Abstract

ODUOR, KENYA FREEMAN. The Effects of Automated Decision Aid Reliability and Algorithm Modality on Reported Trust and Task Performance. (Under the direction of Eric N. Wiebe.)

As IT systems grow more complex and become more prevalent, understanding the collaborative nature of the relationship between humans and automation becomes more important. Several factors influence the human-automation relationship. Research has shown that trust and perceived reliability are key factors in whether a relationship will develop between humans and automation. Presenting automation reliability and automation algorithms are ways to potentially improve this relationship. To explore this question, an experiment was conducted in which an automated decision aid presented suggestions to participants while they managed a simulated city (i.e., Policity). The goal was to maximize the health of the city's population by adding hospitals, housing, businesses, and other facilities and services. Participants were assigned to conditions where the automated decision aid performed with high or low reliability levels. Based on condition, the decision aid's algorithm was presented to participants in a textual or graphical (diagrammatic) format. Results showed that users' perception of the decision aid's reliability directly influenced their trust in the decision aid. Results also showed that presenting the decision aid's algorithm, regardless of modality (i.e., textual or graphical) had a direct impact on reported trust. Both had a direct effect on human performance. Additional results and implications are discussed.

THE EFFECTS OF AUTOMATED DECISION AID RELIABILITY AND ALGORITHM MODALITY ON REPORTED TRUST AND TASK PERFORMANCE

by

KENYA FREEMAN ODUOR

A dissertation submitted to the Graduate Faculty
of North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

PSYCHOLOGY

Raleigh, North Carolina

2006

APPROVED BY:

Eric N. Wiebe, Ph.D.	Robert St. Amant, Ph.D.
Denis O. Gray, Ph.D.	Rupert W. Nacoste, Ph.D.

Michael P. Etgen, Ph.D.

Biography

Kenya was born in Pittsburgh, Pennsylvania and moved to the Bronx, New York as a child. She spent most of her childhood in Queens, New York, where she attended Catholic school through the 12th grade. Kenya attended the University of Maryland, in College Park where she received a Bachelor's of Science degree in Psychology. After spending several years in the physical therapy field, Kenya pursued a graduate degree in Ergonomics. She attended North Carolina State University in Raleigh, where she obtained a Master's of Science degree in Ergonomics and completed the present research as part of the requirement for the Doctoral program in Ergonomics.

Kenya spent most of her graduate school career working as an intern at IBM in Research Triangle Park, North Carolina. The position primarily involved web application software user interface design. Kenya joined the team at IBM full time, after 4 years as a coop.

Kenya currently resides in Morrisville, North Carolina with her husband and dog. She enjoys walking with the dog, exercising, and volunteering in the community.

Acknowledgements

First and foremost, I would like to thank the Almighty Father. Without Him, none of this would have been possible. It took a great deal of faith and prayer to get through the challenging aspects of graduate school and performing research.

I want to thank my parents for all of their encouragement and prayers. I appreciate each and every sacrifice that you made to help me get here. I will never forget all that you have done for me.

I would like to thank my academic advisor for his guidance and support in helping me successfully complete this research. I would also like to thank my thesis advisor and my formal and informal mentors who have aided in my spiritual, academic, and personal growth over the years.

I want to thank each and every one of my friends and family members that kept me encouraged. The countless hours spent studying, typing, researching, talking, and keeping each other going when we wanted to do something else fun will never be forgotten. I don’t know how I would have gotten through school without each of you there to hold me accountable.

Table of Contents

List of Tables

List of Figures

Introduction

Oliner & Sichel (2002) provided estimates of the growth in productivity in the United States economy and attributed the acceleration in labor productivity after 1995 to the greater production and use of information technology (IT). They felt that the future economic increases would depend on the extent to which products embodying IT diffuse through the economy.

Automating aspects of the IT infrastructure can make this technology more marketable as well as more usable. Automating key aspects of IT can improve availability, security, complexity and the management of functions. Automation is not a new concept. However, the utilization of automation in IT, more specifically software applications, is an emerging technology that is experiencing growing pains. Automation involves a shift of control from the user to the IT, and therefore, requires some amount of trust. The development of trust is essential in one's relationship with emerging technologies. Once an exchange takes place between automation and its user, the development of trust may or may not begin. This is dependant upon the nature of the interaction and the production of desired outcomes. Like a new romance, the initial exchange between the automation and the user are what determine if the interaction will continue. Therefore, it is important that one has a positive experience with the automation.

Yet, automation complexity makes it difficult for an effortless exchange to take place and consequently misunderstandings may arise. Finding ways to alleviate the difficulty in interacting with software automation are important. Just as another human can tell you what drives their actions, automation can "tell" you why it acts the way it does. A step-by-step problem solving procedure, or algorithm (advice), is one way of exposing that information making it more transparent to the user. The ways in which algorithm transparency help develop this relationship remains to be seen.

The complexity of automation continues to increase. Recent polls showed that approximately 75% of people in the United States feel that life is too complicated and that technology contributes to this complexity in a significant way (Reeves, 1999). As software automation becomes more prevalent, the development of an appropriate relationship between humans and automation is necessary for the technology to be successfully implemented. The present research will investigate the following aspects of the human-automation relationship:

1. Prior to the development of a relationship with automation, is a human's interaction with automation simply a stimulus-response pattern?
2. Does the presentation of automation's algorithm assist in the development of an appropriate mental model?
3. Does mental model development help facilitate the growth of an appropriate relationship as measured through trust and appropriate usage?
4. Based on experiences with automation and an understanding of the automation's algorithm, will one's ability to make (accurate) predictions of the automation's behavior facilitate appropriate use of the automation?

What is Automation?

Automation is defined as technology that actively selects data, modifies information, makes decisions, or manages processes (Lee & See, 2004). Automation involves a shift in the allocation of function from a human to an automated system or machine. Technology of this kind has tremendous potential to extend human performance, because it transfers functions from a human operator to an automated system. Automation has been implemented in a variety of systems. It can be found in environments as diverse as motor vehicle operation, process control, and information retrieval systems.

Several categorizations of automation exist. The first form of automation executes or controls actions (e.g., levers that control physical processes). This type of automation includes thermostats, cruise control, and power steering (Dougherty, 2003). This form of automation also involves product flowing through a physical process (e.g., a manufacturing process). Within this type of automation there are varying levels of human operator control that ranges from directing manual control, to directing automatic control, to directing multiple levels of automatic control (Llinas, Bisantz, et al., 1998).

An additional function of automation is to present data or advice to human operators. This executive or control automation aids an operator in decision-making and problem solving. Automated decision aids (DA) gather information from the environment, integrate it with other information, and present a recommendation about the best decision to be made by the operator. This form of automation can be found in clocks, fuel gauges, decision aids and expert systems. Automated systems of this form can perform tasks ranging from simple to complex (Dougherty, 2003). A final, less common form of automation assists operators in managing other automation. A flight management system is an example of this form of automation (Dougherty, 2003). For the purposes of this research, the focus will be on automated decision aids.

Kelly et al. (2001) maintain that for something to be considered automation it must not simply include an improvement to the technology, but it must also involve a change in the allocation of function from the human operator to the automation. Additionally, the perception of what is automation changes over time. Parasuraman and Riley (1997) assert that when a reallocation of function from human to machine is complete and permanent, then the function will be perceived as a machine operation, not as automation, making today's automation, tomorrow's machine. It is customary to assign a given level of automation to an automated system. This can be problematic in systems that are not necessarily unitary entities. An automobile contains varying levels of automation in its discrete subsystems. Because this system may have several different subsystems with varying levels of automation, assigning one level of automation to the entire system may be a problem. In this instance, it is not clear how to label the level of automation of the entire system.

The functions performed by automation vary greatly from system to system. Several taxonomies have been developed to help categorize the levels of automation and the allocation of function to the human operator and the automated system.

Fitts (1951) conducted research on function allocation that characterized those functions that a human could perform better than a machine and those that a machine could perform better than a human (see Table 1).

Table 1

<u>Fitts' Allocation of Function List adopted from Fitts (1951)</u>

Property	Machine	Human
Speed	• Much superior	• Lag one second
Power	• Consistency at any level • Large constant standard forces and power available	• 2 HP for about 10 seconds • 0.5 HP for a few minutes • 0.2 HP for continuous work over a day
Consistency	• Ideal for routine, repetitive, or precision tasks	• Not reliable; should be monitored • Subject to learning and fatigue
Complex Activity	• Multi-channel	• Single channel • Low information throughput
Memory	• Best for literal reproduction and short term storage	• Large store multiple access • Better for principles and strategies
Reasoning	• Good deductive power • Tedious to re-program	• Good indicative power • Easy to re-program
Computation	• Fast accurate • Poor error correction	• Slow, subject to error • Good error correction
Input (sensing)	• Some outside human sense range: i.e., non-visible electromagnetic radiation • Insensitive to extraneous variables • Poor pattern recognition	• Wide range (10^{12}) and variety of stimuli dealt with by one unit • Affected by heat, cold, noise, and vibration • Good pattern detection • Low signal detection • Good signal discrimination with high noise levels
Overload reliability	• Sudden breakdown	• Graceful degradation
Intelligence	• None • Incapable of goal switching or strategy switching without specific directions	• Can deal with the unpredicted • Can anticipate • Can adapt
Manipulative abilities	• Task specific	• Great versatility and mobility

Fitts' work would later experience criticism. Moray, Hiskes, Lee, & Muir (1995) felt that Fitts' list wrongly characterized function allocation as a one-time activity that is complete once a system has been designed and implemented. Fitts' list has limited utility in engineering design because it is too qualitative, too general, and lacks fit with engineering concepts. Fitts' list may be useful early on in the automation design process.

Sheridan and Verplank (1978; cited in Levis, Moray, & Hu, 1994) introduced the first categorization of automation levels that described the interaction between humans and machines (see Table 2). The Sheridan-Verplank Scale of Human-Machine Task Allocation (SVL) was designed to describe the task allocation between the human and the automated agent. It was supposed to assist engineers with determining the appropriate level of automation for a human-machine system.

Table 2

Sheridan-Verplank Scale of Human-Machine Interaction from Sheridan & Verplank (1978)

Sheridan-Verplank 10 Levels of Human-Machine Function Allocation
1. The human does all the planning, scheduling, optimizing, etc. and turns tasks over to computer for merely deterministic execution.
2. Computer provides options, but human chooses between them, plans the operations, and then turns the tasks over to the computer for execution.
3. Computer helps to determine options, and suggests one for use, which the human may or may not accept before turning task over to the computer for execution.
4. Computer elects options and plans actions, which human may or may not approve, computer can reuse options suggested by human.
5. Computer selects action and carries it out if human approves.
6. Computer selects options, plans and actions and displays them in time for the human to intervene, and then carries them out in default if there is no human input.
7. Computer does entire task and informs human of what it does.
8. Computer does entire task and informs human only if requested.
9. Computer does entire task and informs human if it believes the human needs to know.
10. Computer performs entire task autonomously, ignoring the human supervisor who must completely trust the computer in all aspects of the decision-making.

Moray, Inogaki, & Itoh (2000) group these ten levels into 3 clusters. In the first five levels, humans are the decision-makers and controllers of the automation. Dynamic collaboration between the human and the machine exists in levels 5 to 7. Systems with automation described in levels 7 to 10 are considered fully automated. Automation at the highest level is able to act autonomously without human intervention.

Parasuraman, Sheridan, & Wickens (2000) utilize a four-stage model of human information processing to devise four functions that must be accomplished to perform most tasks:

1. information acquisition;
2. information analysis;
3. decision and action selection;
4. action implementation.

Since these functions are performed by a human or by automation at various levels, Parasuraman et al. (2000) considers that most human-automation systems consider a mix of levels of automation across the four stages. For example, one system might have a high level of automation across all four sub-functions, and a second system might be highly autonomous in information analysis, but fairly low on the other three functions. According to this model, a parent task is broken down into abstract sub-tasks based on the information processing stages.

The two-dimensional model of automation is a step up from the earlier uni-dimensional models, however it does not extend far enough. The subdivision of the parent task into the four information-processing phases represents only one level of decomposition into an abstract set of task categories. Woods (1996) asserted that automation does not merely shift responsibility for tasks but it can also change their nature. Task decomposition means that sub-tasks may be added as well as eliminated.

Human-computer systems in which the "division of labor" between the human and machine is dynamic, rather than fixed, are referred to as adaptive automation (AA). AA or dynamic function allocation has been categorized as human-centered automation when compared to traditional technology-centered automation. Some of the benefits believed to be associated with AA include alleviation of a loss of situation awareness (SA), of mental workload, and of out-of-the-loop performance issues (Kaber, Riley, Tan & Endsley, 2001). Situation awareness is defined as "*the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future*" (Endsley, 1988, p.97). Situation awareness involves perceiving critical factors within the environment (Level 1 SA), comprehending those factors, particularly when they are integrated with one's goals (Level 2 SA), and at the highest level, understanding what will take place in the system in the near future (Level 3 SA). A theory of human-centered automation closely related to AA asserts that automated systems should be designed to support operator achievement of SA through significant involvement of operators in control processes (Endsley, 1995b, 1996; Kaber & Endsley, 1997). Human involvement may occur at intermediate levels of automation (LOA) or through AA. At intermediate LOA, the human may be more involved in operating the system and more effective at dealing with the automated system when needed. In comparison to full automation, these two techniques may effectively increase human involvement. Human-centered automation is concerned with situation awareness because it has been found to be critical to human performance in automated system processes (Endsley, 1995a).

Automation Complexity

Automation can be characterized as complex for a number of reasons. Complexity may exist because of a number of interrelated components, operation under many different modes, varying levels of automation in discrete subsystems, and complex algorithms. One or more of these and other phenomena can add to a system's complexity. Complex automation increases the likelihood of cognitive complexity (Reeves, 1999). Reeves (1999) defines cognitive complexity (CC) as a reference to external elements that contribute directly to our neural load, and in doing so reduce our capacity to think clearly and comprehend. External factors that make things hard to grasp, hard to comprehend, and hard to use and see create CC; it contributes to an increase in neural load and makes it difficult to learn, make decisions, and solve problems. Similarly, high complexity issues also make it difficult for the automation to extend the performance of the human in the system. Complexity makes it more difficult to find the right computer-based operations that will eventually lead to the correct solution. When complex automation replaces the human in some or all of the tasks of completing goals in a particular problem set, the automation and its interface become a part of the problem and consequently add to the CC. Reeves (1999) states that problems that are complex in nature may possess any number of the following issues:

1. *Intransparency*: Only some of the variables lend themselves to direct observation.

2. *Multiple goals*: With multiple goals, some may be contradictory, and trade-offs are required.
3. *Complexity of the situation*: This may conflict with the limited capacity of the problem solver to think it through.
4. *Connectivity*: Complex problems often contain a high degree of connectivity or interrelationship. In other words, it is very difficult to anticipate all the possible consequences of a given situation.
5. *The dynamic nature of complex problems*: In other words, complex problems can worsen, creating great time pressure and unpredictability.
6. *Time delay*: In complex problems, there can be a delay between the action taken and the response or the appearance of consequences. This places an extra burden on the problem solver.

Funke (1991) provides a list of elements to simplify complex problems which includes:

1. Greater availability of information about the problem;
2. Precision of the goal definition;
3. A condensed number of variables, degrees of connectivity, and the linear relationships between them; and
4. The stability of the properties of the problem or time dependencies in the course of the problem-solving process

Lieberman (1997) defines automation, more specifically automated software, as any program that can be considered by the user to be acting as an assistant or helper. Automated software can be viewed as an assistant by one person and as a tool by another, depending on their view of how the automation is acting. Having automation work directly in the user interface rather than as a background process increases the extent to which the user will perceive the software as acting like an assistant. The automated software will also be seen as an assistant in instances where the user perceives the automation's actions as actions that he or she could have done on their own.

Software, unlike other engineering artifacts, is pure design. Its unreliability is always the result of design flaws, which arise from human intellectual failures (Littlewood & Strigini, 2000). In comparison, the unreliability of hardware systems is seen to be the result of random physical failures of components. Current reliability theories have successfully allowed hardware systems to be built to high reliability requirements, and the final system reliability to be evaluated with acceptable accuracy. Presently, many of these systems have come to rely on software for their correct functioning, making the reliability of software even more important. The problems with software automation reliability are a result of the difficulty and novelty of the problems that are tackled, the complexity of the resulting solutions, the need for short development cycles, and the difficulty in gaining assurance of reliability because of the inherently discrete behavior of software systems. It is particularly difficult to gain assurance of the reliability of software. It is usually impossible to generalize the error-free operation of the automated software in one situation to claim that it will perform appropriately in another situation. However, software must be sufficiently reliable before one can make intelligent decisions about transferring human control to it.

