiling and brow furrowing muscles) through electromyography is a specific example (Larsen, Norris, and Cacioppo, 2003; Benedek and Hazlett, 2005; Hazlett and Benedek, 2007), though the measurement is based on a bipolar affective scale (Larsen et al., 2003). This bipolar affective scale does not agree with the majority of affective research (Ashby et al., 1999; Berridge, 2003; Lewis et al., 2007), but this approach can show useful results nonetheless.

Larsen et al. (2003) studied the effect various pictures, sounds, and words had on affect valence (the degree of positivity or negativity) and the resultant muscle activations in zygomaticus major (the muscle primarily responsible for smiling, seen in Figure 3) and corrugator supercilii (the muscle primarily responsible for brow furrowing in frowns, seen under the shown muscle in Figure 3). This study demonstrated significant correlative activity in both muscles. Positive affect inhibited activity in corrugator supercilii and moderately activated zygomaticus major. Negative affect inhibited activity in zygomaticus major and activated corrugator supercilii. Benedek and Hazlett (2005) found that frustrating computer tasks activated corrugator supercilii and novel or improved software features activated zygomaticus major. Both studies (Benedek and Hazlett, 2005; Larsen et al., 2003) show zygomaticus major activation with positive stimuli and corrugator supercilii activation with negative stimuli. Both studies indicate difficulty measuring muscle activation because the corrugator supercilii and zygomaticus major are located under and among other muscles in the face. Measurement of these muscle activations become noisy and unreliable when the nearby muscles are activated in mildly affective situations. Other researchers are integrating several physiological measures to reduce this noise and make the data more reliable.

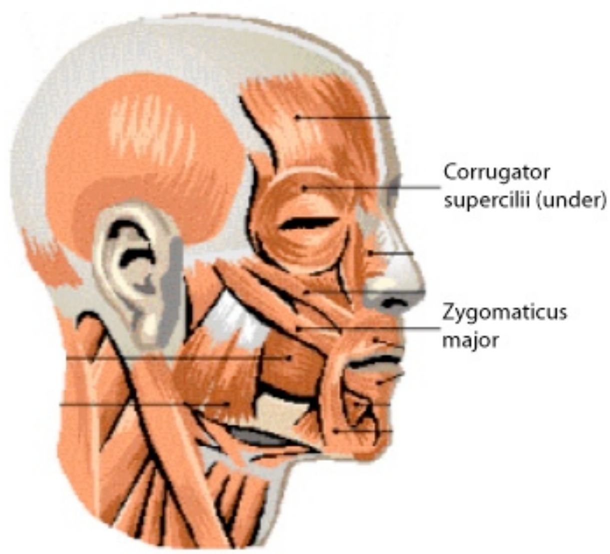

Figure 3: The corrugator supercilii (under the shown muscle) is responsible for furrowing the brow and is shown to reflect an instantaneous negative affect while zygomaticus major is responsible for smiling and is shown to reflect positive affect (adapted from http://www.voetbalacademie.be/spierstelsel.htm).

A recent novel approach aggregates galvanic skin response, heart rate, heart rate variability, interbeat interval, blood pressure, and electromyographic measures of the face to show when an individual is having fun or being creative (Mandryk et al., 2005;

Mandryk et al., 2006; Mandryk and Atkins, 2007). This research investigated classifying emotional reactions while playing video games into boredom, challenge, frustration, excitement and fun categories. These categories are not part of the standard set of emotions that psychologists prefer to study, but this approach was successful at classifying physiological reactions. According to the authors, this approach has not been validated as reliable in domains outside video games and entertainment (Mandryk et al., 2005; Mandryk et al., 2006; Mandryk and Atkins, 2007). This approach is novel compared to other types of affect measurement, which have been studied extensively in the field of psychology.

Subjective Measurements

Using subjective measures to measure affect is the second measurement approach. Affect has been reliably measured by self-report scales in psychological studies. These scales have been shown to be reliably sensitive to mild affective reactions, as opposed to facial EMG measures which are reliable for extremely affective measurements. A successful and widely used scale was developed by Watson, Clark, and Tellegen (1988), which uses ten items to measure each positive and negative affect. The scales, called the Positive and Negative Affect Schedule (PANAS) scales, have been subsequently shown to be very robust and consistent over various time spans and populations. A key component of the PANAS scales is that positive and negative affect are measured independently, with very little correlation. Studies using other measures, including the Thematic Apperception Test (McAdams and Constantian, 1983), Warr et al.'s (1983) 18 item questionnaire, the Nowlis Mood Adjective Check List and Beck Depression Index (Stone et al., 1985), and mood reports (Diener and Emmons, 1985), have shown varying results in different situations. Several studies indicate the independence of positive and negative affect (Warr et al., 1983; Diener and Emmons, 1985), but the correlation in the results are much higher than the PANAS scales (Watson et al., 1988). This aspect of the PANAS scales reflects the neurological evidence of separate affective systems for positive and negative and support its wide use in psychological research. Alternatively, indirect measures of affect include subjective mood and emotion surveys or questions that can be analyzed to infer general affect, such as rating unfamiliar words (Isen et al., 1987). However, since the PANAS scales are so brief and easy to use, it is unclear why an alternative would be used for subjective affective measurements. To select the best affect measurement method, a comparison needs to be made between subjective and physiological approaches.

Comparison of Physiological and Subjective Measurements of Affect

Subjective measures of affect have been studied in a wider range of domains than physiological measurements and therefore have a more robust scientific basis. Though many innovations have made physiological measurements of affect possible, these measures need more research and validation. For instance, the physiological measurements used by Benedek and Hazlett (2005) are based on a bipolar model of affect, which does not truly represent how affect operates in people. Physiological measures do provide interesting insights in different situations, but the data noise and intrusiveness are detrimental to the validity and sensitivity of the results. Subjective measures, like the PANAS scales, have been studied and used extensively in experimental research. Subjective measures are non-intrusive and are more sensitive than physiological measures when measuring minor changes in affect.

Promoting and Inducing Affect

Methods of Promoting or Inducing Positive Affect

Positive affect has been successfully induced in several different ways. Isen and colleagues induced conscious positive affect in participants by giving them a small bag of candy as a gift or by asking participants to watch a five-minute long humorous film (Isen et al., 1987). Affect was measured by rating unfamiliar words or rating words indicating affective dimensions. Participants consciously acknowledged how they felt after receiving the candy or watching the film. Berridge and colleagues successfully induced unconscious positive affect through subliminally presented smiling faces. The faces were shown for 16ms before a neutral face and participants indicated they did not see these expressions after the test. The smiling faces induced unconscious positive affect measured by rating particular products or consumption behavior. Participants indicated they did not consciously feel any different from the beginning of the experiment to the end (Berridge, 2003; Berridge and Winkielman, 2003; Winkielman and Berridge, 2004). Stapel and colleagues (2002) conducted a similar experiment in which participants rated a neutral face on a positive and negative scale after having been subliminally primed with smiling or frowning faces. This experiment indicated similar results as the experiments conducted by Berridge and colleagues.

Besides films, candy, and subliminal cues, positive affect has been induced through pictures, sounds and words presented as text. Larsen et al. (2003) studied the effect pictures, sounds and words had on the affect of participants. The pictures, sounds and words were selected from a cross-section of standard psychological lists. In this study, affect was measured using the affect matrix, an affect measurement tool created by the authors, and a 9-point arousal scale. Interestingly, pictures, sounds and words were shown to induce both positive and negative affect to a similar degree. The results from all of these experiments show that affect can be induced or promoted through a wide variety of stimuli. Inducing positive affect has obvious impact upon human emotion but positive affect has also been shown to benefit human cognition in a variety of ways.

Benefits of Positive Affect

In many simple studies, Isen and colleagues have shown many different cognitive benefits associated with induced positive affect. Positive affect has been shown to promote helpfulness and generosity towards others, to enhance variety seeking in consumer products, to promote positive and non-hostile negotiations, and to enhance efficiency and thoroughness of decision making (Isen, 2001; Isen et al. 1987). Isen and colleagues (1987) also showed that creative problem solving is significantly facilitated through an induced positive affective state. In this case, positive affect was induced through a five-minute comedy film or a small bag of candy. Though the manipulation seems small, participants with a positive affect were significantly more creative and solved the problem faster than subjects who had a neutral or negative affect.

This effect can be seen in the affect infusion model (AIM), developed by Forgas (1995, 2002). AIM describes judgments made by operators in unfamiliar situations are exceptionally prone to affective influence. Additionally, situations that require creative and constructive thinking along with interpreting novel information also increase affective influence (Forgas, 2002). The influence of affect may be largest when complex systems fail, requiring the operator to solve the problem (Lee, 2007). By inducing positive affect, creative problem solving skills of the operator will be enhanced, allowing the operator to solve complicated problems without relying on external resources. An interesting question for human-computer interaction researchers is whether positive affect can be induced or promoted through a computer interface, rather than a comedy film or a gift, to enhance problem solving skills.

Methods of Promoting or Inducing Negative Affect

Negative affect has also been experimentally induced in people using methods similar to inducing positive affect. Isen et al. (1987) asked participants to watch a brief film about concentration camps, thereby inducing a negative affect. Berridge and colleagues successfully induced unconscious negative affect through subliminal cues of frowning faces (Berridge, 2003; Berridge and Winkielman, 2003; Winkielman and Berridge, 2004). Unconscious negative affect was measured by rating particular products or consumption behavior. As with positive affect, negative affect was induced through pictures, sounds and words presented as text (Larsen et al., 2003). Each of these manipulations was measured in the same manner as the change in positive affect.

Benefits of Negative Affect

Negative affect has been shown to narrow and focus concentration on a specific and most obvious problem. To some designers, this effect would be desirable because operators of the system would be more focused and vigilant (Norman, 2004). However, studies of negative affect have shown that though concentration has been narrowed and focused, the operator is less likely to choose a solution that does not readily present itself (Forgas, 1995; 2002). In other words, negative affect promotes convergent thought, which reduces the amount of ideas and solutions the operator can generate. Limiting the generation of possible solutions may be important in situations that have very few solutions. In a complex situation that requires a solution that does not readily present itself, the creative problem solving process cannot be done by an operator with a negative affect.

Comparison of Positive and Negative Affect Benefits

Positive and negative affect each have their own unique advantages and disadvantages. Positive affect enhances creative problem solving processes, making it easier for a person to conceive of more ideas or solutions (Isen et al., 1987). Negative affect, on the other hand, tends to enhance focus and concentration on a singular object or problem (Forgas, 1995; Forgas, 2000; Norman, 2004). The difference is a broad view of the problem space, or divergent thought, promoted by positive affect and a narrow view of the problem space, or convergent thought, promoted by negative affect. The disadvantage for positive affect is the possibility of the need for many iterations to try potential solutions in the problem space. The disadvantage for negative affect is the narrowing of focus on a symptom of the problem, rather than the problem itself, and laboring fruitlessly to solve the problem. Because performance on the creative problem solving task is the goal of a complex problem situation, positive affect should be induced to allow the operator more ability to generate new ideas and solutions. The remainder of this dissertation will focus on positive affect and its benefits to problem solvers. To examine this idea further, the next section will review problem solving and problem solving processes.

Problem Solving

Rasmussen's Skills, Rules, Knowledge (SRK) Taxonomy (1983) can be used as a framework to describe how operators solve complex problems. The first level, skill-based behavior, refers to behavior that happens automatically and without conscious thought. Many people refer to this level of behavior as "muscle memory". The second level, rule-based behavior, refers to behaviors that are based on rules that have been trained or learned. The third level, knowledge-based behavior, is based on formulating and attaining goals and is considered the highest conceptual level of the three. Usually, complex problem solving situations occur within the knowledge-based behavior level. Once a solution is obtained, the solution may become a rule and move to the rule-based behavior level. Designers can use this transition and the SRK Taxonomy in general to create effective aids for operators in complex problem solving situations.

Rasmussen's (1993) decision ladder (Figure 4), based on the SRK Taxonomy, describes the process by which people diagnose and solve problems. The rectangles represent the actions taken in the process while the circles represent "states of knowledge" of the problem solver. The dotted arrows represent shortcuts, found in the SRK Taxonomy, that problem solvers can use based on their expertise in the problem domain. If a person traverses the entire ladder, it is assumed they are using knowledge-based behaviors to solve the problem. Using the short-cuts implies the use of skill- or rule-based behaviors.

To traverse the decision ladder, the process or system is activated which alerts the problem solver. Observations of the system are made, from which problem solvers gain information about the system. This information is used to identify the situation to define the state of the system. The problem solver interprets the available information to generate options, evaluate options, and choose a goal. The goal will be interpreted by the problem solver to select a target. The target will be used to define a specific task needed to solve the problem. A plan will be created with a specific procedure, and finally the problem solver will execute the plan.

Shortcuts within the ladder shorten the process. The shortcuts originate from the observation action. The skill-based shortcut connects an observation directly to a specific procedure or, in a more complex situation, to a specific task. The rule-based shortcut connects an observation with either a specific task or a specific target. There is no shortcut for knowledge-based behaviors.

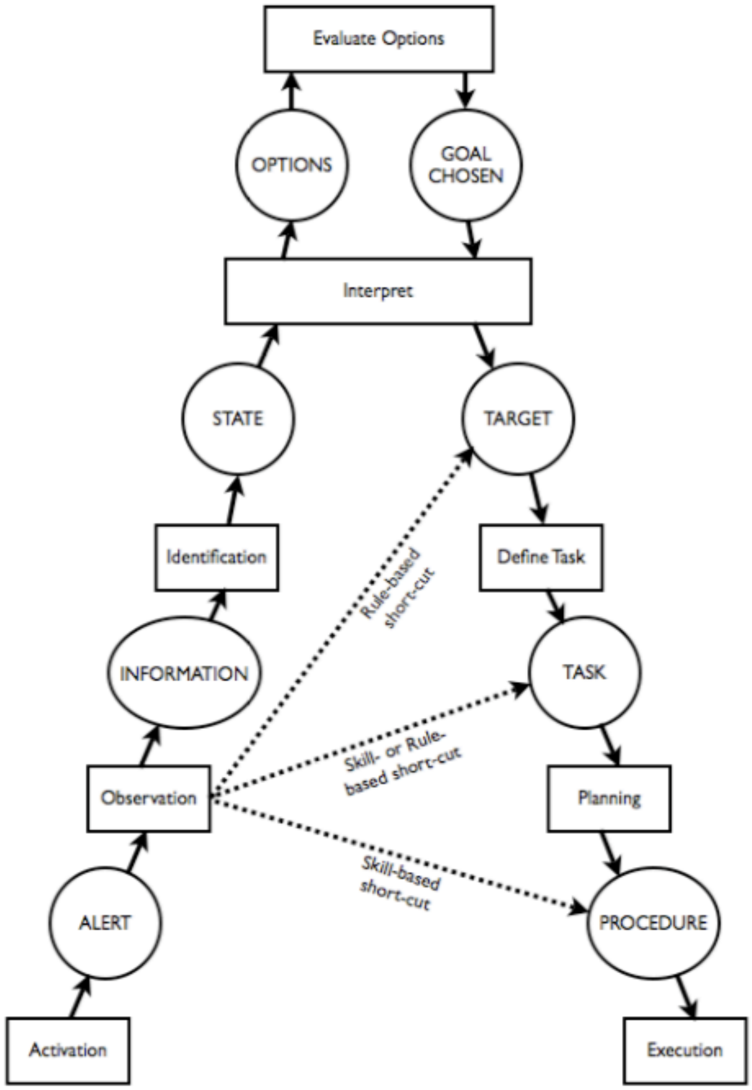

Figure 4: Rasmussen's (1993) decision ladder that models how people diagnose or solve problems.

The SRK Taxonomy (Rasmussen, 1983) and the decision ladder (Rasmussen, 1993) systematically describe the process people use to solve problems in various behavioral levels. Carlson and Bloom (2005) observed the process and performance of people solving analytical problems requiring knowledge-based behaviors. From these observations, they created the Multidimensional Problem-Solving Framework (Table 1). This framework describes the phases of the problem solving process. The first phase is Orienting, in which problem solvers will examine the problem and organize the facts associated with the problem. Planning is the second phase, in which people will create a plan for solving the problem by conjecturing and imagining possible solution spaces. The third phase is Executing the plan created in the second stage. This stage involves the actual computation or implementation needed to accomplish the plan. Checking is the fourth phase of the problem solving framework, in which people verify the results of the possible solution. At this point, the problem solver decides whether the solution has solved the problem or if a new plan is required. If a new plan is required to create a feasible solution, the problem solver will return to the Planning phase and continue the cycle. Additionally, the framework includes the dimensions of Resources, Heuristics, Affect, and Monitoring for each phase of the cycle. The different dimensions in the framework describe, for each phase, the cognitive resources demanded, the heuristics employed, typical emotions, and how problem solvers monitor the process.

Interestingly, each phase within the Multidimensional Problem-Solving Framework (MPSF) can be represented by a series of actions and states from the behavior-oriented decision ladder. Though the MPSF is not as specific as the decision ladder concerning behavior, the MPSF includes cognitive and emotional dimensions that should be considered in problem solving.

Table 1: Carlson and Bloom's (2005) Multidimensional Problem-Solving Framework which describes the phases of the problem solving process.

Phase • Behavior	Resources	Heuristics	Affect	Monitoring
Orienting • Sense making • Organizing • Constructing	Mathematical concepts, facts and algorithms were accessed when attempting to make sense of the problem. The solver also scanned her/his knowledge base to categorize the problem.	The solver often drew pictures, labeled unknowns and classified the problem. (Solvers were sometimes observed saying, "this is an X kind of problem.")	Motivation to make sense of the problem was influenced by their strong curiosity and high interest. High confidence was consistently exhibited, as was strong mathematical integrity.	Self-talk and reflective behaviors helped to keep their minds engaged. The solvers were observed asking, "What does this mean?"; "How should I represent this?"; "What does that look like?"
Planning Conjecturing Imagining Evaluating	Conceptual knowledge and facts were accessed to construct conjectures and make informed decisions about strategies and approaches.	Specific computational heuristics and geometric relationships were accessed and considered when determining a solution approach.	Beliefs about the methods of mathematics and one's abilities influenced the conjectures and decisions. Signs of intimacy, anxiety, and frustration were also displayed.	Solvers reflected on the effectiveness of their strategies and plans. They frequently asked themselves questions such as, "Will this take me where I want to go?", "How efficient will Approach X be?"
Executing • Computing • Constructing	Conceptual knowledge, facts and algorithms were accessed when executing, computing and constructing. Without conceptual knowledge, monitoring of constructions was misguided.	Fluency with a wide repertoire of heuristics, algorithms, and computational approaches were needed for the efficient execution of a solution.	Intimacy with the problem, integrity in constructions, frustration, joy, defense mechanisms and concern for aesthetic solutions emerged in the context of constructing and computing.	Conceptual understandings and numerical intuitions were employed to reflect on the sensibility of the solution progress and products when constructing solution statements.
Checking • Verifying • Decision making	Resources, including well-connected conceptual knowledge informed the solver as to the reasonableness or correctness of the solution attained.	Computational and algorithmic shortcuts were used to verify the correctness of the answers and to ascertain the reasonableness of the computations.	As with the other phases, many affective behaviors were displayed. It is at this phase that frustration sometimes overwhelmed the solver.	Reflections on the efficiency, correctness and aesthetic quality of the solution provided useful feedback to the solver

Problem Solving Performance

This Multidimensional Problem-Solving Framework can only address different problem solving strategies in terms of the dimensions of the framework. Researchers have examined behaviors of problem solvers in terms of divergent and convergent thought and strategies. A prominent theory related to divergent and convergent thought is referred to as the spreading-activation theory (Collins and Loftus, 1975). This theory models ideas in a continuum in which related concepts are close and disjointed concepts are distant. The theory contends that with closely related ideas, spreading activation is high, and with ideas that are remotely related, the spreading activation is low. Each idea can be visualized as a node in a network. The closer the nodes, the more related the ideas. A problem-solver searches this space for new ideas by moving from one node to another. The spreading activation theory explains that when searching for ideas, a person's movement in the space may be convergent or divergent. Convergent thinking searches with small steps, moving among closely related ideas. Divergent thinking takes large steps, moving among loosely connected ideas. For example, the word *car* can be closely associated with the word *truck*, less associated with the word *street*, and even less associated with the word *toy*. In this example, the *car* and *truck* nodes would be located quite close, whereas the *car* and *toy* nodes would be located further from each other. A person with a more divergent thought process is more likely to associate *car* and *toy* than someone with a more convergent thought process. The convergent thinking person would most readily associate *car* and *truck*, but would not create the association between *car* and *toy*. This idea is the basis of the Remote Associates Test (Mednick, 1962).

The Remotes Associates Test (RAT) is a measure of creativity (Mednick, 1962), based on the spreading-activation theory. The test consists of items with three seemingly unrelated words. The task for someone taking the RAT is to link the three words with a fourth. Consistent with the spreading activation theory, RAT item words that are located close together encourage a convergent answer. These items on the RAT are generally easier to answer. For example, the words “cottage swiss cake” would have a response of “cheese”. RAT item words that are located farther apart require divergent thought to generate the appropriate answer. The typical RAT contains 30 items. The number correct is a general measure of creativity and by association, a measure of divergent thought. The higher the RAT score, the more creative or divergent-minded an individual is at the time of testing.

People thinking divergently typically find solutions to a problem faster than people thinking convergently (Isen et al., 1987). Divergent thinkers generate more potential solutions than convergent thinkers, increasing the probability of discovering a correct solution. Several studies (Brophy, 1998; Vosburg, 1998; Clapham, 2000), have demonstrated that divergent thinkers are more capable of solving “real-world” problems contrasted with problems that can be solved through simple deductive reasoning. Building on this problem solving background, the following sections examine how human-computer interfaces support user problem solving.

