2008

Sensemaking In Information Systems: Toward A Sensemaking Inquiring System

James Parrish
University of Central Florida

SENSEMAKING IN INFORMATION SYSTEMS: TOWARD A SENSEMAKING INQUIRING SYSTEM

by

JAMES L. PARRISH, JR.

B.S.B.A University of Central Florida 2002

M.S. University of Central Florida 2004

A dissertation submitted in partial fulfillment of the requirements

for the degree of Doctor of Philosophy of Business Administration

in the Department of Management Information Systems

in the College of Business Administration

at the University of Central Florida

Orlando, Florida

Summer Term

2008

Major Professor: James F. Courtney

ABSTRACT

Complexity and uncertainty have long been problems for organizations of all types. Organizational members do not do a very good job of dealing with the complexity and uncertainty as research shows that when faced with complex situations humans often turn to the same sources of information repeatedly (a practice that will eventually betray them), and/or reduce the amount of scanning that they do (Weick 1995; Boyd and Fulk 1996). Organizations often turn to information systems to help them deal with the complexity, but they often take a techno-centric view of knowledge that does not incorporate the human qualities needed for unstructured decisions (Malhotra 1997; Courtney 2001; Malhotra 2001). Additionally, there are times when the information systems that we are using may hinder the processes of dealing with the complexity (Weick and Meader 1993).

Weick's (1995) concept of sensemaking is believed to help us to deal with this complexity. In his work with Meader (1993) he wonders what the effects of a sensemaking support system would have, but he does not have the answer because they state that it has not been asked. This dissertation answers the call of Weick and Meader as well as other scholars that have called for sensemaking and human intuition to be included in our information systems. This is accomplished by viewing sensemaking from an inquiring systems perspective (Churchman 1971) to develop a kernel theory that will be used in the context of design science to develop design requirements and principles for a sensemaking system. These design principles are then used to

build an instantiation of the system in the form of SenseMan, a system designed to help a local government agency deal with complexity in the context of software updates. Finally the design is evaluated for its effectiveness in dealing with the complexity of in this context using both quantitative and qualitative methods.

This dissertation is dedicated to my wife, Natalia, and my children Kaitlyn and Brendan for their love and sacrifice throughout this process. I doubt that I will ever fully realize the level of sacrifices that they have made for me. However, I do realize that without them, I would not be the person that I am today.

It is also dedicated to my Parents and my Parents-in-Law for without their support and guidance, this work would not be possible.

ACKNOWLEDGEMENTS

I would like to acknowledge the outstanding level of support and dedication contributed to this work by the employees of the Lake County Clerk of Circuit and County Clerk. I would especially like to acknowledge the efforts of Hon. James C. Watkins and Hon. Neil R. Kelly, the former and present Clerk of the Circuit and County Court of Lake County, FL, Mr. Brent R. Holladay, Chief Deputy of the Information Resources Department of the Lake County Clerk, and Mr. Charles Reed, Information Resources Project Manager for the invaluable amounts of time, insight, and assistance that they provided to this dissertation.

I would also like to acknowledge the dedication and many hours of work put in on this project by the members of my dissertation committee. The work that Dr. James Courtney, Dr. Paul Cheney, Dr. David Paradice, and Dr. Lawrence West contributed to this project significantly raised its quality and for that I am very thankful.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

ANOVA	Statistical analysis of variance technique.
DAGS	Methodology from Adams and Courtney (2004) that stands for design science, action research, grounded theory, and software engineering.
DFD	Data flow diagram, a conceptual data modeling tool used by systems analysts.
DSS	Decision support system.
EKP	Emergent knowledge process.
GDSS	Group decision support system.
GSS	Group support system.
IS	Information system(s).
KMS	Knowledge management system(s).
NHC	National Hurricane Center.
SERM	Software engineering research methodology.

CHAPTER ONE: INTRODUCTION

"Computers are incredibly fast, accurate and stupid. Human beings are incredibly slow, inaccurate and brilliant. Together they are powerful beyond imagination." ~ Albert Einstein

In 1971 C. West Churchman released his seminal work *The Design of Inquiring Systems: Basic Concepts of Systems and Organization*. In this book, he presents his views on design and introduces us to the concept of inquiring systems. Inquiring systems are teleological (goal seeking) systems with the objective of creating knowledge for the betterment of the human condition. Churchman, who started his career as a mathematician and philosopher, believed in the power of the scientific method in its application to the problems of society. He is known as a founding figure in the field of operations research and he helped to write the first textbook in that area. To the information systems community, however, he is best known as being one of the founders of the "Systems Approach." He believed that the world was really just one system that was inseparable, a point that he argues in his widely cited books *Challenge to Reason* (1968) and *The Systems Approach* (1968). In these books he begins to formulate some of the concepts for his inquiring systems, such as the guarantor, and discusses the systems approach and its limitations. Churchman refined the concepts presented in these books into his inquiring systems. He did this by taking the works of Leibniz, Locke, Kant, Hegel, and his own teacher Singer, and

recasting their views on how to make the world better (design) and on the creation of knowledge (inquiry) into the language of systems. In the 36 years since their release, Churchman's inquiring systems and his views on design have become cornerstones upon which much IS research of today has been built. This is especially true for the IS specialization fields of decision support systems, knowledge management, and systems design. In fact, Churchman's contributions to the IS field are so great that he was honored as being one of the first recipients of the LEO award. The LEO, named after the first business computing system, "recognize[s] truly outstanding individuals in the Information Systems community, both academics and practitioners, who have made exceptional contributions to research in and/or the practice of Information Systems (Systems 2007)."

Despite all of the work that has drawn upon the insight of Churchman and his inquiring systems, little of it has ventured from Churchman's original five inquirers. One example of an effort to move beyond the realm of Leibniz, Locke, Kant, Hegel, and Singer is found in a series of eleven brief reviews entitled "Design of the Modern Inquiring System" that appeared in the journal Systems Research from 1989-1994. This series, primarily authored by John P. van Gigch, encouraged authors to submit reviews on work that could serve as new epistemologies for inquiring systems. More specifically, the intention was summed up in the forward of each review: "…the thinking of past and present philosophers and thinkers are surveyed and discussed to see how their logic and methods of reasoning can be used to design the Modern Inquiring

System – a system dedicated to the acquisition and production of knowledge and to the solution of contemporary problems (Snell 1988; van Gigch 1988; van Gigch 1988; van Gigch 1988; Herrscher 1989; Pavesi and Pavesi 1989; van Gigch 1990; Pavesi and Pavesi 1991; van Gigch 1993; van Gigch 1993; van Gigch 1994). ” Among the individuals featured in the series were the rationalist philosopher and scientist Renee Descartes, the poet and novelist Herman Hesse, and philosopher and mathematician Ramon Lull, who had a great influence on Leibniz. While these reviews provided some examples of how the inquirers could be expanded, they did not go as far as to actually translate the works into the language of inquiring systems and to generate design principles based on the translation.