Patterns of Automation Usage

Automation use decisions are based on a complex interaction of many factors and are subject to strongly divergent individual considerations. Attitudes toward automation vary widely among individuals (Helmreich, 1984; McClumpha & James, 1994). Recognizing these attitudes – positive and negative, general and specific – is the first step toward understanding human use of automation. Use refers to the voluntary starting or stopping of automation by human operators. Factors affecting one's ability to monitor automation influences use, includes automation reliability and consistency, workload, and the saliency of automation state indicators. Automation use is also influenced by trust, mental workload, and risk, but individual differences and interactions between factors make prediction of automation use difficult.

An over-reliance on automation, which can result in failures of monitoring or decision biases, is automation misuse. Disuse, or underutilization of automation, is caused by a perception of unreliability and might be caused by alarms that continually activate falsely. This often occurs because the base rate of the condition to be detected is not considered in establishing the trade-off between false alarms and omissions. The automation of functions by designers and implementation by managers that disregard the consequences for human performance and that tend to define the operator's roles as by-products of the automation constitute automation abuse. Automation abuse can also promote automation misuse and disuse by human operators. Understanding the factors associated with each of these aspects of human-automation interaction can lead to improved system design, effective training methods, and reasonable policies and procedures involving automation use (Parasuraman & Riley, 1997).

Overview

As key aspects of systems become more highly automated, it is increasingly important that the human's interaction with these systems be well understood. System and machine automation are not new. However, there are many aspects of software-based automation that are still flawed. With the persistence of software reliability issues, the addition of automation increases complexity and increases the potential for automation misuse, disuse, or abuse to arise. The human's development of trust in the automation is essential to appropriate usage. Therefore, ways to improve the trust relationship must be found. One way to potentially increase user trust is to make the automation's algorithm transparent to the user. Not making the automation's algorithm transparent can lend to complexity because of the limited ability of variables that can be directly observed (Reeves, 1999). Presenting the automation's algorithm may help increase one's comprehension of the automation's underlying functions, reducing cognitive complexity, and subsequently giving the user the ability to discern whether or not the automation is acting appropriately. Software automation may never attain perfect reliability, but in instances where it fails to act as expected, one's ability to troubleshoot its behavior may be helpful. This predictive ability may also foster an appropriate relationship that is built on trust. This research will look be looking at the relationship between humans and automation and how the automation's algorithm transparency may or may not facilitate trust and appropriate use.

The Relationship Between Humans and Automation

When a human comes together with automation to form a complex whole, automation is expected to improve the throughput, safety, and efficiency of the environment. Within this system, software automation can operate in the interface, as well as the back end of an application. Traditional interface design is geared towards conversational interfaces, where the human and the automation "take turns" acting. Automated software can also act "autonomously", as opposed to an exchange taking place. Automation that has any level of automaticity may need to interact with the interface while the user is also interacting with the interface. The user may or may not be aware of the automation's activities at any given moment. To appropriately rely upon the automation, a user must be given some amount of information about its activity. As our dependence on software increases, the associated automation is expected to enhance rather than complicate the resulting system. Littlewood & Strigini (2000) discuss automated software's move from a supportive role to a primary role in providing critical services. Some of the applications of automated software imply extremely high dependability requirements especially in the deployment of new systems. Like 'electronic commerce' or e-commerce, automated software is becoming the only way to perform some functions; failure would deeply affect individuals or groups that depend on it. Automated software is also becoming an accepted part of everyday life and in some cases, is integrated into systems without effective human control. In this instance, these sorts of systems are built where software failures can spread their effects more quickly and with less opportunity for human correction.

Similar to an interpersonal relationship, one must get acquainted with automation in order to develop an appropriate human-automation relationship. One must have a positive perception of the relationship with automation in order to allow it to assist in task completion. Prior to the development of this relationship, automation use may be based on a simple exchange, similar to the stimulus-response model of behaviorism (Schwartz, B., Wasserman, E.A., & Robbins, S.J., 2002). If a user has no prior experience with or perception of a certain type of automation, in a sense, actions can be explained entirely as responses to stimuli. Observable behavior may be the only suitable measure of the human-automation relationship until the user has had sufficient interaction with the automation to form an opinion about its capabilities and its role.

Reeves & Nass (1996) maintain that humans respond socially to technology and responses to computers can be analogous to reactions to human collaborators. There is a phenomenon in social psychology known as the similarity-attraction hypothesis. The hypothesis states that people with similar personality traits will be attracted to one another (Nass & Lee, 2001). Some people develop this sort of attraction to technology, such as automation.

These findings can be generalized to user acceptance of software (Nass & Lee, 2001; Nass, Moon, Fogg, Reeves, & Dryer, 1995). Software that demonstrates personality characteristics similar to those of the user tends to be more easily accepted. For example, dominant users will find software with directive language such as “You should definitely do this” more appealing, while software that uses less authoritative language, such as “You may want to do this” will tend to appeal to more submissive users (Nass & Lee, 2001). This collection of research implies that the attitudinal and emotional factors that influence human-human relationships may also contribute to the relationship between humans and automation.

Anthropomorphization involves “attributing human characteristics to non-human phenomena” (Guthrie, 1993). Nass et al. (1995) found that subjects could be persuaded to behave as if computers were human, even though users knew they were not. There is a tendency to believe that the human brain is like a computer and therefore computers can also have human qualities (Muir 1988).

Caporael (1986), Nass, Steuer, Tauber, & Reeder (1993), Reeves & Nass (1999), and Watt (1997) maintain that the anthropomorphization of technical devices or animals is a common phenomenon. Even if a person is aware that they are addressing an inanimate object, the anthropomorphization persists. Several studies support the notion that the interaction with computers is primarily social in nature (e.g., Nass, Steuer, & Tauber, 1994; Takeuchi & Ketagiri, 1999), even if the amount and mode of computer use is moderated by other factors (e.g., attitudes toward computers or computer skill level; Levine & Donitsa-Schmidt, 1998).

Similar to human-human interaction, users that perceive automated software to be more reliable than manual operation are more likely to place greater trust in the automated system, and rely on it. If the automated system is truly more reliable than manual operation, appropriate automation use will take place. However, when users inaccurately estimate the reliability of the automation or their manual operation, then inappropriate automation reliance may occur (Pomranky, Dzindolet, & Peterson, 2001).

A comparison of the perceived reliability of an automated aid (trust in aid) and the perceived reliability of manual control (trust in self) determines automation use. The outcome of the decision making process has been named the “perceived utility” of the automated system and is expected to be directly related to the trust of the automated aid and subsequent use, misuse, disuse, or abuse (Dzindolet, Pierce, Beck, & Dawe, 2000).

The perceived utility of the automation will be most accurate when the *actual* ability of the automation and the *actual* ability of the manual operator are compared. Unfortunately, the actual reliability of the automated aid and the manual operator are unlikely to be perceived accurately by the user. Realistically speaking, biases and errors are likely to occur. The greater the biases and errors, the more likely disuse and misuse are to appear (Pomranky et al., 2001).

Users often expect and predict near-perfect performance from automation. This sort of expectation can lead to disuse of the automation it if does not perform as expected. Prior to interaction, users often have a schema regarding automation that it should be perfectly reliable and accurate. It is this schema that leads users to expect perfect performance from the automation. Cognitive psychologists have found that information inconsistent with a schema is likely to be remembered and will greatly influence information processing (Smith & Graesser, 1981; Ruble & Stangor, 1986).

When a user is presented with a simple task, promptly decides a course of action, and makes a decision with a high degree of confidence, the user assumes the automation will be in agreement. When the automation reaches an opposing decision, the user is likely to detect the obvious error just committed by the automation and will question the automation’s reliability. Repeated occurrences of the automation committing obvious errors may result in a distorted view of the automation’s perceived utility (Pomranky et al., 2001).

When events occur that are in direct opposition to a user's original expectation, the user will be more likely to remember the event. Pomranky et al. (2001) assert that errors committed by the automated aid are inconsistent with the user's schema that the automation is accurate and reliable. Therefore, every error made by the automation is likely to be highly salient. As the task progresses, it may be difficult for the user to maintain an accurate picture of the aid's reliability. The contradictory information may be exaggerated and prominent in one's mind resulting in a distorted negative view of the automation's ability. This may lead the user to underestimate the automation's performance. Consequently, an equally salient form of advice or feedback is necessary to overcome the effects of the inconsistencies.

Lee & Moray (1992) defined the factors that influence the development of trust and an appropriate relationship between humans and automation. They determined performance, process, and purpose as the bases of trust in automation.

Performance is the past and present operation of the automation that includes characteristics such as ability, reliability, and predictability. Performance information defines what the automation does and refers to the expertise and capability as demonstrated by its ability to accomplish the user's goals. Like Sheridan's (1992) concept of robustness as a basis for trust in automation, performance demonstrates the task-dependant nature of trust. When automation performs in a manner that reliably accomplishes user's goals, it is likely to be trusted.

Process is the extent to which the automation's algorithms are relevant to the situation and are capable of achieving user goals. Process information defines how the automation functions. Process as a basis for trust is a shift of focus away from explicit behaviors and toward the characteristics and attributes of the automation. In other words, trust is shifted from the specific actions of the automation to the automation itself.

In the context of automation, algorithm presentation is the process basis of trust. This is similar to Sheridan's (1992) notion of understandability of one's relationship with automation. Users tend to trust automation if its algorithm can be understood and is a feasible means of achieving the user's goals in a given situation.

Purpose is the extent to which the automation is being used within the domain in which it was designed as intended. Purpose explains why the automation was created. Like benevolence and faith, purpose shows the perception that the user has a positive orientation toward the automation. The development of this relationship will depend on whether the designer's intent has been clearly communicated to the user. If so, there is a tendency for the user to trust that the automation will achieve the goals it was designed to achieve.

Trust in Automation

A number of definitions of trust exist. A universal definition has yet to emerge. One definition of trust asserts that it is the attitude that an agent will help achieve an individual's goal in a situation that is vulnerable or uncertain. In this definition, an agent can be another person or automation that dynamically interacts with the environment on behalf of the individual. Substantial human-human interaction research has shown the importance of the attitude of trust in mediating how people rely on each other (Deutsch, 1958; Rempel, Holmes, & Zanna, 1985; Ross & LaCroix, 1996; Rotter, 1967).

Within theory and research of the relationship between humans and automation, several definitions of trust have appeared. Madsen & Gregor (2000) define trust in decision aids as the extent to which a user is confident in, and willing to act based on the recommendations, actions, and decisions of an artificially intelligent agent. Moray & Inagaki (1999) define trust in automation as an attitude which includes the idea that the collaborator will perform as expected, and can, within the boundaries of the designers' intentions, be relied upon to achieve the design goals.

Lee & See (2004) assert that trust guides, but does not entirely determine reliance. A large body of current research exists on the topic of trust and reliance, however it has produced many contradictory and confusing findings.

Despite contradictory findings, it is clear that the development of trust in automation is important to its rate of adoption. *Rate of adoption* is the relative speed with which an innovation (e.g., an automation tool) is adopted. A number of attributes of an innovation affect its rate of adoption. The *relative advantage* is the degree to which an innovation is perceived better than its predecessor, in this case, manual control. The relative advantage of automation will be apparent only if it assists a user in performing a task rather than complicates it. *Compatibility* is the degree to which the innovation is perceived as consistent with past experiences, existing values, and the needs of potential adopters. If automation succeeds in incorporating the values, needs, and beliefs of users, it is more likely to be adopted. The degree to which an innovation is perceived as relatively difficult to use and understand is *complexity*. Finding ways to minimize the complexity of the interaction with automation is essential to its adoption (Rogers, 1985).

Developing trust in automation can be challenging when automation can be "brittle": only working appropriately in situations for which it is designed (Gregory, 1986; Will, 1991); making it difficult for users to maintain vigilance (Moray, 1986), situation awareness (Endsley, 1996), or effectively adjusting the automation's state (Lee & Moray, 1992). Because of the uncertainty that a relationship with automation presents, it is necessary to develop appropriate levels of trust.

Trust as a Social Psychological Construct

There are a number of theories on trust as a social psychological construct. Several theorists provide definitions on the basis of close relationships in the interpersonal trust domain. Thibault & Kelley (1959) developed a theory of how people decide what to do in their relationships. The Social Exchange Theory, is a theory based on an individual's exchange of rewards and costs that quantifies the values of outcomes from different situations. The outcome of an interaction is the combination of rewards and costs. People attempt to minimize costs and maximize rewards, and then base the likeliness of developing a relationship with someone on the perceived possible outcomes. When the outcomes are perceived to be positive, an individual will disclose more and develop a closer relationship with that other person. Boon & Holmes (1991) define trust as a state involving confident predictions of another's motives in situations that entail some level of risk.

A definition of trust from an organizational theory perspective, defines trust as an individual's willingness to be vulnerable to the outcomes of another party based on the expectation that the other will perform a desired action, regardless of the ability to monitor or control the other party (Mayer, Davis, & Schoorman, 1995).

Trust evolves within the context of the individual, the organization they belong to, and their culture. The individual context consists of individual differences such as the inclination to trust. These differences affect one's initial level of trust and their interpretation of new information. The individual context also includes an individual's experiences that have led to a particular level of trust (Lee & See, 2004).

Individual differences are what make some people more inclined to trust than others (Gaines, Panter, Lyde, Steers, Rusbult, Cox, et al., 1997; Stack, 1978). Rotter (1967) identified trust as a persistent personality trait. This definition of trust follows a social learning theory approach, in which expectations for a given situation are determined by previous experiences with situations that are perceived to be similar (Rotter, 1971). People form beliefs about others that are generalized and taken from one interaction to the next. In this case, trust is a generalized expectancy that is independent of a specific situation. Instead, it is based on the generalization of a considerable number of diverse experiences. Individual differences regarding trust have strong implications for the study of human-automation trust because they may impact reliance in ways that are not related to the characteristics of the automation.

Considerable evidence shows that the tendency to trust, considered as a personality trait, can be measured reliably and can influence behavior in a systematic way. Rotter's (1980) Interpersonal Trust Scale distinguishes people on their inclination to trust others. Rotter (1971) found that individuals with a high level of trust are seen as more trustworthy by others and exhibit more truthful behavior. People with a high tendency to trust predicted others' trustworthiness better than those with a lower tendency to trust (Kikuchi, Wantanabe, & Yamasishi, 1996). Similarly, low- and high- trust individuals respond differently to feedback regarding collaborators' intentions and to situational risk (Kramer, 1999).

The previous findings may explain why individual differences and the general propensity to trust automation (as measured by a complacency scale; Parasuraman, Singh, Molloy, & Parasuraman, 1992) are not obviously related to misuse of automation. Singh, Molloy, & Parasuraman (1993) found that high-complacency individuals detected more automation failures in a constant reliability condition (53.4% compared with 18.7% for low complacency individuals). Wiener (1981) defines complacency as a psychological state characterized by a low index of suspicion. The condition is a result of working in environments with highly reliable automation, in which the operator serves as a monitor of system states looking for occasional automation failure. Complacency is exhibited as a false sense of security, which the operator develops when working with highly reliable automation; however, no machine is perfect and has the potential to fail without warning. Studies have shown that automation-induced complacency can negatively affect an operator's ability to monitor an automated system (Parasuraman, Molloy, & Singh, 1993). This unexpected outcome is similar to findings in studies of interpersonal trust, in which highly trusting individuals were found to trust more appropriately. A number of studies of trust in automation show that some peoples trust changes drastically as the automation's capability changes and for other people trust changes minimally (Lee & Moray, 1994; Masalonis, 2000). One possible explanation is that high-trust individuals may better adjust their trust to situations where automation is highly capable as well as to situations in which it is not.

While trust can be described as a personality trait, most research has focused on trust as an attitude (Jones & George, 1988). Within the realm of interpersonal relationships, trust has been considered as a dynamic attitude that evolves as the developing relationship progresses (Rempel, et al. 1985). Individual differences influence the predisposition to trust when a situation is new or ambiguous and generalizations dominate. Individual differences become less important as the relationship evolves (McKnight, Cummings, & Chervany, 1998). Trust as an attitude is a history-dependant variable that depends on the prior behavior of the trusted person and the information that is shared (Deutsch, 1958). The initial level of trust is determined by past experiences in related situations; some of these experiences may be indirect, as with gossip.

A framework developed by Azjen & Fishbein (1980; Fishbein & Azjen, 1975) helps to reconcile the conflicting definitions of trust. Their framework describes how behaviors result from intentions and that intentions are a function of attitudes. Beliefs are the basis for attitudes. According to their framework, perceptions and beliefs characterize the information base that determines attitudes. The availability of information and the individual's experiences influence beliefs. When considering trust and reliance, trust is an attitude and reliance is a behavior. The framework keeps beliefs, attitudes, intentions, and behaviors distinct and can help describe the influence of trust on reliance. Trust affects reliance as an attitude rather than as a belief, intention, or behavior. Beliefs are the basis for trust, and various intentions and behaviors may result from different levels of trust. The framework and its relationship to trust and reliance are depicted in Figure 1.