Human-Computer Interaction

Facilitating problem solving in complex systems has become more prominent in human factors research in recent years as automation increasingly takes over tasks that are tedious or require high amounts of repeatability (Lee, 2007). Humans in these systems are delegated more higher level responsibilities, usually as supervisory roles (Lee, 2007; Sheridan, 1992). These roles require the operator to monitor the automation and solve problems the automation cannot. A common failure among highly automated systems is the situation in which the automation does not know how to respond or react (Lee, 2007). At these times, the human supervisor is required to take over and remedy the situation by creatively solving the problem. Since the problem is beyond the system's automation, the problem usually has a high amount of complexity. The need for the human to effectively solve problems like these is paramount because of the high complexity and usually high cost of these systems (Lee, 2007).

Solutions for aiding operators in complex problem solving situations include stress mitigation through stress measurement and reduction, affective computing solutions including detecting operator mood and compensating, or developing different types of interface interactions for different situations (Picard, 1997; Picard and Daily, 2005; Lazar et al., 2006; Hudlicka, 2003). The common theme through all of these operator aids is that of "detection and reaction" from the system standpoint, or rule-based concentration. Additionally, many systems focus on avoiding negative reactions from the operator, such as frustration, but only delegate limited thought to promoting positive interactions. Unfortunately, in the HCI domain, many of these solutions are driven by technology, in that detection and display technologies promote the directions solutions take for operator aids (Hudlicka, 2003). These approaches ignore much of the psychological research that has been done on emotion and human perception, leading to a "reinventing the wheel" in a technological standpoint while neglecting people's Knowledge-based abilities (Aboulafia and Bannon, 2004).

The question then becomes how to promote (and support) positive affect for a human operator to enhance knowledge-based behavior. Since many complex systems are represented through a computer interface, it is reasonable to start at the interface level. To determine whether positive affect can be induced through an operator interface, an examination of the design and evaluative processes of human-computer interface design is required.

Human Computer Interfaces

Interface Design

The design of an interface concerns, in part, the "look and feel" or aesthetics. Aesthetics are considered by graphic designers to be an essential part of the design, whereas programmers generally consider aesthetics to be "icing on the cake". To show that aesthetics play a larger role in interfaces than "looking good", Kurosu and Kashimura (1995) and Tractinsky (1997, 2000) compared ATM interface layouts based on participant ratings of apparent usability and beauty. These studies showed that apparent usability ratings are influenced by beauty ratings and the two scales are positively correlated.

These results indicate that if function is a central concentration of the field of HCI, form (aesthetics) should be a major focus in the field, as form influences function.

Designing for positive affect in products is the key component to the recently termed field of hedonomics (Helander & Tham, 2003; Murphy et al., 2003; Hancock et al., 2005). Hedonomics focuses on the pleasurable aspects of interacting with products and how to design products for pleasure that were previously designed for safety and performance. This field only studies pleasure, and by association, positive affect, for the sole purpose of enhancing the quality of user interaction. The only benefit studied in this field is that the user feels good about or enjoys the interaction they have with a given product and are satisfied with the interaction. The purpose of the present research is to bring the cognitive benefit of positive affect to interaction design and specifically human-computer interaction.

Hedonomics is the only research field focused on designing for positive affect of the user. Other fields, such as affective computing or cognitive-affective engineering, focus on introducing emotion detection devices, algorithms, or models of emotional interaction to make systems "more emotional" and responsive to human emotion. Affective engineering, or Kansei engineering, focuses on the reaction individuals have to certain products or product aspects from an emotional point of view. Engineering aesthetics, a term coined by Yili Liu (2003), focuses on introducing aesthetics into the engineering design process to determine the effect aesthetics has on engineered systems. These fields are concerned with inducing a certain emotion or the detection of a certain emotion but neglect the study of the potential benefits of a specific emotion, or more generally, affect.

Interface Effectiveness

The GOMS (goal, operator, method, selection rules) model, which can be used to determine the amount of time a particular task will require of a human operator (John and Marks, 1997), is commonly used to examine tasks to be completed by an interface user. Use of the GOMS model does not necessarily help with interface development, but is designed for interface evaluation in terms of human performance. Within the human-computer interaction domain, performance is commonly disguised as usability. There currently exists a wide spectrum of usability examinations, ranging from the very subjective to fairly objective; however, the most popular and widely used usability measures focus on user preference (subjective) and user performance (objective).

Usability methods, such as task analysis, user-centered analysis, and usability testing analyze the interaction humans have with systems based on performance. For example, task analysis reviews the procedures and steps that are necessary to complete a particular task so the system may be designed to support these procedures and steps (John and Marks, 1997). User-centered analysis considers the user from a cognitive and perceptual standpoint, promoting designs that take into account human performance capabilities (John and Marks, 1997). Usability testing analyzes a system based on user performance and occasionally user comments. Each of these methods examine user performance and attempt to increase user performance by manipulating aspects of the design. For example, the task analysis method may reveal an unnecessary step the system requires from the user, which increases task completion time. By removing this step, task completion time can be reduced and the user will be more productive with the saved time. Performance based analyses such as these reveal important design issues but are not able to detect emotion-based impacts on performance. A widely studied example of this is frustration.

Frustration with computer systems is commonly measured as a deviant from usability and is a detriment to task performance (Hudlicka, 2003; Ceaparu et al., 2004; Lazar et al., 2006). Ceaparu et al. (2004) found that over 38% of time spent working with a computer was lost due to frustrating experiences while Lazar et al. (2006) found that users lost 42-43% of time due to frustration. Examples of the most frustrating experiences include difficulties with features while browsing the web, checking email, and word processing. Suggestions for reducing user frustration include keeping the user in the iterative design process, designs which follow design standards, and better word choice in interfaces and error messages (Lazar et al., 2006).

Better word choice for interfaces and error messages has not been a priority in HCI until recently. The first human-computer interfaces were text-based interfaces that required the user to know many commands to properly operate a particular program. Since the beginning of human-computer interaction, examination of the actual text within interfaces has been neglected on the whole. Studies examining the effect words, pictures and sounds have on affect have shown that positive or negative words have a similar impact on affect as positive or negative pictures (Larsen et al., 2003), indicating that word choice can be just as important in an interface as the aesthetics. In particular, words have been shown to have specific affective valences, refering to the degree to which a word is perceived as positive or negative (Bradley and Lang, 1999). Bradley and Lang (1999) developed the Affective Norms for English Words (ANEW), which contains standardized emotional ratings for a very large number of words. It is conceivable that designers could use this list to make appropriate word choices to promote the desired operator affect.

Words in human-computer interfaces are especially important for providing user feedback, particularly in error messages. However, error messages typically do not consider the user's cognitive abilities and most certainly do not consider the user's emotion or affect (Schemenaur and Pawlick, 2007). Since words can have such an impact on affect (Lewis, et al., 2007), it is logical that designers of text-based error messages should carefully consider the wording of those error messages to be sure negative affect is not inadvertently induced. In the same line of logic, if an effort is made not to induce negative affect, it is a small step further to attempt to promote positive affect. Promoting positive affect through design is not a new idea, but has yet to be applied to human-computer interactions at this level. The following section examines the impact positive affect has on problem solving.

Affect and Problem Solving

Effect of Affect on Problem Solving

The effect found by Isen et al. (1987) can be viewed in the context of the spreading-activation theory in which the participants in the positive condition could associate ideas that participants in the negative or neutral conditions could not. The RAT is one measurement tool Isen and colleagues (1984; 1987) used to measure creativity in individuals after inducing positive or negative affect. In this case, divergent thought was promoted through positive affect, leading individuals to make remote associations which allowed the discovery of creative solutions. Isen and others found that positive affect promotes divergent thinking while negative affect promotes convergent thinking (Isen et al. 1984; 1987; Vosburg, 1998; Clapham, 2000; Fredrickson, 2004). Consequently, it is reasonable to expect that people with high positive affect will more quickly solve problems that require solutions that do not readily present themselves. Similarly, promoting negative affect in a problem solving situation would benefit a deductive, analytical type problem (Norman, 2004).

The Multidimensional Problem-Solving Framework describes common problem solving steps, resources, and emotions, but does not explain why some problems are harder to solve than others. However, the Orienting and Planning phases seem as though they are areas in which the problem solver must access and search a wide variety of concepts and ideas to formulate a problem solving strategy. The spreading-activation theory suggests that the ability to quickly search for a variety of potential solutions depends on whether one is thinking convergently or divergently. Combining these two ideas is the basis for the model presented in the next section.

<u>Modified Multidimensional Problem Solving Model</u>

To describe the impact and interaction affect has on and between the phases in the Multidimensional Problem-Solving Framework (MPSF) (Carlson and Bloom, 2005), the framework was modified to illustrate the affect effect. This modified Multidimensional Problem-Solving Model, or m-MPS Model, is shown in Figure 5. The m-MPS Model, developed for this dissertation, is a simplified and expanded version of the original framework developed by Carlson and Bloom (2005). Each cell contains important features for each phase. The original Affect column was replaced by an Emotion / Valence column, because emotions were contained in the original framework. The general valence (positive or negative) of each emotion is indicated with a "+" or "-", to show the progression of emotion and general affect through the problem solving process. Finally, an Affect Impact column was added to describe how affect is changed through the course of problem solving.

The m-MPS Model can be used to trace a problem solver's process while predicting the effect affect will have on problem solving performance. The problem solver will begin with a certain level of positive affect that, in the Orienting phase, will either increase or decrease depending upon the problem solver's perception of the problem at hand. If the problem seems to be easy or is within the problem solver's expertise, positive affect may increase. If the problem appears to be very complex or difficult, positive affect may decrease. An increase in positive affect will promote divergent thought, thereby allowing the problem solver to generate more plans or potential solutions (Isen et al., 1987) in the Planning phase. A decrease in positive affect will not promote divergent thought and may even suppress it. This lack of divergent thought could be characterized as convergent thought, which would encourage the problem solver to focus on the most readily available solution or plan in the Planning phase. The m-MPS Model only considers the creative problem solving benefits that positive affect has been shown to provide, along with the absence of benefit in the reduction or absence of positive affect.

Phase	RESOURCES	HEURISTICS	EMOTION / VALENCE		MONITORING	AFFECT IMPACT
ORIENTING	Knowledge Base	Classify Unknowns	Curiosity Interest Confidence	+ + +	Reflective Behaviors	Orienting phase impacts *initial* state of positive affect
PLANNING	Conceptual Knowledge Facts	Computational Heuristics	Confidence Intimacy Anxiety Frustration	+ + - -	Reflect on Strategies and Effectiveness	Divergent thought - wide plan scope Convergent thought - narrow plan scope
EXECUTING	Conceptual Knowledge Facts Algorithms	Heuristics Algorithms Computational Approach	Intimacy Joy Frustration Defense Mechanisms Concern	+ + - - -	Conceptual Reflection and Intuition	Successful plan execution - increase positive affect Unsuccessful execution - decrease positive affect
CHECKING	Conceptual Knowledge	Computational and Algorithmic Shortcuts	Joy Frustration	+ -	Reflect on Efficiency, Correctness, and Aesthetics of Solution	Successful solution - increase positive affect Unsuccessful solution - decrease positive affect

Increase Positive Affect - Divergent thinking
Decrease Positive Affect - Convergent thinking

Resultant Affect

Next Problem

Figure 5: The modified Multidimensional Problem-Solving (m-MPS) Model which includes the interactions affect promotes between the phases of the model, modified from Carlson and Bloom (2005).

After creating a plan, successes or failures in the Execution phase will adjust affect accordingly. For instance, if the execution of the plan goes smoothly, positive affect could increase. Conversely, if the plan execution encounters problems or is more difficult than anticipated, positive affect could decrease. In the Checking phase, if the solution is successful, positive affect increases and the problem solver moves on to the next problem, beginning another cycle with the resultant affect. If the solution is unsuccessful, positive affect will decrease and the problem solver will cycle back to the Planning phase to attempt a new plan.

The model describes cycles in not only the problem solving phases, but also changes in affect. A successful problem solver will begin the next problem with the advantage of an increased positive affect, unless they have been cycling through unsuccessful solutions. Cycling through unsuccessful solutions results in a lower positive affect with each try. Attempting to use solutions used previously is a clear sign of decreased divergent thought, usually resulting from a series of unsuccessful solutions. Finally generating a successful solution will increase positive affect, but the additive result from the beginning to the end of the problem solving cycle could be a decrease in positive affect.

Overall, the m-MPS Model describes an additive change in affect, in which one phase has a direct influence on affect in the following phase. Specifically, the m-MPS Model can be used as an "affective roadmap" of problem solving, describing affect during and between the phases of the model. Additionally, the m-MPS Model accounts for the divergent or convergent thought processes promoted by positive or negative affect. The cyclical nature of the model accurately describes the iterative process necessary to solving many problems. The model does not suggest that any relationships between affect, divergent thought, and problem solving performance are guaranteed in a linear fashion. The model only predicts that which is most likely, drawing on the research described in the previous sections.

The modified Multidimensional Problem-Solving Model outlines a cyclical pattern of problem solving, indicating that once a particular course of action or potential solution fails to solve the problem at hand, another course of action is chosen in the attempt to solve the problem. This cycle will continue until the problem is solved or the problem solver gives up on the problem.

Theoretical Roadmap

Based on the m-MPS Model, a theoretical roadmap of problem solving was created to describe how problem solving is affected by affect within the HCI domain. The valence-based roadmap, shown in Figure 6, describes the problem solving process, taking into account the m-MPS Model described in the previous section and interface message wording, which is especially important in human-computer interactions. The roadmap assumes the person in the system takes the role of supervisor, in which the person monitors the system for anomalies and failures. When a failure occurs, the supervisor is notified through an interface message. This roadmap describes two separate paths for the message occurrence: a positively worded message and a negatively worded message. The positively worded message will promote an increase in positive affect, thereby promoting divergent thinking. On the other hand, a negatively worded message will promote a decrease in positive affect, thereby promoting convergent thinking. Divergent thinking will allow the problem solver to view a wide section of the problem space, increasing the probability the solution will be found. Convergent thinking will limit the view of the problem space for the problem solver, focusing on one potential problem and reducing the probability that the solution is found. Through the problem solving process, as described in the m-MPS Model, the problem solver's thinking becomes increasingly convergent on a potential solution. When the problem is solved, the problem solver resumes the role of system supervisor.

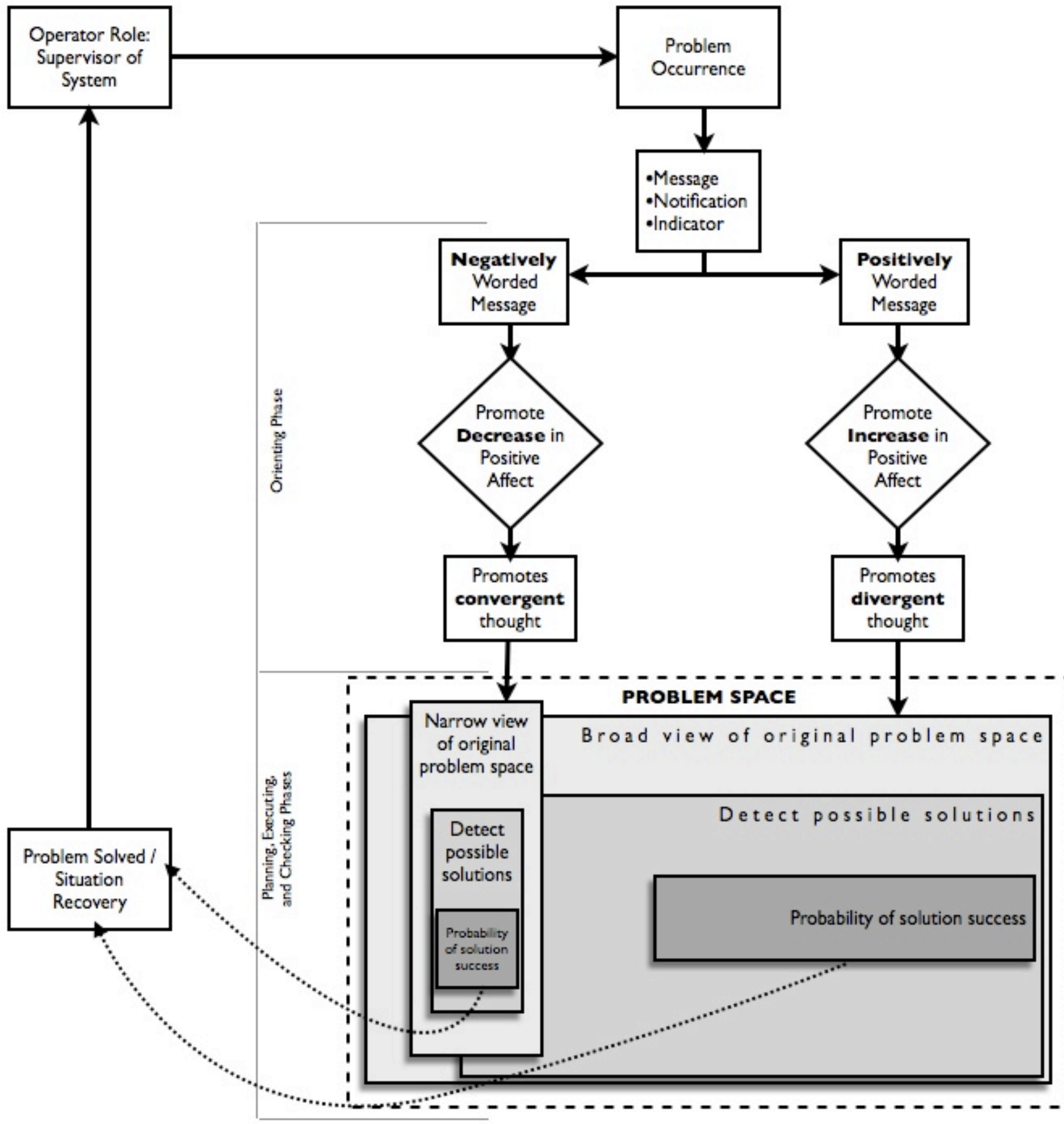

Figure 6: Theoretical valence-based affective roadmap of problem solving, developed for this dissertation research.

Interfaces, Problem Solving, and Affect

Gaps in Literature

Current research on affect or emotion within the HCI domain is focused on using technology to detect and classify users' emotions. The classification of emotion allows the system to properly react to the users' state, usually to reduce frustrating feelings. In short, the paradigm in current research can be summarized as detection and reaction. However, some researchers have suggested a paradigm shift toward preventing the negative emotions from the outset through system design and careful consideration of the user (Ceaparu et al., 2004; Lazar et al., 2006). These researchers have shown that up to 43% of possible productive time is lost due to frustration with very common tasks when using computers. Preventing frustration in this context would allow less time to be lost to frustration, rather than detecting frustration and reacting to the situation.

Positive emotions, on the other hand, have been widely overlooked in performance based research within the HCI domain. Given the large amount of research focused on problem solving performance and positive affect, it is interesting that positive affect in terms of task performance has not been extensively studied. This research seeks to show that promoting positive affect through human-computer interaction can have a positive impact on task performance and should be considered when creating system designs.

Research Objective

The overall objective of this research is to describe the interaction between affect and problem solving, facilitated through human-computer interfaces. More specifically, positively worded interface dialogs, specifically error messages, promote positive user affect and the resulting positive affect can have a measurable impact on problem solving performance. This research intends to show two specific aims: (1) positively worded error messages in interfaces promote positive affect and (2) positively worded error messages in interfaces have a measurable positive impact on problem solving performance. The m-MPS Model is used to describe the interaction affect and problem solving have on one another with respect to these specific aims. Figure 7 shows the theoretical valence-based affective roadmap of problem solving including the specific aims and their locations within the model. Figure 7 also includes annotations of measurements to be used for specific aspects of the roadmap.

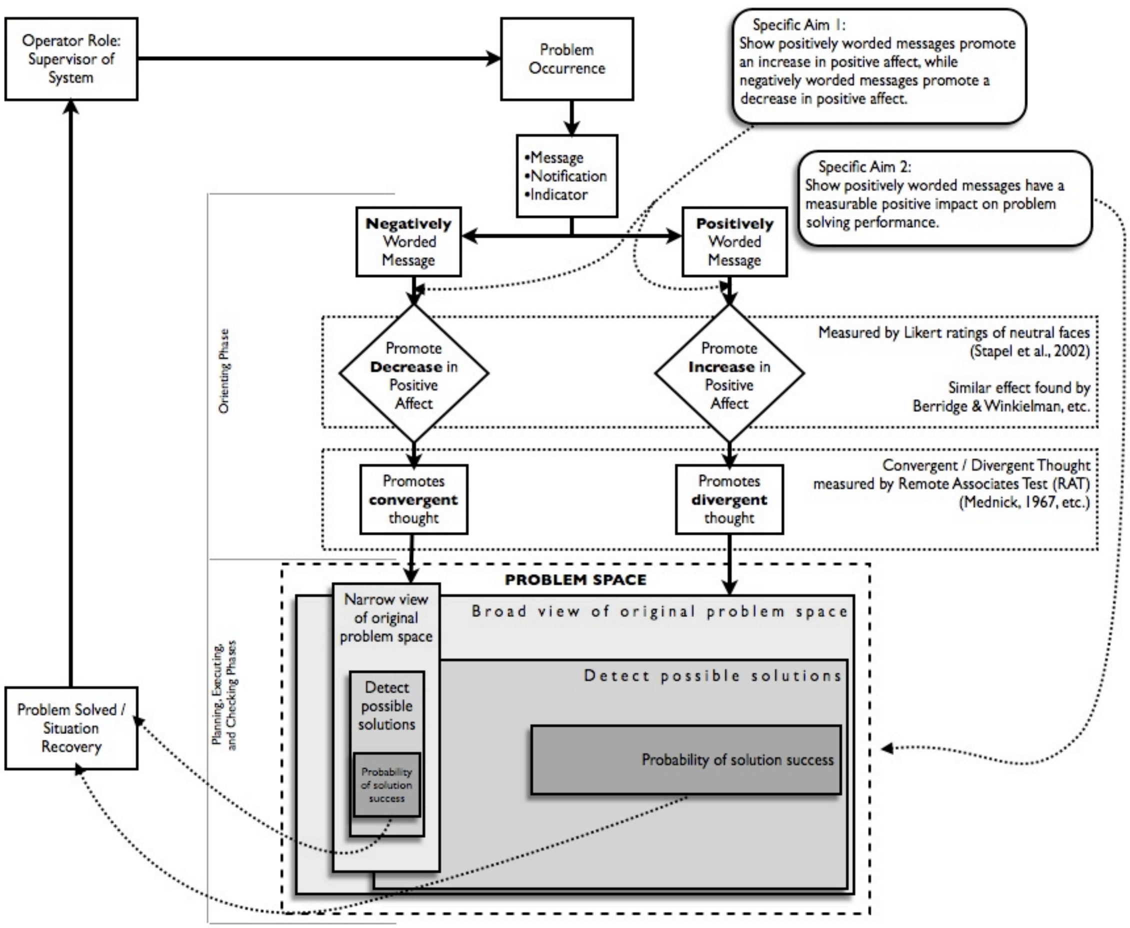

Figure 7: Theoretical valence-based affective roadmap of problem solving, developed for this dissertation research, including specific aims and measurement notations.