Another example of an attempt to expand on the epistemological foundation of the inquiring systems is the work of Guo and Sheffield (2006). They integrate the critical social theory of Habermas with Churchman's inquiring systems into what they define as a Habermasian inquiring system. The epistemological stance taken by Guo and Sheffield is that knowledge is created by a bidirectional interaction of persons with the organizational world that helps to form their values and the technical world of material facts (Guo and Sheffield 2006). They utilize their inquiring system as a framework for knowledge management research that they argue will "provide a philosophically grounded, universally pragmatic framework useful in managing the complexity, and conceptualizing the richness, of knowledge phenomena (Guo and Sheffield, 2006, p.1)."

The work of Habermas is also used by Asif and Klein (2007), who argue that the success of recent phenomena such as blogging and social networking Internet sites is evidence that it is time to look for new epistemological foundations for information systems. They state that systems are moving away from supporting instrumental inquiry and towards a concept that they call deliberative inquiry. According to Asif and Klein, deliberative inquiry improves upon the best ideas of the Kantian and Hegelian inquiring systems. They believe that their notion of the inquiring system advances the epistemological foundations of the inquiring systems and overcomes weaknesses in the Kantian and Hegelian inquirers (Asif and Klein 2007). This proposal agrees with both Guo and Sheffield and Asif and Klein that it is time to expand on the epistemological foundations of the inquiring systems, especially given the dramatic increase of complexity in information technology and our environments that have occurred since Churchman penned his inquiring systems.

Today's environments are characterized by radical change and increasingly complex and wicked problems (Courtney 2001; Malhotra 2001). Wicked problems are problems that are so unstructured, that actually formulating the problem **is** the problem. They are also characterized by having no true or false answers (only good or bad), no stopping rule, no immediate test of the solution, irreversibility of the selected solution, and other characteristics listed below in Table 1 (Rittel and Webber 1973; Courtney 2001).

Table 1: Characteristics of "wicked" problems (Rittel et al., 1973; Courtney, 2001).

Characteristics of "Wicked" Problems
1. The problem is formulating the problem.
2. No stopping rule exists for wicked problems.
3. The solutions to wicked problems are either good or bad.
4. There is no immediate test for the solution, and any solution may have other consequences for an unbounded period of time.
5. It is impossible to learn from trial and error when wicked problems are concerned because each solution is final and cannot be undone.
6. The set of actions to solve wicked problems is not well defined. In fact, no solution may exist for some wicked problems.
7. Even though two wicked problems may seem similar, every wicked problem is unique.
8. Other wicked problems may be exacerbated by solving one wicked problem. The interconnectedness of wicked problems can allow them to be seen as symptoms of other problems.
9. There are many ways to explain differences between actual and desired states. The one that is chosen is the one most plausible to the decision maker.
10. Unlike scientists who can test their hypothesis and find it to be false, the planner who has to solve wicked problems has no right to be wrong.

Because of the dynamic nature of environments and problems, it is increasingly evident that the use of static information to deal with issues in these environments is insufficient, at best. Compounding this problem is the fact that when our environmental complexity increases, we tend to focus in on familiar information and environmental cues that blind and mislead us (Weick 1995). Dynamic and complex environments also necessitate a movement from the old paradigm of predicting changes to our environments and then reacting to the changes when they happen to anticipating the environmental changes and increasing the speed by which we create actionable knowledge (Malhotra 1997).

In order to make this change it will be necessary to develop the sensemaking capabilities of individuals and organizations. I am not only speaking of the sense that we make of the information that we receive, but also the sense that we make of our environments. Many times, unexpected events do not take place within the context of a large crisis; instead they are a result of taking action with a goal in mind and not having a clear picture of the environment in which the action is taken. This misunderstanding of the environment causes the actual events that take place to differ from the intentions of the person or organization initiating the action, thus creating the unexpected (Weick and Sutcliffe 2001). So, making proper sense of the environment is a critical factor in any decision making process because it is necessary to fully understand the problems that require a decision. Weick and Meader (1993) state that without sensemaking the information that is utilized in decision making is not as informative and rich as if it had undergone a sensemaking process.

Despite the need to make sense of information and the environment to create knowledge, many of our information systems take a techno-centric view of knowledge (Malhotra 1997; Courtney 2001). Malhotra (2001) argues that it is important to tie the static information that is stored in our databases to the dynamic nature of the humans that make sense of it. He suggests that the Kantian and Hegelian inquiring systems be used in lieu of the Leibnizian or Lockean. Although these models are more inclusive of human sensemaking, they do not incorporate it explicitly in the design. He also presents a knowledge management model that differentiates the processing

of information from the construction of meaning. Malhotra states that the processing of information by technology involves the pre-determination of meaning for pre-programmed results. In his model, he argues that the construction of knowledge involves human sensemaking and incorporates creativity and innovation, and that this model is more effective for non-structured or non-routine sensemaking (Malhotra 2004).

Possibly even more compelling is the argument that we are not only designing our information systems without regard to sensemaking, we are designing some types of systems in such a way that inhibits it. Weick and Meader (1993) write that the design of group support systems (GSS) are biased by the preoccupation with decisions possessed by western culture. Designs rooted in this bias support decisions primarily and partially support some methods of sensemaking while "short-circuiting" others. The result of this type of design is information used in the decision process that is not grounded in the sensemaking of the persons making the decisions and therefore not as rich as it could have been had sensemaking occurred. They wonder what effects a sensemaking system would have on groups, but they do not have an answer because the question has not been asked (Weick and Meader 1993).

Churchman seemed to recognize the importance of characteristics such as intuition and sensemaking with regards to inquiring systems when he wrote,

"Thus intuition is always important in the development of the mind of the inquiring system, but the challenge to the thinking designer is to rationalize the operations of intuition, so that the creativity of one man becomes the methodology of another; the great idea of one generation becomes the mundane operating basis of the following." (Churchman, 1971, pg. 262)

Research Questions

Thus Malhotra, Weick and Meader, and others have called for both the inclusion of sensemaking, creativity, and intuition in information systems and for information systems designed to support the sensemaking process. This dissertation is an attempt to answer those calls by asking whether IS can assist individuals and organizations with this environmental sensemaking in the form of an inquiring system. In order to answer this question, three contingent questions must be addressed:

1. Is sensemaking an appropriate epistemology for an inquiring system?
2. If sensemaking is an appropriate basis for an inquiring system, can design principles for such a system be derived to guide the construction of an IT artifact supporting sensemaking?

3. If the design principles can be derived to guide the construction of an artifact, how effective will the artifact be in enhancing sensemaking in individuals and organizations?

While the first of the three questions posed in this dissertation is conceptual and leans more toward behavioral science, the second two questions are more rooted in design science.

Design of the Study

In the forthcoming chapters, all three of these questions were answered by developing a kernel theory that builds upon Churchman's (1971) inquiring systems, and then taking that kernel theory to construct an information system design theory (Walls, Widmeyer et al. 1992; Walls, Widmeyer et al. 2004) for a system with the goal of supporting sensemaking. This design theory was then used to create an actual instantiation of the theory in the form of an information system that was deployed and evaluated in an organizational setting.