Within the trust in automation domain, there is evidence that cognitive processes play an important role in determining trust expectations. Muir (1994) demonstrated that the expectation of competence about a machine seemed to best describe what people meant when they said they trusted a machine. The perceived trustworthiness of automation may be influenced by the extent to which it does what it is expected to do to complete the task for which it was designed.

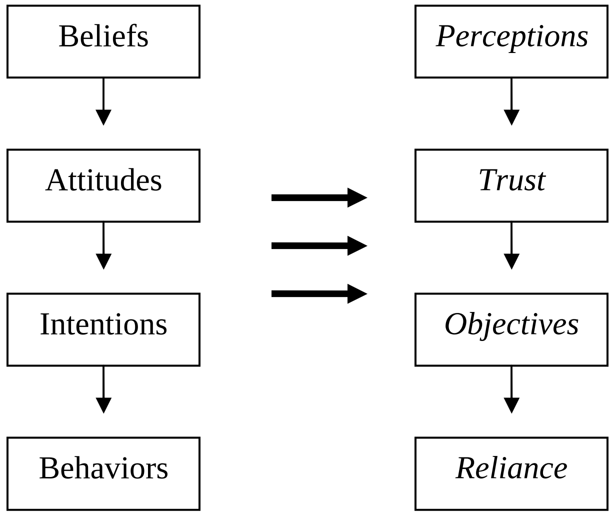

Figure 1. Azjen & Fishbein’s (1980) framework of the relationship between trust and reliance.

Therefore, trust in automation can be defined as the attitude that automation will help achieve an individual’s goals in a situation characterized by uncertainty and vulnerability (Lee & See, 2004). Trust is a nonlinear function of automation performance and the dynamic interaction between the user and the automation. Trust tends to be influenced by the worst behaviors of an automated system (Muir & Moray, 1996), which is always the case with people. That is, negative interactions have a greater influence than positive interactions (Kramer, 1999). Initial experiences have an unwavering effect on trust where an initial low level of reliability leads to lower trust and reliance, even when automation performance improves. Similarly, trust is more resilient if automation reliability starts high and decreases than when it starts out low and increases (Fox &

Boehm-Davis, 1998). No single level of reliability can be recognized as the cause for distrust and disuse. Trust depends on the consequence, timing, and expectations associated with failures of the automation (Lee & See, 2004).

Self-confidence is a particularly important variable that interacts with trust to influence automation reliance. Self-confidence is an important factor in general decision-making (Bandura, 1982) and more specifically, in mediating the effect of trust on reliance (Lee & Moray, 1994). When a user's self-confidence is high and trust in the system is low, they are more inclined to rely on manual control. Alternatively, users with low self-confidence are more inclined to rely on the automation (Lee & Moray, 1994). Riley (1996) compared students and aircraft pilots in a function allocation task. Students had a tendency to have a greater level of self-confidence and were less inclined to allocate tasks to automation, which they tended to distrust. The pilots, who were more accustomed to using the automation, trusted and relied on the automation more than the students. Thus, appropriate reliance on automation can be strongly affected by biases in self-confidence (Lee & See, 2004).

Within the domain of interpersonal relationships, trust is seen as involving affective processes. Developing trust requires seeing others as personally driven by sincere concern and care to protect our interests (Lewis & Weingert, 1985; McAllister, 1995). Trust evolves as people invest emotionally in relationships, express genuine concern for other's well being, and come to believe these feelings are reciprocated. This concept of investment and mutual concern is typically somewhat less prominent in a human's relationship with automation. Still, it is clear that trust in automation can have an affective component. To some extent, both our attitudes and our behavior toward automation are predicted not just on our knowledge and beliefs about it, but on how it makes us feel.

Lewandowsky, Mundy, & Tan (2000) found that the delegation to automation was different than the delegation to human collaborators. They found that the delegation to the humans, but not to automation, is dependent upon people's assessment of how others perceive them: People are more likely to delegate if they perceive their own trustworthiness to be low. They also discovered the degree of trust is more strongly related to the decision to delegate to the automation, as compared with the decision to delegate to the human. One possible explanation for these results is that users may perceive the final responsibility in a human-automation partnership to lie with the user, whereas individuals in a human-human partnership may perceive the ultimate responsibility as being shared. Similarly, people are less likely to disuse human aids than they are automated aids, even though self-reports imply a preference for automated aids (Dzindolet, Pierce, Beck, & Dawe, 2002).

In the context of interpersonal relationships, Rempel et al. (1985) contend that trust changes as a result of the relationship that develops. Early in the relationship, trust in based on the *predictability* or regularity or the degree to which future behavior can be anticipated. As the relationship matures, the nature of trust changes and becomes based upon the operator's attribution of *dependability*. This is the degree to which the behavior is consistent. At the highest level of trust, the user is capable of projecting beyond what can be observed to a broader attribution about the future dependability of the other person. At this point, *faith* has developed and the person is judged as being reliable. Muir (1987, 1994) has argued that trust in automation develops in similar stages.

However, trust in automation can follow an opposing pattern, in which faith is important early in the interaction, followed by dependability, and then by predictability (Muir & Moray, 1996).

The process by which trust in automation develops over time is similar to the way trust develops in the interpersonal domain. In the interpersonal domain, trust is seen to develop over time, and generally as the product of prolonged interaction with other people (e.g., Rempel et al., 1985). Muir (1994) agrees that in models of trust in automation, trust is also seen to develop over time and because of the same processes inherent in interpersonal trust (faith, dependability, and predictability). Subtle shifts in trust have been exhibited in response to the properties and performance of automation (Muir, 1989). When the automation performed reliably, operator trust increased over time. However, when the automation performed unreliably, trust quickly decreased. Trust in automation appears to be dynamic like that of interpersonal trust.

It is important to consider that the very goal of the relationship with automation may be different. Unlike interpersonal relationships, we do not interact with automation because of the need for personal relationships. We interact because we have a task to accomplish, and we begin our "relationship" with automation purely because it is capable of assisting in completing the task. The scope of the "relationship" is often confined to whether the automated system functions appropriately or not and there sometimes may be no need for expectations outside of this narrow scope to be even considered. In this context, trust can be based purely on a judgment that one can predict the automation's function, and there is little need for analysis or attribution processes.

Studies of human-automation trust (e.g., Muir & Moray, 1996) show that it is possible to quantify the subjective degree of trust experienced by users and to track meaningful changes in trust over time. Khasawneh, Bowling, Jiang, Gramopadhye, & Melloy (2004) sought to discover methods and metrics for the measurement of trust. The approach they followed illustrated that trust can be mathematically predicted by knowing system errors, an objective approach which provides a more accurate measurement of human trust.

Masalonis (2003) adapted Cohen, Parasuraman, & Freeman's (1998) model that predicts that training on the factors affecting reliability should lead to more appropriate trust assessments. Training was not found to affect overall performance, but it was found to impact bias. Trained individuals were more likely to report both real and perceived conflicts, possibly because knowing the automation had variable reliability led them to be more cautious overall. The trained individuals were more likely to unquestioningly accept the automation's judgments, which was generally the appropriate decision. This indicated that the training enabled users to capitalize on the workload savings provided by reliable decision aids, without a performance decrement.

Appropriate levels of trust and reliance are a function of the users' understanding of how the context affects the automation's capability. Training should focus on showing how situations interact with the characteristics of the automation to affect its capability (Lee & See, 2004). Comprehension of the automation's capability can be improved by revealing the automation's algorithms and operations.

Algorithms and Mental Model Development

When time does not permit adequate training or training is not available, presenting pertinent information about the automation is necessary. Presenting information graphically converts it to a form that is more easily processed by the user, aiding the necessary mental analysis and synthesis of the information (Bertoline & Wiebe, 2003). Information regarding the automation's capability can be defined in terms of detail and abstraction. Abstraction refers to information regarding the automation's performance, purpose, and process. Detail is the functional specificity of the trust. Detail specifies whether trust is focused on the mode of the automation, the automation as a whole, or a group of automated components (Lee & See, 2004). The automation's algorithm is capable of conveying this particular information and facilitating the development of an appropriate mental model. If this information is not available in the user interface or if it is not properly formatted, trust may not develop appropriately. Algorithms have the greatest utility in instances where problems are highly structured and well defined. Unfortunately, it is often the case that no algorithm exists or the algorithm is so cumbersome that it would be of no real help to use it.

Leahey & Harris (2001) define algorithms as strategies used in problem solving that are guaranteed to produce a solution. The utility of presenting a system's algorithm can be questioned when the algorithm is too complex and requires a great deal of cognitive effort to understand and to learn. It can also lack utility if the information presented is not potentially relevant to the task or if the underlying processes are complex and difficult to illustrate (Guerlain, et al., 2002).

An algorithm, or conceptual model, of a system is created to provide an appropriate representation of the target system. Conceptual models are created with the intention of being accurate, consistent, and complete (Norman, 1983). Conceptual models are created as tools for the understanding or teaching of physical systems. Mental models are what people have in their heads and what guide their use of a system. Ideally, there ought to be a direct relationship between the conceptual model and the mental model (Norman, 1983).

Norman (1983) characterizes mental models as naturally evolving models. Mental models enable users to predict the operations of a particular system. Reeves (1999) define mental models as a version of a knowledge structure, found in long-term memory that contains the understanding of a topic or procedure. In user-centered design, one of the key design and performance goals is to facilitate the development of a mental model of how the system or application performs. Through interaction with a system, users formulate mental models of that system. The models do not have to be technically accurate, but they must be functional. A person's mental model will continue to be modified as they interact with the system in order to get a workable model. Mental models are limited by the user's technical background, previous experience with similar systems, and the structure of the human information processing system. Norman (1983) reported a few general observations about mental models:

1. Mental models are incomplete.
2. User's abilities to "run" their models are severely limited.
3. Mental models are not stable. People forget the details of the system they are using.
4. Mental models do not have clear boundaries: similar mechanisms and operations get confused with one another.

5. Mental models are not "scientific": People maintain behavior patterns, or heuristics, even though they know they are used because they require little physical effort and save mental effort.
6. Mental models are parsimonious: Oftentimes, users do extra physical operations rather than the mental planning that would allow them to avoid those actions.

People's understanding of the devices and systems they interact with is surprisingly inadequate, imprecise, and full of inconsistencies, gaps, and personal quirks. Users often feel uncertain of their own knowledge – even when it is correct and complete – and their mental models include information about the degree of certainty they feel for their knowledge. A person's mental model can include heuristics even if they make no sense (Norman, 1983). Heuristics are problem-solving procedures that often work in solving everyday problems. A heuristic is a rule-of-thumb or guideline for coming up with a solution (Best, 1989).

Learning and Mental Model Development

An interface can be a gateway to learning or a severe barrier (Reeves, 1999). The layout of information determines the ease with which the human visual system can navigate around the display. Edward Tufte (1990, 1997) is known for using information design to convey ideas about things that are not materially in our presence. This is achieved by inducing (in graphical form) an appropriate mental representation of the ideas and concepts. Effective design represents concepts or objects by supplying only essential elements that are most characteristic of the concepts or objects.

Individuals' knowledge structures change with learning and experience. As people become more familiar with a particular domain, their knowledge representation changes in systematic ways to provide better support for their actions (McDougall, Curry, & Bruijn, 2001). Such changes have been seen in the computer software field, among others (Adelson, 1981). The extent of convergence of knowledge structures may depend on the individual's perspective on learning. Previous research has shown that if individuals experience the same problem-solving task via different visual interfaces, the knowledge structures may develop systematically different as a result of the different perspective one gets from the interface (McDougall, et al., 2001).

Design that is focused on the user, is concerned with the active and explicit construction of stable, long-term mental models of topics or procedures that may or may not involve representations of images (Reeves, 1999). Learning and the development of mental models has been studied extensively in the context of work on expert knowledge. As one becomes more competent in a particular domain, one develops a richer model of the subject. A key difference between the way an expert and a novice reason is found in the structure of their mental models. Novices' mental models represent objects and concepts as they are seen in the real world, whereas experts construct more abstract models (Reeves, 1999).

There are three kinds of learning one might engage in while gaining an understanding of a new system. The first type of learning involves establishing a connection between the structure of the device, in this case automation, and its function. The second type entails making the structure-function connection more robust by making implicit assumptions explicit. Finally, the last is the "caching" or the storing, of the results of the projection problem solving on the inherent mechanism (de Kleer & Brown, 1983).

According to de Kleer & Brown (1983), a user trying to acquire a deep understanding of automation will be likely to employ all three types of learning. Early in the learning process the user is given a collection of component models and from there synthesis takes place, connecting the structure and function of the automation. Producing this connection is also known as “constructing a mechanistic mental model of the device.” A user’s preliminary mental model includes many implicit assumptions that may or may not be correct. As the learning process progresses, the user identifies and makes explicit many of the assumptions. Progress is made by discovering violations of the consistency, correspondence, and robustness. Violations to the robustness may be encountered in situations where the mental model fails to explain the system’s behavior (when the system is experiencing a fault or causality), or by observing the results of a hypothetical modification to the model that results in contradictory results. Every time violations of this type take place, the user has a chance to identify an underlying or implicit assumption in one of his mental models. The thrust for the second type of learning is to increase the robustness of the mental model by making the implicit assumptions explicit.

As the component models of a machine get stripped of their implicit assumptions about the overall functioning of the automation, the problem-solving work performed by projection increases. A component model is a description of the input-output behavior of a particular component. The third form of learning concerns a technique for preserving this “work” so that it can be evoked only when needed and otherwise remain transparent. In other words, the user can cache the results of projection (specifically problem-solving) by noting the aspects of the component models that were actually used in the automation’s causal model. From this knowledge the user can create new distinctive component models, which make explicit just those aspects, thereby eliminating the development of any ambiguities. Using only the distinctive component models, the user can repeatedly predict the automation extremely efficiently. The distinctive models can be linked to their embedded assumptions, which act as caveats, whereas the non-robust models do not articulate assumptions. If the caveats are violated, then the user calls upon the original models (de Kleer & Brown, 1983).

Visual Representation of Information

Legrenzi & Girotto (1996) concluded that it is critical for information designers to focus on design principles that help build the desired representation of an interface, module or information architecture in such a way that users understand it from first interaction. Text has the power to focus or defocus the user's attention toward the more accurate representation. It would be advantageous of designers to create representations or textual content that do not generate more than one competing model of an interaction or understanding of the situation. The result of competing models is increased uncertainty and complexity for the user. Similarly, text-based displays that only show raw data values in numerical format force the user to remember the set points and relationships of interest, make a comparison of the values that may or may not be displayed, and perform mental calculations to determine the necessary information. This is an example of cognitive load obstructing representational development.

Schnotz, Bennert, & Seufert (2002) assert that text and pictures can be seen as complementary sources of information and users can choose to focus on one source more than the other. One important issue is whether the use of verbal and pictorial information follows the principles of cognitive economy. According to this principle, users attempt to construct a mental image that sufficiently handles tasks with minimal cognitive effort. Constructing a mental model from a diagram requires less effort than constructing a mental model from text, because text comprehension requires a transfer from a symbolic propositional representation to an analog representation. However, there may be differences with regard to the ease of interpretation of diagrams depending upon its complexity.

The presentation of information in more than one medium can help individuals who have trouble encoding information from either text or diagrams alone. Many types of information can be characterized in more than one way. If two representations contain identical information, they may be considered informationally equivalent. However, extracting the information from the two media requires different types of processes and so they are not equivalent (Larkin & Simon, 1987). When text is accompanied by a diagram, the comprehension process is more involved because the information that readers have to incorporate is presented in two different media. When text is supplemented by a diagram, the important information could be reactivated by inspecting the diagram rather than reading the text. A diagram might be particularly beneficial if the different pieces of information were presented in adjacent, or easily identified locations (Hegarty, et al., 1991). However, diagrams can be used to compensate for limitations to working memory only if the reader can process the diagram. In summary, diagrams may facilitate the development of a mental model because they lead to the formation of a more detailed, robust representation of the material presented to the individual (e.g., Glenberg & Langston, 1992).

When asked the question are pictures and words encoded in a single underlying mental representation or separate memory systems, Paivio (1986) proposes a dual coding model. The model demonstrates the existence of a verbal system specialized for processing and storing linguistic information and a separate nonverbal system for spatial information and mental imagery. The two systems can operate independently, but they are also connected so that, for example, a person looking at a picture might verbalize its description. When the results of research on the memory for pictures is compared with memory for verbal stimuli, the recurrent finding is that memory for pictures is better than memory for words, an outcome known as the "pictorial superiority effect" (Levie, 1987). Paivio argues that dual coding is more likely to take place with pictures than words, and since two memory traces are better than one, dual coding justifies the pictorial superiority effect. Haber & Myers (1982) demonstrated that the memory for a picture-word combination was superior to memory for text alone or pictures alone.