CHAPTER III. PILOT STUDY: IMPACT OF AFFECTIVE MESSAGES ON PROBLEM SOLVING PERFORMANCE

Introduction

A pilot study was conducted to determine the existence of an affect effect on task performance resultant from positive and negative messages. The task scenario, computer programming, was chosen because of the semi-structured nature of the problem solutions, along with the frequent occurrence of compiler error messages. The programming task is semi-structured because of the many potential solutions for a particular problem, though there are few which require the least amount of time and effort. The frequency of compiler error messages is beneficial for this particular type of experiment, ensuring frequent presentation of positive or negative stimuli.

Experimental Methods

Participants

For this pilot study, a total of 14 participants were recruited through the University of Iowa College of Engineering. Participants were between 18 and 32 years old with a mean age of 22.6 (s.d. 4.27) and had taken one but not more than two C-language programming classes, including a currently enrolled class.

Apparatus

The focus of this pilot study was centered on fixing, or debugging, a computer program written in the C computer programming language. A screen shot of the windows the participants interacted with is in Figure 8. The program each participant attempted to debug contained several logic errors, such as using the wrong variable for a piece of evaluative code. The programming errors, called bugs, in this program were typical programming problems suffered by novice programmers. There were six bugs that needed to be corrected to entirely fix the program. The program was written to read a separate text file, count particular characters in that text file, and print the results to the screen. The program can be found in Appendix A. An additional program, written in Python for the purpose of this study, served as a message modifying interlink program which intercepted messages from the compiler program and modified them (Figure 9). The interlink program presented compile messages based on two wording conditions: positively-oriented and negatively-oriented. Each message contained specific word and phrase substitutions to make the original compiler message more positive or negative. The list of messages in each category can be found below. The positive and negative words were selected from the Affective Norms for English Words (ANEW) list and have been shown to elicit positive or negative reactions in people (Bradley and Lang, 1999).

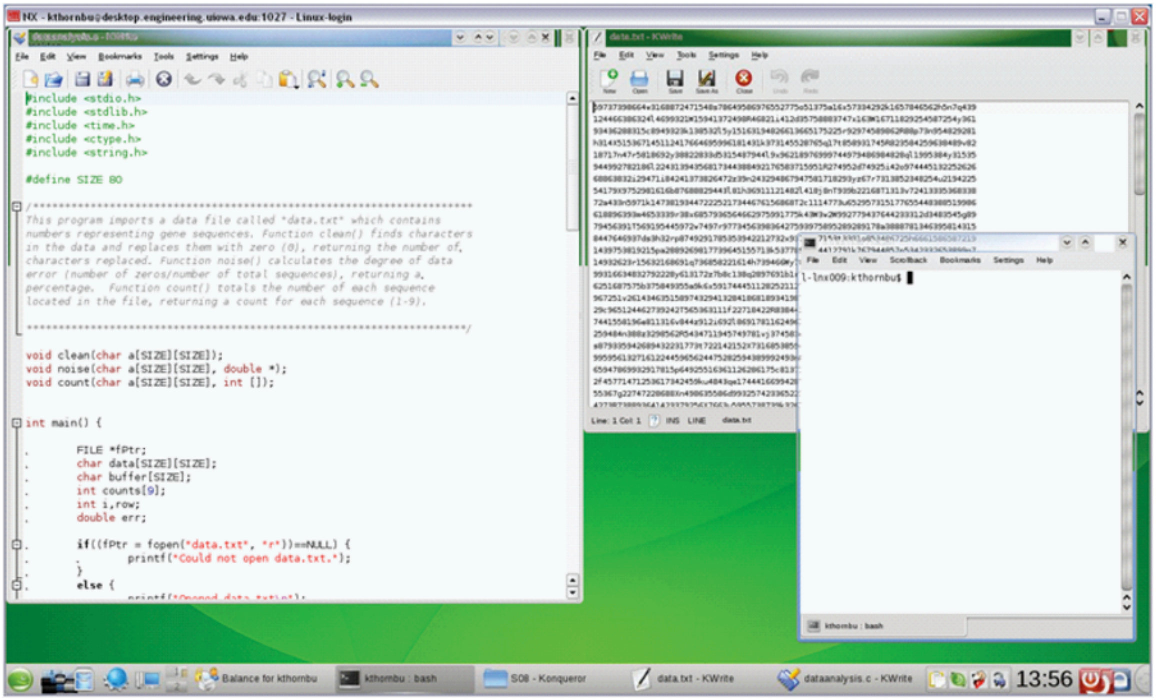

Figure 8: Screen shot of the interactive windows participants manipulate to complete the programming task.

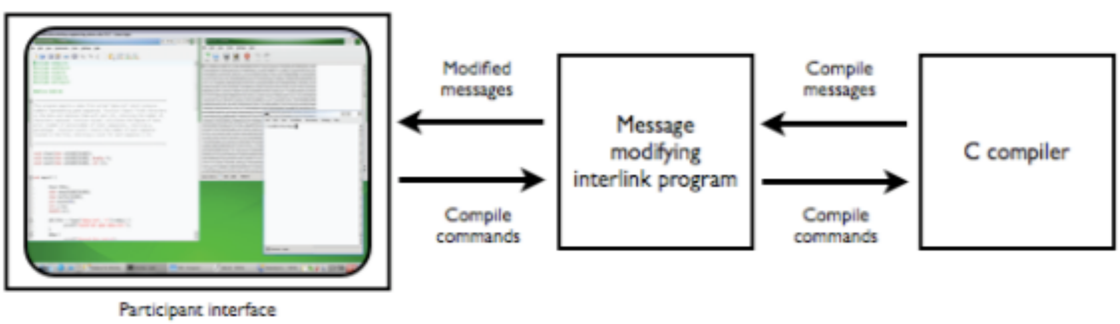

Figure 9: Illustration of the flow of messages through the interlink program between the participant and the C compiler program.

The modified error messages were designed to present the same amount of information but may present more information than the original error messages. The amount of information provided in the modified messages was increased to provide more opportunity to present affective words and to guide participants through the problem solving task. A semantic breakdown of the original message, the negative counterpart, and positive counterpart is shown in Table 2. Each column in Table 2 represents equivalent statements for each group of messages.

Table 2: Semantic breakdown for the original, negative, and positive message wording of the four standard error messages.

	Error Type	Equivalent Statements		
Error Message 1	warning:	passing	argument 1 of 'sprintf'	from incompatible pointer type
Negative Message	Execution will result in wrong output:	Avoid disaster by being sure	argument 1 of 'sprintf'	is not declared differently by mistake
Positive Message	Program output may not be correct:	Improve the program by ensuring	parameter 1 of 'sprintf'	is consistent with the declared type
Error Message 2	error:		incompatible	type for argument 2 of 'noise'
Negative Message	Execution failure!	To avoid disaster,	be sure there are not multiple	types for argument 2 of 'noise'
Positive Message	Program compiling had a problem	To improve, use this idea:	ensure consistent	types for parameter 2 of 'noise'
Error Message 3	error:		incompatible	types in assignment
Negative Message	Execution failure!	To avoid disaster,	be sure there are not multiple	types for variables
Positive Message	Program compiling had a problem	To improve, use this idea:	ensure consistent	types for variables
Error Message 4	Segmentation fault (core dumped)			
Negative Message	Fault destroyed program execution, cored dumped.	Avoid program abortion by	not attempting to access	outside array limits
Positive Message	Program progress stopped.	Improve the program with this idea:	ensure access	to arrays are within limits

The valences reported on the ANEW list are based on a 9-point scale with 1 being very negative and 9 being very positive. The words from the ANEW list chosen for the modified messages and associated valences are shown in Table 3. The mean valence for each of the messages are shown in Table 4. This leaves differences in valences for corresponding messages in the range between 4.33 and 5.40. Any compounding effects seen by combining words into sentences will increase these differences.

Table 3: Valences for each of the words selected from the ANEW list for the modified messages.

Word	Valence
abortion	3.50
destroy	2.64
disaster	1.73
dump	3.21
execution	2.37
failure	1.70
fault	3.43
idea	7.00
improve	7.65
mistake	2.86
progress	7.73

Table 4: Mean valences for the negative and positive versions of the four standard error messages.

Message	Negative Mean Valence	Positive Mean Valence	Valence Difference
Message 1	2.32	7.65	5.33
Message 2	1.93	7.33	5.40
Message 3	1.93	7.33	5.40
Message 4	3.13	7.46	4.33

Procedure

In this pilot study, participants filled out a PANAS scale (see Appendix B for the scale) to establish a baseline affect measure for both positive and negative affect. Participants were then asked to compile and debug the experimental program and were presented with either positive or negative error messages. Performance measures (total task time, time spent reviewing individual error messages, number of compile attempts, and number of errors corrected) were recorded in this experiment. Orienting phase time was determined as the time spent reviewing the error messages, while Planning and

Executing phase time was measured after reviewing the messages to compiling. After completing or abandoning the programming task, participants filled out a second PANAS scale to compare to the baseline measurement.

For this pilot study, the hypotheses reflected the specific aims: 1) that participants using the positively worded compiler will show an increase in positive affect as shown by the PANAS scales, while the participants using the negatively worded compiler will not show an increase in positive affect, and 2) that participants will have higher overall performance when using the positively worded compiler than when using the negatively worded compiler.

The performance measures recorded were:

1. Total task time - Amount of time taken to finish or abandon the task.

2. Time spent reviewing individual error messages - Indicates the amount of time spent in the Orienting phase for each problem.

3. Number of compile attempts - Indicates the number of iterations through the entire problem solving cycle.

4. Number of errors corrected - A direct measure of task performance in a programming environment.

Results

The positive and negative conditions were equally represented with seven participants each. The change in positive affect for all participants was calculated by subtracting the pretest score from the posttest score, resulting in a difference score of positive affect (ΔPA). There were no significant changes in positive affect for the positive wording condition ($t(12) = -0.86$, $p > 0.05$) or the negative wording condition ($t(12) = -0.15$, $p > 0.05$). For each of the performance measures, the positive and negative conditions were compared using the TTEST procedure in SAS 9.2. There was no significant difference in the total task time in the positive and negative conditions ($t(12) = 0.16$, $p > 0.05$), but interestingly, the number of errors corrected was higher for the positive condition than the negative condition ($t(12) = -4.77$, $p < 0.01$). This difference is highlighted by the fact that none of the participants in the negative wording condition finished the task (corrected all six errors). Additionally, the average amount of time spent in the Orienting phase was higher for the negative condition than the positive condition ($t(12) = 1.99$, $p < 0.05$), but the number of compile attempts was not different between conditions ($t(12) = 0.724$, $p > 0.05$).

Discussion

The wording of error messages has an impact on task performance. Unfortunately, a clear change in positive affect was not promoted in any of the participants in the positively worded condition. This may have been that the effect was too subtle to see with the number of participants used in the study or the large variability of affect experienced by the participants. Although the total task time was not different between conditions, more errors were corrected in the positively worded condition than the negatively worded condition. This indicates that the participants in the positively worded condition were more efficient and effective in correcting the programming errors. Additionally, none of the participants in the negatively worded condition finished the task, while 4 of the 7 participants in the positively worded condition finished and the remaining three participants fixed at least 50% of the errors.

The first hypothesis, that the message wording would change the affect of the participants was not supported, though it was not entirely refuted either. The second hypothesis was supported: the participants clearly performed the task more successfully with the positively worded messages. In terms of the m-MPS Model, participants in the negatively worded condition spent more time orienting to the problem, which could indicate a struggle to generate possible solutions for the problem. Also, where the participants in the negatively worded conditions made as many compile attempts, they were less successful. This suggests that their attempts were less well targeted towards the solution, perhaps because they repeated failed attempts or tried variations on an unsuccessful strategy.

There are several limitations in this study that should be addressed. The first limitation is the structure of the positive and negative messages in the task. Though only slightly different, it is possible the structures of the messages differ enough to cause the effect seen on performance. A second limitation is that the measurement of affect may not be sensitive enough. It is surprising that no significant difference in affect change was detected between the two groups, although one group was far less successful in the task than the other. It seems reasonable that succeeding in the task would increase one's positive affect, while giving up on the task would result in a reduction in positive affect, but such changes were not observed. It is possible the measuring instruments were not sensitive enough for this type of experiment. The third limitation is that the data collected during the experiment does not provide a clear picture of each participant's activities. Affect is only measured twice during the experimental procedure, which only allows for a linear analysis of affect. Not enough information is available to draw specific conclusions about the type of impact affect has on human-computer interactions, especially in the context of dynamic problem solving interactions, which necessitates procedural modifications for the following experiment.

CHAPTER IV. EXPERIMENT: IMPACT OF AFFECTIVE MESSAGES ON PROBLEM SOLVING PERFORMANCE

Introduction

In order to better understand the dynamic affect effects felt by the participants during the problem-solving process, the pilot study was modified to include more frequent measurements of affect. Also, to measure the participants' level of divergent thinking during the problem solving process, the Remotes Associates Test was administered. This experiment differs from the pilot study in the following ways:

1. Measurements of affect were taken periodically throughout the problem solving task by asking participants to rate the expression of emotionally neutral faces.
2. Measurements of divergent and convergent thinking were taken periodically throughout the task by periodically administering Remote Associates Test items.
3. The error message wording structures were the same for the positive and negative conditions: the positive and negative words were encapsulated in phrases prepended to the compiler error.
4. Each compile attempt taken by problem solvers was individually recorded.
5. Participants were restricted to fix each bug in a set order.

These modifications to the experimental design allow participants' problem solving behavior to be tracked at a higher resolution throughout the dynamic task and linked to the Modified Multidimensional Problem Solving Model.

<u>Experimental Methods</u>

Participants

A total of 36 participants (10 female, 26 male) were recruited through the University of Iowa College of Engineering and randomly assigned by gender to one of the two conditions. Participants were between 18 and 29 years old with a mean age of 19.8 (s.d. 2.43) and all participants had taken only one C programming class in a university setting.

Apparatus

The apparatus was the same as pilot study, with the following exceptions:

1. A dialog box was included after the onset of a *new* set of error messages that required the participant to quickly rate a neutral face for positivity or negativity on a 5-point Likert scale (Figure 10).
2. A second dialog box was included after the neutral face ratings that required the participant to quickly complete an easy, medium, and hard Remote Associates Test item sequentially.
3. The error messages were the original messages from the compiler, preceded by a positive or negative message which had no bearing on the error message.
4. A new experimental program (Appendix C) was developed to eliminate the runtime error (last bug) that appeared in the pilot study program. The program was designed to "deal" two hands of ten cards from a standard deck of playing cards with no repeating occurrences.

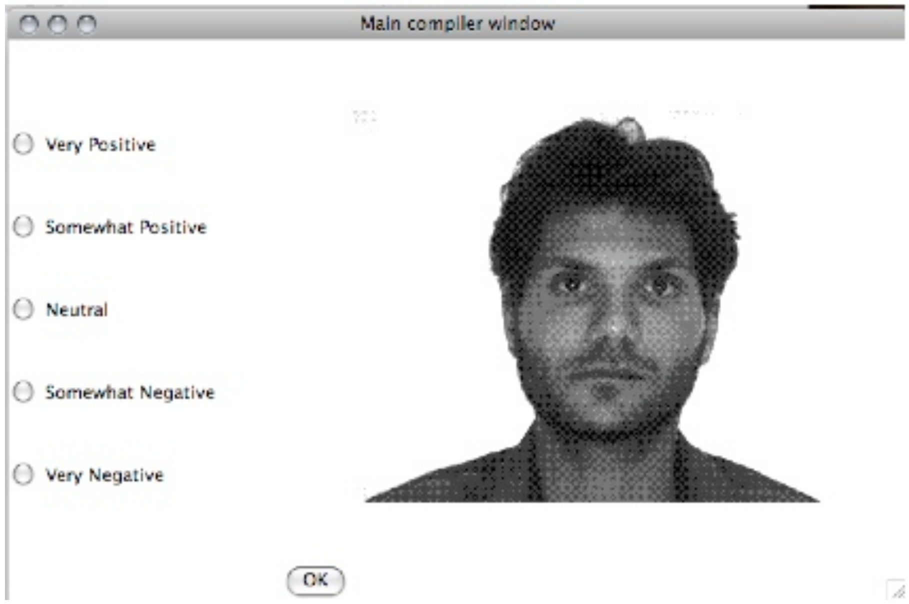

Figure 10: Example of a neutral face rating dialog box.

The neutral faces were selected from the AR Face Database (Martinez and Benavente, 1998). The database is a collection of 120 faces in various facial expressions. A set of 20 neutral faces were selected from the database based on ratings from five unbiased raters. The RAT items were randomly selected from the set developed and tested by Bowden and Jung-Beeman (2003). The normative data presented by Bowden and Jung-Beeman (2003) served as the basis for the selection of easy, medium difficulty, and hard items for each RAT set presented to the participant. The set of RAT items used in this study are in Appendix D. The messages were presented in a different format from the pilot study as well. Below are the messages which preceded the error message during the programming task. Words taken from the ANEW list that have a measured valence are presented in all capital letters. The pound symbol (#) represents the remaining number of compile errors.

Positive messages:

`PROGRESS!  Just # bug(s) remaining. Generate a new IDEA by reviewing the compiler response to IMPROVE your program:`

`New IMPROVEMENT IDEA! The compiler may have a helpful suggestion to RESCUE the program by overcoming one of the # remaining bug(s). See if you can SATISFY the compiler’s concerns:`

`GOOD PROGRESS! The compiler has a suggestion to IMPROVE your program. You have # issue(s) remaining. ACE the program by ACHIEVING this message:`

`Program IMPROVEMENT! The compiler has identified # challenge(s) remaining. Perhaps this IDEA is an EXCELLENT one for addressing these challenges:`

`QUALITY IDEA! The compiler detected a bug. You have # bug(s) remaining. Use your CAPABLE skills to IMPROVE the program:`

Program IMPROVEMENT! You have made PROGRESS, but the compiler detected a bug. You have # bug(s) left. Use this IDEA to IMPROVE the program:

Program IDEA! The compiler has identified # challenge(s) remaining. RESCUE the program by SATISFYING this statement:

Program PROGRESS! The compiler had an issue. There are # issue(s) left to fix. IMPROVE the program by using this IDEA:

Negative messages:

EXECUTION FAILURE! The program FAILED to compile. You have # mistakes(s) left to fix. To avoid DISASTER, fix this error:

The compiler CRASHED! The program was REJECTED by the compiler. You have # fault(s) left to fix. Use this DAMAGE report to resolve the CRISIS:

EXECUTION DISASTER! The program CRASHED the compiler. You have # problem(s) left to fix. To avoid a CRISIS, fix this MISTAKE:

```
ABORT compile! The program DESTROYED the compile attempt.
You have # mistake(s) left. To avoid compiler REJECTION,
fix this FAULT:

REJECTED EXECUTION! The compiler ABORTED the program. You
have # fault(s) left. Use this FAILURE message to fix this
MISTAKE:

Program CRISIS! The compiler REJECTED the program. You have
# mistake(s) remaining. Use this CRASH message to fix the
FAILURE:

EXECUTION CRASH! The program ABORTED the compile attempt.
You have # fault(s) remaining. To avoid TERRIBLE
consequences, fix this FAILURE:

Compiler CRISIS! The program was a DISASTER for the
compiler. You have # problem(s) remaining. Use this FAULT
message to fix the compiler REJECTION:
```

The interlink program was modified from the pilot study to record each compile attempt by a participant, along with randomly selecting an affective message to be presented with the compile message. The interlink program also randomly selected a neutral face and a RAT item from the easy, medium, and hard sets when appropriate. Figure 11 illustrates the purpose of the interlink program in this study.