In adherence to the guidelines of Hevner et al. (2004), any product of design science must show sufficient rigor in its construction and evaluation. To demonstrate that the information system in this dissertation meets these guidelines, the process of developing the kernel theory, design theory, and artifact will all be discussed. The evaluation of the artifact will be guided by the

testable hypotheses that are developed as a part of the design theory (Walls, Widmeyer et al. 1992; Walls, Widmeyer et al. 2004). Since sensemaking is the real goal construct of this evaluation and cannot be measured directly, perceived complexity was chosen as a proxy measure.

Perceived complexity, as operationalized by Boyd and Fulk (1996) is a construct comprised of three dimensions: perceived adequacy of information, perceived analyzability of cause and effect relationships, and perceived predictability. In Boyd and Fulk's study, perceived adequacy of information was a measure of how adequate the participants believed their information to be in a given context. Perceived analyzability was the feelings that the participants had with their ability to discern cause and effect relationships as a result of some action in a given context. Perceived predictability was defined as the participant's assessment of their ability to identify environmental forces that could affect events in a given context. An increase in any or all of these measures would have a negative effect of how complex the participants perceived their environment to be. This being the case, if the system in this dissertation is to be viewed as effective in reducing perceived complexity in the context of software updates, at least one of the following hypotheses should hold under statistical analysis to demonstrate quantitative evidence of system effectiveness.

- H1: System use will cause a posttest increase in perceived adequacy.
- H2: System use will cause a posttest increase in perceived analyzability .
- H3: System use will cause a posttest increase in perceived predictability.

These hypotheses will be further developed and tested later in the dissertation. In an effort to ensure that the requisite rigor is included in the evaluation of these hypotheses, methodological triangulation will be used as a method to check the validity of our findings. In the case of this dissertation, we will conduct a survey research project (quantitative) in parallel with a analysis of interview data (qualitative) to see if there is convergence in the research findings (Hesse-Biber and Leavy 2006).

The remainder of this dissertation is structured as follows. First, there is a review of the relevant literature. Next the process of developing the information system design will be discussed in the context of design science as well as a description of the artifact that resulted from the design science research project. Then the process evaluating the information system that resulted from the design will be presented from both the quantitative and qualitative perspectives as well as the results of those evaluations. Following, there will be a discussion about the results in the context of the entire project that will attempt to find convergence in the results from the studies and to evaluate the overall study in the context of design science. Finally, there are some concluding remarks about the study, its limitations, and the future of research in this area.

CHAPTER TWO: LITERATURE REVIEW

To answer the questions that have been proposed in this dissertation, it is first necessary to review the concepts that are going to be used to develop the answers. To this end, this chapter will review the work of Karl Weick, and his concepts of loose coupling, mindfulness, and sensemaking that all play a part in the development of his evolutionary epistemology. The inquiring systems of C. West Churchman will also be reviewed along with his views on design. Finally, the literature on design science will be presented to acquaint readers with the concept and the prior work that has been done in this area.

Karl Weick - Loose Coupling, Mindfulness and Sensemaking

Karl Weick is a noted organizational theorist and researcher at the University of Michigan School of Business. Throughout his career, Weick has been known for making many contributions to organizational studies. One such contribution is the notion of "loosely coupled"

organizations. Those of us familiar with computing technology may recognize this as a term that refers to systems that are created to interact with other systems via an open architecture. This loose coupling of the systems allows the systems to undergo dynamic changes without affecting the relationship that they have. This same type of robust relationship is found in loosely coupled organizations. In loosely coupled organizations, the requirements of the relationship are stated explicitly and few assumptions are made about the nature of the other organizational partner. This allows for changes to be made in either organization that will not jeopardize the relationship (Weick 2001).

Mindfulness

Weick also provides us with the notion of mindfulness as a way of managing the inevitable unexpected events that an organization will experience. In *Managing the Unexpected,* Weick examines what he calls "high reliability organizations" or HRO's, and asks why these organizations experience far fewer unexpected events or crises than other, more traditional, organizations. HRO's can be defined as those organizations where "failure is not an option." Examples of these types of organizations include aircraft carriers and nuclear power plants. Weick attributes the lack of unexpected events in these types of organizations to a concept that

he calls mindfulness. Mindfulness not only helps organizations anticipate unexpected events, but it also assists in mitigating the damage those events cause when they eventually do happen (Weick and Sutcliffe 2001). He then argues that all organizations can follow the steps that these HRO's have followed to become more mindful. This is accomplished through a five step process that is divided into two logical groups. The five steps are listed here and discussed in Table 2 below: (1) preoccupation with failure, not with success, (2) a resistance to simplify interpretation, (3) being sensitive to operations, (4) making a commitment to resilience, and (5) deferring to organizational expertise in decision making scenarios. The first three steps of the process are dedicated to anticipating the unexpected events before they occur and the second two steps involve limiting the damage that the eventual unexpected event does once it has happened (Weick and Sutcliffe 2001). Taken together, the mindfulness that the process creates promotes a level of awareness that can enhance the organization's ability to detect and address the "little things" that, when left unattended, can culminate in a crisis situation (Weick and Sutcliffe 2001).

Table 2: The Process of Developing Mindfulness (Weick and Sutcliffe 2001).

Process Step	Description
Preoccupation with failure.	Success makes organizations complacent because they feel that what they are doing is the best way of doing it. This leads them to become intolerant of other ideas and interpretations that can blind them to the little events that could become crises.
Resistance to simplify interpretations.	Simplifying interpretations leads to a dependence on expectation that can lead to ignoring evidence that the unexpected is about to occur.
Sensitive to operations.	When the focus is what is going on at the operational level, small events get big attention and seldom blossom into crisis situations.
Commitment to resilience.	Don't ignore errors that have already occurred. Correct them before they become bigger errors that can cause greater damage.
Defer to organizational expertise.	Flexible leadership structures allow the person with the most expertise to be empowered to make decisions, allowing organizational proficiencies to be made use of in a crisis. However, higher level managers are readily accessible should events become more than the local experts can address.

Sensemaking

Sensemaking, however, is what Weick is best known for in the academic community. Weick's concept of sensemaking quite literally means "the making of sense." It is a tool that has long been used in the management field to assist in managerial decision making and strategic planning. In the book *Sensemaking in Organizations* (1995), Weick does not provide any hard and fast procedures for how sensemaking should be performed. He does, however, give us seven guidelines for the sensemaking process. The first guideline is that sensemaking is grounded in identity construction. This means that there must be a sensemaker to initiate the sensemaking process and that much of the process is determined by the sensemaker. The second guideline is that sensemaking is a retrospective process. In sensemaking, examining the past events allows us to, in a way, justify the present and to predict a plausible future. The third principle deals with the sensemaker enacting sensible environments. Here, Weick is not saying that you can change the course of future events by undergoing a sensemaking process, but you can have some influence over future events by gaining an understanding of the present. The fourth principle is that sensemaking is an ongoing process. There is no stopping rule with sensemaking. The retrospective process is continuously being fed new data as time passes. The fifth principle is that sensemaking is a social process. As we make contact with other people and interact with them, we can gain some perspective on how they view events that could change our perspectives.