Charts, graphs and diagrams reside at the center of the continuum that extends from pictures to words. Charts, graphs, and diagrams comprise a family of graphic forms that share the common attributes of abstraction and the exploitation of space. The current research focuses on diagrams, more specifically flow diagrams, which are different than graphs and charts in function and complexity (Winn, 1987). Diagrams describe whole structures and processes often at levels of high complexity. Diagrams explain sequences by illustrating sequences of actions, rules, and chains of reasoning. Schematic diagrams depict very abstract concepts and rely on conventions that depict both the components and their organization (Hegarty, et al., 1991). In this instance, the automation's algorithm, or flow diagram, is depicted as a schematic diagram.

Many tasks involve a sequence of operations that must be performed in a specific order. While these orders can be described by a textual list, it is also possible to represent them visually so that the spatial position corresponds to the sequence of actions. The best example of this is the flow diagram or flowchart (Haber & Wilkinson, 1982). Mapping between visual representations and a user's mental model is essential in the comprehension and effective use of algorithms. A learnable image that portrays the underlying system or database is necessary for a user to have a starting point in the organization of a visual mental map (Sutcliffe, Ennis, & Hu, 2000). Bennett & Flach (1992) assert that performance can be improved by providing displays that allow the user to make use of the more efficient processes of perception and pattern recognition instead of requiring the user to utilize the cognitively taxing processes of memory, integration, and interference (Rasmussen & Vincente, 1989; Vincente & Rasmussen, 1990; Woods, 1991). One method of accomplishing this is by collecting and combining information in centralized displays.

Haber & Wilkinson (1982) assert that programmers begin with some type of flowchart. Flowcharts can be used to learn structure that is unsuspected or unknown. Programmers typically begin with a flowchart of the task performed by the program. The steps in the flowchart are then translated into code. The program may fail to operate for several reasons, including failure to follow the proper syntax of the programming language, failure to follow the flowchart in creating the instructions, or mistakes in the flowchart itself that prevent the program from serving its intended purpose (Haber & Wilkinson, 1982). In a related experiment, subjects were asked to learn the recipe for a sponge cake. Half saw it in the traditional cookbook format; while the other half saw it in a flowchart. After studying the recipe, the subjects were asked to recall the ingredients, to recall the sequence of steps, and to recognize the correct or incorrect procedures. On all measures, the group of subjects who reviewed the visual flowchart did better than those that looked at the set of verbal instructions, such as those given over the telephone.

Guerlain et al., (2002) maintain that a major strength of graphical representations of data as opposed to text-based displays is that many relationships can be conveyed directly using visual properties of the display. This allows access to embedded knowledge through perception of the display, rather than doing mental calculations to deduce the desired information. Visual representations are also useful in the preservation of spatial, topological, and geometric properties that are important for certain kinds of tasks. When trying to understand the representational benefits of a diagram, Larkin and Simon (1987) asserted that object properties could be indexed by their location, rather than by an explicit label. Additionally, a number of properties can be represented simultaneously in the same location, when these properties may all be relevant to the problem (Guerlain, et al., 2002).

Visual information can stimulate learning by presenting more than just data. Information is cognitively usable when presented as process, form, relationships, and functionality. Instead of appearing disjointed, information appears as a unified whole, spanning time and space, depicting cause-and-effect, and even visualizing abstract concepts. Information presented visually allows users to better comprehend complex ideas.

Impact of Automation Purpose, Process, and Performance on User Trust

The previously defined dimensions of purpose, process, and performance provide a set of distinctions that describe the basis of trust. These dimensions classify three types of goal-oriented information that contribute to developing an appropriate level of trust (Lee & See, 2004). Early in the relationship with automation, there may be little history of performance, but there may be an obvious statement of purpose. Therefore, early in the relationship trust can depend on the automation's purpose and not its performance. The particular evolution of the relationship depends on the type of information provided by the human-computer interface and the training and documentation (Lee & See, 2004). Observing a system's performance can support presumptions regarding internal mechanisms associated with the dimension of process and analysis of internal mechanisms via an algorithm, can support inferences regarding the designer's intent associated with the dimensions of purpose. Similarly, knowledge of the designer's intent can reveal the underlying process, and performance can be estimated from comprehension of the underlying process. If these inferences support trust, then the system design more likely facilitates an appropriate level of trust. If inferences are not in agreement with observations, then trust will probably suffer because of a weak correspondence with the observed agent. Trust that is established on understanding the agent's motives will be less fragile than trust based only on the reliability of the agent's performance (Rempel et al., 1985).

Trust and affect are in fact two separate constructs. Beliefs and affect about another are reflected in one's trust for another (Holmes & Rempel, 1986). In other words, one's positive affect often derives from the process of attaching meaning to another's positive behavior. This in turn has a positive impact on one's trust in another. Because user trust has a significant impact on automation reliance, automation designers should design automation in ways that facilitate trust. The design goals should not be greater trust, but more importantly, appropriate trust. Making the agent's motive's transparent to the user may be one way to achieve appropriate trust and reliance.

As the automation of aspects of IT becomes more prevalent and the shift of control moves from the user to the IT, there is an increasing need to improve the relationship between the two. Trust in automation is critical, because of the interpersonal nature of the relationship between humans and automation. Research has shown that humans respond to automation as they would to other human collaborators (Reeves & Nass, 1996). However, their expectations of near-perfect performance from automation are unrealistically high (Smith & Graesser,1981; Ruble & Stangor, 1986). Because of the unusual nature of this human-automation relationship, it is necessary to find ways to ensure its success. Developing appropriate levels of trust in the automation are necessary. One way to improve a user's trust in automation is through the development of a suitable mental model of the automation and its algorithm. There are a number of ways in which the automation's algorithm can be presented, including text or graphic. The modality that lends to a more appropriate mental model and appropriate usage pattern remains to be seen.

A more robust mental model of the automation or the decision aid's (DA) algorithm will lead to higher levels of trust in the DA. Higher affective ratings of trust in the DA and more appropriate use of the DA will be seen as a result. The present research is designed to investigate the following hypotheses:

1. The DA interface that uses a graphical (diagrammatic) representation of the automation's algorithm will generate higher subjective ratings of trust than the interface using a textual description. The textual representation, in turn, will show higher ratings than no representation of the algorithm.

2. The DA interface that uses the graphical (diagrammatic) representation of the automation's algorithm will result in significantly higher rates of appropriate use than the interface using a textual description. The textual representation, in turn, will show higher rate of appropriate use than no representation of the algorithm. Appropriate use is exhibited by choosing to rely on one's own decisions instead of the automation's decisions in instances where it is not correct and choosing to rely on the automation's decisions in instances when it is correct.

Method

Participants

North Carolina State University undergraduate Psychology students were used as participants. The experiment was posted in the psychology experiment database. Students enrolled in the Introduction to Psychology course participated. All participants received course credit that will fulfill a course requirement. In addition to course credit, participants were told that individuals with the highest scores were entered in a raffle to win a prize valued at one hundred dollars. A power analysis was conducted to determine the appropriate number of participants. The value for sigma, the common standard deviation for both populations, used to calculate power was determined from a preliminary study using similar procedures. A power value of .80 was used. It took a sample size of 90 (15 per cell) to reasonably detect a treatment effect.

Materials

A standard laptop computer with the Windows operating system on it was used in this study. A signal detection task was performed on the computer with the assistance of an automated decision aid. The theory of signal detection is that nearly all reasoning and decision-making takes place in the presence of some uncertainty (Heeger, 2003). The general approach of signal detection theory has direct application in sensory experiments.

Policity

Policity is a city simulation game where a policy set is used to manage the city. The version of Policity used in this research is the same as in previous experiments (Campbell, C.S., Kandogan, E., November, A., Barrett, R., Maglio, P.P., 2005), except that the policies are presented as DA suggestions that participants could execute if they chose to do so (see Figure 2). This condition shows the DA providing a suggestion using a textual algorithm.

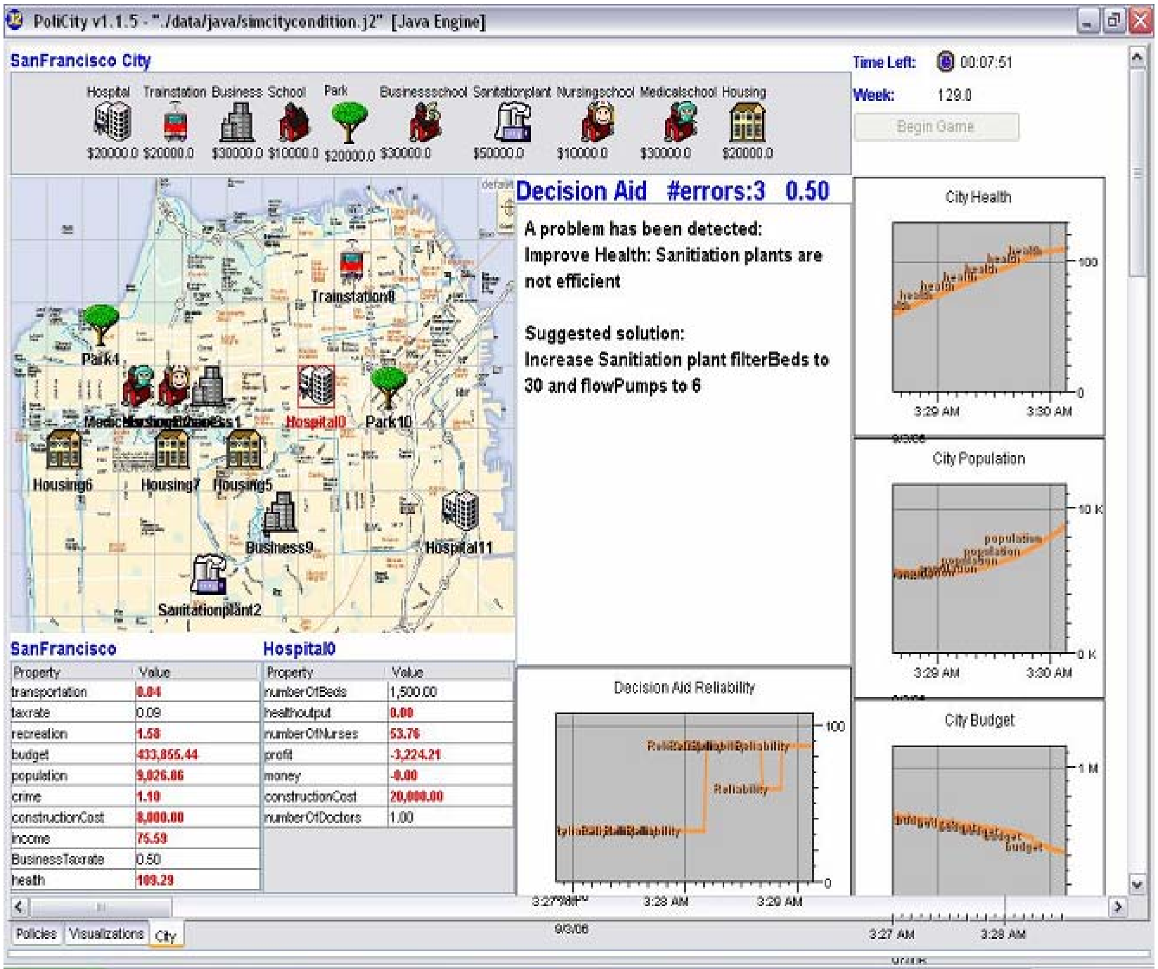

Figure 2. Policity user interface showing the city and buildings (top-left) in San Francisco, the Decision Aid (center), City and a selected Hospital properties (bottom-left), and city properties (right).

Policity was developed to resemble distributed systems where different types of processes can be added and configured, working together to provide a service. In this case, the processes were facilities of 10 different types – businesses, business schools, hospitals, housing, medical schools, nursing schools, parks, sanitation plants, schools, and train stations. Each facility type had adjustable properties and other properties that showed the facilities' status. For example, hospitals had properties like *number of beds*, *nurses*, and *doctors*. The *number of beds* property was adjustable, but the *nurses* property was purely informational. The city had a total of 60 unique properties across all facility types that interacted in various ways to create a unique set of dynamics. The city itself also had its own set of properties. Two of these properties (i.e., *individual tax rate* and *business tax rate*) were adjustable. The remaining seven showed the state of other aspects of the city (i.e., budget, crime, health, personal income, population, recreation, and transportation). On the left side of the game's interface, a map displayed the physical layout of San Francisco city. As buildings and facilities were added, icons representing them were randomly placed on the map. The exact location of facilities had no direct effect on the city or on other buildings. On the right side of the game's interface were three dynamic graphical displays of the city's budget, health, and population.

Policity has a standard GUI-based interaction in which participants can click on objects to act on them and select text boxes to edit property values. Figure 2 shows the user interface with the DA embedded in the center panel. The top-left menu bar provides a way to add facilities to the city and includes the initial purchase price for each facility. Every game started with no facilities on the map. Once added to the map, clicking on a facility displays that building's properties in the lower property-box. Figure 2 shows that a hospital has been selected. The properties of that hospital (i.e., Hospital0) are shown in the lower property-box. If participants clicked on an editable property in the lower property-box, the cursor would gain focus and allow the participants to edit the value.

The DA is shown in the center of the user interface in Figure 2. Figure 3 shows the three versions of the DA's algorithm that were displayed based on experimental condition (i.e., no algorithm (right), graphical (center), and textual (left)). The results of the experiment are presented below. Preliminary pilot testing revealed that a graphic-only depiction of the decision aid's algorithm was incomprehensible without the accompanying suggestion (as described below). Participants reported that there was not enough information available in the graphic alone to provide direction on what to do next. The textual algorithm included an additional suggestion that was an explicit statement on what to add or edit in the city. This same suggestion was added to the graphical algorithm interface and was found to make the algorithm more comprehensible. This information was presented in the same location as it was in the textual algorithm conditions. Therefore, any references of the graphical conditions from this point forward include a textual component.

The DA sampled the city's condition once every 1000 milliseconds to search for a problem. The DA suggestions had a problem (conditional) component and a solution (action) component. In the textual and graphical algorithm conditions, the problem represented the algorithm the DA used to determine what was wrong with the city. The graphical algorithm also provided a diagrammatic representation of the problem. Figure 3 shows the relationships among the properties of the city and its facilities represented as ovals and arrows. The health of the city, which is the target for improvement, is colored gold to identify that it is the target property. The properties that need adjusting to alleviate the problem are colored green to indicate that they need to be increased and red if they had to be decreased. For the purposes of this experiment, diagrammatic algorithm and graphical algorithm were used interchangeably. The algorithm had to be complex enough to warrant the use of a diagram. The complexity of the algorithm was verified during pilot testing. The DA in the no algorithm conditions did not provide a representation of the algorithm it used.

The DA's suggestions were the original set of policies used in previous experiments (Campbell et al., 2005). There were a total of 18 different policies that were created to address specific problems in the city (see the list below) to some extent. Every policy was thoroughly tested independent of the others to verify that each provided a benefit to the city in at least one situation or another. For example, the policy to improve the city's budget only does so if certain buildings (i.e., businesses) have been added to the city. Each policy was also tested to ensure that none were "super" powerful. The total list of policies includes rules that require adding every building type available, as well as, making adjustments to facilities and city properties.

There were 18 policies across five different tasks:

1. Improve Health (9 Policies)
2. Improve Population (2 Policies)
3. Improve Budget (5 Policies)
4. Improve Income (1 Policy)
5. Improve Transportation (1 Policy)

The reliability of the DA was displayed in the title bar as the number of errors and the percent correct suggestions. Reliability was also presented in a graph below the DA (see Figure 2) as a percentage between 0 and 100. The DA was characterized as having high reliability or low reliability. The reliability was manipulated based on condition and was determined during pilot testing. In the high reliability conditions, there was a 75% chance a high reliability suggestion being chosen for each presentation. For the low reliability conditions, there was a 25% chance of a high reliability suggestion being chosen. Results from Dzindolet, et. al.(2000), reveal that the more types of reliability feedback that participants received, the more likely they were to rely on a superior automated aid. Continual feedback of the DA's reliability was presented as the number of errors it makes during a trial and was presented in the interface as text above the DA. The number of DA errors was presented as well as the respective proportional value of these errors.