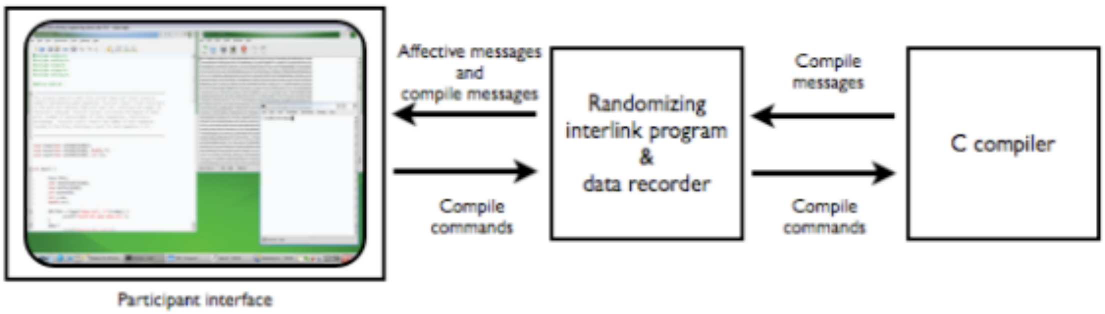

Figure 11: A representation of the interlink program's interaction between the participant and C compiler.

Procedure

After brief introductions, participants filled out a short questionnaire to ensure their eligibility to participate and a PANAS scale to establish a baseline measure for both positive and negative affect. A brief orientation to the task followed, introducing the programming task, as well as the intermittent face ratings and Remote Associates Test (RAT) items. The two non-programming tasks were framed as distractions to the participant, but the participant was asked to do their best on the tasks. The participant was then asked to compile and debug the experimental program using the modified compiler with either positive or negative messages. If the compiler discovered several errors in the program, it only displayed the first compile message in the set. This forced participants to solve each bug serially in order to control the problem solving process across each participant for analysis purposes. Each compile message was preceded by a randomly selected affective message depending upon condition. Each set of messages was required to be reviewed and acknowledged by the participant by clicking a button on the screen.

Performance measures, including total task time, time spent on each bug, total number of compile attempts, number of compile attempts for each bug, and number of errors corrected, were recorded in this experiment. Each modification to the experimental program was saved, along with the associated compile messages presented to the participant. After completing or abandoning the programming task, participants filled out a second PANAS scale to anchor the positive affect measurements.

For this experiment, the hypotheses were similar to the pilot study. The first hypothesis was that participants using the positively modified compiler will show an increase in positive affect as indicated by the PANAS scales and neutral face ratings, while the participants using the negatively modified compiler will not show an increase in positive affect as shown by the PANAS scales and neutral face ratings. The second hypothesis was that participants will have higher overall performance and associated divergent thought as measured by the RAT when using the positively modified compiler than when using the negatively worded compiler. Additionally, a third hypothesis was included, predicting that the m-MPS Model reflects the problem solving mechanism displayed by problem solvers.

The performance measures that were recorded are:

1. Neutral face ratings - An indirect measure of affect.
2. RAT scores - A direct measure of divergent thinking.
3. Task time (total and per bug) - Amount of time taken to finish or abandon the task.
4. Number of compile attempts (total and per bug) - An indicator of problem solving efficiency.
5. Number of bugs corrected - A direct measure of task performance in a programming environment.

Programming solutions used more than once by a participant (solution cycles) were recorded by a comparative analysis of each modification performed by the participant. This analysis was done after the experiment's conclusion.

Results

This study examined the effect non-task related messages had on positive affect, divergent thought, and problem solving performance in two conditions: positive-wording and negative-wording. The following sections will describe the results from these three areas of interest and then review their interaction with reference to the modified Multidimensional Problem Solving Model. Please note that data from two participants in the positive-wording condition were removed because of data collection failure.

Positive Affect

The first specific aim of this research was to show that positively worded messages presented throughout the task would increase positive affect, while negatively worded messages would not. Positive affect was measured pre-task and post-task using the Positive Affect and Negative Affect Schedule (PANAS) scales, resulting in a positive affect score between 10 and 50 for each measurement. The change in positive affect (ΔPA) was calculated by subtracting the pre-task score from the post-task score. A positive result indicates an increase in positive affect over the duration of the task, while a negative result indicates a decrease in positive affect.

The positive affect data was analyzed using a between-subjects ANOVA model implemented using the GLM procedure in SAS 9.2 to examine the effects that wording condition, gender, and their interaction had on ΔPA. The results of this ANOVA are in Table 5. Though not explicitly stated in the specific aims, the same test was conducted for the change in negative affect (ΔNA), also measured by the PANAS scales. The results of the ΔNA ANOVA are in Table 6.

Table 5: ANOVA results for ΔPA.

Dependent Variable	**ΔPA**				
Source	**DF**	**Sum of Squares**	**Mean Square**	**F Value**	**Pr > F**
Model	3	200.32	66.77	1.59	0.211
Condition	1	76.52	76.52	1.83	0.187
Gender	1	29.59	29.59	0.71	0.407
Condition*Gender	1	30.94	30.94	0.74	0.397
Error	30	1256.42	41.88		
Corrected Total	33	1456.74			

Table 6: ANOVA results for ΔNA.

Dependent Variable	ΔNA				
Source	**DF**	**Sum of Squares**	**Mean Square**	**F Value**	**Pr > F**
Model	3	107.76	35.92	1.08	0.373
Condition	1	18.38	18.38	0.55	0.463
Gender	1	28.61	28.61	0.86	0.361
Condition*Gender	1	80.70	80.70	2.42	0.130
Error	30	998.71	33.29		
Corrected Total	33	1106.47			

The positive affect data was also analyzed as their raw scores without normalizing, using a mixed ANOVA model implemented using the MIXED procedure in SAS 9.2 to examine the effects wording condition, gender, initial positive affect, and the associated interactions had on the ending positive affect measurement. The results from this analysis are in Table 7. The same analysis was conducted for the negative affect measurements, the results of which are in Table 8. These results suggest the post-task measurements of affect only significantly rely on the pre-task measurements.

Table 7: Mixed ANOVA results for raw PA scores.

Dependent Variable	End PA			
Source	**Num DF**	**Den DF**	**F Value**	**Pr > F**
Condition	1	26	0.32	0.574
PA Start	1	26	9.10	0.006 **
Gender	1	26	0.24	0.627
Condition*Gender	1	26	0.02	0.877
PA Start*Condition	1	26	0.13	0.719
PA Start*Gender	1	26	0.07	0.792
PA*Gender*Condition	1	26	0.09	0.766
**** Denotes significance at α = 0.05**				

Table 8: Mixed ANOVA results for raw NA scores.

Dependent Variable	**End NA**			
Source	**Num DF**	**Den DF**	**F Value**	**Pr > F**
Condition	1	26	0.34	0.568
NA Start	1	26	11.47	0.002 **
Gender	1	26	0.00	0.944
Condition*Gender	1	26	0.62	0.438
NA Start*Condition	1	26	0.52	0.479
NA Start*Gender	1	26	0.23	0.636
NA*Gender*Condition	1	26	1.32	0.261
**** Denotes significance at α = 0.05**				

The previous analysis only examines the change in positive affect (and negative affect) from the beginning of the trial to the end, which can only be used to generally characterize the overall impact on positive affect. This study included intermittent neutral face ratings as an indicator of the increase or decrease of positive affect at that particular moment in time. The participants could rate the neutral faces as Very Positive, Somewhat Positive, Neutral, Somewhat Negative, and Very Negative. These ratings were coded for analysis as 2, 1, 0, -1, and -2, respectively. These data show a dynamic view of positive affect over the course of the problem solving task. A second analysis was conducted which examines the repeated aspect of the face rating measurements and the iterative nature of the problem solving task. A repeated measures ANOVA was conducted using the MIXED procedure in SAS 9.2 which examines how the repeated face ratings were affected by the condition, the particular bug being solved, and the interaction between condition and bug. The MIXED procedure allows for the repeated measures analysis to be blocked by participant, which will account for participants having different numbers of trials. The results of this analysis are shown in Table 9. These results suggest that the condition and individual bug being solved do not significantly impact face ratings as individual effects, but do so as an interaction. Figure 12 shows the least-squares means plot for each condition by bug, indicating an interaction at the fourth bug. This interaction could possibly indicate that participants in the negative condition felt better about their solutions for the fourth bug than participants in the positive condition did, which could be related to the participants' perception of their problem solving performance. Additionally, the fourth bug is essentially a consistency error, wherein participants could solve the problem simply by comparing the problem line (the function definition) with the defining line (function prototype). Ensuring the two lines contain the same information can be done relatively quickly.

Table 9: Outcome of the repeated measures ANOVA for face ratings.

Repeated ANOVA Face Ratings			
Effect	**DF**	**F Value**	**Pr > F**
Condition	1	0.11	0.744
Bug	4	1.77	0.143
Condition*Bug	4	2.64	0.039 **
*** Denotes nearing significance**			
**** Denotes significance at α = 0.05**			

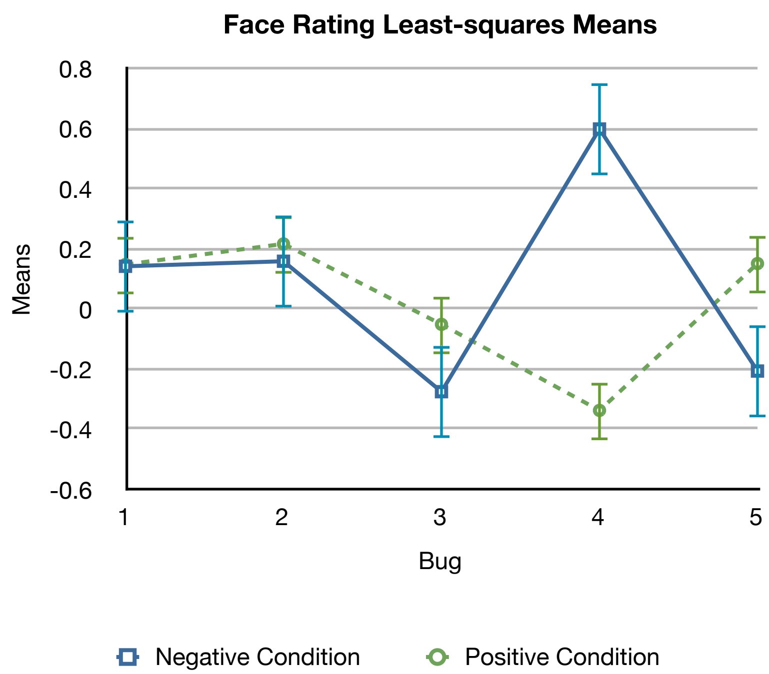

Figure 12: Least-squares means for face ratings by each bug with standard error.

<u>Divergent Thought</u>

Divergent thought was measured in this study through participant responses to a series of Remote Associates Test (RAT) items. The RAT items were presented in groups of three which included an easy, medium, and hard difficulty level item. The difficulty level was determined based on the normative data collected by Bowden and Jung-Beeman (2003) and each level was weighted according to difficulty (1 for correct easy response, 2 for correct medium response, and 3 for correct hard response). For each set of RAT items, the possible score ranged from 0 (no correct responses) to 6 (all correct responses). This score represents the participant's disposition for divergent or creative thought at the particular moment in time the measurement was taken.

The RAT scores were averaged for each participant across their entire trial. The average RAT score data was analyzed using a between-subjects ANOVA model implemented using the GLM procedure in SAS 9.2, considering the condition, participant gender, and the interaction. Table 10 contains the results of this ANOVA, indicating no significant effects.

Table 10: ANOVA results for average RAT scores.

Dependent Variable	**Average RAT Score**				
Source	**DF**	**Sum of Squares**	**Mean Square**	**F Value**	**Pr > F**
Model	3	2.33	0.78	0.58	0.635
Condition	1	0.01	0.01	0.01	0.923
Gender	1	1.23	1.23	0.91	0.347
Condition*Gender	1	1.07	1.07	0.79	0.381
Error	30	40.49	1.35		
Corrected Total	33	42.82			

A second analysis was conducted to examine the effect the repeated nature of the problem solving process had on RAT scores. Since a series of measurements were taken throughout the task, a repeated measures ANOVA was conducted using the MIXED procedure in SAS 9.2, examining the effect the condition, bug being solved, and their interaction had on particular RAT scores. This analysis was blocked by participant, the results of which can be found in Table 11. This analysis indicates that the bug effect is approaching significance for RAT scores, but the condition effect is not. Figure 13 shows the least-squares means plot for each condition by individual bug.

Table 11: Outcome of the repeated measures ANOVA for RAT scores.

Repeated ANOVA RAT Scores			
Effect	**DF**	**F Value**	**Pr > F**
Condition	1	0.02	0.889
Bug	4	2.20	0.076 *
Condition*Bug	4	0.34	0.853
*** Denotes nearing significance**			
**** Denotes significance at α = 0.05**			

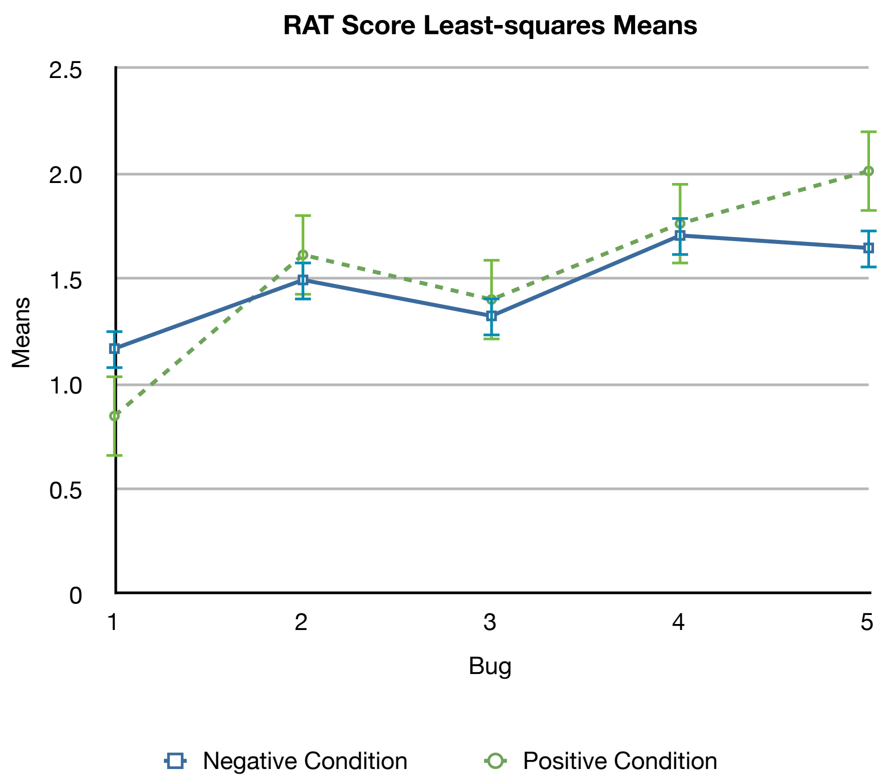

Figure 13: Least-squares means for RAT scores by condition for each bug with standard error.

Problem Solving Performance

The second specific aim of this research was to show that positively worded non-task related messages presented throughout the task would have a positive impact on problem solving performance when compared with negatively worded messages. In this study, problem solving performance was measured in terms of task completion (0 for incomplete, 1 for complete), number of bugs solved (0 through 5 bugs), total task time, and time spent on each bug. Additional performance measures include number of solution cycles (using a solution more than once) and number of compile attempts.

The performance measures were analyzed in two stages. The first stage comprised a series of ANOVAs designed to examine each performance measure in terms of the condition, participant gender, and the interaction between the two. These analyses were conducted using the GLM procedure in SAS 9.2. Table 12 contains the results for the Total Task Time measure, indicating a significant effect from gender in the model, though the overall model is not significant. Table 13 shows the results for the Compile Attempts measure, indicating no significant effects. Table 14 summarizes the results for the Solution Cycles measure, also resulting in no significant effects.

Table 12: Results of the ANOVA for Total Task Time.

Dependent Variable	Total Task Time				
Source	**DF**	**Sum of Squares**	**Mean Square**	**F Value**	**Pr > F**
Model	3	21.87	7.29	1.96	0.142
Condition	1	237330.6	237330.6	0.23	0.633
Gender	1	6256366.1	6256366.1	6.14	0.019 **
Condition*Gender	1	407950.9	407950.9	0.40	0.532
Error	30	111.7	3.72		
Corrected Total	33	133.53			
*** Denotes nearing significance**					
**** Denotes significance at α = 0.05**					

Table 13: Results of the ANOVA for Compile Attempts.

Dependent Variable	Compile Attempts				
Source	**DF**	**Sum of Squares**	**Mean Square**	**F Value**	**Pr > F**
Model	3	58.58	19.53	0.11	0.953
Condition	1	0.23	0.23	0.00	0.971
Gender	1	34.66	34.66	0.20	0.662
Condition*Gender	1	23.29	23.29	0.13	0.720
Error	30	5322.95	177.43		
Corrected Total	33	5381.53			

Table 14: Results of the ANOVA for Solution Cycles.

Dependent Variable	Number of Solution Cycles				
Source	**DF**	**Sum of Squares**	**Mean Square**	**F Value**	**Pr > F**
Model	3	31.59	10.53	1.69	0.191
Condition	1	6.18	6.18	0.99	0.328
Gender	1	0.80	0.80	0.13	0.723
Condition*Gender	1	12.58	12.58	2.01	0.166
Error	30	187.38	6.25		
Corrected Total	33	218.97			

The second stage comprised two specialized analyses which would account for the distribution of the data. First, a non-parametric median t-test using the NPAR1WAY procedure in SAS 9.2 was conducted to examine the difference between conditions for the number of bugs solved. The non-parametric method was chosen to account for the bimodal aspect of the number of bugs solved. Next, a logistic test was conducted using the LOGISTIC procedure in SAS 9.2 to determine the difference in binary task completion between conditions. The logistic method was chosen to accommodate the binary outcome in the analysis. The results of these tests are in Table 15. Figure 14 shows the percentage of participants that solved individual bugs.

Table 15: Analysis of number of bugs solved and completed tasks between conditions.

Performance Measure Results	**Positive Basic Stats**	**Negative Basic Stats**	**Statistic**
Number of Bugs Solved	3.69 (2.02)	2.61 (1.91)	$X^2 = 4.13$, $p = 0.042$ **
Completed Task	11	6	Wald $X^2 = 4.06$, $p = 0.044$ **
*** Denotes nearing significance**			
**** Denotes significance at $\alpha = 0.05$**			

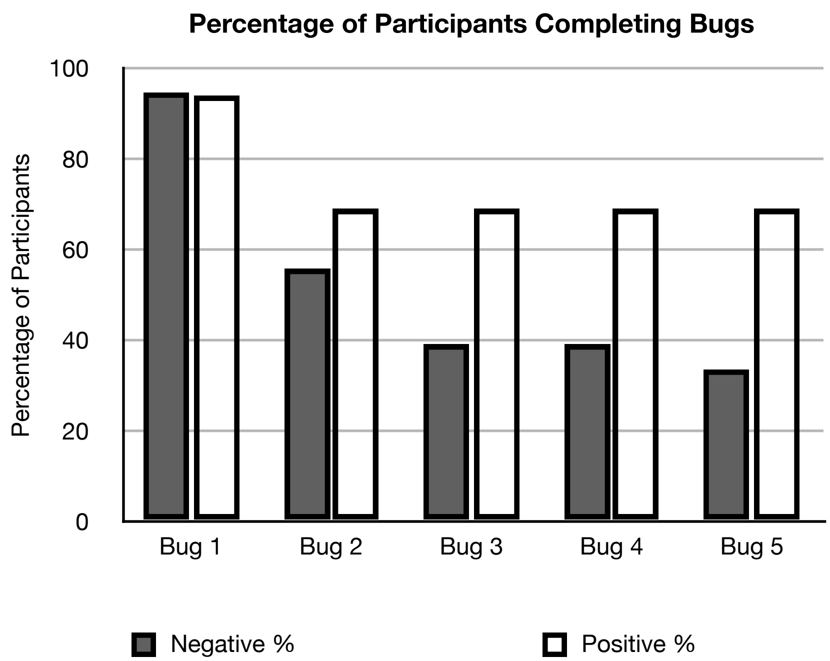

Figure 14: Percentage of participants completing each bug in each condition.

Since the overall models for Total Task Time and Compile Attempts were not significant and because it is clear there is a difference between conditions for the number of bugs solved, an analysis of performance measures for each bug was also conducted in terms of time spent on each bug and number of compile attempts for each bug. The MIXED procedure with the Satterthwaite approximation in SAS 9.2 was used to conduct a repeated ANOVA for each of the measures in terms of the condition, the bug being solved, and the interaction. Table 16 contains the results from this analysis for the compile attempts for each bug, indicating a significant effect from the bug. Figure 15 shows the number of compile attempts for each bug per condition. Table 17 contains the repeated ANOVA results for the time spent per bug, revealing no significant effects.

Table 16: Outcome of the repeated ANOVA for bug compile attempts.