Additionally, our interactions with them can serve to change theirs as well. The sixth principle is that the process of sensemaking is based on extracted cues. As people, we are bombarded by various cues in our environment. As sensemakers, it is up to us to choose which of these cues are important enough to be included in the sensemaking process. Finally, the seventh principle is that sensemaking values plausibility over accuracy. There are no hard and fast "correct" or "true" answers that come from a sensemaking process. Only pictures of the present and future that pass the test of face validity (Weick 1995). The principles of sensemaking are summarized in Table 3 below.

Table 3: Principles of Sensemaking (Weick 1995; Parrish Jr. and Courtney 2007).

	Principles of Sensemaking	
1	Grounded in Identity Construction	A sensemaker is needed and the results are based on the perspective of that sensemaker.
2	Retrospective	Accounts of the present are made possible by reflecting on the past.
3	Enacts Sensible Environments	The sensemaker can partially influence his or her future environment.
4	Social	Our interactions with others shape the results of our sensemaking.
5	Ongoing	There is no stopping rule for sensemaking.
6	Based on Extracted Cues	We will choose to focus on certain cues out of the many potential cues that exist in our environment.
7	Focused on plausibility rather than accuracy	Finding the exact true answer is not the goal in sensemaking, we just need to find something that is plausible.

Sensemaking in IS

Sensemaking has a rich history in the IS literature. Choo (1996) states that it is one of the three strategic uses for information in organizations, along with decision making and knowledge creation, and is used to assign meaning to organizational actions and events. Swanson and Ramiller (1997) posit that sensemaking is central to the creation of what they call an "organizing vision" that guides the diffusion of an IS innovation through both its early and late stages. Taking a line from Weick, they state that the organizing vision that is based on sensemaking "talks the walk" with regard to the IS innovation and, without it, the IS innovation is doomed to be misunderstood (Swanson and Ramiller 1997).

Malhotra argues for the expansion of the paradigms governing the development of artificial intelligence and expert systems to include the human sensemaking processes that he believes are complementary to the learning processes of machines (Malhotra 2001). He also advocates a knowledge management paradigm that includes sensemaking and allows for the construction of meaning and action based on human creativity and interpretation instead of accepting the static meaning of the information in knowledge management systems (KMS) based on the old paradigm. This static meaning leads to pre-programmed actions and is not reflective of the

reality of today's organizational environments. He argues that non-reflection of reality is a reason why KMS fail (Malhotra 2002).

Weick and Meader (1993) stated that the varying results in GSS that researchers had experienced to that date may be because of misplaced focus instead of the methodological shortcomings that had often been deemed the reason. They believed that the focus should not be on the decision, but rather on defining the questions. Because the focus has always been on the decision, most GSS only peripherally support sensemaking. They offer five strategies that can be employed in GSS to enhance sensemaking: action, triangulation, deliberation, contextualization, and affiliation. They also wonder what a sensemaking system would look like. They say that they do not know what it would look like as no one to that date had asked the question.

Finally, Parrish and Courtney describe a sensemaking approach taken by a local government agency to facilitate its IS strategy (Parrish Jr and Courtney 2007). They also use sensemaking as a source for perspective and combine it with the DSS paradigm first developed by Courtney (2001) and then further by Elgarah, et al. (2002) as the basis for a DSS that fosters collaboration in the context of making control practice decisions for electronic records (Parrish Jr and Courtney 2007)

Churchman's Inquiring Systems

The many contributions of C. West Churchman to the IS community were discussed briefly in the earlier chapter of this proposal, so our focus in this chapter will be on reviewing his thoughts on systems and, more specifically, his inquiring systems. This section will first discuss Churchman's requirements for systems, and then review each inquirer along with providing practical examples of each type of inquirer in use today.

Churchman and Systems

Churchman had distinct requirements on what constitutes a system. Because the central figure in *The Design of Inquiring Systems* is the designer, Churchman leaves it up to the designer to choose whether something is a system or not. However, according to Churchman, for something to be conceptualized as a system, it must meet the following criteria (Churchman and Buchanan 1969; Churchman 1971):

1. It must be teleological, meaning that it must exist to serve some goal.

2. It must have a measure of performance that describes how well the system actually performs with regards to its goal.

3. It serves a client in such a way that the better the system performs, the better the interests of the client are served.

4. It is comprised of goal seeking components that have their own measures of performance and that together serve to co-produce the measure of performance for the entire system.

5. The system has an environment that also serves to co-produce the measure of performance of the whole system.

6. It has a decision maker that can produce changes in the measures of performance via system resources, and by doing so, can produce changes in the measure of performance for the whole system.

7. It has a designer that conceptualizes the system in such a manner that the concepts that the designer presents could cause the changes to be made by the decision maker and, therefore, affect the measure of performance.

8. The goal of the designer is to design the system in such a way that it maximizes its value to the client.

9. There is a guarantee that the goals of the designer are ultimately realizable.

After presenting this list of criteria, Churchman states that even though they are all necessary for something to be called a system, he wonders if they are also sufficient. He states that the question of sufficiency is one of the basic questions addressed in *The Design of Inquiring Systems* (Churchman 1971). As we move forward in this proposal, we will revisit these criteria in order to see if the work of Weick can be considered an appropriate basis for an inquiring system.

A Review of Inquiring Systems

In order to create his inquiring systems, Churchman took the philosophical views of some of the great western philosophers (Liebniz, Locke, Kant, Hegel, and Singer) and proposed systems based on their views of knowledge and of the world that each of the philosophers espoused. Being true systems, all of the inquirers have inputs, processes, and outputs and, consistent with his requirements, they are all teleological or goal seeking. Another feature of Churchman's inquiring systems is the feature called the guarantor. The guarantor serves to ensure that the knowledge created by the system is consistent with the philosophy on which the system is based and "true" to the extent that it is not believed to be false (Courtney, Croasdell et al. 1998).

Having reviewed the basic components of inquiring systems in general we can now examine each of the systems in more detail.

The Liebnizian Inquirer

The Liebnizian inquirer can best be described as a closed system. Churchman (1971, pp. 34-35) details the following features of the Liebnizian inquirer.

1. Innate ideas i.e. no inputs.
2. Capability of producing strings of symbols that can be broken down into recognizable units.
3. Capability of classifying any unit as a tautology, self-contradiction, or contingent truth.
4. Capability of forming nets of units.
5. Capability of ranking the nets based on a prescribed criterion.
6. A method of processing symbols and building nets such that the system will ultimately arrive at the optimal net and know when it has reached that point.