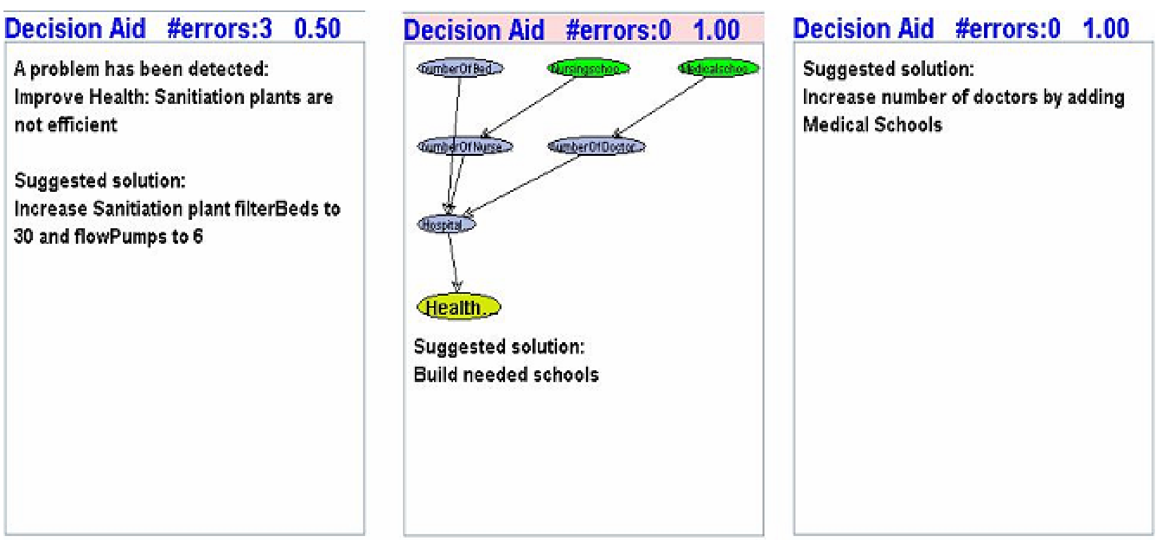

Figure 3. Decision Aid Algorithm Representations for Textual (left), Graphical (center), and No algorithm (right)

Treatment conditions

Participants were randomly assigned to one of the 3 between-subject (Algorithm presentation: None, Text, or Graphic) X 2 between-subject (Automation reliability: Low or High) conditions shown in Table 3.

Table 3

Experimental Conditions

	Algorithm Presentation		
Automation Reliability	None	Text	Graphic
Low	*None-Low*	*Text-Low*	*Graphic-Low*
High	*None-High*	*Text-High*	*Graphic-High*

Procedure

Participants were told that they were recently elected the mayor of San Francisco and were asked to manage aspects of a simulated city using Policity. Participants were told that an automated decision aid (DA) would be assisting them in maximizing the city's health. The experimenter explained that decisions made interacted with one another and every decision would have subsequent implications on the outcome of the city. They were also told to use as little of the city's money as possible to accomplish this goal. Participants were also told that the DA uses a predefined algorithm or problem solving procedure to suggest the correct answer. The DA assisted participants by providing suggestions on how to adjust the city's resources or tax rates. The same algorithm was used in the low and high reliability conditions.

After reviewing and signing a consent form, participants were asked to sit in front of the laptop computer. The experimenter read aloud the on-screen instructions.

Participants were given 2 practice trials. The DA was 100% reliable during the practice trials. This meant that every suggestion was the best course of action for maximizing the city's health. Once the practice trials ended, participants were told they successfully completed them. They completed the HCT questionnaire that investigated their perceptions of the automation, their perception of the relationship they may have with the automation and their inclination to trust. Upon completion of the questionnaire, they proceeded with the experimental trials. Depending on experimental condition, participants were shown the DA's algorithm in a textual format, a graphical format, or not at all. While completing the practice and experimental trials, participants saw the DA's errors. This information was presented as a means of conveying the automation's reliability. The DA's suggestions were presented throughout the practice and experimental trials. Each suggestion was displayed for 30 seconds, giving them enough time to read the suggestion (should take about 10 seconds to read the text or interpret the graph) and to make a decision to act on the suggestion. This time was also chosen because it ensured that subjects would see nearly all of the 18 different suggestions during each session.

After completing half of the experimental trials, they completed the HCT a second time. A comparison of differences in subjective ratings on the two questionnaires was analyzed to find potential group differences. Participants completed the remaining experimental trials. After completing all of the experimental trials, participants in the algorithm conditions were asked to report how they decided when to use the DA, how they would improve the DA, and ways in which they would improve the DA's algorithm. Participants that did not see the DA's algorithm were asked to report how they decided when to use the DA, how they would improve the DA, and how they reasoned through the task. All participants were asked to complete a demographic questionnaire.

Dependant Variables

Data were collected on subjective measures using questionnaires. Several subjective rating scales measuring human trust in automation were examined. Jian, Bisantz, and Drury (2000) developed a twelve-item trust scale that was developed as part of a three-phased experimental study. A word elicitation study, where various words related to concepts of trust and distrust, was conducted during the first phase. A questionnaire study was conducted during the second phase. It investigated how closely each of the words was related to trust or distrust. Participants took part in a paired comparison study where they rated the similarity of pairs of words during the third phase. Data from the questionnaire study and the paired comparison study were used to create a multi-dimensional measurement scale for trust.

The SHAPE Automation Trust Index (SATI) was developed to measure different aspects and elements of trust in real-time air traffic management (ATM) simulations. It was determined that the SATI could be used to measure trust in any given real-time automation simulation. Construct validity of the SATI was assessed by subject-matter experts (SMEs).

Madsen and Gregor (2000) developed a measure of trust of computers that was based on earlier work by Rempel, et al. (1985), Sheridan (1988), Muir & Moray (1996) and others. The measure was named the Human-Computer Trust (HCT) scale. It consists of five main constructs each with five sub-items. The five items were extracted from an original list of ten trust constructs because they were found to have the highest predictive validity. Madsen and Gregor claim the HCT has been empirically proven to be valid and reliable. Cronbach's alpha was used to measure the reliability of the HCT and was found to be 0.94. The HCT assumes that the user may not know the underlying technology or the specifics of how the system formulates its output (i.e., its algorithm). It also assumes that the users may or may not be experts in the specific task domain (Madsen & Gregor, 2000).

For the purposes of the current research, Madsen and Gregor's HCT scale was used to collect subjective measurements of trust in automation before and after the experimental trials. The HCT scale was chosen over the SATI and other questionnaires because it measures the five main constructs of trust (i.e., perceived reliability, perceived technical competence, perceived understandability, faith, and personal attachment). A five point Likert-type scale where 1 = Strongly Disagree to 5 = Strongly Agree was used. Responses on the HCT scale permit analyses on the five items (i.e., perceived reliability, perceived technical competence, perceived understandability, faith, and personal attachment) that help to comprise the concept of trust as shown in Appendix A. For the purposes of this research, the word *system* was replaced with *decision aid* on the HCT, so that we could capture users' ratings of the DA only and not the entire game.

To determine their understanding of the automation and algorithm complexity, at the end of the experiment, participants were asked open-ended questions regarding their reported usage of the DA and suggestions for improvements to the DA and its algorithm, as shown in Appendix B. Subjective data were collected after the practice trials and halfway through the experimental trials as shown in Table 4.

Table 4

Experimental Design

Practice Block	**Test 1A** (HCT)	Experimental Block 1	**Test 1B** (HCT)	Experimental Block 2	**Test 2** (Subjective measure of complexity)

Objective data on automation *usage* was measured by observing those instances when participants rely on the DA and whether it is providing correct or incorrect suggestions and when they choose not to rely on the DA and whether or not it is correct in those instances as shown in Table 5.

Table 5

Users' Patterns of Decision Aid Usage

	Decision Aid's Suggestion	
User Complies with DA	Correct	Incorrect
Yes	*Use*	*Misuse*
No	*Disuse*	*Use*

Preliminary pilot testing was performed which resulted in a number of considerations included in the current research. Ten minutes of training was determined to be sufficient for participants to learn how to use the game, the information in the interface, and to develop an understanding that each decision had an impact on the city's health and budget. During pilot testing it was found that participants needed an opportunity to ask questions before proceeding to the experimental trials. Therefore, after 5 minutes of practice they were given the opportunity to ask questions. Giving them the opportunity to ask questions and then proceed with the practice trials allowed them to better understand any questions that they had as it relates to the game's interface and playing the game. Pilot tests also revealed that a graphic-only algorithm was not sufficient in providing the same level of detail as the textual algorithm. The textual and no algorithm conditions were developed with a suggestive component included. It was found that the graphical algorithm condition should also include the suggestive component. The suggestion provided information on the specific value to change various attributes of the facilities or the city.

Results

Ninety individuals participated in this study. Fifty-three of the 90 participants (59%) were males. The median age of the participants was 19. Full-time students made up 91% of the participants (n = 82), with an average 12.9 years of school completed. Participants varied greatly in their video game experience. Participants were grouped into 3 categories of video game experience as shown in Table 6.

Table 6

Video Game Experience

Experience	n	%
None	20	22.2
1-4 yrs	16	17.8
>4 yrs	54	60

One participant had as many as 21 years of experience playing video games. Thirty-eight participants reported that they did not play video games on a regular basis. Thirty-two participants played video games an average of 0.5 to 3 hours per week. Of the 90 participants, 10 reported playing video games 4 to 8 hours per week. Six of the participants play video games 10-15 hours per week. Three of the participants played video games more than 20 hours per week. Forty-seven percent of the participants have played the Simcity video game before (n = 42).

The ethnicities of the participants are shown in Table 7.

Table 7

Ethnicity/Race of Participants

Ethnicity	n	%
Caucasian	63	70
African-American	12	13
Asian	6	7
Mixed race	3	3
African	2	2
Latino	2	2
East Indian	1	1
Middle Eastern	1	1

To investigate group differences between the change in the average reported trust ratings of the DA's algorithm presentation modality groups and the reliability groups, two-way analysis of variances (ANOVAs) were computed with algorithm presentation (none, textual, and graphical) and reliability (low, high) as between-subjects factors and the difference in average reported trust in the decision aid after practicing with the game for 10 minutes (Practice Block) and after a 10 minute experimental trial (Experimental Block 1), as dependant variables. Even though the two-way ANOVA captures the essence of the experimental design, it does not factor in demographic information. Participants provided demographic information including their educational status (i.e., full- or part-time student), highest level of education completed, overall years of video game experience, amount of experience with the SimCity video game, number of hours spent playing video games per week, age, gender, and ethnicity. A backward elimination process was used in order to minimize the effects of selected confounding factors. All possible covariates were included and then sequentially omitted beginning with the covariate with the highest p-value. There were no confounding factors identified from this process for each of the analyses.

Analyses were conducted on scores reported on the Human-Computer Trust Scale (HCT) at time 1 and time 2 for each of the five main constructs of the HCT (i.e., Perceived Reliability, Perceived Technical Competence, Perceived Understandability, Faith, and Personal Attachment).

HCT Questionnaire Responses

Perceived Reliability (R)

A 3 (None/Textual/Graphical) X 2 (Low/High) ANOVA with the difference in the average reported perceived reliability (R) of the DA from time 1 to time 2, as the dependent variable did not yield any significant interactions. A significant main effect for reliability level [F(1, 84) = 3.94, p = .05] was found. Reported perceived reliability of the DA decreased at a significantly greater rate for participants in the low reliability conditions (M = -.64) than in the higher reliability conditions (M = -.34). Table 8 shows the mean subjective ratings at time 1, time 2, and the change from time 1 to time 2 for the perceived reliability (R) of the DA for participants in the high and low reliability by algorithm modality conditions.

Table 8

Change in Perceived Reliability (R) of the Decision Aid for High and Low Reliability Groups by Algorithm Modality from Time 1 to Time 2

	Low Reliability			High Reliability		
Algorithm Modality	M_1(SD)	M_2(SD)	☚M(SD)	M_1(SD)	M_2(SD)	☚M(SD)
None	3.26(.59)	2.41(.49)	-.85(.45)	3.32(.63)	2.84(.62)	-.48(.70)
Text	3.59(.60)	3.07(.64)	-.52(.63)	3.04(.69)	2.80(.91)	-.24(1.1)
Graphic	3.29(.68)	2.75(.78)	-.55(.72)	3.05(.49)	2.76(.55)	-.29(.64)

Note. n = 15 for each condition

Faith (F)

A 3 (None/Textual/Graphical) X 2 (Low/High) ANOVA with the difference in the average reported faith (F) in the DA from time1 to time 2, as the dependant variable yielded no significant interactions or main effects for algorithm modality condition and DA reliability condition. The mean subjective ratings at time 1 and time 2, as well as, the change from time 1 to time 2, for participants in the low and high reliability by algorithm modality conditions are shown in Table 9.

Table 9

Change in Faith (F) in the Decision Aid for High and Low Reliability Groups by Algorithm Modality from Time 1 to Time 2

	Low Reliability			High Reliability		
Algorithm Modality	M_1(SD)	M_2(SD)	☚M(SD)	M_1(SD)	M_2(SD)	☚M(SD)
None	2.81(.97)	2.29(.64)	-.51(.77)	2.73(.56)	2.37(.65)	-.36(.70)
Text	3.29(.80)	2.71(1.0)	-.58(.73)	2.77(.82)	2.39(.99)	-.39(.79)
Graphic	2.67(.83)	2.40(.95)	-.27(.54)	2.69(.75)	2.49(.54)	-.20(.49)

Note. n = 15 for each condition

Perceived Technical Competence (T)

A 3 (None/Textual/Graphical) X 2 (Low/High) ANOVA with the difference in the average reported perceived technical competence (T) of the DA from time 1 to time 2, as the dependant variable did not yield any significant interactions or main effects for algorithm modality condition and DA reliability condition. Table 10 shows the mean subjective ratings for time 1, time 2, and the change from time 1 to time 2 for participants in the low and high reliability by algorithm modality conditions.

Table 10

Change in Perceived Technical Competence (T) of the Decision Aid for High and Low Reliability Groups by Algorithm Modality from Time 1 to Time 2

	Low Reliability			High Reliability		
Algorithm Modality	M_1(SD)	M_2(SD)	☚M(SD)	M_1(SD)	M_2(SD)	☚M(SD)
None	3.15(.56)	2.59(.53)	-.56(.47)	3.32(.52)	2.83(.73)	-.49(.63)
Text	3.49(.63)	2.97(.81)	-.52(.59)	2.92(.70)	2.84(.94)	-.08(.82)
Graphic	3.32(.75)	2.92(.68)	-.43(.51)	3.05(.62)	2.63(.50)	-.43(.60)

Note. n = 15 for each condition

Perceived Understandability (U)

A 3 (None/Textual/Graphical) X 2 (Low/High) ANOVA with the difference in the average reported perceived understandability (U) of the DA from time 1 to time 2, as the dependant variable did not yield any significant interactions, but did yield a significant main effect for algorithm presentation modality [$F(2,84) = 3.66$, $p < .05$]. Reported perceived understandability of the DA decreased at the greatest rate for participants in the no algorithm conditions ($M = -.49$), followed by the graphical presentation conditions ($M = -.40$), and finally the textual presentation conditions ($M = -.05$). Post-hoc tests revealed a significant difference between the graphic and text conditions [$F(1,84) = 4.12$, $p < .05$]. A significant difference was also found between the text and no algorithm conditions [$F(1,84) = 6.58$, $p < .05$]. A significant difference was not found between the graphic and no algorithm conditions [$F(1, 84) = 0.29$, $p > .05$]. Table 11 shows the mean subjective ratings for time 1 and time 2, as well as, the change from time 1 to time 2 for participants in the high and low reliability by algorithm modality conditions.

Table 11

Change in Understandability (U) of the Decision Aid for High and Low Reliability Groups by Algorithm Modality from Time 1 to Time 2

	Low Reliability			High Reliability		
Algorithm Modality	M_1(SD)	M_2(SD)	☚M(SD)	M_1(SD)	M_2(SD)	☚M(SD)
None	3.35(.75)	2.68(.74)	-.67(.55)	3.29(.71)	2.97(.77)	-.32(.49)
Text	3.49(.90)	3.32(.89)	-.17(.72)	3.28(.77)	3.36(.93)	.08(.70)
Graphic	3.52(.82)	3.08(.93)	-.44(.87)	3.20(.67)	2.84(.58)	-.36(.66)

Note. n = 15 for each condition

Personal Attachment (P)

A 3 (None/Textual/Graphical) X 2 (Low/High) ANOVA with the difference in the average reported personal attachment (P) to the DA from time 1 to time 2, as the dependant variable did not yield any significant interactions. It did yield a significant main effect for algorithm presentation modality [$F(2,84) = 3.90$, $p < .05$]. Reported personal attachment to the DA decreased at the greatest rate for participants in the no algorithm conditions ($M = -.59$), followed by the textual presentation conditions ($M = -.47$), and finally the graphical presentation conditions ($M = -.13$). Post-hoc tests reveled a significant difference between the graphic and text conditions [$F(1,84) = 3.96$, $p = .05$]. A significant difference was also found between the graphic and no algorithm conditions [$F(1,84) = 7.24$, $p < .01$]. A significant difference was not found between the textual and no algorithm conditions [$F(1, 84) = 0.49$, $p >.05$]. The mean subjective ratings at time 1 and time 2, as well as, the change from time 1 to time 2, for participants in the low and high reliability by algorithm modality conditions are shown in Table 12.