Repeated ANOVA for Compile Attempts				
Effect	**Num DF**	**Den DF**	**F Value**	**Pr > F**
Condition	1	113	0.38	0.538
Bug	4	112	11.68	<0.001 **
Condition*Bug	4	112	0.72	0.581
*** Denotes nearing significance**				
**** Denotes significance at α = 0.05**				

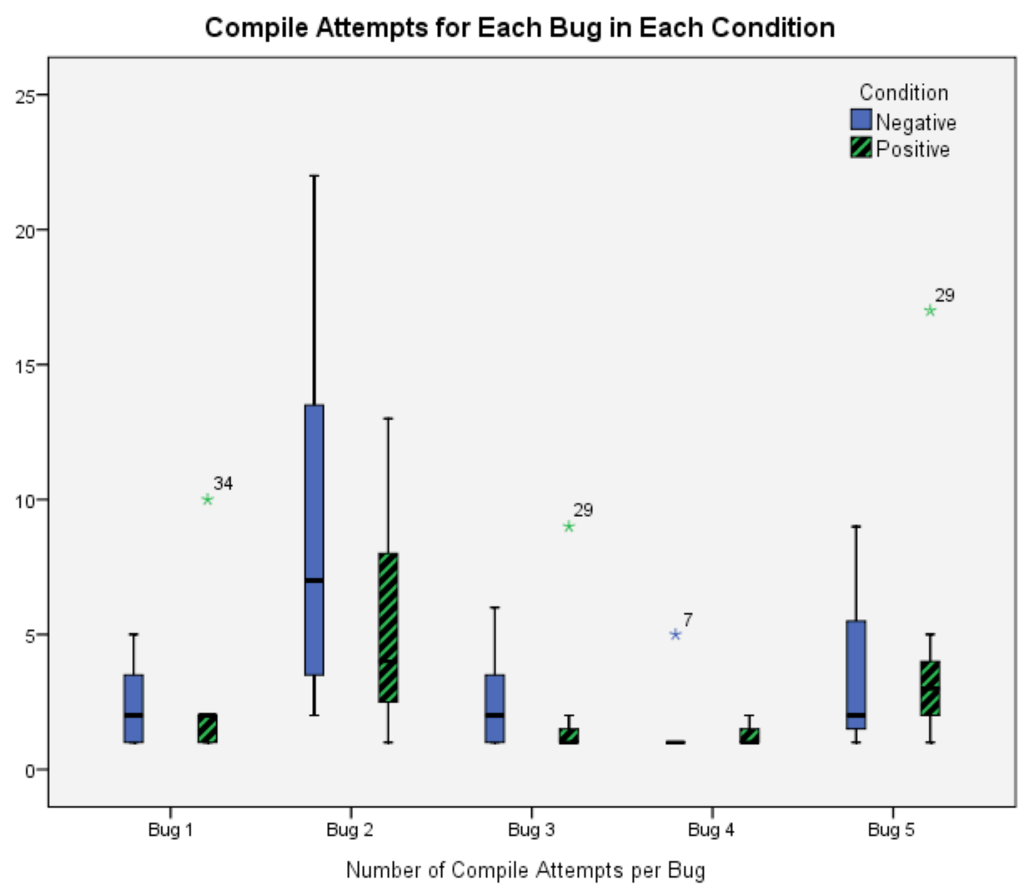

Figure 15: Number of compile attempts for each bug in each condition.

Table 17: Outcome of the repeated ANOVA for time spent on each bug.

Repeated ANOVA Time per Bug			
Effect	**DF**	**F Value**	**Pr > F**
Condition	1	0.40	0.534
Bug	4	0.90	0.466
Condition*Bug	4	0.18	0.949
*** Denotes nearing significance**			
**** Denotes significance at α = 0.05**			

Model Predictions

Figure 16 shows the underlying theory behind the modified Multidimensional Problem Solving (m-MPS) Model, along with the variables recorded during the experiment that represent each theoretical construct. The variable blocks have rounded corners and italicized text in the figure. Each link identified by a letter was examined to identify the particular relationship exhibited by that link to lend support to this underlying theory.

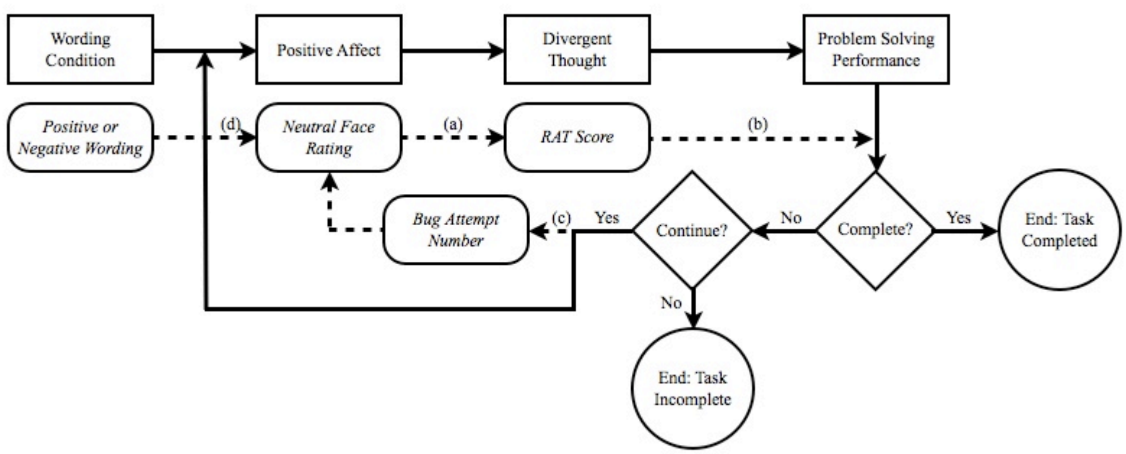

Figure 16: The underlying theory of the m-MPS Model including the variables associated with each theoretical block.

Link (a) represents the relationship between positive affect and divergent thought. Positive affect was measured by neutral face ratings on a Likert scale (-2, -1, 0, 1, 2) and divergent thought was measured by responses to three Remote Associates Test (RAT) items, resulting in a RAT score between 0 and 6 inclusive. Research suggests that increased positive affect results in increased divergent thought, indicating a positive relationship. This relationship was examined by performing a one-way between-within repeated measures mixed ANOVA using the MIXED procedure in SAS 9.2. The mixed model is represented as:

$$\text{Face rating} = \text{RAT Score} * \beta + \text{Participant Effect} * \gamma + \varepsilon$$

where β is the unknown vector of fixed-effects, γ is the unknown vector of random-effects and ε is the unknown random error vector.

The results from the mixed model indicate a positive relationship between face ratings and RAT scores that is nearing significance, shown in Table 18. Though the relationship shown through this experiment may not necessarily be causal, a relationship between the two variables may exist, which would support the first prediction of the theoretical model and the literature in general. Figure 17 shows the face ratings with associated RAT scores for each bug. Figure 18 shows the trend of face ratings and RAT scores for each error across all participants.

Table 18: Results of the mixed model analysis for face ratings and RAT scores.

Mixed ANOVA of RAT Scores				
Effect	**Num DF**	**Den DF**	**F Value**	**Pr > F**
Face Rating	1	423	3.66	0.056 *
*** Denotes nearing significance at α = 0.05**				

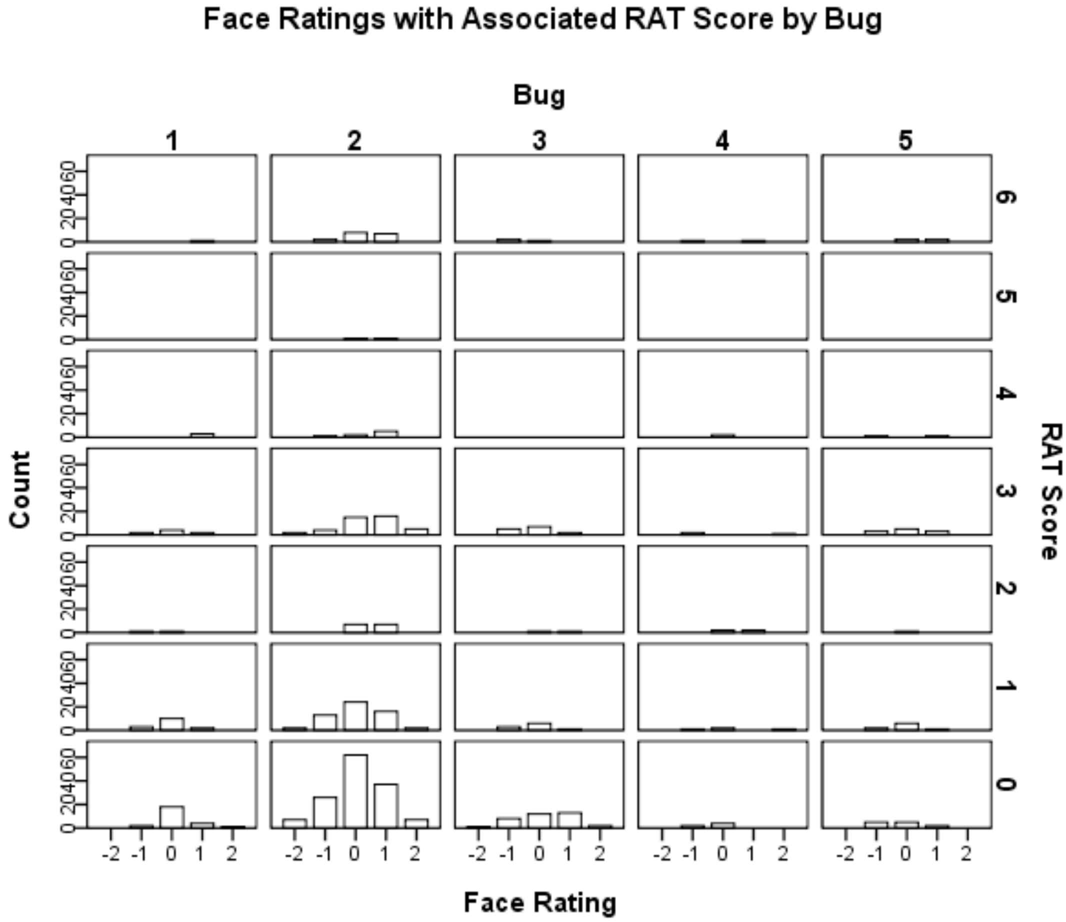

Figure 17: Face ratings and associated RAT scores for each bug.

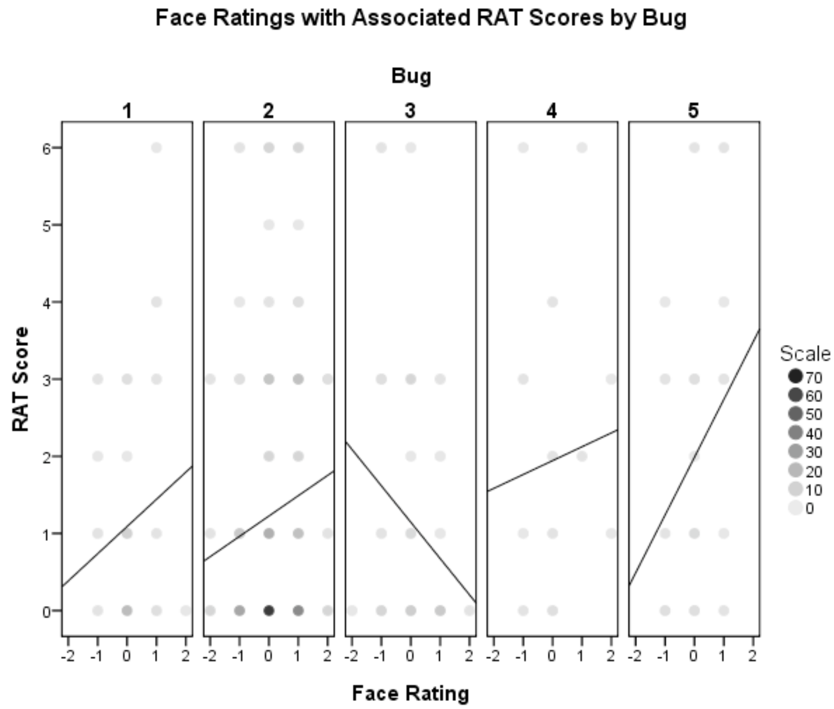

Figure 18: Face ratings and associated RAT scores for each bug for all participants.

Figure 18 indicates a general positive trend between face ratings and RAT score for all bugs with the exception of Bug 3. The negative relationship reflected in Bug 3 is most likely the reason the regression between face ratings and RAT score is nearing significance and is not decidedly significant. To examine this further, a general linear model was developed using the GLM procedure in SAS 9.2 which models RAT scores based on face rating and bug, along with the interaction between face rating and bug. In addition, each bug was analyzed individually. The comprehensive results are shown in Table 19, which indicate a significant overall model including significant effects of bug and the interaction between face rating and bug. Additionally, Bug 2 and Bug 3 indicate significant relationships between face ratings and RAT scores, though Figure 18 suggests opposite relationships.

Table 19: Results of the General Linear Model for RAT Scores.

General Linear Model for RAT Scores					
Source	**DF**	**Sum of Squares**	**Mean Square**	**F Value**	**Pr > F**
Model	9	69.28	7.70	2.70	0.004 **
Face Rating	1	6.48	6.48	2.27	0.132
Bug	4	32.32	8.08	2.83	0.024 **
Face*Bug	4	28.59	7.15	2.51	0.042 **
Error	448	1277.25	2.85		
Corrected Total	457	1346.53			
Model of Bug 1	1	2.90	2.90	1.41	0.241
Error	52	107.19	2.06		
Corrected Total	53	110.09			
Model of Bug 2	1	15.72	15.72	5.35	0.021 **
Error	277	813.58	2.94		
Corrected Total	278	829.30			
Model of Bug 3	1	10.43	10.43	4.18	0.045 **
Error	63	157.32	2.50		
Corrected Total	64	167.75			
Model of Bug 4	1	0.55	0.55	0.15	0.705
Error	19	70.40	3.71		
Corrected Total	20	70.95			
Model of Bug 5	1	11.15	11.15	3.21	0.082
Error	37	128.74	3.48		
Corrected Total	38	139.90			
*** Denotes nearing significance**					
**** Denotes significance at $\alpha = 0.05$**					

Link (b) represents the relationship between divergent thought, measured by the RAT score, and the number of attempts a participant took to solve a particular bug. A special case of a solution attempt is when a participant uses a solution previously attempted. This occurrence is termed a solution cycle. Theoretically, higher RAT scores should be associated with fewer solution cycles, suggesting the more divergently someone is thinking, the more unique solutions will be generated. On the other hand, the lower the RAT score, the higher the number of solution cycles, which could potentially lead to more trials than are necessary to solve the problem. This relationship was examined through a one-way between-within repeated measures mixed model using the MIXED procedure in SAS 9.2, using RAT score as the predictor variable and cycles as the dependent variable. The mixed model is represented as:

$$\text{Solution Cycles} = \text{RAT Score} * \beta + \text{Participant Effect} * \gamma + \varepsilon$$

where β is the unknown vector of fixed-effects, γ is the unknown vector of random-effects and ε is the unknown random error vector.

Table 20 indicates a significant relationship between RAT scores and solution cycles. Figure 19 indicates the relationship between RAT scores and solution cycles is a negative relationship, suggesting that higher RAT scores will less likely have an occurrence of one or more solution cycles. Each point is labeled with the number occurrences of solution cycles at each level of RAT score.

Table 20: Regression output for RAT scores and solution cycles.

Dependent Variable	**Solution Cycles**			
Effect	**Num DF**	**Den DF**	**F Value**	**Pr > F**
RAT Scores	1	423	12.39	0.0005 **
**** Denotes significance at $\alpha = 0.05$**				

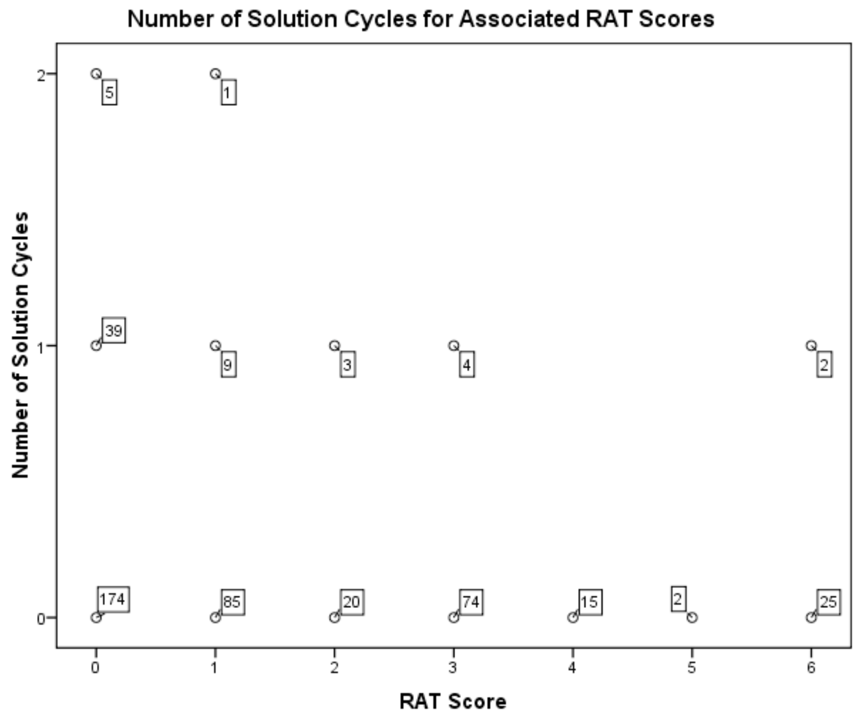

Figure 19: Number of solution cycles within each RAT score over all participants.

Higher RAT scores, though, will not necessarily lead to fewer attempts for a particular bug because higher RAT scores should lead to more unique solutions. Therefore, following the natural convergence of the problem solving process, it is expected that extended efforts (many unsuccessful attempts) to generate a solution will result in lower RAT scores. In other words, the natural convergence of problem solving will be reflected in the RAT scores. This relationship was examined by correlating the RAT score and associated attempt number using the GENMOD procedure with repeated measures in SAS 9.2. The results suggest a significant negative relationship between RAT scores and the number of attempts to solve a particular bug ($Z = -3.17$, $p = 0.002$).

This test lends support to the relationship described, supporting the second prediction of the theoretical model. Figure 20 shows the relationship between the RAT scores and the attempt number to solve a bug along with the general trend line. Figure 21 shows the relationship by each bug.

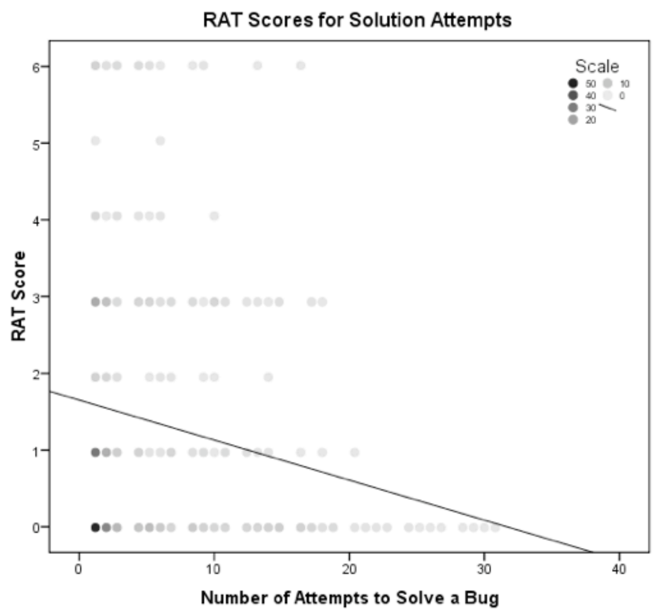

Figure 20: RAT scores and associated attempt number for all bugs over all participants.

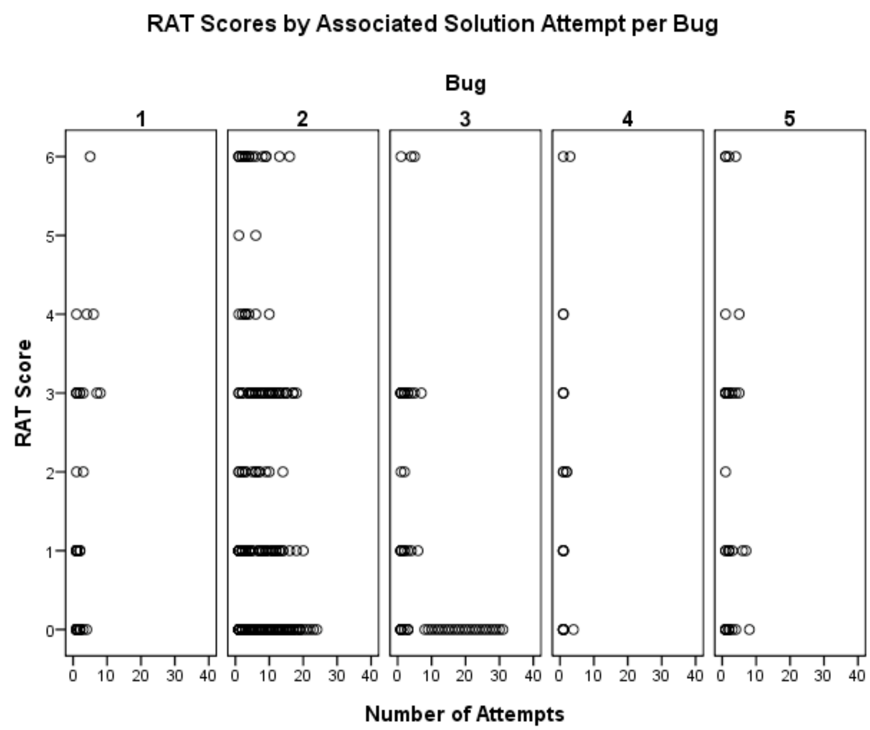

Figure 21: Attempts for each bug and the associated RAT score over all participants.

To examine this relationship further and account for random effects from individual subjects, a general linear model of RAT scores was generated using the GLM procedure in SAS 9.2 considering the number of attempts on particular bugs, a random subject factor, and the interaction between the two. Table 21 shows the results from this analysis, indicating that the overall model and all factors are significant.

Table 21: Outcome of the general linear model of RAT scores.