As seen in the features provided by Churchman, the Liebnizian inquirer does not have inputs, per se, but is created with a set of basic axioms. A sentence generator is used to generate hypotheses which are then tested against the axioms for validity and formal logic also plays a role in testing the hypothesis. Knowledge is created in the Liebnizian inquirer in the form of "fact-nets" comprised of contingent truths. The guarantor of the knowledge is its consistency with the basic axioms (Churchman 1971; Courtney, Croasdell et al. 1998; Courtney 2001).

Examples of Liebnizian inquiring systems in use today include many types of expert systems, theorem-proving systems, problem solvers, and algorithm generating systems (Linden, Kuhn et al. 2008).

The Lockean Inquirer

The Lockean inquirer stands in almost direct contrast to the Liebnizian inquirer. Whereas the Liebnizian inquirer is a closed system, the Lockean inquirer is completely open and takes its input in the form of environmental observations. The system is given a set of elementary labels with which to begin. Knowledge is then created in the form of taxonomies by a process of assigning labels to the observations made by the system with the goal being to create a "storehouse" of knowledge" (Churchman 1971; Courtney, Croasdell et al. 1998) . The Lockean

inquirer does not operate alone. In fact, the labels that the Lockean inquirer assigns are only deemed valid if a consensus is reached as to the label's validity by a community of Lockean inquirers. This consensus acts as the primary guarantor of the system's knowledge. Additionally, knowledge is guaranteed by the Lockean inquirer's capability of self monitoring through a process called reflection. In this process, labels can be traced backwards from the complex to the most elementary ones to ensure internal validity (Churchman 1971; Courtney, Croasdell et al. 1998; Linden, Kuhn et al. 2008).

A wonderful example of the Lockean inquirer in practice can be found on the image search site from Google (http://images.google.com). This site has a program that allows humans to act as Lockean inquiring systems. Users are paired together anonymously and then shown an image from the millions of stored images in the Google database. The users are then asked to assign labels to the image. When the users reach consensus on a label, that label is assigned to the image and the users move to another image. This process helps Google to manage their information about the images by creating a more effective taxonomy by which to search for images on the WWW (Linden, Kuhn et al. 2008).

The Kantian Inquirer

In some ways, the Kantian inquirer can be seen as a combination of the Liebnizian and Lockean inquirers. The Kantian inquirer takes some empirical input which is assigned time and space data via a kinematic clock internal to the inquirer. The data is then tested against several mathematical models to see which one provides the best "fit" for the data. Knowledge comes in the form of models and the degree of fit acts as the guarantor (Churchman 1971; Courtney, Croasdell et al. 1998; Linden, Kuhn et al. 2008).

It seems that most traditional decision support systems can be supported by the Kantian inquirer. Forecasting applications also come to mind as being amenable to support from the Kantian inquirer. For example, the National Hurricane Center (NHC) feeds several pieces of data about the wind speed, movement, barometric pressure, etc. of storms into its systems. The system then fits the data to several forecast models such as the NHC98, NOGAPS, UKMET, and the FSU Super ensemble. The NHC's official forecast model is generally chosen from these models based on how well it is performing or, in other words, how well the degree of fit has been between the storm's behavior and the model .

The Hegelian Inquirer

The Hegelian inquirer operates on the epistemological premise that true knowledge is created through the conflict of ideas. The process begins with a worldview, the thesis, which has been deemed insufficient for some reason. This insufficiency leads to the creation of an alternate worldview, the antithesis, which stands diametrically opposed to the original worldview. A debate then takes place between the opposing views. An overseer in the form of a "bigger mind" that has a neutral position with respect to the debate observes it and takes the most salient points from each position. The overseer then synthesizes these points into a new worldview, which consumes the opposing ones and is more valid. The new worldview is the knowledge created by the system and it is guarantor is the overseer (Churchman 1971; Courtney, Croasdell et al. 1998; Linden, Kuhn et al. 2008).

Since the Hegelian inquirer is based in conflict, conflict resolution and negotiation support software seems to be tailor made to be supported by this type of system. Group decision support systems (GDSS) could also fall within this category. A final example is the dialectical decision support methodology presented by Elgarah, et al. (2002).

The Singerian Inquirer

Churchman saved his most complex inquiring system for the philosophy of his own mentor Singer. The Singerian inquirer is based on two premises: a system of measures and a strategy of disagreement. The system of measures is central and is used to settle disputes in the community. When models can no longer adequately explain some phenomenon in the world, the Singerian inquirer engages in this process of "sweeping-in" variables into the models from outside the problem domain to better explain the phenomenon. However, the process does not end here as this explanation will soon be found to be inadequate and more variables will then be swept into the analysis. Churchman referred to this system as having a "grand teleology with an ethical base" (Churchman, 1971, p. 200) and it has the lofty goal of creating exoteric knowledge, or knowledge that can be shared by all humankind. The system of measures as well as the Hegelian overseer acts as the guarantors for the created knowledge (Churchman 1971; Courtney, Croasdell et al. 1998). The complex nature of the Singerian inquirer makes it difficult to find any real examples of this type of inquirer in practice although it has been used in conjunction with other theories such as Simon's decision types or Habermas' theory of communicative action to form the basis for KMS design ((Hall and Paradice 2005; Richardson, Courtney et al. 2006; Linden, Kuhn et al. 2008)

With the Singerian inquirer, Churchman also revisits his nine requirements for systems, this time recasting them in the context of the Singerian inquirer (Churchman, 1971, p. 200).

1. The system has the purpose of creating knowledge that is described to be "exoteric."

2. The system's measure of performance is the "level" of scientific and educational excellence of all society.

3. The client is humankind, i.e., all human teleological beings.

4. The components of the system have traditionally been "disciplines;" this is incorrect if the goal is knowledge that is to be useful to the humans in every society.

5. The system has "fuzzy" boundaries that are necessary for the cooperation that creates inquiry, and in turn, the inquiry that creates cooperation.

6. The decision makers are everyone – in the ideal; the most important of which are the "heroes."

7. The designers are everyone – in the ideal. Progress can be measured in terms of the degree to which the client, decision maker, and designer are become a single entity.

8. The designer's intention is to change the system so as to maximize its value to the client (everyone).

9. There is a built-in guarantor that gives a sense of optimism.

Considering Churchman used this approach to define his most complex inquiring system, this proposal will use a similar method to define the Weickian inquirer in the next section. Comparing the inquirers against Churchman's system requirements, however, is not the only way to assess inquiring systems.

Alternate Views of the Inquirers

In an attempt to make the inquirers more accessible to systems designers, Parrish, Jr. and Courtney translated the features of the inquirers into the language of object oriented programming (Parrish Jr and Courtney Forthcoming in 2008). In the object oriented perspective, inquiring systems are viewed as a class with each inquiring system a subclass of the main class. The processes of each inquirer are viewed as the methods that the subclass possesses, and their inputs and outputs the attributes. Every inquiring system subclass possesses a method called "validation" that will be the object oriented incarnation of the guarantor. The principle of polymorphism allows us to refer to the different validation methods by the same name despite the fact that the guarantor acts differently in each of the systems (Parrish Jr and Courtney Forthcoming in 2008).