Table 12

Change in Personal Attachment (P) to the Decision Aid for High and Low Reliability Groups by Algorithm Modality from Time 1 to Time 2

	Low Reliability			High Reliability		
Algorithm Modality	M_1(SD)	M_2(SD)	☛M(SD)	M_1(SD)	M_2(SD)	☛M(SD)
None	3.08(.74)	2.37(.84)	-.71(.55)	2.95(.78)	2.48(.72)	-.47(.78)
Text	3.17(1.0)	2.64(.99)	-.53(.76)	3.00(.90)	2.60(1.0)	-.33(.64)
Graphic	2.91(.93)	2.84(1.0)	-.07(.46)	2.72(.89)	2.53(.69)	-.19(.69)

Note. n = 15 for each condition

Open-ended Questionnaire Responses

Participants were asked to provide their opinions on 4 open-ended questions related to the DA, its algorithm, and their line of reasoning. Their responses were recorded as text passages. Unique labels were assigned to the text passages that contained references or interpretations of specific categories of information (Bernard, 1994; Miles and Huberman, 1994). Based on participants' responses, codes were created using an inductive process. Two reviewers went through the open-ended questions and compiled a list of codes for each question. As each new thought or idea was encountered, it was added to the list of codes.

Inter-rater reliability was conducted to ensure an unweighted observed Kappa value of at least 0.80 between the two raters. The Kappa value for the first question was 0.94. This question asked participants to explain how they decided when to use the DA. The percentages of respondents for each coded category are shown in Table 13. A Kappa value 0.98 was found for the second question shown in Table 14. In the second question, participants were asked to provide suggestions on ways to improve the DA. Only participants in conditions where the algorithm was presented in textual or graphical format were asked to provide suggestions on ways to improve the DA's algorithm. This question had a Kappa value of 0.97. The percentages of participants in each category are shown in Table 15. Participants in the conditions where no algorithm was present were asked to explain how they reasoned through the task. The Kappa value for this question is 0.95 and the percentages for each of the responses are shown in Table 16. The percentage of the total responses for each question is reported. The percentage of participants in each condition that provided a response to a particular question is also presented.

Table 13

Open-ended Responses to Decision Aid (DA) Usage in Percentages

		Algorithm Modality					
		None		Text		Graphic	
	Reliability	Low	High	Low	High	Low	High
	%	NL	NH	TL	TH	GL	GH
How did you decide when to use the decision aid ?							
1. Based on DA's characteristics	**32.7**	15.6	28.1	15.6	6.3	18.8	15.6
2. Based on the participant's ability or certainty	**25.5**	24.0	12.0	24.0	20.0	8.0	12.0
3. Based on characteristics of the city	**23.5**	13.0	17.4	17.4	13.0	30.4	8.7
4. Used it more frequently in the beginning	**9.2**	0	33.3	33.3	22 .2	0	11.1
5. Used it all the time	**4.1**	25.0	0	0	0	25.0	50.0
6. Never used it	**3.1**	66.7	0	0	0	0	33.3
7. Other	**2.0**	0	0	0	100	0	0

Note. % = Percentage of total responses; n = 90 for total percentage and n = 15 for each condition

Table 14

Open-ended Responses on How to Improve the Decision Aid (DA) in Percentages

		Algorithm Modality					
		None		Text		Graphic	
	Reliability	Low	High	Low	High	Low	High
How would you improve the DA?	%	NL	NH	TL	TH	GL	GH
1. Expose, change or include additional considerations of the city to its algorithm	**34.0**	19.4	22.6	16.1	9.7	6.5	25.8
2. Improve the DA's reliability or accuracy	**19.8**	16.7	5.6	22.2	27.8	16.7	11.1
3. Provide several suggestions	**14.3**	15.4	7.7	7.7	23.1	30.1	15.4
4. Improve the method to change things or improve the interface	**7.7**	14.3	28.6	0	14.3	42.9	0
5. No changes	**6.6**	0	33.3	50.0	0	16.7	0
6. Make the DA more clear/comprehensible	**5.5**	20.0	0	0	0	20.0	60.0
7. Response does not answer the question	**4.4**	50.0	0	25.0	25.0	0	0
8. Have the DA rank the importance of its suggestions	**3.3**	0	33.3	0	66.7	0	0
9. Have the algorithm expose or consider things to remove from the city	**1.1**	0	0	100	0	0	0
10. Remove the associated algorithm	**1.1**	0	0	0	0	100	0
11. Completely destroy or disable the DA	**1.1**	100	0	0	0	0	0
12. Other	**1.1**	0	100	0	0	0	0

Note. % = Percentage of total responses; n = 90 for total percentage and n = 15 for each condition

Table 15

Open-ended Responses on How to Improve the Decision Aid's (DA) Algorithm in Percentages

	Reliability	Algorithm Modality: Text Low	Text High	Graphic Low	Graphic High
How would you improve the DA's algorithm?	%	TL	TH	GL	GH
1. No suggestions	**50.0**	26.7	26.7	16.7	30.0
2. Provide better explanations on decisions suggested	**20.0**	33.3	16.7	16.7	33.3
3. Take time into consideration when making suggestions	**10.0**	16.7	50.0	33.3	0
4. Present the algorithm in a textual format	**6.7**	0	25.0	50.0	0
5. Improve the algorithm	**3.3**	50.0	0	50.0	0
6. Improve algorithm presentation in the interface	**3.3**	0	0	50.0	50.0
7. Provide suggestions on ways to improve the city's health without spending money	**3.3**	0	0	50.0	50.0
8. Provide multiple suggestions or alternatives	**1.7**	100	0	0	0
9. Provide suggestions from a monetary standpoint	**1.7**	0	0	100	0

Note. n = 60; % = Percentage of total responses; Only participants in the algorithm conditions (i.e., textual or graphic) provided responses to this question.

Table 16

Open-ended Responses to How they Reasoned Through the Task

		Algorithm Modality None	
	Reliability	Low	High
How did you reason through the task?	%	NL	NH
1. Used any information presented in the game's interface	**55.6**	60.0	40.0
2. Used their own logic	**19.4**	57.1	42.9
3. Considered what the DA suggested	**13.9**	60.0	40.0
4. Other, (including no response)	**11.1**	0	100

Note. n = 30; % = Percentage of total responses; Only participants in the no algorithm conditions (i.e., no algorithm) provided responses to this question.

Objective Data

Objective data was collected while participants were playing Policity. Data was collected on DA usage and non-DA patterns of usage. Analyses were performed on this data and the results are presented below.

The game was designed to provide approximately the same number of suggestions per session. The game was tuned before the experiment so that the suggestions would timeout giving each subject an opportunity to view almost every suggestion at least once during the experiment. Each participant had 2 experimental blocks. Figure 4 shows that in most cases, the number of suggestions presented for each block was between 21 and 30. The median value was 24. Several experimental blocks had more than 30 suggestions because the subjects were executing actions at a greater than average rate.

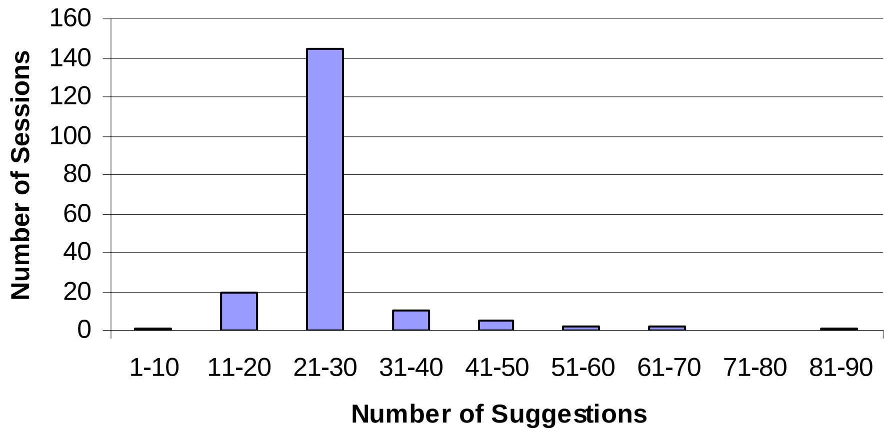

Figure 4. Number of Suggestions Provided by the Decision Aid for each Experimental Session

Figure 5 shows the mean number of DA suggestions that participants executed. To investigate group differences between the mean number of suggestions that participants executed, a two-way analysis of variance (ANOVA) was computed with algorithm presentation (None/Textual/Graphic) and reliability (Low/High) as between-subject factors. The two-way ANOVA yielded a significant interaction for reliability condition $F(1,121) = 3.70$, $p = .056$. Two-sample t-tests were conducted to compare reliability group differences. When comparing individuals in the graphical algorithm conditions, a significant effect for reliability was found, $t(60) = 1.95$, $p = .05$, where participants in the high reliability condition (M=8.77) executed a higher number of suggestion than participants in the low reliability condition (M=5.94). When comparing individuals in the textual algorithm and no algorithm conditions, a significant effect for reliability was not found.

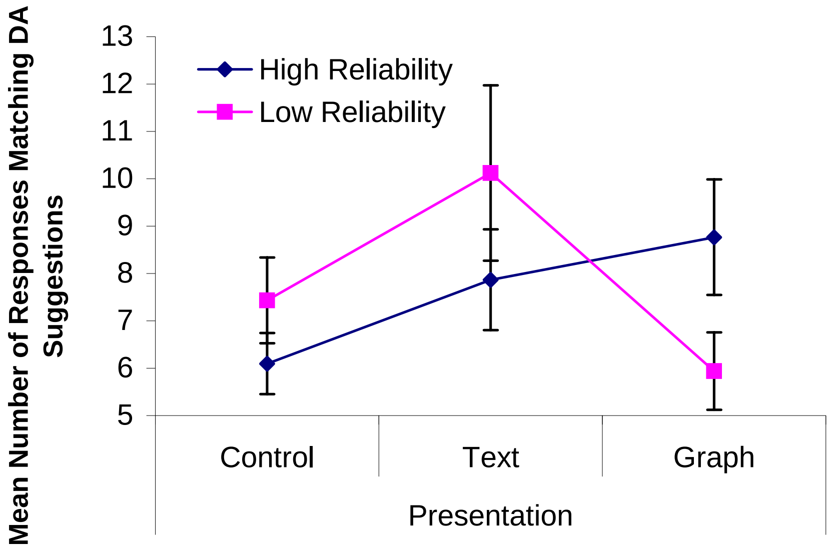

Figure 5. Mean Number of Responses Matching DA Suggestions for Participants in the High and Low Reliability Groups by Algorithm Modality

Participants' DA usage patterns and the resulting mean health levels of the city are shown in Figure 6. Individuals were categorized according to the number of times they chose to use the DA and the number of non-DA related actions that were observed. Low-trust individuals were those that used the DA's suggestions 5 or fewer times during the game. High-trust individuals took the DA's suggestions more than 5 times. Participants who had more than 22 non-DA related actions were characterized as having high manual action. Those with less than 22 had low manual action. The threshold values 5 and 22 were the median values of the distributions for number of suggestions (6) and number of non-DA actions (24). Table 17 shows the mean health levels for participants as they relate to the Parasuraman and Riley's (1997) characterizations of human-automation interaction. Participants who showed low levels of manual action when the DA was highly reliable (M=54.69) or who showed high levels of manual action when the DA had low reliability (M=62.50) showed appropriate usage patterns. Participants with a high rate of manual action when the DA was highly reliable underutilized, or disused, the DA (M=44.92). An over-reliance, or misuse, of the DA was seen when participants showed low levels of manual action when the DA's reliability was low (M=38.38). Participants who used the DA appropriately performed significantly better at managing the city than those who misused the DA, ($t(30) = 1.7$, $p < .05$), ($t(35) = 2.7$, $p < .05$). Participants that appropriately used the DA also performed better then those who disused the DA. However, this was only statistically significant for the high manual users ($t(32) = 1.3$, $p < .01$),but not the high DA users ($t(27)$, $p>.05$).

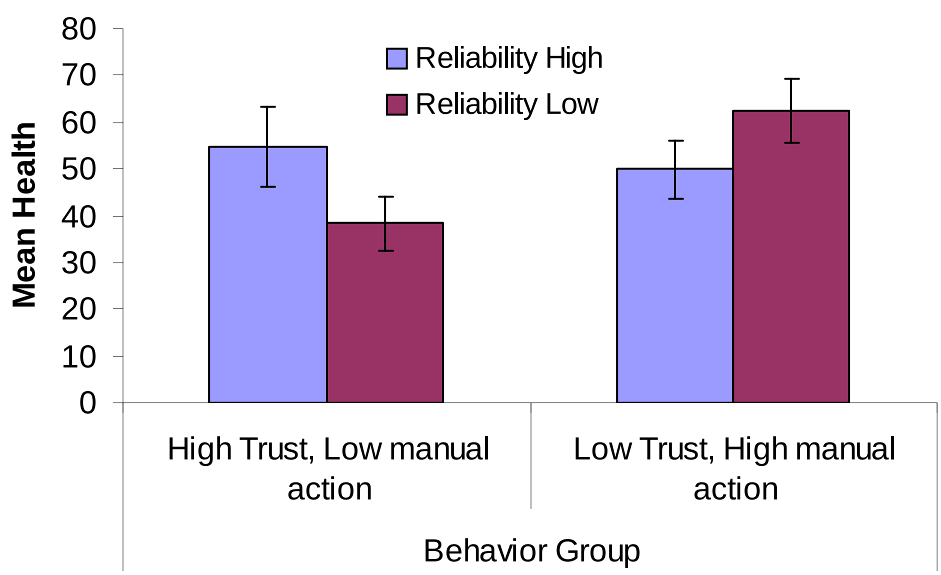

Figure 6. Mean Health Levels for Participants Based on DA Trust Levels and Usage Patterns for Low and High Reliability Conditions

Table 17

Mean Health Levels as They Relate to Characterizations of Usage Patterns

	Decision Aid's Suggestion	
User Complies with DA	Correct (high reliability)	Incorrect (low reliability)
Yes (low manual action)	54.69 *(Use)*	38.38 *(Misuse)*
No (high manual action)	44.92 *(Disuse)*	62.50 *(Use)*

Discussion

The following will discuss any demographic or behavioral aspects of the findings followed by a discussion of the subjective and objective findings. Subjective findings (i.e., reported ratings on the HCT) as they relate to whether the graphical algorithm produced higher subjective ratings than the textual algorithm and whether the textual algorithm produced higher results than not presenting the algorithm will be discussed. Objective findings as they relate to whether the graphical algorithm produced more appropriate usage patterns than the textual algorithm and whether the textual algorithm produced more appropriate usage patterns than presenting no algorithm will also be discussed. Finally, practical applications and implications on future research will be addressed.

The results of the present study showed that a large number of participants had previous video game and SimCity experience. Nearly 78% of the participants had more than 1 year of video game experience, with 60% of them possessing more than 4 years of experience. More than half (n = 51) played video games at least 30 minutes per week, with three playing more than 20 hours per week. Nearly 50% of participants have played Simcity in the past (n=42). Of the 90 participants, 59% of them were male. In a study by Lucas & Sherry (2004), female respondents of a large-scale survey (n = 534) reported less frequent play and motivation to play video games in social situations. It may be concluded that the reports of video game experience and amount of time spent playing video games in the present study, for the most part, could be attributed to the large number of male participants.

Past research has found that both affective and cognitive components of trust can be measured (Madsen & Gregor, 2000). A study by Madsen and Gregor (2000) found that the affective components were the strongest indicators of trust. Madsen and Gregor (2000) found perceived reliability to be a component of affect-based trust. The decline in the reported perceived reliability of the DA was significantly greater for participants in the low reliability conditions when compared to participants in the high reliability conditions. The decline in mean subjective ratings for participants in the low reliability conditions from time 1 to time 2 was nearly twice that of their counterparts in the respective high reliability conditions. This expected result showed that participants' perception of the DA's reliability was congruent with the actual changes in the DA's reliability. The greatest decreases in reported perceived reliability were in the no algorithm conditions. The next greatest levels of decline were found in the graphic algorithm conditions followed by the textual algorithm conditions. The decline in the algorithm modality conditions was very similar. These findings were contrary to what was hypothesized. We can conclude that the algorithm's modality has less to do with how individual's perceive the automation's reliability than the actual reliability itself.