General Linear Model of RAT Scores					
Source	**DF**	**Sum of Squares**	**Mean Square**	**F Value**	**Pr > F**
Model	66	644.09	9.76	5.43	<0.01 **
Attempts	1	46.35	46.35	25.80	<0.01 **
Subject	33	464.86	14.09	7.84	<0.01 **
Attempts*Subject	32	132.89	4.15	2.31	<0.01 **
Error	391	702.43	1.79		
Corrected Total	457	1346.53	74.56		
*** Denotes nearing significance**					
**** Denotes significance at α = 0.05**					

Link (c) represents the relationship between the problem solver's willingness to continue with the problem solving process and the number of attempts to solve a particular bug. This relation is derived from the hypothesis that the more unsuccessful attempts encountered will result in a lower positive affect and a higher probability of abandoning the task. Previously, it was shown that participants in the positive condition were more likely to solve all five bugs in the problem solving task, though there was no statistical difference between the conditions for the number of compile attempts overall or for each bug. This analysis takes the previous tests a step further by analyzing the probability of continuation, given the number of previous attempts to solve the bug. This analysis was conducted using the GENMOD procedure with the logit option in SAS 9.2, predicting continuation (a binary dependent variable) using the solution attempt number as the independent variable. The results indicate that the probability of continuing will decrease with each additional unsuccessful solution attempt, as shown in Table 22. This suggests that the more unsuccessful solutions attempted, the less likely a participant will choose to continue with the task. Figure 22 shows the predicted and actual probabilities of continuing based on solution attempt.

Table 22: Logistic continuation model results.

Logistic Continuation Model				
Parameter	**Estimate**	**Standard Error**	**Z Stat**	**Pr > |Z|**
Intercept	4.581	0.540	8.49	<0.001 **
Attempts	-0.105	0.035	-2.98	0.003 **
**** Denotes significance at α = 0.05**				

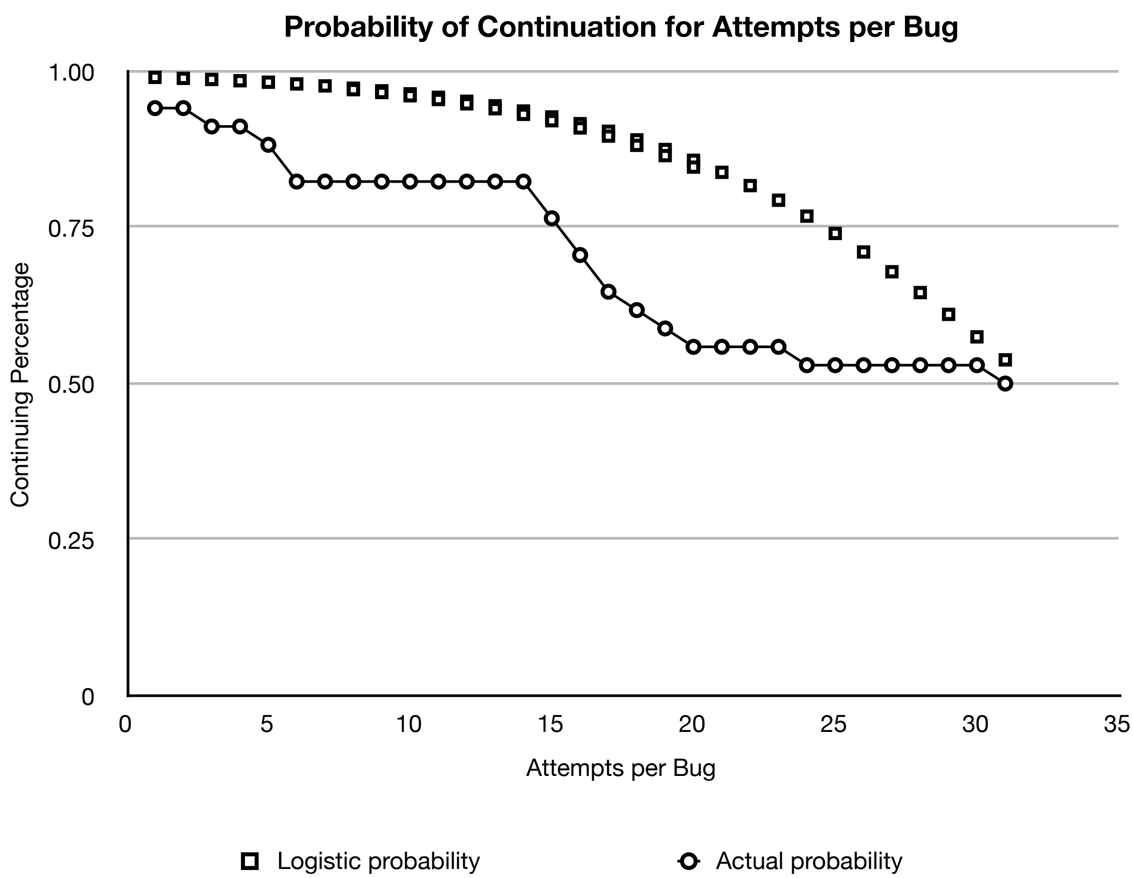

Figure 22: Predicted and actual probability of continuing with the problem for each attempt.

Link (d) represents the relationship between the wording condition and associated neutral face ratings. Theoretically, face ratings made in the positive condition should be higher (more positive) than face ratings made in the negative condition. This relationship, though, is influenced by the feedback loop from the number of solution attempts for each bug. To examine this relationship, a repeated measures ANOVA was conducted using the MIXED procedure in SAS 9.2, modeling actual face ratings to the wording condition and the number of solution attempts while blocking by subject. The results indicate that the number of solution attempts is the only factor approaching significance, as shown in Table 23.

Table 23: Repeated measures analysis for face ratings considering wording condition and attempts for each bug.

Repeated ANOVA for Face Ratings			
Effect	**DF**	**F Value**	**Pr > F**
Condition	1	0.11	0.741
Attempts per Bug	1	3.53	0.061 *
Condition*Attempts	1	0.02	0.875
*** Denotes nearing significance**			
**** Denotes significance at α = 0.05**			

Transitional Solution States

The m-MPS Model suggests that when a problem (bug) is solved, there will be an associated increase in positive affect, which should be reflected through an increase in neutral face rating measurements. This relationship was examined by taking the difference in face rating measurements from the rating immediately before solving the bug and the rating immediately after solving the bug. This new ΔFace Rating variable was used as the dependent variable in a repeated measures ANOVA conducted in SAS 9.2 with the MIXED procedure, using the wording condition, the specific bug, and the interaction between condition and bug as the predictor variables while blocking by subject. The results of this analysis are in Table 24 and the data is plotted in Figure 23. The fitted line in Figure 23 has an R^2 value of 0.0005, indicating the variance accounted for by the model is very low. Each point is labeled with the number of occurrences of the particular face rating at each transition.

Table 24: Repeated measures analysis results for transitional face ratings.

Repeated ANOVA for Transition Face Ratings			
Effect	**DF**	**F Value**	**Pr > F**
Condition	1	0.52	0.476
Bug	3	2.98	0.040 **
Condition*Bug	3	2.98	0.040 **
*** Denotes nearing significance**			
**** Denotes significance at α = 0.05**			

As discussed previously, the m-MPS Model predicts an increase in face ratings should lead to an increase in RAT scores. To test this prediction on transition states, a regression model was built using the ΔFace Rating as the predictor variable and the ΔRAT scores as the dependent variable. This analysis was conducted using the REG procedure in SAS 9.2. The regression suggests that ΔFace Rating has a significant linear relationship with the ΔRAT scores ($F(1,88) = 7.42$, $p = 0.008$), supporting the general prediction of the model. Figure 24 shows the relationship between the ΔFace Rating and

ΔRAT scores, with an R^2 value of 0.077 for the regression line. Each point is labeled with the number of occurrences of the particular face rating and RAT score combination.

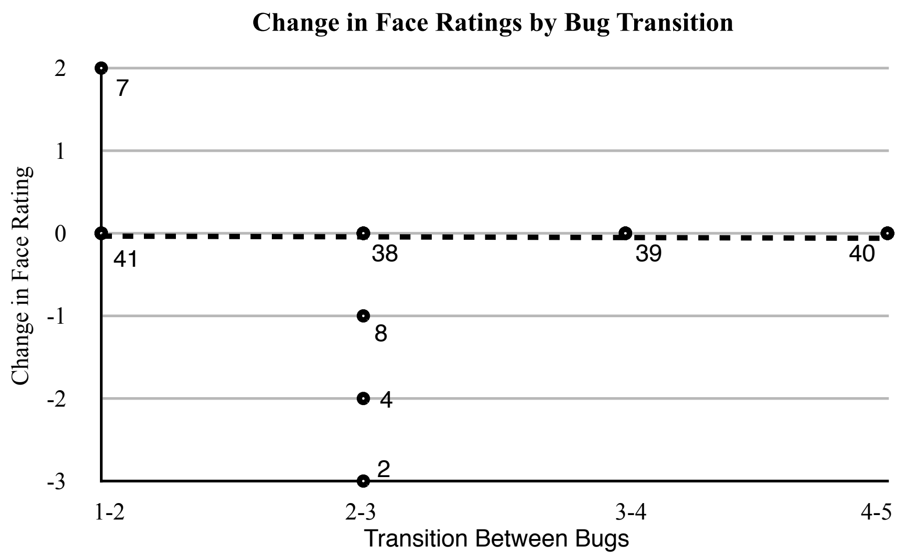

Figure 23: Transitional ΔFace Ratings by bug transition.

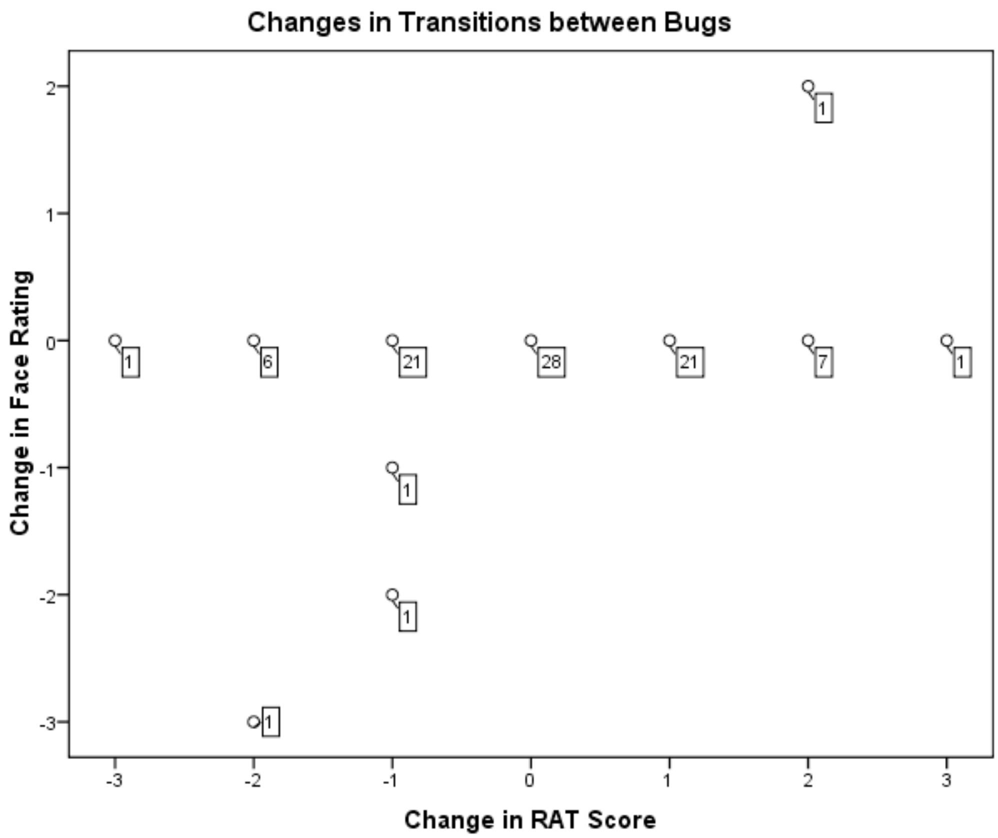

Figure 24: Changes in transitions between bugs for face ratings and RAT scores.

Summary of Key Findings

This experiment examined three hypotheses: that positive messages presented to problem solvers would increase positive affect throughout the problem solving task, that problem solvers receiving positive messages would perform the task better and have increased divergent thought, and that the predictions made by the m-MPS Model reflect actual behavior. The notable outcomes based on these hypotheses and predictions are summarized in this section.

Positive messages presented to problem solvers did not induce a measurable increase in positive affect. There was no difference between the positive and negative messages for either positive or negative affect from the beginning of the task to the end. However, problem solvers who received positive messages were significantly more likely to solve more bugs and solve the entire task than participants who received negative messages. There was no difference between the wording conditions for divergent thought scores. This result may be because of the relationship positive affect and divergent thought exhibit. Since there was no difference in positive affect between conditions, it is expected there should not be a difference in divergent thought.

The performance measures collected during this study indicate differences only in the number of bugs solved and task completion. Problem solvers receiving positive messages solved significantly more bugs and significantly completed the entire task more frequently. This finding suggests an effect exists based on the positivity or negativity of unrelated task messages. Table 25 summarizes the performance measure results recorded in this study.

Table 25: Performance measure results indicating difference between the conditions.

Performance Measures	Test	Test Statistic	p Value	Difference between conditions?
Total Task Time	ANOVA	F(3,30) = 1.96	p = 0.142	No
Compile Attempts	ANOVA	F(3,30) = 0.11	p = 0.954	No
Solution Cycles	ANOVA	F(3,30) = 1.69	p = 0.191	No
Number of Bugs Solved	Non-parametric t	X2 = 4.13	p = 0.042	**Yes**
Completed Task	Logistic	Wald X2 = 4.06	p = 0.044	**Yes**

The m-MPS Model makes several predictions about interactive problem solving. Some of these predictions were supported, based on the data collected in this study. The supported predictions include a positive relationship between face ratings (positive affect) and RAT scores (divergent thought), higher RAT scores are associated with fewer solution cycles, RAT scores follow the natural convergence of problem solving, and that each additional unsuccessful problem solving attempt reduces the probability of continuing the pursuit of a solution. Table 26 summarizes the results of these predictions. However, since positive affect was not successfully induced through this study, some of the predictions made by the model could not be adequately tested. Clearly, a different task and different measurement techniques could provide better data to test the model.

Table 26: Summary of m-MPS Model prediction results.

Prediction	Test	Test Statistic	p Value	Sig. Prediction?
Positive messages increase positive affect	ANOVA	$F(3,30) = 1.59$	$p = 0.211$	No
Positive messages increase average divergent thought	ANOVA	$F(3,30) = 0.58$	$p = 0.635$	No
Positive messages increase task performance (completed task)	Logistic	Wald X2 = 4.06	$p = 0.044$	**Yes**
Positive relationship between Face Ratings and RAT scores	Mixed ANOVA	$F(1,423) = 3.66$	$p = 0.056$	No
Higher RAT scores are associated with fewer solution cycles	Mixed ANOVA	$F(1,423) = 12.39$	$p < 0.001$	**Yes**
RAT scores reflect natural convergent problem solving behavior	Repeated Correlation	$Z = -3.17$	$p = 0.002$	**Yes**
Probability of continuing declines with each unsuccessful attempt	Repeated Logistic	$Z = -2.98$	$p = 0.003$	**Yes**
Face ratings are higher in positive condition	Repeated ANOVA	$F(1,32) = 0.11$	$p = 0.741$	No

CHAPTER V. DISCUSSION

Hypothesis 1

The first hypothesis of this research predicted that positively worded messages presented throughout a task on a computer interface would increase positive affect. This study collected measurements of affect at the beginning and end of the experiment, along with neutral face ratings throughout the task. Neutral face ratings have been shown to be a good (though coarse) indicator of affect at a particular moment in time (Stapel et al., 2002). There was not a significant difference in PANAS scores, face ratings, or the number of solution attempts between the wording conditions. However, there was a significant interaction between the wording condition and solution attempts for face ratings, particularly for Bug 4. This interaction could be the result of many participants in the negative condition solving the bug quickly, resulting in a boost in face rating.

The literature suggests a link between positive affect and divergent or creative thought (Isen, 2001; Isen et al. 1987; Forgas, 2002). The average Remote Associates Test (RAT) score, the measure of divergent thought, was not dependent on the wording condition. Since there was no difference in measured positive affect between the positive wording and negative wording conditions, it is reasonable to predict that divergent thought would not differ as well. The absence of a significant difference in RAT scores is consistent with the relationship between affect and divergent thought described in the literature.

Though the literature suggests humorous videos, small gifts, or subliminal smiling faces will increase positive affect (Isen et al., 1987; Berridge, 2003; Berridge and

Winkielman, 2003; Winkielman and Berridge, 2004; Stapel et al., 2002; Larsen et al., 2003), this study showed that positive affect was not increased when comparing positive to negative messages. There are several plausible explanations of this lack of effect. One explanation for this finding is that the problem solving task in this study, debugging a computer program, is not very pleasant to many people. The nature of the task itself could potentially overwhelm any effect the positive messages or the negative messages might have on the participants. Much of the cited research was focused on generating the effect, not measuring the effect as a result of the task. Alternatively, this finding may suggest the absence of any impact on affect from the positive or negative messages, indicated by the lack of difference between positive or negative affect. The final possibility is that the measurement tools used in this study were not sensitive enough to the changes exhibited by participants during the problem solving task. Regardless, it is clear that the first hypothesis is not supported in the context of this study. The pattern of evidence presented by all three dependent variables related to the relationship between wording, positive affect, and divergent thinking (the PANAS scale results, the face ratings, and the RAT scores) are consistent with the idea that the wording manipulation had either no effect on positive affect or an effect existed that was too small to measure.

Hypothesis 2

The second hypothesis in this research predicts higher problem solving performance for participants who received the positive messages. Participants in the positive wording condition solved significantly more bugs and a significantly higher number of participants completed the entire task. One possible explanation for this finding is that, given no difference in the general performance measures, participants in the positive condition used more efficient problem solving methods in their attempt to complete the task. Along that line of reasoning, participants in the positive wording condition may have been able to organize their problem solving strategy in a directed manner more frequently than the participants in the negative condition. This finding supports the second hypothesis in terms of problem solving performance.

Several other general performance measures were examined in this study, including total task time, number of compile attempts, and the number of solution cycles. The analyses conducted considered the effect wording condition, gender, and the interaction of condition and gender had on the specific performance measure (Table 25). The analysis for total task time indicates the only significant factor is gender, suggesting females were significantly faster to complete or abandon the task. The analysis for the number of compile attempts shows no significant differences between conditions or gender. The number of solution cycles analysis also contains no significant factors. These results offer support for the second hypothesis only in terms of overall performance which can be described as solving the presented problems. The evidence exhibited by the performance measures related to the wording condition suggest that positive messages allow problem solvers to solve more bugs than negative messages, while not providing any advantage in time, the number of solution attempts, or the number of solution cycles.

Hypothesis 3

The third hypothesis predicts the modified Multidimensional Problem Solving (m-MPS) Model reflects actual problem solving behavior. The m-MPS Model makes a series of predictions about how several of the measures previously discussed are related. These predictions include 1) increased positive affect will result in increased divergent thought, 2) extended efforts to solve a problem will result in lower RAT scores, 3) an unsuccessful solution attempt will lower the likelihood of another attempt, 4) positive messages will promote higher neutral face ratings while negative messages will promote lower neutral face ratings, 5) when a problem is solved successfully, there will be an associated increase in neutral face ratings and a resultant increase in RAT scores, and 6) increased RAT scores will result in fewer reused solutions (solution cycles). The evidence concerning each of these predictions is considered separately below.

The first prediction is that divergent thought will increase with an increase in positive affect. These two variables are represented by neutral face ratings and the RAT scores taken at particular moments throughout the problem solving task. The mixed model results do not confirm the predicted positive relationship between neutral face ratings and RAT scores ($F(1,423) = 3.66$, $p = 0.056$), though the relationship is approaching significance. There are several possible explanations for this result, the first of which is that in the context of this problem solving task, the relationship does not exist. This relation has primarily been studied in the context of puzzles (example: Isen et al., 1987) rather than slightly unpleasant, “work” related tasks. However, it is conceivable that the measurement tools used in this study were not sensitive enough to record the changes in affect or divergent thought that may have occurred, especially considering the relationship is approaching significance. The evidence gathered in this study does not support a positive relationship between positive affect and divergent thought, but the evidence is approaching significance suggesting more research is needed in this area.

The second prediction the m-MPS Model makes is that in extended efforts to solve a particular problem, RAT scores will lower, following the natural convergence in the problem solving process. This prediction is based on the problem solving literature, which suggests that as potential solutions are ruled out, the problem solver narrows the solution space, converging on the solution (Brophy, 1998; Vosburg, 1998; Clapham, 2000; Norman, 2004). The correlation results indicate that a negative relationship does exist between RAT scores and the number of attempts to solve a particular bug (rho = -0.186, $p < 0.001$), indicating that RAT scores trend down as the number of attempts rises. This suggests that the participants narrowed their thinking patterns as they iteratively approached a solution, as predicted by the m-MPS Model. On the whole, the evidence supports the second prediction.

The third prediction of the m-MPS Model is that the probability that a problem solver will succeed on the next attempt decreases with the number of previously unsuccessful problem solving attempts. This prediction is based on the previous prediction, in which problem solving is naturally convergent along with the assumption that a problem solver will only generate a finite number of solutions before becoming frustrated to an unrecoverable level. The logistic regression results indicate that each additional attempt lowers the probability that the problem solver will continue in the event of an unsuccessful attempt (Wald $X^2 = 15.07$, $p < 0.001$). Thus, the experiment supports the claim that the more unsuccessful attempts a participant made to solve a particular bug, the less likely they were to try again.