As an example of this object oriented perspective, let's revisit the characteristics of the Liebnizian inquirer discussed earlier in this chapter. Here are the characteristics as presented by Churchman (1971, p. 34-35):

1. Innate ideas, i.e., no inputs.
2. Capability of producing strings of symbols that can be broken down into recognizable units.
3. Capability of classifying any unit as a tautology, self-contradiction, or contingent truth.
4. Capability of forming nets of units.
5. Capability of ranking the nets based on a prescribed criterion.
6. A method of processing symbols and building nets such that the system will ultimately arrive at the optimal net and know when it has reached that point.

From an object oriented perspective, this description would break down into the Liebnizian subclass having an attribute of innate ideas. It would also possess the methods of: (1) produce symbols, (2) classification, (3) form fact net, (4) rank fact net, and (5) process optimal net. It is represented graphically in Figure 1 below.

Figure 1: The Liebnizian Inquirer from an Object Oriented Perspective.

Design Science

Bridging the gap from rigor to relevance takes us into the realm of design, which is seen by some as being the central focus of the IS discipline (Markus, Majchrzak et al. 2002). Given the fact that this dissertation is discussing the design of an inquiring system, it is only prudent that Churchman's ideas on design be presented along with the design views of other IS scholars.

Churchman believes that design belongs to the teleological, or goal seeking, family of behaviors (Churchman 1971) In general, Churchman saw design as a thinking process that involved the

selection of an alternative from several possible alternatives in order to attain some goal. More specifically, he believed that design was an activity that is used to better the human condition. Churchman saw design activities as having four characteristics: (1) distinguishing in thought the difference between various alternatives, (2) choosing the alternative that has the best conceptual chance of achieving the stated goals, (3) communicating the alternative in such a way that others can take the conceptual alternative and use it to accomplish the goals, and (4) generalizing the goal to wider applicability. The fourth characteristic is, simply stated, that the more classes of problem situations that can be addressed with a design, the greater the explanatory power that the design has (Churchman 1971).

The design views of Walls, Widmeyer et al. (1992) focused on IS design theory. They believed that an IS design theory consisted of two components, the design product and the design process. The design product begins with the kernel theory, from which meta-requirements are derived. They called their requirements, meta-requirements as opposed to system requirements because they were designed to address not just one problem, but rather a class of problems. These meta-requirements are then utilized to come up with a meta-design. Again, the term "meta" is used because the design is not for a single system, but a class of systems. In addition to specifying the aspects of the design product, the meta-requirements also played a part in the choice of the design method used in the design process (Walls, Widmeyer et al. 1992; Walls, Widmeyer et al.

2004). Finally, the meta-design drove the formulation of testable hypotheses to validate the design theory.

The meta-design however, was not the only thing that determined the design method. The design process itself was also grounded in kernel theory. This kernel theory was used in conjunction with the meta-design to choose the design method. Testable hypotheses were then generated that were applicable to the design method. The design theory process is shown graphically in Figure 2 below.

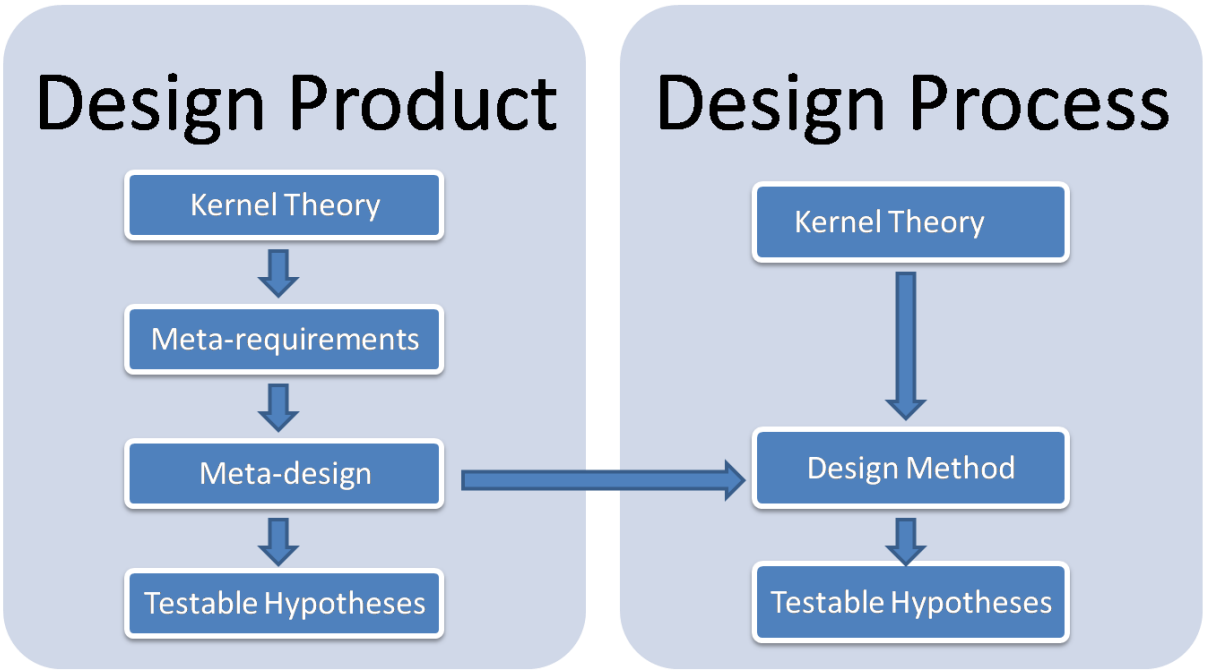

Figure 2: Components of IS Design Theory (Walls, Widmeyer, et al 2004)

Walls, Widmeyer et al., also believed that there were two characteristics to IS design theories. IS design theories had to be grounded in theory (the theory could come from academia or practice) and they had to be relevant to practice (Walls, Widmeyer et al. 1992). Some scholars state that the conception of IS design theory presented by Walls, Widmeyer et al. is not really a radical shift from design thinking at the time (Markus, Majchrzak et al. 2002). However, its main contribution is the extension of design theories to more specialized systems and an extension of the labeling of these types of systems (ex. TPS, EIS, GDSS).

While the approach of Walls, et al. focused on the artifact that the design created, the work of Gregg, Kulkarni et al. (2001) was more focused on the research methods used to attain an artifact. They argue that many times it is difficult to discern software engineering research from application development in the literature. They believe that there is a distinct difference between the two, so they created their software engineering research methodology (SERM). SERM, according to Gregg, Kulkarni et al., consisted of three phases. The first phase was conceptualization, where conceptual requirements were generated. The second phase, formalization, is where the concepts were formalized in the form of DFDs, etc. The final phase, implementation, dealt with the construction of a prototype. Within SERM, the conceptualization phase informed both the formalization and implementation phases, while formalization and implementation informed each other. By laying out SERM in this manner, they believed that

software engineering research could be differentiated from application development and even meet the criteria of testability (or falsifiability) laid out by Sir Karl Popper (Gregg, Kulkarni et al. 2001).