There were no significant interactions or main effects for the change in participants' reported faith in the DA. The overall decline in mean reported scores from time 1 to time 2 was greater in the low reliability conditions when compared to the high reliability conditions. Rempel et al. (1985) found that trust changes as interpersonal relationships develop. Trust in interpersonal relationships is based on predictability, dependability, and faith. Once faith has developed, the person is judged as reliable. Similarly, Muir (1987,1994) argued that trust in automation develops in a similar fashion, but that it follows an opposing pattern. Faith is important early in the interaction followed by dependability, and then predictability (Muir & Moray, 1996). Subtle shifts in trust have been exhibited in response to the properties and performance of automation (Muir, 1989). When the automation performed reliably, operator trust increased over time. However, when the automation performed unreliably, trust quickly decreased. Therefore, DA's reliability levels seemed to have a direct effect on participants' faith in the DA's performance.

There were no significant interactions or main effects for the change in reported perceived technical competence of the DA. As expected, the overall change in mean reported scores from time 1 to time 2 was greater in the low reliability conditions when compared to the high reliability conditions, except in the graphical algorithm conditions where the delta was the same for the 2 reliability conditions. The change in the mean subjective rating from time 1 to time 2 for the high reliability condition where the textual algorithm was presented was -.08, whereas the change in the graphical algorithm and no algorithm conditions were -.43 and -.49, respectively. This slight change in the textual condition versus the other 2 conditions may be an indication that one's perception of the DA's technical competence was better communicated and comprehended when the algorithm was presented in a textual format.

There were no significant interactions for the change in the mean reported scores for perceived understandability of the DA. There was a significant main effect for algorithm presentation condition. Contrary to the hypotheses, the smallest change in the perceived understandability was found in the textual algorithm conditions. An increase in the mean reported perceived understandability from time 1 to time 2 was found in the textual algorithm condition when the DA's reliability was high. The textual algorithm conditions were also significantly different than the graphic and no algorithm conditions. This finding shows that the textual algorithm may have facilitated a greater understanding of the DA over time when compared to presenting it in a graphical format or not at all.

No significant interactions were found for the change in mean reported scores for personal attachment to the DA. There was a significant main effect for algorithm presentation modality. The expected result was seen where the change from time 1 to time 2 was greatest for the no algorithm conditions, followed by the textual algorithm conditions, and finally the graphical algorithm conditions. Participants' attachment to the algorithm was as hypothesized. Participants in the graphical algorithm conditions felt a greater sense of personal preference for using the DA and would feel a sense of loss without it when compared to participants in the textual and no algorithm conditions.

Participants in each condition were asked to provide responses to 3 open-ended questions. The first 2 questions given to all participants asked how they decided when to use the DA and how would they improve the DA. The third question was only given to participants in the conditions where the DA's algorithm was present. Participants were asked how they would improve the DA's algorithm. In conditions where the DA's algorithm was not present, participants were asked how they reasoned through the task.

Nearly one-third (32.7%) of reports of DA usage were based on the DA's characteristics. Characteristics of the DA considered were reliability level and ways in which the DA helped or hindered their decision-making ability. If automation does not live up to the expectation of near-perfect performance, disuse can occur. Prior to interaction, users have a schema regarding automation that it should be perfectly reliable and accurate. This schema leads users to expect perfect performance from the automation. When events occur that are in direct opposition to a user's original expectation, the user will be more likely to remember the event. Every error made by the automation is likely to be remembered. As the task progresses, it may be difficult for the user to maintain an accurate picture of the aid's reliability. The contradictory information may be exaggerated and prominent in one's mind resulting in a distorted negative view of the automation's ability. This may lead the user to underestimate the automation's performance and reliability. Declines in perceived reliability (R) on the HCT were greater for participants in the low reliability conditions when compared to their respective counterparts in the high reliability conditions. This supports the notion that users have the expectation of near-perfect performance (seen in the Practice Block prior to the Test 1A) and when automation performs in a less than perfect manner (seen in Experimental Block 1 prior to the Test 1B) they will remember this and this will negatively affect their perception of the automation.

A little more than 25% of participants' reported DA usage was based on the participants' perception of their personal ability or certainty of an event. When comparing students and aircraft pilots in a function allocation task, students had a tendency to have a greater level of self-confidence and were less inclined to allocate tasks to automation, which they tended to distrust. The pilots, who were more accustomed to using the automation, trusted and relied on the automation more than the students (Riley, 1966). Thus, appropriate reliance on automation can be strongly affected by biases in self-confidence (Lee & See, 2004).

When asked how to improve the DA, 34% of the total responses were to expose, change, or include additional considerations of the city to the algorithm. Lee & Moray (1992) determined performance, process, and purpose as the bases of trust in automation. Process is the extent to which the automation's algorithms are relevant to the situation and are capable of achieving user goals. Process information defines how the automation functions. Process as a basis for trust is a shift of focus away from explicit behaviors and toward the characteristics and attributes of the automation. In other words, trust is shifted from the specific actions of the automation to the automation itself. In the context of automation, algorithm presentation is the process basis of trust. This is similar to Sheridan's (1992) notion of understandability of one's relationship with automation. Users tend to trust automation if its algorithm can be understood and is a feasible means of achieving the user's goals in a given situation.

Nearly 20% of the responses were to improve the DA's reliability or accuracy. Performance is the past and present operation of the automation that includes characteristics such as ability, reliability, and predictability. Performance information defines what the automation does and refers to the expertise and capability as demonstrated by its ability to accomplish the user's goals. Like Sheridan's (1992) concept of robustness as a basis for trust in automation, performance demonstrates the task-dependant nature of trust. When automation performs in a manner that reliably accomplishes user's goals, it is likely to be trusted.

Participants in conditions where the algorithm was presented in a textual or graphical format were asked how to improve the DA's algorithm. No suggested changes were reported for 50% of the total responses. This may be an indication that participants did not fully understand the algorithm or that they did not have a comprehensive mental model of the algorithm. Of the total responses, 20% were to provide better explanations on the decisions suggested. This may further support the premise that participants did not understand the algorithm. Of the 20% of the total responses to this question, an equal number (50%) of participants in the graphical and textual algorithm conditions provided this response, potentially demonstrating that the graphical algorithm was not more comprehensible than the textual algorithm. These reports contradict research findings (e.g., Glenberg & Langston, 1992) that state that diagrams may facilitate the development of a mental model because they lead to the formation of a more detailed representation of the material presented to the individual. These results may support the need for a more comprehensible graphical algorithm. Future research of this kind should consider presenting graphical algorithm alternatives with the goal of finding the most understandable design possible.

Participants in the no algorithm conditions were not shown the DA’s algorithm, so they were asked how they reasoned through the task. A large number of the total responses (55.6%) were that they used any information presented in the game’s interface and 13.9% were reports of considering the DA’s suggestions. Therefore, any information in the interface should be comprehensible and useful, lending more support to the importance of presenting the automation’s algorithm. These findings may also suggest that even if users don’t rely on the DA, they may consider some of its information when making decisions. In 19.4% of the responses, participants reported using their own logic to reason through the task. This group of respondents represents those that are less likely to use (i.e., disuse) the DA, even if it is highly reliable.

A two-way ANOVA conducted on the number of DA's suggestions that participants executed yielded a significant interaction for reliability condition. Participants in the low reliability conditions where no algorithm or a textual algorithm was presented executed the DA's suggestions more than participants in the respective high reliability conditions. Whereas, participants in the high reliability condition that saw a graphical algorithm, on average, executed more of the DA's suggestions than participants in the respective low reliability condition. When asked how they reasoned through the task, a larger percentage of participants in the no algorithm, low reliability condition (60%) used any information presented in the game's interface or considered what the DA suggested, then their counterparts in the no algorithm, high reliability condition (40%). In both instances, this misuse of the DA was not anticipated.

A comparison of the perceived reliability of an automated aid (trust in aid) and the perceived reliability of manual control (trust in self) determines automation use. The outcome of the decision making process has been named the "perceived utility" of the automated system and is expected to be directly related to the trust of the automated aid and subsequent use, misuse, disuse, or abuse (Dzindolet, Pierce, Beck, & Dawe, 2000). Figure 6 illustrates how participants' usage patterns influenced mean health while playing the game. Usage was based on the number of times they used the DA's suggestions (i.e., low <=5; high > 5) and the number of times they chose to perform an action manually (low manual action < 22 non-DA related actions; high manual action > 22 non-DA related actions) and their associated trust levels. Participants who used the DA did not make a lot of manual changes and those that did not use the DA manually configured the city. The best performers were those that relied on the DA when the reliability was high and those that manually configured the city when the reliability was low. Table 17 shows that these 2 groups had the highest mean health scores. These 2 groups also demonstrated appropriate use of the automation. But, the opposite was true for individuals that relied on the DA when the reliability was low and those that manually configured the city when the reliability was high. The second bar in the graph shows the lowest mean health levels and represents participants who wrongly showed high levels of trust and low manual action, when the DA's reliability was low. These participants erroneously relied on the DA in instances where it did more harm then good when making decisions. This behavior represents misuse of the automation. In instances where the DA was useful, but participants failed to use it, disuse occurred. This is characterized by low trust and high manual action when the DA's reliability is high. Disuse of the DA meant a greater effort on the part of the users to arrive at a decision of their own. The mean health for this group is shown as the third bar in the graph and is the second lowest mean health score. The results show that the cost of disuse, or deciding not to allow any type of human-system relationship to develop, means a lower health score when compared to individuals who showed appropriate usage patterns (i.e., use when it was reliable and not using it when it was unreliable).

DA reliance levels varied greatly among the 6, algorithm modality and reliability, conditions. Participants in the high reliability conditions showed DA reliance levels as hypothesized. DA reliance decreased from the graphical, to the textual, to the no algorithm conditions. DA reliance levels in the low reliability conditions were not quite as consistent. Participants in the no algorithm and graphical algorithm conditions showed very similar levels of DA reliance. Participants in the textual algorithm condition executed fewer DA suggestions. Participants exposed to the low reliability DA in the no algorithm and graphical algorithm conditions relied on the DA more than their counterparts in the high reliability conditions. This could have been a result of their failure to form an appropriate mental model of the DA and therefore led to over-reliance, or misuse of the automation. After realizing that the DA could not be trusted in the low reliability conditions, these individuals may have also employed alternative strategies for playing the game.

Based on the subjective research findings, the presentation of an automated DA's algorithm, regardless of its modality, assists in the development of a more robust mental model of the automation. However, the findings were inconsistent in supporting the greater utility of a graphical algorithm over a textual algorithm. Participants in the no algorithm conditions showed the greatest decrease in reported trust in the DA for the five sub-constructs of the HCT. The changes in reported trust varied for the textual and graphical algorithm conditions. In some cases, subjective ratings of the DA on the 5 sub-constructs of trust were similar for the textual and graphical algorithm conditions. Sometimes there was a greater decline in the subjective ratings for participants in the graphical algorithm conditions when compared to participants in the textual algorithm conditions.

Practical Implications

Overall, these findings can be used to support the inclusion of a DA's associated algorithm. Consideration must be given to the user population and their knowledge and ability as it relates to the automation, as well as, the complexity of the automation and it's algorithm when deciding what algorithm modality to consider in the implementation. If a graphical algorithm is so complex that it is difficult to comprehend, consider presenting the algorithm in natural language that spells out the relationships among variables in a way that the target audience can understand. When considering an expert population with knowledge and training in a particular area (e.g., aircraft pilots), a graphical algorithm can be considered.

Future Research

Several areas of future investigation came out of the present research. The graphical algorithm used in the present study was revised during a series of pilot tests. However, there are several potential improvements to the graphical algorithm that could be considered. The algorithm's size, information presented, and the relationships among pieces of information could be made more intuitive through further investigation. Automated DAs and their associated algorithms can be useful in non-gaming environments and as a part of an individual's job tasks (e.g., automated software applications that assist in problem determination or performance tuning). Research of this kind, performed in a non-gaming environment, might produce a very different set of findings.

In order to ensure that participants are implementing the DA's suggestions when selecting an action to perform, future research could include the inclusion of a button that allows the user to directly implement the DA's suggestion. This would ensure that objective data on DA usage was more accurate.

Bibliography

Adams, B.D., Bruyn, L.E., Houde, S., & Angelopoulos, P. (2003). Trust in automated systems literature review, (7747-10), Toronto, Ontario, CA: Department of National Defense.

Adelson, B. (1981). Problem solving and the development of abstract categories in programming languages. *Memory and Cognition, 9*, 422-433.

Azjen, I., & Fishbein, M. (1980). *Understanding attitudes and predicting social behavior*. Upper Saddle River, NJ: Prentice Hall.

Bandura, A. (1982). Self-efficacy mechanism in human agency. *American Psychologist, 37*, 122-147.

Bennett, K.B. & Flach, J.M. (1992). Graphical displays: An evaluation of the basic tasks model of graphical perception. *Human Factors, 34*(5), 513-534.

Bernard, H.R. (1994). *Research methods in anthropology: Qualitative and quantitative approaches*. 2nd Edition. Thousand Oaks, CA: Sage Publication.

Bertoline, G. R., Wiebe, E. N. (2003). *Technical Graphics Communication* (3rd Ed.). New York: McGraw-Hill.

Best, J.B. (1989). *Cognitive Psychology*. US: West Publishing.

Boon, S. & Holmes, J. (1991). The dynamics of interpersonal trust: Resolving uncertainty in the face of risk. In Hindle, R., & Groebel, J. (Eds.). *Cooperation and Prosocial Behavior.* (pp. 167-182). New York: Cambridge University Press.

Campbell, C.S., Kandogan, E., November, A., Barrett, R., Maglio, P.P. (2005). Policity: An Experimental Evaluation of Policy-Based Administration in a City Simulation. *IEEE 6th International Workshop on Policies for Distributed Systems and Networks*.

Card, S., Moran, T., & Newell, A. (1983). *The Psychology of Human Computer Interaction,* Mahwah, NJ; Lawrence Erlbaum Associates.

Carey, J.W., (?). Intercoder agreement in analysis of responses to open-ended interview questions: Examples from Tuberculosis Research. *Cultural Anthropology Methods, 8*(3), 1-5.

de Kleer, J. & Brown, J.S. (1983). Assumptions and ambiguities in mechanistic mental models. In Gentner, D., & Stevens, A.L. (Eds.). *Mental Models.* (pp. 155-190). Hillsdale, NJ; Lawrence Erlbaum Associates.

Deutsch, M. (1958). Trust and suspicion. *Journal of Conflict Resolution, 2,* 265-279.

Dougherty, S. (2003). Automation. Retrieved June 18, 2004 from http://faculty.erau.edu/dohertys/410/410_16_automation.ppt

Dzindolet, M.T., Pierce, L.G., Beck, H.P., & Dawe, L.A. (2000). A framework of automation use, (ARL-TR-2412), Aberdeen Proving Ground, MD: Army Research Laboratory.

Dzindolet, M. T., Beck, H.P. & Pierce, L.G. (2000). Encouraging human operators to appropriately rely in automated decision aids. *Proceedings of the 2000 Command and Control Research and Technology Symposium*, Monterey, CA: Department of Defense Cooperative Research Program.

Endsley, M. (1988). Design and evaluation for situation awareness enhancement. In *Proceedings of the 32nd Annual Meeting of the Human Factors and Ergonomic Society Annual Meeting*, pp. 1388-1392. Santa Monica, CA: Human Factors Society.

Endsley, M. (1995a). Towards a new paradigm for automation: Designing for situation awareness. In *Proceedings of the 6th IFAC/IFIP/IFOR/IEA symposium on Analysis, Design and Evaluation of Man-Machine Systems* (pp.421-426). Cambridge, MA: MIT Press.

Endsley, M. (1995b). Towards a theory of situation awareness in dynamic systems. *Human Factors, 37,* pp 36-64.

Endsley, M. (1996). Automation and Situation Awareness. In R. Parasuraman & M. Mouloua (Eds.), *Automation and human performance: Theory and applications* (pp.163-181). Mahwah, NJ: Lawrence Erlbaum Associates Inc.

Fishbein, M., & Azjen, I., (1975). *Belief, attitude, intention, and behavior.* Reading, MA: Addison-Wesley.

Fitts, P. (1951). Human engineering for an effective air navigation and traffic control system. Washington, DC: National Research Council.

Fox, J.M., & Boehm-Davis, D.A. (1998). Effects of age and congestion information accuracy of advanced traveler information on user trust and compliance. *Transportation Research Record, 1621*, 43-49.

Gaines, S.O., Panter, A.T., Lyde, M.D., Steers, W.N., Rusbult, C.E., Cox, C.L., et al. (1997). Evaluating the circumplexity of interpersonal traits and the manifestation of interpersonal traits in interpersonal trust. *Journal of Personality and Social Psychology, 73,* 610-623.