The fourth prediction is that neutral face ratings are impacted by wording condition. Specifically, participants in the positive condition should have higher neutral face ratings than participants in the negative condition. However, over the course of the task, the neutral face ratings will be influenced by unsuccessful or successful solution attempts. The repeated ANOVA results from this experiment indicate that this prediction does not hold. There is no particular effect of the wording condition on face ratings through the task ($p > 0.05$, Table 23), though the effect of the number of attempts for each bug on face ratings is approaching significance ($p = 0.061$, Table 23). This result is consistent with the previous findings of this study, indicating that either message wording has no impact upon affect as measured by the neutral face ratings, or the neutral face rating measurement is not sensitive to the changes potentially exhibited during the task.

The fifth prediction contains two parts. The first is that when a problem is solved successfully, the model predicts an increase in positive affect as indicated by the neutral face ratings. The results suggest a significant effect for the interaction between wording condition and bug ($p = 0.04$, Table 24), however the vast majority of face rating changes were 0, meaning the face ratings did not change from one bug to the next. One possible explanation is that in the actual task, the problem solver was unaware the problem was successfully solved before the measurement was taken. The measurement tool was designed to gain insights into the problem solving task before the participants were exposed to the messages. Since the messages were designed to impact affect, taking the measurements before the messages were presented was logical. However, the messages were the only indication that the particular bug had been solved. Regardless, the evidence clearly indicates only a few data points make the relationship significant, which cannot be used to support the prediction.

The second part of the fifth prediction is that neutral face ratings and RAT scores will increase together when a problem solver transitions from one problem to the next. The regression of transitional states suggest a significant relationship between the change in face ratings and the change in RAT score ($F(1,88) = 7.42$, $p < 0.01$). However, the same issue exists in the collected data that the previous prediction encountered: the majority of transitional face rating differences are zero. The evidence suggests only a few data points make this relationship significant, indicating this prediction is not supported by the data.

The sixth prediction is that higher RAT scores will result in lower numbers of solution cycles (reused solutions). The regression analysis indicates a significant negative relationship between RAT scores and solution cycles ($F(1,423) = 12.39$, $p < 0.001$). Higher RAT scores imply more divergent thought, suggesting a larger pool or variety of potential solutions would be available to the problem solver. On the same token, lower RAT scores will result in higher numbers of solution cycles since the number of potential solutions would be increasingly limited. This evidence supports the sixth prediction.

The evidence presented by this experimental investigation suggests an effect exists between interface messages and problem solving performance, though the proposed theoretical mechanism linking the two factors may or may not be accurate. As previously discussed, the theoretical mechanism is based upon years of previous research from a wide variety of domains, including human-computer interaction, neuroscience, psychology, and human factors. Each of these domains provide evidence for elements in the theory, though some elements were not supported in this investigation. The evidence does not refute any of the relationships described in previous research, but may indicate one of two situations. The first is that the evidence presented in this study suggests the relationships may not exist in the narrow context of the programming task, while the second suggests a lack of evidence to support the relationships described in previous research. Though the proposed theoretical mechanism is not supported through this study, the link between interface messages and problem solving performance is supported, suggesting there may be alternative explanations of the effect.

The difference in problem solving performance between positive messages and negative messages may be the result of negative messages accelerating feelings of frustration through the problem solving task. This alternative explanation is supported by a simple frustration mitigation that Lazar et al. (2006) suggest: choosing better words for interface and error messages will reduce frustration. Reducing frustration may not actually aid the problem solving process, but instead allow the problem solver more time or attempts to solve a problem before giving up. A second alternative explanation is that the differences between the messages could amount to the difference between positive and negative reinforcement. Though the presented positive or negative messages were not task related, their influence on task performance was evident, indicating an effect on problem solving activities.

Taking all the results together, Figure 25 highlights the predictions made by the m-MPS Model through the third hypothesis and indicates whether the predictions were supported or not. The analysis indicates that the first hypothesis is not supported. However, the performance aspect of the second hypothesis is supported, along with several predictions of the underlying theory of the m-MPS Model included in the third hypothesis. Considering these results, a new theoretical flow model was created that only represents the predictions that were clearly supported by the data collected in this experiment. This new model is presented in the following section.

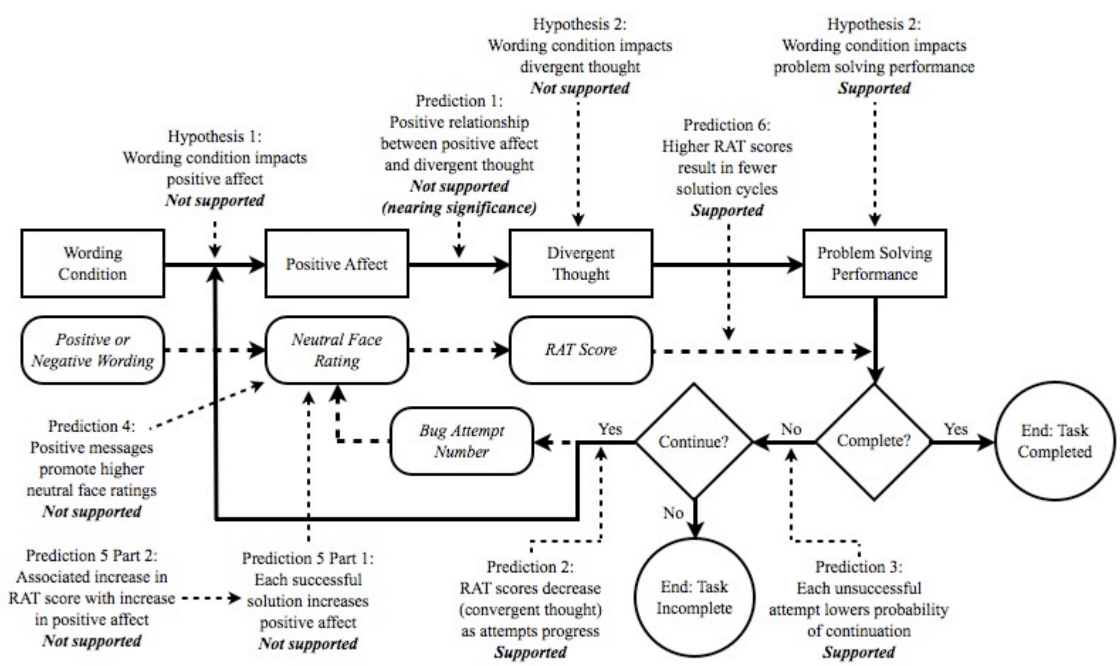

Figure 25: Theoretical prediction accuracies made by the m-MPS Model.

Updated Theoretical Model

The new flow model, shown in Figure 26, indicates the shared relationship between positive messages and task performance based on the results of this experimental programming task. This model is based on the model shown in Figure 25, though it has been updated as a conventional flow chart to reflect the findings of this study. The flow chart begins with a new problem for which the problem solver attempts to find a solution. The problem solver generates a solution and, in this case, tests the solution by compiling the solution. The problem solver is then presented a message about the solution that can be worded positively or negatively. The data collected in this study suggest that problem solvers who receive the positive messages are significantly more likely to solve the entire task (Table 25). The odds ratio from the logistic regression indicates that problem solvers receiving negative messages will be 2.098 times more likely to fail to complete the task.

If the problem solver fails in an attempt to solve the problem, the data indicates the probability of continuing to try to solve the problem will decrease, irrespective of the type of messages being received (Table 26). The odds ratio indicates that each additional unsuccessful attempt will increase the likelihood, by 1.16 times, of abandoning the task. Additionally, the Remote Associates Test scores indicate that problem solvers will exhibit convergent thought through the problem solving process (Table 26). Convergent thought will lead to an increased likelihood of attempting to reuse an unsuccessful solution (Table 26). The evidence based modifications of this new flow model suggest that a new model should be created to more accurately reflect these findings.

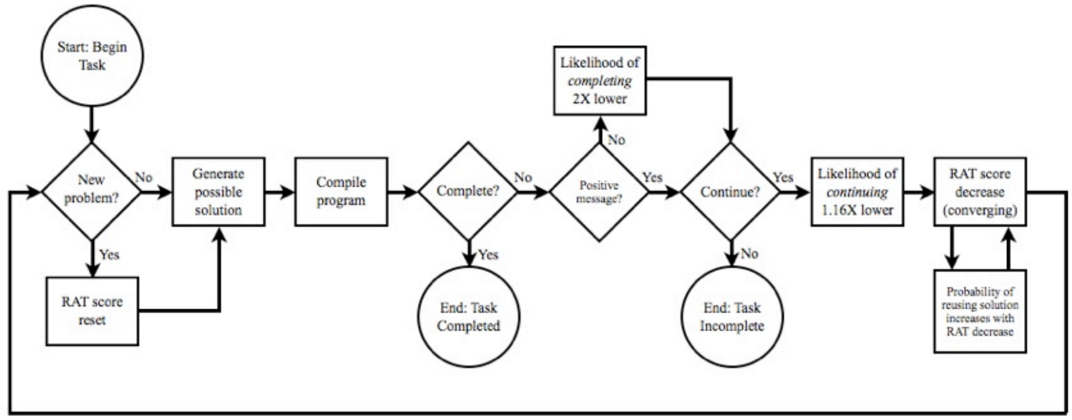

Figure 26: The new underlying theoretical flow chart based on the supporting data.

The new theoretical flow chart has several limitations which undermine its usefulness and applicability. The flow chart is based on a programming task and is limited in its generalizability. It is possible to generalize this flow chart to problem solving situations that contain text-based or verbal feedback. However, one of the main objectives of this research was to show how positive (as compared to negative) messages affect problem solving performance in order to begin to close this research gap in the human factors domain. As a descriptive model, this theoretical flow chart takes a small step in the appropriate direction, though the structure is limited to the specific programming domain. A generalizable model should also be able to be applied to a variety of problem solving domains. Though the flow chart style model is undoubtably useful in describing a process like problem solving, a model that combines process flow and various visualizations is useful to visualize both the theory and the process in an evidence-based manner.

The limitations the theoretical flow chart exhibits are reduced or resolved in a new model called the Cyclical Problem Solving (CPS) Model, shown in Figure 27. This new model is based on the theoretical roadmap model (shown in Figure 7) which described the predictions of the m-MPS Model and the original hypotheses of this dissertation. The CPS Model combines the experimental results with the underlying theory shown in Figure 26, and the iterative nature of problem solving as described in the literature. It represents the basic relationships shown by the experimental data.

The CPS Model assumes the problem solver is the supervisor of a system in which a problem occurs. In the case of this programming experiment, the problem solver can be classified as the supervisor of the compiler system. When a problem occurs in the system, the model assumes a warning or message is provided to the supervisor regarding the problem which is worded in a positive or negative way. The problem in the programming task is a compiler error, indicated by a compiler error message and accompanied by a non-task related positive or negative message. The model indicates that problem solvers receiving negative messages are two times more likely to fail to find a solution than problem solvers who receive positive messages. This model prediction is based on the evidence presented by the experimental results which indicate problem solvers receiving positive messages were significantly more likely to solve the entire task (Table 25). The odds ratio specifies that problem solvers receiving negative messages are 2.09 times more likely to fail solving the problem.

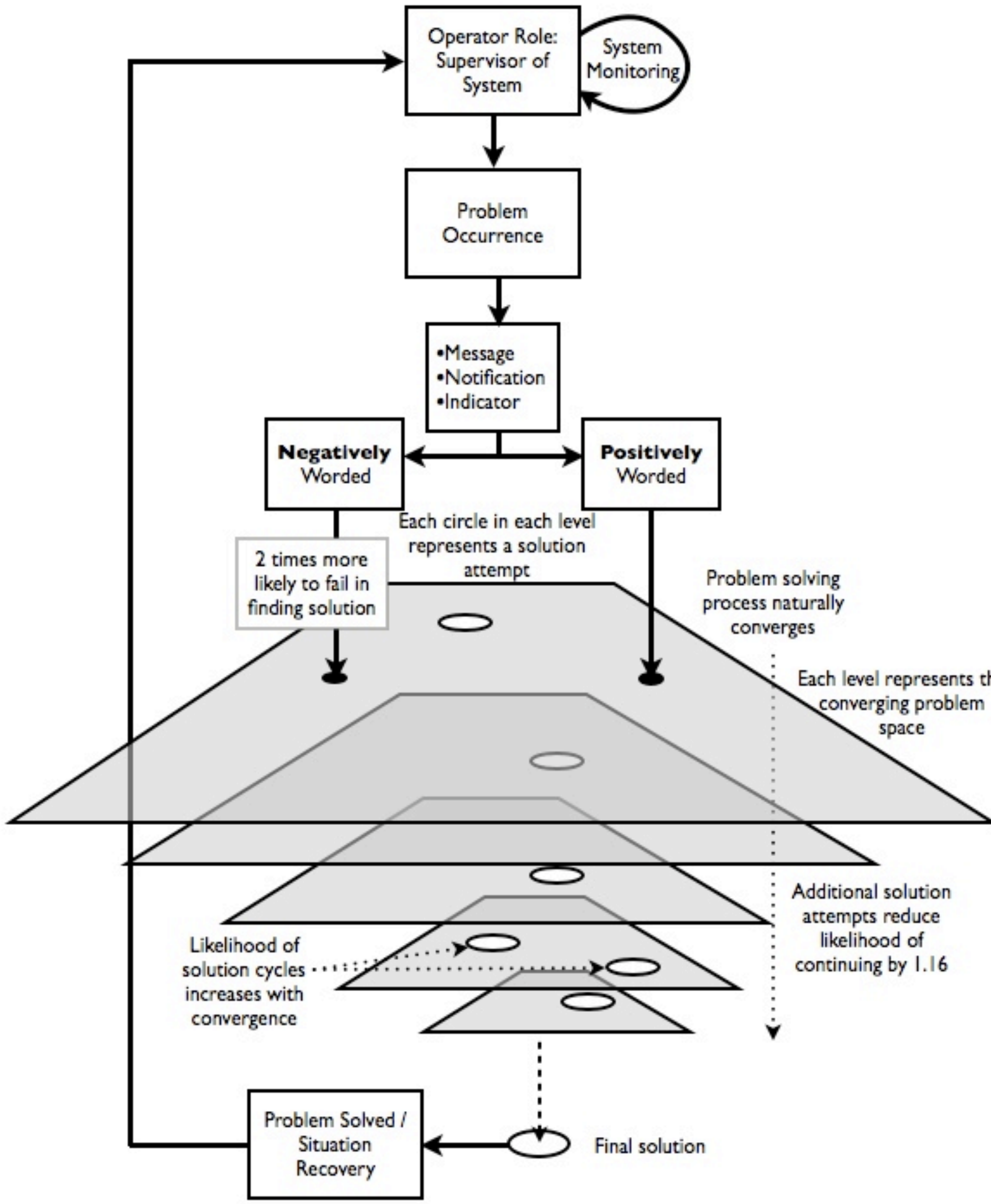

Figure 27: The CPS Model based on experimental evidence.

The problem solver then enters into a problem solving strategy within the problem space and will begin to converge on a solution, which has been described as narrowing the problem space (Brophy, 1998; Vosburg, 1998; Clapham, 2000; Norman, 2004). The converging problem solving process is represented by the shrinking problem space levels, which is similar to the “swiss cheese” metaphor model of accidents (Reason, 1990). The decreasing level visualization is useful to represent the converging problem solving process as narrowing the problem space and targeting a solution. Each solution attempt, represented in the model by small empty circles, will be met with a positive or negative message. Each additional unsuccessful solution attempt will reduce the likelihood of continuing to solve the problem by 1.16 times. This prediction is based on the odds ratio from the experimental results which examined the relationship between the number of unsuccessful attempts and the decision to try again or quit (Table 26).

Following the convergent problem solving process, each solution attempt will narrow the problem space, converging on an ultimate solution. As the problem solver converges on the solution, the likelihood of reusing a previous solution increases ($p < 0.01$, Table 24). Reusing a solution could be attributed to the limited number of possible solutions available within the narrowed problem space. In the model, this possible event is represented by two solution symbols on a particular solution level. If the problem solver converges the problem space enough that the actual solution falls outside the boundaries of the space, the problem solver will most likely not be able to solve the problem. Eventually, the problem solver will solve the problem and return to monitoring the system, or the problem solver will abandon the problem and face the associated consequences.

The flow chart shown in Figure 26 can be essentially embedded in the CPS Model as an in-depth description of the problem solving process within the computer programming domain. However, generalizing the flow chart can be accomplished by replacing the "Compile program" process block with an "Apply solution" block. In this way, the flow chart can provide a structured process to the visualization of the converging problem solving process represented in the CPS Model.

The CPS Model contains several limitations, the first of which is the assumption of either positive or negative messages. Messages in interfaces can conceivably range from very positive to very negative, resulting in a continuous scale of wording. This scale is not captured in the current model representation. The range of messages will need to be examined in future work in order to accurately represent a message scale in the model. A second limitation is the representation of the solution within the problem space. The decreasing levels representation of the convergent problem solving process is novel and can be easily understood, though it fails to satisfactorily capture the dynamic behaviors of problem solvers within the problem space. The current representation provides a static understanding of a hypothetical problem solving state, but the problem solving process is more closely aligned with a stochastic system.

Several advantages exist in representing the problem solving process within the combination of the flow chart (Figure 26) and the CPS Model. The first advantage is that the structure of the CPS Model and the flow chart allows a representation that clearly shows the path of the problem solving process. The path the problem solver takes to gain the solution can be traced from the onset of a problem to the return to normal state. The second advantage is that objective, evidence based predictions can be clearly shown within each of these representations. Including these objective predictions is important to clearly show the difference in wording conditions for the problem solving process. The third advantage is that the CPS Model represents the converging problem solving process in a way that has not been demonstrated in the literature. Though the literature has described the converging problem solving process as a narrowing of the problem space (Brophy, 1998; Vosburg, 1998; Clapham, 2000; Norman, 2004), this process has not been represented in an objective, evidence-based visualization. Additionally, reusing solutions in a problem solving process is a new and innovative way to identify convergence within the problem space. The final advantage presented by the CPS Model is that even though the model originated from theoretical relationships, the model can easily be understood by designers who develop systems in which problems need to be solved.

CHAPTER VI. CONCLUSIONS

Human factors engineers seek to design systems that support human performance. It is surprising, then, that the human factors community has largely left affect alone as a means of supporting human performance. In supervisory control systems, supporting human performance is very important because of the unique situations in which human operators may find themselves. These situations may require a certain amount of creative problem solving or, at the very least, a certain measure of divergent thought to solve a problem that has not been seen or solved before. Since increasing positive affect has been shown to support creative problem solving and divergent thought, it is logical to use positive affect as an additional design aspect to support human operators in a supervisory control situation.

Inducing positive affect has been successfully achieved by showing humorous films or giving candy to people. These are not practical solutions in a supervisory control system, such as controlling a nuclear power plant or responding to a warning light for an unmanned aerial system. However, many supervisory control systems use text-based messages to relay important information about current system states, which provides an opportunity to induce or promote positive affect without providing the operator a five minute break to watch a comedy film. Specific positive words have been shown to promote an increase in positive affect. Including these positive words in system messages have the potential to increase positive affect in a way that does not intrude upon the actual task at hand. This is the central idea of the experiment presented in this dissertation.

To describe this relationship in terms of affect, divergent thought, and problem solving performance, the modified Multidimensional Problem Solving (m-MPS) Model was created. This model makes a series of predictions about the interactions between affect and problem solving performance which ultimately would support the human operator in a problem solving situation. The central idea to this model, in practical terms, is that specific positive words in task messages will promote an increase in positive affect, resulting in increased problem solving performance. This central idea is also the main gap in the human factors research.

This dissertation research examined two specific aims. The first aim was to show that positively worded messages promote positive affect throughout a problem solving task. The second aim was to show that positively worded messages will have a measurable positive impact on problem solving performance. Each of these aims translates into the central hypotheses that this dissertation tested: 1) positively worded messages promote positive affect and 2) positively worded messages have a measurable positive impact on problem solving performance. Each of these hypotheses were examined against negatively worded messages.

The first hypothesis and associated aim was not supported in this study. Positive affect was not significantly different from the beginning to the end of the task for participants in either the positive wording or negative wording condition. However, the problem solving task used in this study (computer program debugging) is commonly regarded as a generally unpleasant task. Unfortunately, the unpleasantness of the task itself could have negated any effect the positive messages may have exhibited over the negative messages. Though the hypothesis was not supported in the context of this study, it remains unclear whether the task confounded the measurements or the messages had no effect. Future research should reexamine this hypothesis using a less unpleasant problem solving task.

The second hypothesis and associated aim was supported in terms of problem solving performance. Participants receiving positive messages solved significantly more bugs in the debugging task and significantly more participants receiving positive messages finished the entire task. This finding suggests that positive messages are more useful for problem solving performance than negative messages. The second hypothesis is not supported in terms of divergent thought. As previously mentioned, since positive affect was not different between the two conditions, it is not expected that divergent thought would be different between conditions, based on the underlying mechanism as described by the m-MPS Model.

The m-MPS Model provides a mechanism for attempting to understand the underlying theory of the interaction between affect and problem solving. The model predictions were the focus of the third experimental hypothesis. The specific predicted relationships shown in this experiment include a positive relationship between positive affect and divergent thought, increased divergent thought is associated with fewer solution cycles, and each additional unsuccessful problem solving attempt reduces the probability of continuing the pursuit of a solution. Since positive affect was not successfully induced through this study, some of the predictions made by the model could not be adequately tested. Therefore, the new CPS Model was developed based on the outcome of the experimental study that highlights the effect positive messages have on performance over negative messages.

The current experiment contained several limitations. The first is that the programming task is slightly unpleasant to many people which could have influenced the measurements taken through the experiment and perhaps overwhelm any effect the messages may have had on the problem solvers. The second limitation is the methods by which affect and divergent thought were measured throughout the task were intrusive upon the task. Participants were required to interrupt their problem solving activities to conduct a seemingly unrelated task which may have influenced their affective state. The third limitation is that the actual measurement tools may not have been satisfactorily sensitive to changes in affect or divergent thought. These tools may have limited the ability to measure the proposed theoretical mechanism between interface messages and problem solving performance.