While believing that they had a solid research methodology, Gregg, Kulkarni et al. did not believe that it fit into a single research paradigm. SERM did not really fit into the positivist paradigm, an epistemology that stated that we could know about the outside world through science (or at least know it to some statistical degree), and in which the researcher should remain totally objective. Nor did it fit really into the interpretive paradigm that viewed reality as being socially constructed and in which the researcher should be interactive (Gregg, Kulkarni et al. 2001). Because of this, they created the Socio-Technologist/Developmentalist paradigm. The socio-technologist/developmentalist paradigm stated that reality was technologically created within multiple socially constructed realities. It also stated that researchers could create the context of research, but could inject whatever values that they deemed important. The methodology they chose for this paradigm was development.

The design work of Markus, Majchrzak et al. (2002) differed from earlier work in that instead of having views of the IS design focusing on the type of system, or whether or not it was differentiated from application development, their design theory focused on the type of problem

environment. In environments such as strategic planning or new product development, user requirements and actions have extremely high levels of unpredictability. They believe that these types of environments were beyond what Simon would call unstructured in his continuum of decision types (Simon 1977). They are characterized by a lack of structure for the processes in the environment, user types that are difficult or unable to be predicted, and user information needs that are complex and evolving. Markus, Majchrzak et al. argue that the term "emergent" more adequately describes the knowledge processes that occur in such environments (Markus, Majchrzak et al. 2002). In order to develop their design theory, they used an action research methodology to come up with a set of six guiding principles for the design of systems that deal with emergent knowledge processes (EKP). The design principles that came from their work were:

1. Recruiting naïve users into the design process in order to increase user engagement.
2. Translating knowledge and refining the design through the use of many functional prototypes.
3. Designing the system such that the system output would be used offline as well as online.
4. Integrating the knowledge of the systems experts with the expertise of local employees.
5. Using a dialectical design methodology to implicitly guide the knowledge workers (as opposed to explicitly, which they would have resisted).

6. The componentization of everything in the system so it would be easier to adapt to the changing requirements.

Whereas, the design science research of Walls, Widmeyer et al. guides the design theory for a specific type of design product, is important to note that the principles of Markus, Majchrzak et al. are more applicable to the design process provided that the artifact being designed is to support an EKP.

Design science is research that is intended to both further the academic field and to be relevant to practitioners through the creation of IS artifacts (in the form of theories, constructs, methods, frameworks, or instantiations) using rigorous methods. These artifacts also provide guidance to practitioners (Hevner, March et al. 2004). Because of this, the IS theory that results from design science research can be seen as normative, meaning that it must not only pass the tests of academia, but it must also pass the tests of practice (Markus, Majchrzak et al. 2002).

Following this theme, it is a goal of this research to achieve not only an IS theory that can be deemed rigorous, but also one that is relevant. Adams and Courtney (2004) believe that design science can be used in conjunction with other methodologies such as action research, grounded theory, and software development to create better IS theories and to gain more relevance to IS practice. The DAGS (D – Design Science, A – Action Research, G – Grounded Theory and, S-

Software Engineering) framework incorporates the aforementioned methodologies into a framework similar to that of Nunamaker's (1991) multi-methodological research framework. Adams and Courtney make a real distinction between the two activities supported by the DAGS framework, theory building and theory testing. According to Adams and Courtney (2004) design science and grounded theory are appropriate for theory building activities and action research and software development are appropriate for testing and refining theories. The methodologies used for each are chosen based on what is more appropriate for the situation.

Hevner, March, et al. (2004) submit that there are two distinct research paradigms in information systems. There is the behavorial paradigm that has at its core the search for truth and the design science paradigm that seeks to find utility. They feel that the goal of IS research can also serve the dual roles of rigor and relevance by taking business needs from the environment and applicable knowledge from the knowledge base of academia and using them in conjunction with one another to guide the construction of design theories and or artifacts that are assessed and refined through the use of rigorous evaluation methods. The product of this research is relevant because it can be applied to help remedy the business needs that drove its creation. It is also rigorous because it is often based in theory and the evaluation of the design theory can advance academic knowledge (2004). Hevner, March, et al. (2004) provided seven guidelines for design science research that are summarized in Table 4 below and described in more detail in the following paragraphs.

Table 4: Design Science Guidelines of Hevner, et al. (2004).

Guideline	Summary
Design as an Artifact	Design science research should produce a viable IS artifact.
Problem Relevance	Design science research should address important business problems.
Design Evaluation	The characteristics of the design science research artifact should be evaluated by rigorous methods.
Research Contributions	The product of the design science research must make a clear contribution that is also verifiable.
Research Rigor	The construction and evaluation of design science artifacts must adhere to rigorous methods.
Design as a Search Process	Design science research uses any available means to find a solution. It is often an iterative process that may not achieve optimal results.
Communication of Research	The product of design science research must be able to be communicated to both academics and practitioners in an understandable manner.

The seven design science guidelines of Hevner, et al. (2004) begin with the fact that design science research must produce an artifact in the form of a construct, a method, a model or an instantiation. They feel that the information systems artifact is at the core of the subject matter examined by all information systems research and that they define the characteristics that provide for all phases of information systems development including analysis, design, implementation and use. Problem relevance speaks to the notion that the problem should be interesting and relevant to business and must be the primary motivation for the research. Rigorous design evaluation provides feedback to the design science researcher that allows them to generate valuable feedback that will assist with the artifact's refinement. There are several methods that can be used to evaluate artifacts such as:

- Observational – such as case studies or field studies.
- Analytical – such as static or dynamic analysis.
- Experimental – such as controlled experiments or simulations.
- Testing – such as black or white box testing.
- Descriptive – such as informed arguments or scenario generation.

Research contribution addresses the value of the contribution to the business and academic environments. One part of the criteria regarding the value of the contribution ties back whether

or not the problem is relevant. In other words, if the information systems artifact has value to businesses in the context of some problem, then the researchers have made a contribution to practice. Research rigor dictates that the artifact must be constructed and evaluated using rigorous techniques applicable to the artifact being constructed. Note however, that despite the fact that the artifact must be constructed and evaluated rigorously, it still must be relevant. Designing as a search process recognizes that good design is an iterative process that is subject to the principle of bounded rationality (Simon, 1996). This means that the researcher will never be able to reach the optimal solution because considering all possible solutions is not feasible or even possible. Therefore, the design science researcher may have to practice satisficing and take the best solution available, even if it is only satisfactory. Finally, the fruits of the design science researcher's labor should be able to be communicated to both academics and practitioners in such a way that practitioners can replicate the artifact in their own organizations and academics can recognize the contribution that has been made to the overall pool of knowledge (Hevner, et al., 2004). The guidelines of Hevner, et al. serve as guideposts to plan and evaluate the entire process of design science research to help achieve results that are both rigorous and relevant.