Glenberg, A.M., & Langston, W.E. (1992). Comprehension of illustrated text. Pictures help to build mental models. *Journal of Memory and Language, 31*, 129-151.

Gregory, D. (1986). Delimiting expert systems. *IEEE Transactions on Systems, Man, & Cybernetics, SMC-16*, 834-843.

Guerlain, S., Jamieson, G.A., Bullemar, P., & Blair, R. (2002). The MPC Elucidator: A case study in the design for human-automation interaction. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, 32*, 1, 25-40.

Guthrie, S.E., (1993). *Faces in the cloud: A new theory of religion.* New York: Oxford University Press.

Haber, R.N. & Myers, B.L. (1982). Memory for pictograms, pictures, and words separately and all mixed up. *Perception, 11*, 57-64.

Haber, R.N. & Wilkinson, L. (1982). Perceptual components of computer displays. *IEEE Computer Graphics and Applications, 2*, 23–35.

Heeger, D. (2003). Signal detection theory. Retrieved July 31, 2006 from http://www.cns.nyu.edu/~david/sdt/sdt.html.

Hegarty, M., Carpenter, P.A., & Just, M.A. (1991). Diagrams in the comprehension of scientific texts. In R. Barr, M. L. Kamil, P. Mosenthal, & P. D. Pearson (Eds.), *Handbook of reading research* (Volume II, pp. 641-668). Mahwah, NJ: Ehrlbaum.

Helmreich, R.L. (1984). Cockpit management attitudes. *Human Factors, 5*, 583-589.

Jones, G.R. & George, J.M. (1998). The experience and evolution of trust: Implications for cooperation and teamwork. *Academy of Management Review, 23*, 531-546.

Kaber, D., & Endsley, M. (1997). The combined effect of level of automation and adaptive automation on human performance with complex, dynamic, control systems. In *Proceedings of the 41st annual meeting of the Human Factors and Ergonomic Society* (pp. 205-209). Santa Monica, CA: Human Factors and Ergonomic Society.

Kaber, D., Riley, J., Tan, K., & Endsley, M. (2001). On the design of adaptive automation for complex systems. *International Journal of Cognitive Ergonomics, 5(1)*, pp. 37-57.

Kelly, C., Boardman, M., Goillau, P., & Jeannot, E. (2003). Guidelines for trust in future ATM systems: A literature review. *European Organization for the Safety of Air Navigation:* 49pp.

Kikuchi, M., Watanabe, Y., & Yamasishi, T. (1996). Judgment accuracy of other's trustworthiness and general trust: An experimental study. *Japanese Journal of Experimental Social Psychology, 37,* 23-36.

Kramer, R.M. (1999). Trust and distrust in organizations: Emerging perspectives, enduring questions. *Annual Review of Psychology, 50,* 569-598.

Larkin, J., & Simon, H. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science, 11,* 65-99.

Leahey, T.H. & Harris, R.J. (2001). Learning and Cognition. New Jersey: Prentice Hall, Inc.

Lee, J.D., & Moray, N. (1992). Trust, control strategies and allocation of function in human-machine systems. *Ergonomics, 35*, 1243-1270.

Lee, J.D., & Moray, N. (1994). Trust, self-confidence, and operators' adaptation to automation. *International Journal of Human-Computer Studies, 40,* 153-184.

Lee, J. & See, Katrina (2004). Trust in automation: Designing for appropriate reliance. *Human Factors, 46* (1), pp. 50-80.

Legrenzi, P., & Girotto, V. (1996). Mental models in reasoning and decision making processes. In J. Oakhill & A. Garnham (Eds.), *Mental models in cognitive science* (pp. 95-118). Hove, UK: Psychology Press.

Levie, W.H. (1987). Research on pictures: A guide to the literature. In D.M. Willows & H.A. Houghton (Eds.), *The Psychology of Illustrations Vol. I: Basic Research* (pp. 1-50). New York: Springer-Verlag.

Levine, T. & Donitsa-Schmidt, S. (1998). Computer use, confidence, attitudes, and knowledge: A causal analysis. *Computers on Human Behavior, 14,* 125-146.

Levis, A., Moray, N., & Hu, B., (1994). Task decomposition, and allocation problems and discrete event systems. *Automatica*, *30*, 203-216.

Lewandowski, S., Mundy, M., & Tan, G. (2000). The dynamics of trust: Comparing humans to automation. *Journal of Experimental Psychology: Applied*. Vol 6, 2, 104-123.

Lewis, J.D. & Weingart, A. (1985). Trust as a social reality. *Social Forces, 63,* 967-985.

Littlewood B. & Strigini L.(2000). *A discussion of practices for enhancing diversity in software designs,* DISPO project technical report, Centre for Software Reliability, City University.

Llinas, J., Bisantz, A., Drury, C., Seong, Y., & Jian, J. (1998). Studies and analyses of aided adversarial decision making. Phase 2: Research on human trust in automation. Wright-Patterson AFB, OH, Air Force Research Laboratory, Human Effectiveness Directorate: 117.

Lucas, K., & Sherry, J. L. (2004). Sex differences in video game play: A communication-based explanation. *Communication Research, 31*, 499 - 523.

Luczak, H., Roetting, M., & Schmidt, L. (2003). Let's talk: anthropomorphization as means to cope with stress on interacting with technical devices. *Ergonomics, 46,* 1361-1374.

Madsen, M. and Gregor, S. (2000). Measuring human-computer trust. In G. Gable and M. Viatle (Eds.). *Proceedings of the 11th Australasian Conference on Information Systems*, Brisbane, 6-8 Dec., p. 53.

Mayer, R., Davis, J., & Schoorman, F. (1995). An integrative model of organizational trust. *Academy of Management Review, 20*(3), 709-734.

McClumpha, A. & James, M. (1994). Understanding automated aircraft. In M. Mouloua & R. Parasuraman (Eds.) *Human performance in automated systems: Current research and trends* (pp. 183-190), Hillsdale, NJ:LEA.

McDougall, S.J., Curry, M.B., & de Bruijn, O. (2001). The effects of visual information on users' mental models: An evaluation of pathfinder analysis as a measure of icon usability. *International Journal of Cognitive Ergonomics, 5*(1), 59-84.

McKnight, D.H., Cummings, I.L., & Chervany, N.L. (1998). Initial trust formation in new organizational relationships. *Academy of Management Review, 23,* 473-490.

Miles, M. B. & Huberman, A.M., (1994). *Qualitative data analysis*. 2nd Edition. Thousand Oaks, CA: Sage Publications.

Miller, C. & Parasuraman, R. (2003). Beyond levels of automation: An architecture for more flexible human-automation collaboration. *Proceedings of the Human Factors and Ergonomics Society 47th Annual Meeting,* pp.182-186.

Moray, N. (1986). Monitoring behavior and supervisory control. Boff, K., Kaufman, L., & Thomas, J. (Eds.), *In Handbook of Perception and Human Performance.* (pp. 40.1-40.51) New York: Wiley.

Moray, N., Hiskes, D., Lee, J., & Muir, B. (1995). Trust and human intervention in automated systems. *In Expertise and technology: Cognition and human-computer cooperation* (pp.183-194). Hillsdale, NJ: Lawrence Erlbaum Associates.

Moray, N., Inagaki, T., & Itoh, M. (2000). Adaptive automation, trust and self-confidence in fault management of time-critical tasks. *Journal of Experimental Psychology: Applied*, 6(1), 44-58.

Muir, B.M. (1989). Operator's trust in and use of automatic controllers in a supervisory process control task. Unpublished doctoral dissertation. University of Toronto.

Muir, B.M. (1994). Trust in automation: Part I. Theoretical issues in the study of trust and human intervention in a process control simulation. *Ergonomics, 37*(11), 429-460.

Muir, B.M., & Moray, N. (1996). Trust in automation: 2. Experimental studies of trust and human intervention in automated systems. *Ergonomics, 37,* 1905-1922.

Muir, D.E. (1988). Assessing the psychosocial impact of computers: the von Neumann effect. *Sociological Forum, 3,* 606-612.

Nass, C., & Lee, K.N. (2001). Does computer-synthesized speech manifest personality Experimental tests of recognition, similarity-attraction, and consistency-attraction. *Journal of Experimental Psychology – Applied, 7,* 171-181.

Nass, C., Moon, Y., Fogg, B.J., Reeves, B., & Dryer, D.C. (1995). Can computer personalities be human personalities? *International Journal of Human-Computer Studies, 43,* 223-239.

Nass, C., Steuer, J.S., & Tauber, E. (1994). Computers are social actors. *Proceedings of the CHI Conference.* 72-77. Boston, MA.

Norman, D.A. (1983). Assumptions and ambiguities in mechanistic mental models. In Gentner, D., & Stevens, A.L. (Eds.). *Mental Models.* (pp. 7-14). Hillsdale, NJ; Lawrence Erlbaum Associates.

Oliner, S.D. & Sichel, D.E. (2002). Information technology and productivity: Where are we now and where are we going? *Economic Review 87*(3), 15-44.

Paivio, A. (1986). Mental Representations. New York: Oxford University Press.

Parasuraman, R., Molloy, R., & Singh, I.L. (1993). Performance consequences of automation-induced complacency. *International Journal of Aviation Psychology, 3*(1), 1-23.

Parasuraman, R. & Riley, V. (1997). Humans and automation: Use, misuse, disuse, and abuse. *Human Factors, 39,* 230-253.

Parasuraman, R., Sheridan, T. & Wickens, C. (2000). "A Model for Types and Levels of Human Interaction with Automation," IEEE Trans. on Systems, Man and Cybernetics – Part A: System and Human, volume 30, 286-297.

Parasuraman, S., Singh, I.L., Molloy, R., & Parasuraman, R. (1992). Automation-related complacency: A source of vulnerability in contemporary organizations. *IFIP Transactions A – Computer Science and Technology, 13,* 426-432.

Pomranky, R.A., Dzindolet, M.T., & Peterson, S.A. (2001). Violations of expectations: A study of automation use. *Proceedings of the 2001 Command and Control Research & Technology Symposium.* Annapolis, MD.

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

Reeves, B. & Nass, C. (1996). *The media equation: How people treat computers, television, and new media like real people and places.* New York: Cambridge University Press.

Reeves, W. (1999). *Learner-centered design*. California: Sage Publications, Inc.

Rempel, J.K., Holmes, J.G., & Zanna, M.P. (1985). Trust in close relationships. *Journal of Personality and Social Psychology, 49*(1), 95-112.

Riley, V. (1996). Operator reliance on automation: Theory and data. In R. Parasuraman & M. Mouloua (Eds.), *Automation theory and applications* (pp. 19-35). Mahwah, NJ: Erlbaum.

Ross, W. & LaCroix, J. (1996). Multiple meanings of trust in negotiation theory and research: A literature review and integrative model. *International Journal of Conflict Management, 7,* 314-360.

Rotter, J.B. (1967). A new scale for the measurement of interpersonal trust. *Journal of Personality, 35,* 651-665.

Rotter, J.B. (1971). Generalized expectancies for interpersonal trust. *American Psychologist, 26,* 443-452.

Ruble, D.N., & Stangor, C. (1986). Stalking the elusive schema: Insights from developmental and social-psychological analyses of gender schemas. *Social Cognition, 4,* 227-261.

Schnotz, W., Bannert, M., & Seufert, T. (2002). Toward an integrative view of text and picture comprehension: Visualization effects on the construction of mental models. In J. Otero, J.A. Leon, & A.C. Graesser (Eds.), *The Psychology of Science Text Comprehension* (pp. 385-416). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Schwartz, B., Wasserman, E.A., & Robbins, S.J. (2002). Psychology of Learning and Behavior. New York: W.W. Norton & Company.

Sheridan, T. B. (1988). Trustworthiness of command and control systems. *Conference on Analysis; Design and Evaluation of Man-Machine Systems*, 427-431.

Singh, I.L., Molloy, R., & Parasuraman, R. (1993). Individual differences in monitoring failures of automation. *Journal of General Psychology, 120,* 357-373.

Smith, D.A., & Graesser, A.C. (1981). Memory for actions in scripted activities as a function of typicality, retention interval, and retrieval task. *Memory & Cognition, 9,* 550-559.

Stack, L. (1978). Trust. In H. London & J.E. Exner Jr. (Eds.), *Dimensions of personality* (pp.561-599). New York: Wiley.

Sutcliffe, A.G., Ennis, M., & Hu, J. (2000). Evaluating the effectiveness of visual user interfaces for information retrieval. *International Journal Human-Computer Studies, 53*, 741-763.

Takeuchi, Y. & Katagiri, Y. (1999). Identity perception of computers as social actors. *Proceedings of ICCS/JCSS,* 247-252.

Thibault, J.W., & Kelley , H.H. (1959). The Social Psychology of Groups. NY: John Wiley & Sons, Inc.

Tufte, E. (1990). *Envisioning information.* Cheshire, CT: Graphics Press.

Tufte, E. (1997). *Visual explanations.* Cheshire, CT: Graphics Press.

Vicente, K.J., & Rasmussen, J. (1990). The ecology of human machine systems. II: Mediating “direct perception” in complex work domains. *Ecological Psychology, 2*(3), 207-249.

Wiener, E.L. (1981). Complacency: Is the term useful for air safety? In *Proceedings of the 26th Corporate Aviation Safety Seminar* (pp. 116-125). Denver: Flight Safety Foundation, Inc.

Will, R. (1991). True and false dependence on technology: Evaluation with an expert systems. *Computer Human Behavior, 7*, 171-183.

Winn, B. (1987). Charts, graphs, and diagrams in educational materials. In D.M. Willows & H.A. Houghton (Eds.), *The Psychology of Illustrations Vol. I: Basic Research* (pp. 1-50). New York: Springer-Verlag.

Woods, D.D. (1991). The cognitive engineering of problem representations. In G.R.S. Weir and J.L. Alty (Eds.), *Human-computer interaction and complex systems* (pp. 169-188). London: Academic.

Woods, D. (1996). “Decomposing automation: Apparent simplicity, real complexity,” in *Automation and Human Performance: Theory and Applications*, R. Parasuraman and M. Mouloua, Eds., Mahwah, NJ: Erlbaum. pp. 1-16.

Appendices

Appendix A: Human-Computer Trust (HCT) rating scale (Madsen & Gregor, 2000)

Use the scale below (1=Strongly Disagree to 5=Strongly Agree) to answer the following questions. Please record your responses on the lines below.

1	2	3	4	5
Strongly Disagree	Disagree	Neither Agree Nor Disagree	Agree	Strongly Agree

______ R1. The decision aid always provides the advice I require to make my decision. ______ R2. The decision aid performs reliably. ______ R3. The decision aid responds the same way under the same conditions at different times. ______ R4. I can rely on the decision aid to function properly. ______ R5. The decision aid analyzes problems consistently.
______ T1. The decision aid uses the appropriate methods to reach decisions. ______ T2. The decision aid has sound knowledge about this type of problem built into it. ______ T3. The advice the decision aid produces is as good as that which a highly competent person could produce. ______ T4. The decision aid correctly uses the information I enter. ______ T5. The decision aid makes use of all the knowledge and information available to it to produce its solution to the problem.
______ U1. I know what will happen the next time I use the decision aid because I understand how it behaves. ______ U2. I understand how the decision aid will assist me with decisions I have to make. ______ U3. Although I may not know exactly how the decision aid works, I know how to use it to make decisions about the problem. ______ U4. It is easy to follow what the decision aid does. ______ U5. I recognize what I should do to get the advice I need from the decision aid the next time I use it.
______ F1. I believe advice from the decision aid even when I don't know for certain that it is correct. ______ F2. When I am uncertain about a decision I believe the decision aid rather than myself. ______ F3. If I am not sure about a decision, I have faith that the decision aid will provide the best solution. ______ F4. When the decision aid gives unusual advice I am confident that the advice is correct. ______ F5. Even if I have no reason to expect the decision aid will be able to solve a difficult problem, I still feel certain that it will.
______ P1. I would feel a sense of loss if the decision aid was unavailable and I could no longer use it. ______ P2. I feel a sense of attachment to using the decision aid. ______ P3. I find the decision aid suitable to my style of decision-making. ______ P4. I like using the decision aid for decision-making. ______ P5. I have a personal preference for making decisions with the decision aid.

Appendix B: Subjective Questionnaire of Understanding of Automation's Complexity

Please answer the following questions. If you do not understand any of the questions, please ask the experimenter to provide further clarification.

1. How did you decide when to use the decision aid?

2. How would you improve the decision aid?

3. Can you suggest ways to improve the decision aid's algorithm?

(No algorithm participants only) Please explain how you reasoned through the task.