Understanding the underlying theory and relations that positive affect and problem solving performance share is an important next step to design operator support systems. The model and the theoretical background presented in this dissertation can serve as a broad roadmap for discovering new methods of interaction that support both positive affect and problem solving performance. These connections could be explored in many areas of human-computer interaction. Future research in this area should focus on refining the measurements for affect and divergent thought used in this study. Extending from this experiment directly, including a baseline condition with neutrally worded messages could provide information about specific benefits or detractions provided by positive and negative messages. Identifying non-intrusive, task related interactions that would promote positive affect in a measurable way would provide a means to an observable result and the ability to accurately verify and extend the new model.

Current human-computer interaction research is focused on using technology to detect or infer individuals' emotional state, particularly frustration, to modify the interaction parameters in an effort to reduce that frustration. Though this research is moving to acknowledge the influence affect has on people's cognitive processes, the solutions are driven by sensing technologies that react to particular states, rather than proactively changing designs. This research builds upon research in other domains by suggesting that it is important to understand how performance can be influenced by emotion, affect, or simple interface messages. This research showed specifically that message wording has an impact on problem solving performance in the programming domain. Understanding this effect can lead to better designs which accommodate and capitalize on people's affect, ultimately leading to better performance from the outset, rather than relying on technology to reliably detect and react to changes in affect.

REFERENCES

Aboulafia, A. and Bannon, L. (2004). Understanding Affect in Design: An Outline Conceptual Framework. Theoretical Issues in Ergonomics Science, 5(1), pp. 4-15.

Ashby, F.G., Isen, A., and Turken, A. (1999). A Neuropsychological Theory of Positive Affect and Its Influence on Cognition. Psychological Review, 106(3), pp. 529-550.

Benedek, J. and Hazlett, R. (2005). Incorporating Facial EMG Emotion Measures as Feedback in the Software Design Process. Proceedings of the 2005 Human Computer Interaction Consortium, Colorado.

Berridge, K. (2003). Pleasures of the Brain. Brain and Cognition, 52, pp. 106-128.

Berridge, K. and Winkielman, P. (2003). What Is an Unconscious Emotion? (The Case for Unconscious "Liking"). Cognition and Emotion, 17(2), pp. 181-211.

Bowden, E. and Jung-Beeman, M. (2003). Normative Data for 144 Compound Remote Associate Problems. Behavior Research Methods, Instruments, & Computers, 35(4), pp. 634-639.

Bradley, M.M., and Lang, P.J. (1999). Affective norms for English words (ANEW): Stimuli, instruction manual and affective ratings. Technical report C-1, Gainesville, FL. The Center for Research in Psychophysiology, University of Florida.

Brophy, D. (1998). Understanding, Measuring, and Enhancing Individual Creative Problem-Solving Efforts. Creativity Research Journal, 11(2), pp. 123-150.

Carlson, M. and Bloom, I. (2005). The Cyclic Nature of Problem Solving: An Emergent Multidimensional Problem-Solving Framework. Educational Studies in Mathematics, 58, pp. 45-75.

Ceaparu, I., Lazar, J., Bessiere, K., Robinson, J., and Shneiderman, B. (2004). Determining Causes and Severity of End-User Frustration. International Journal of Human-Computer Interaction, 17(3), pp. 333-356.

Clapham, M. (2000). The Effects of Affect Manipulation and Information Exposure on Divergent Thinking. Creativity Research Journal, 13(3&4), pp. 335-350.

Collins, A. and Loftus, E. (1975). A Spreading-Activation Theory of Semantic Processing. Psychological Review, 82(6) pp. 407-428.

Croson, R. and Donohue, K. (2006). Behavioral Causes of the Bullwhip Effect and the Observed Value of Inventory Information. Management Science, 52, pp. 323-336.

Dalgleish, T. (2004). The Emotional Brain. Nature Reviews: Neuroscience, 5, pp. 582-589.

Diener, E. and Emmons, R. (1985). The Independence of Positive and Negative Affect. Journal of Personality and Social Psychology, 47(5), pp. 1105-1117.

Dijksterhuis, A. and Smith, P. (2002). Affective Habituation: Subliminal Exposure to Extreme Stimuli Decreases Their Extremity. Emotion, 2(3), pp. 203-214.

Fossati, P., Hevenor, S., Graham, S., Grady, C., Keightley, M., Craik, F., and Mayberg, H. (2003). In Search of the Emotional Self: An fMRI Study Using Positive and Negative Emotional Words. American Journal of Psychiatry, 160(11), pp. 1938-1945.

Fredrickson, B. (2004). The Broaden-and-Build Theory of Positive Emotions. Philosophical Transactions of the Royal Society of London - B, 359, pp. 1367-1377.

Forgas, J. (1994). The Role of Emotion in Social Judgments: An Introductory Review and an Affect Infusion Model (AIM). European Journal of Social Psychology, 24, pp. 1-24.

Forgas, J. (1995). Mood and Judgement: The Affect Infusion Model (AIM). Psychological Bulletin, 117(1), pp. 39-66.

Forgas, J. (2002). Feeling and Doing: Affective Influences on Interpersonal Behavior. Psychological Inquiry, 13(1), pp. 1-28.

Hancock, P., Pepe, A., and Murphy, L. (2005). Hedonomics: The Power of Positive and Pleasurable Ergonomics. Ergonomics in Design, 13(1), pp. 8-14.

Hazlett, R. and Benedek, J. (2007). Measuring Emotional Valence to Understand the User's Experience of Software. International Journal of Human-Computer Studies, 65, pp. 306-314.

Helander, M.G. and Tham, M.P. (2003). Hedonomics - Affective Human Factors Design. Ergonomics, 46(13/14), pp. 1296-1272.

Hudlicka, E. (2003). To Feel or Not to Feel: The Role of Affect in Human-Computer Interaction. International Journal of Human-Computer Studies, 59, pp. 1-32.

Isen, A. (2001). An Influence of Positive Affect on Decision Making in Complex Situations: Theoretical Issues With Practical Implications. Journal of Consumer Psychology, 11(2), pp.75-85.

Isen, A., Daubman, K., and Nowicki, G. (1987). Positive Affect Facilitates Creative Problem Solving. Journal of Personality and Social Psychology, 52(6), pp. 1122-1131.

Isen, A. and Daubman, K. (1984). The Influence of Affect on Categorization. Journal of Personality and Social Psychology, 47(6), pp. 1206-1217.

John, B.E. and Marks, S.J. (1997). Tracking the Effectiveness of Usability Evaluation Methods. Behaviour & Information Technology, 16(4-5), pp. 188-202.

Kurosu, M. and Kashimura, K. (1995). Apparent Usability vs. Inherent Usability. CHI '95 Conference Companion, pp. 292-293.

Larsen, J., Norris, K., and Cacioppo, J. (2003). Effects of Positive and Negative Affect of Electromyographic Activity Over Zygomaticus Major and Corrugator Supercilii. Psychophysiology, 40, pp. 776-785.

Lazar, J., Jones, A., and Shneiderman, B. (2006). Workplace User Frustration with Computers: An Exploratory Investigation of the Causes and Severity. Behaviour & Information Technology, 25(3), pp. 239-251.

Lee, J.D. (2007). Affect, Attention, and Automation. Attention - From Theory to Practice, Kramer, A., Wiegmann, D., and Kirlik, A. (eds.), Oxford University Press, New York.

Lewis, P.A., Critchley, H.D., Rotshtein, P., and Dolan, R.J. (2007). Neural Correlates of Processing Valence and Arousal in Affective Words. Cerebral Cortex, 17, pp. 742-748.

Liu, Y. (2003). Engineering aesthetics and aesthetic ergonomics: Theoretical foundations and a dual-process research methodology. Ergonomics, 46(13), pp. 1273-1292.

Maddock, R., Garrett, A., and Buonocore, M. (2003). Posterior Cingulate Cortex Activation by Emotional Words: fMRI Evidence From a Valence Decision Task. Human Brain Mapping, 18, pp. 30-41.

Mandryk, R. and Atkins, M.S. (2007). A Fuzzy Physiological Approach for Continuously Modeling Emotion During Interaction with Play Technologies. International Journal of Human-Computer Studies, 65, pp. 329-347.

Mandryk, R., Inkpen, K., and Calvert, T. (2006). Using Psychophysiological Techniques to Measure User Experience With Entertainment Technologies. Behaviour & Information Technology, 25(2), pp. 141-158.

Mandryk, R., Atkins, M.S., and Inkpen, K. (2006). A Continuous and Objective Evaluation of Emotional Experience with Interactive Play Environments. Proceedings of the 2006 Computer Human Interaction Meeting.

Martinez, A.M. and Benavente, R. (1998). The AR Face Database. CVC Technical Report #24.

McAdams, D. and Constantian, C. (1983). Intimacy and Affiliation Motives in Daily Living: An Experience Sampling Analysis. Journal of Personality and Social Psychology, 45(4), pp. 851-861.

McNeese, M. (2003). New Visions of Human-Computer Interaction: Making Affect Compute. International Journal of Human-Computer Studies, 59, pp. 33-53.

Mednick, S. (1962). The Associative Basis of the Creative Process. Psychological Review, 69(3), pp. 220-232.

Murphy, L., Stanney, K., and Hancock, P. (2003). The Effect of Affect: The Hedonomic Evaluation of Human-Computer Interaction. Proceedings of the 47th Annual Meeting of the Human Factors and Ergonomics Society, pp. 764-768.

Norman, D. (2004). *Emotional Design: Why We Love (or Hate) Everyday Things*. New York: Basic Books.

Picard, R. (1997). *Affective Computing*. Cambridge, MA: The MIT Press.

Picard, R. and Daily, S. (2005). Evaluating Affective Interactions: Alternatives to Asking What Users Feel. CHI Workshop on Evaluating Affective Interfaces: Innovative Approaches.

Rasmussen, J. (1983). Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models. IEEE Trans. on Systems, Man, and Cybernetics, 13(3), pp. 257-266.

Rasmussen, J. (1993). Diagnostic Reasoning in Action. IEEE Trans. on Systems, Man, and Cybernetics, 23(4), pp. 981-992.

Reason, J. (1990). *Human Error*. New York: Cambridge University Press.

Schemenaur, P.J. and Pawlick, C. (2007). Evaluating Guidelines for Writing User Interface Text. Proceedings of the Special Interest Group on Design of Communication, 2007, pp. 216-220.

Sheridan, T. (1992). Telerobotics, Automation, and Human Supervisory Control. The MIT Press, Cambridge, MA.

Stapel, D., Koomen, W., and Ruys, K. (2002). The Effects of Diffuse and Distinct Affect. Journal of Personality and Social Psychology, 83(1), pp. 60-74.

Stone, A., Hedges, S., Neale, J., and Satin, M. (1985). Prospective and Cross-Sectional Mood Reports Offer No Evidence of a "Blue Monday" Phenomenon. Journal of Personality and Social Psychology, 49(1), pp. 129-134.

Tellegen, A., Watson, D., and Clark, L.A. (1999). On the Dimensional and Hierarchical Structure of Affect. Psychological Science, 10(4), pp. 297-303.

Tractinsky, N. (1997). Aesthetics and Apparent Usability: Empirically Assessing Cultural and Methodological Issues. CHI ’97, pp. 22-27.

Tractinsky, N., Katz, A., and Ikar, D. (2000). What is Beautiful is Usable. Interacting with Computers, 13, pp. 127-145.

Vosburg, S. (1998). The Effects of Positive and Negative Mood on Divergent-Thinking Performance. Creativity Research Journal, 11(2), pp.165-172.

Warr, P., Barter, J., and Brownbridge, G. (1983). On the Independence of Positive and Negative Affect. Journal of Personality and Social Psychology, 44(3), pp. 644-651.

Watson, D. and Clark, L.A. (1997). Measurement and Mismeasurement of Mood: Recurrent and Emergent Issues. Journal of Personality Assessment, 68(2), pp. 267-296.

Watson, D., Clark, L.A., and Tellegen, A. (1988). Development and Validation of Brief Measures of Positive and Negative Affect: The PANAS Scales. Journal of Personality and Social Psychology, 54(6), pp. 1063-1070.

Watson, D., Wiese, D., Vaidya, J., and Tellegen, A. (1999). The Two General Activation Systems of Affect: Structural Findings, Evolutionary Considerations, and Psychobiological Evidence. Journal of Personality and Social Psychology, 76(5), pp. 820-838.

Winkielman, P. and Berridge, K. (2004). Unconscious Emotion. Current Directions in Psychological Science, 13(3), pp. 120-123.

APPENDIX A: EXPERIMENTAL PROGRAM FOR PILOT STUDY

Experimental program for the Pilot Study. The error lines are marked in bold text with the direct solution commented on the same line.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <ctype.h>
#include <string.h>

/*******************************************************************
This program imports a data file called "data.txt" which contains
numbers representing muscle activations. The data file contains 80 rows of 80
digits each.  Function clean() finds characters in the data and replaces them
with zero (0), returning the number of characters replaced. Function noise()
calculates the degree of data error (number of zeros/number of total sequences),
returning a percentage.  Function count() totals the number of each sequence
located in the file, returning a count for each sequence (1-9).
*******************************************************************/

#define SIZE 80 // #define SIZE 81

void clean(char a[SIZE][SIZE]);
void noise(char a[SIZE][SIZE], double *);
void count(char a[SIZE][SIZE], int []);


int main() {

   FILE *fPtr;
   char data[SIZE][SIZE];
   char buffer[SIZE];
   int counts[9];
   int i,j;
   double err;

   if((fPtr = fopen("data.txt", "r"))==NULL) {
      printf("Could not open data.txt.");
   }
   else {
      printf("Opened data.txt\n");
      j=0;
      while(!feof(fPtr)){
         fscanf(fPtr, "%s", buffer);
         sprintf(data, "%s", buffer);    // sprintf(data[row], "%s", buffer);
         j++;
      }

      fclose(fPtr);

      clean(data);
      noise(data,err);      // noise(data, &err);
```

```
49         count(data,counts);     // count(data,counts[]);
50
51         printf("Error rate: %.2f %\n", (err*100.0));
52         for(i=0;i<9;i++){
53             printf("Number of %d sequences: %d\n", (i+1), counts[i]);
54         }
55
56     }
57     return 0;
58 }
59
60 /* Function clean() */
61 void clean(char a[][SIZE]) {
62     int i, j;
63
64     // Loop through data array to clean up random characters
65     // and replace them with 0.
66     for(i=0;i<SIZE;i++){
67         for(i=0;i<SIZE;i++) {
68             if((isalpha(a[i][j]))!=0) {
69                 a[i][j] = '0';
70             }
71         }
72     }
73
74 }
75
76 /* Function noise() */
77 void noise(char a[][SIZE], double *err) {
78     int i, j;
79     double countnum = 0.0, counterr = 0.0;
80
81     for(i=0;i<SIZE;i++){
82         for(j=0;j<SIZE;j++){
83             if(a[i][j]==0){
84                 counterr++;
85             }
86             countnum++;
87         }
88     *err = (counterr/countnum);       // err = (counterr/countnum);
89 }
90
91 /* Function count() */
92 void count(char a[][SIZE], int counts[9]) {
93    int i, j;
94
95     for(i=0;i<SIZE;i++){
96         for(j=0;j<SIZE;j++){
97             switch(a[i][j]){
98                 case '1':
99                     counts[1] += 1;      // counts[0] += 1;  This error continues in lines 102-122.
100                     break;
101                 case '2':
102                     counts[2] += 1;
```

```
                break;
            case '3':
                counts[3] += 1;
                break;
            case '4':
                counts[4] += 1;
                break;
            case '5':
                counts[5] += 1;
                break;
            case '6':
                counts[6] += 1;
                break;
            case '7':
                counts[7] += 1;
                break;
            case '8':
                counts[8] += 1;
                break;
            case '9':
                counts[9] += 1;
                break;
            default:
                break;
        }
    }
  }
}
```

APPENDIX B: PANAS SCALE

Subject______ Run_______

This scale consists of a number of words that describe different feelings and emotions. Read each item and then mark the appropriate answer with an X in the space next to that word. Indicate to what extent you *feel this way right now*, that is, *at the present moment*. Use the following scale to record your answers.

	very slightly or not at all	a little	moderately	quite a bit	extremely
interested					
distressed					
excited					
upset					
strong					
guilty					
scared					
hostile					
enthusiastic					
proud					
irritable					
alert					
ashamed					
inspired					
nervous					
determined					
attentive					
jittery					
active					
afraid					

APPENDIX C: EXPERIMENTAL PROGRAM FOR THE EXPERIMENT

Experimental program for the experiment. The lines with errors are marked in bold text with the solution following in comments.

```
/****************************
 * cardDealer.c             *
 * This program deals two   *
 * hands of 10 cards from   *
 * one deck (52 cards).     *
 ****************************/

#include <stdio.h>
#include <time.h>
#include <string.h>

#define SIZE 13

int randomSuit( void );
int randomCard( void );
int checkCard( int usedDeck[][SIZE], int suit, int card);
char determineCard( char deck[][SIZE], int suit, int card);
void determineSuit( int suit, char cardSuit[] );
void setupDeck( char deck[][SIZE] );

int main(void){

  char deck[4][SIZE];
  int usedDeck[4][SIZE] = {{0}};
  int moreCards = 1, suit, card, used = 0, hand = 1;
  char cardVal = '0', cardSuit[9] = "";

  srand(time(NULL));

  setupDeck(deck);

  while ( hand < 3 ) {
    printf("Hand #%d\n", hand);
    while ( moreCards <= 10 ) {
         while ( used == 0 ) {
             suit = randomSuit( suit );   // suit = randomSuit();
             card = randomCard();
             used = checkCard( deck, suit, card );  // used = checkCard( usedDeck, suit, card );
         }
         cardVal = determineCard( deck, suit, card );
         cardSuit = determineSuit( suit );   // determineSuit( suit, cardSuit);
         printf("Card %d: %c of %s\n", moreCards, cardVal, cardSuit);
         moreCards++;
         used = 0;
    }
    moreCards = 1;
    hand++;
```

```
  }
  return 0;
}

/* Get random suit number */
int randomSuit( void ) {
  return (rand()%4);
}

/* Get random card number */
int randomCard( void ) {
  return (rand()%13);
}

/* Check to see if card has been used yet */
void checkCard( int deck[4][SIZE], int suit, int card) {  // int checkCard( int deck[4]
[SIZE], int suit, int card) {
  int check;

  if( deck[suit][card] == 1 ) {
    check = 0;
  }
  else {
    deck[suit][card] = 1;
    check = 1;
  }
  return check;
}

/* Get the card */
char determineCard( char deck[4][SIZE], int suit, int card) {
  return deck[suit][card];
}

/* Figure out what suit the number corresponds to */
void determineSuit( int suit, char cardSuit[9]) {
  switch(suit) {
        case 0:
          strcpy(cardSuit, "Hearts");
          break;
        case 1:
          strcpy(cardSuit, "Diamonds");
          break;
        case 2:
          strcpy(cardSuit, "Clubs");
          break;
        case 3:
          strcpy(cardSuit, "Spades");
          break;
  }
  return 0;
}

/* Set up the new deck of 52 cards (4 suits with 13 cards each) */
```

```
void setupDeck( char deck[][] ) {  //  void setupDeck( char deck[4][SIZE] ) {
  int suit, card;

 for( suit = 0; suit < 4; suit++ ) {
   for( card = 0; card < 13; card++ ) {
     switch(card) {
        case 0:
                deck[suit][card] = 'A';
                break;
        case 1:
                deck[suit][card] = '2';
                break;
        case 2:
                deck[suit][card] = '3';
                break;
        case 3:
                deck[suit][card] = '4';
                break;
        case 4:
                deck[suit][card] = '5';
                break;
        case 5:
                deck[suit][card] = '6';
                break;
        case 6:
                deck[suit][card] = '7';
                break;
        case 7:
                deck[suit][card] = '8';
                break;
        case 8:
                deck[suit][card] = '9';
                break;
        case 9:
                deck[suit][card] = 'X';
                break;
        case 10:
                deck[suit][card] = 'J';
                break;
        case 11:
                deck[suit][card] = 'Q';
                break;
        case 12:
                deck[suit][card] = 'K';
                break;
       }
     }
   }
}
```

APPENDIX D: SELECTED REMOTE ASSOCIATES TEST ITEMS

The Remote Associates Test items used in the experiment, sorted by the standardized solvability values. Each item contains three words, followed by the correct solution.

EASY RAT Items (66-100% solvability)
cottage swiss cake : cheese
cream skate water : ice
loser throat spot : sore
show life row : boat
night wrist stop : watch
duck fold dollar : bill
rocking wheel high : chair
dew comb bee : honey
fountain baking pop : soda
preserve ranger tropical : forest
aid rubber wagon : band
flake mobile cone : snow
cracker fly fighter : fire
safety cushion point : pin
cane daddy plum : sugar

MEDIUM RAT Items (33-65% solvability)
dream break light : day
fish mine rush : gold
political surprise line : party
worm shelf end : book
flower friend scout : girl
river note account : bank
pie luck belly : pot
hound pressure shot : blood
food forward break : fast
water mine shaker : salt
home sea bed : sick
sage paint hair : brush
pike coat signal : turn
wheel hand shopping : cart
right cat carbon : copy

HARD RAT Items (1-32% solvability)
fly clip wall : paper
age mile sand : stone
health taker less : care
lift card mask : face
down question check : mark
tail water flood : gate
way board sleep : walk
marshal child piano : grand
time blown nelson : full
pile market room : stock
fence card master : post
tooth potato heart : sweet
wise work tower : clock
pea shell chest : nut
pet bottom garden : rock