It is important to discuss design in this dissertation to show the varying perspectives from which it has been addressed and to set the context for the remainder of the dissertation. Churchman viewed it from a philosophical and ethical standpoint, and argued that its purpose was to better the human condition, Walls, Widmeyer et al. focused in on the theory behind the design and the

methods used to generate such theory, the importance on differentiating design research from application development was addressed by Gregg, Kulkarni et al., Markus, Majchrzak et al. showed that the context of the design environment was also something that had to be considered. Finally, both Adams and Courtney (2004) and Hevner, et al. (2004) argue that design science can be employed to achieve not only the goal of academic rigor, but it can also provide a contribution that is relevant to industry. Adams and Courtney argue that achieving this dual goal can be achieved by combining and applying appropriate methodologies to the problem space and Hevner, et al. (2004) believe that it can be achieved by adhering to a set of guidelines that govern the research process.

Despite the fact these different scholars approach design science from different angles, this dissertation takes the position that the approaches are not mutually exclusive. In fact, this dissertation uses the work of Walls, Widmeyer, et al. (1992; 2004) to build an IS design theory that is then used to create an artifact in the form of an instantiation of the theory that adheres to the guidelines of Hevner, March, et al. (2004). The next chapter details how the characteristics of the Weickian system were derived and how they were used along with the work of design scientists to develop the system requirements and design principles that guided the development of an information system that reflects these principles named the SenseMan system.

CHAPTER THREE: THE DESIGN PRODUCT AND DESIGN PROCESS

Design science often involves the use of kernel theories to guide the development of the requirements and principles of a design theory (Walls, Widmeyer et al. 1992; Markus, Majchrzak et al. 2002) and this dissertation is no exception. However, for this paper there was no readily available kernel theory that could, by itself, provide adequate guidance for the development of a design theory. That being said, it was necessary to build a kernel theory for an information system that would generate knowledge via sensemaking. It was decided that this could be achieved by synthesizing the concept of sensemaking with the concept of information systems to come up with a new type of inquiring system that would serve as an applicable kernel theory.

The process to specify the design features of the system in this dissertation is very similar to the process advocated by Walls, Widmeyer, et al. (1992; Walls, Widmeyer et al. 2004). The kernel theory that was developed was used to guide the construction of the system requirements and those requirements then guided the creation of the system design principles. The work of Markus, Majchrzak, et al. (2002) on EKPs was used as the kernel theory for the design process, which drove the selection of the applicable design method for systems that support that type of knowledge. This chapter begins with a discussion of the kernel theory and how it can be justified as a system. This is followed by details on the design requirements and the initial design

principles used to begin the development of the information system. After setting the context for the design process by providing a description of the organization that agreed to be a partner in the research study, the process of the design that lead to the final design principles is discussed. The chapter concludes with a presentation of SenseMan, the actual information system that resulted from the design science project.

Building the Kernel Theory

Considering that this dissertation is attempting to justify the creation of a new type of inquiring system, I can see no better way to answer the first question than to evaluate the principles of sensemaking against Churchman's system requirements to see if it is appropriate for inclusion as a basis for an inquirer. Sensemaking will be compared to each of the requirements, the results of which are summarized in Table 5 and then discussed in the paragraphs below.

Table 5: Sensemaking and Churchman's System Requirements (Churchman 1971).

Churchman's Requirements of Systems	How the requirement is satisfied by sensemaking
1. *S* is teleological.	The goal is to make "sense" of a problem environment. To gain knowledge about the environment thereby reducing uncertainty.
2. *S* has a measure of performance.	The reduction of perceived uncertainty due to the sensemaking product is the measure of performance of *S*.
3. There exists a client whose interests are served by the system in such a manner that the higher the measure of performance, the better the interests are served and, more generally, the client is the standard of the measure of performance.	Any person or organization that is affected by the complexity and uncertainty is considered a client. The more clearly the sensemaking product depicts the problem environment, the greater the reduction in uncertainty will be.
4. There exists an environment which co-produces the measure of performance of S.	The environment and its level of complexity have a great effect on the way sensemaking is performed, and on its results.
5. S has teleological components which co-produce the measure of performance of *S*	The principles of sensemaking are the teleological components that co-produce the measure of performance of *S*. For example, if the sensemaker is not effectively extracting cues from the environment, the sensemaking will be affected.
6. There exists a decision maker who – via his resources – can produce changes in the measures of performance of *S*'s components, and hence changes in the measure of performance of *S*.	The Organization managers are the decision makers in *S* because it can wield its resources to produce changes in the system.
7. There exists a designer, who conceptualizes the nature of *S* in such a manner that the designers concepts potentially produce actions in the decision maker, and hence changes in the measures of performance of *S*.	The sensemaker is also the designer. The sensemaking process states that the sensemaker can enact sensible environments, therefore influencing the decision maker.

Churchman's Requirements of Systems	How the requirement is satisfied by sensemaking
8. The designer's intention is to change *S* so as to maximize *S*'s value to the client.	The sensemaker seeks to form as plausible a depiction of the environment as possible in order to base actions.
9. *S* is stable with respect to the designer, such that there is a built in guarantee that the designer's intention is ultimately realizable.	The point of sensemaking is to come up with a plausible explanation of the decision environment. Because the explanation must only be plausible, and not accurate, the intentions are realizable. The guarantor of the process is consensus.

However, the table above only presents how sensemaking meets the requirements of churchman in the general sense. The following paragraphs examine each requirement as it relates to sensemaking in more detail.

Is it teleological? – Weick (1993) writes "The basic idea of sensemaking is that reality is an ongoing accomplishment that emerges from efforts to create order and make retrospective sense of what occurs." This quote demonstrates that sensemaking is teleological in that it has the goal of creating order and making sense.

Does it have a measure of performance? – In describing sensemaking, Maltis (2005) states that "…sensemaking allows people to deal with uncertainty and ambiguity by creating accounts that enable action. Sensemaking thus precedes decision making and follows it…" Since

sensemaking is to enable decision making through the reduction of complexity, it is reasonable to assume that the reduction of perceived complexity in the environment could serve as a proxy for the measure of performance for the sensemaking process.

Is there a client? – Again, given that sensemaking is tied to a reduction of uncertainty and complexity, it seems that anyone that stands to be affected by the uncertain and complex environments could be considered a client. However, in the organizational sense, the organization that is benefiting from the sensemaking products of the individual sensemakers would be the client.

Does it have teleological components? – This proposal argues that the principles of sensemaking themselves can be considered teleological components. For example, the degree to which a sensemaker is effective in extracting cues from his or her environment affects the degree of performance of the entire sensemaking process. Weick states that in times of increased complexity, we tend to go with the "tried and true" cues from our environment. However, by doing this, we often blind ourselves to other cues that can help us to construct a more effective account (Weick 1995). The process is also teleological itself because it has the goal of extracting cues from the environment which will be used to construct the sensemaking product.

Does it have an environment? – The answer here is, of course, yes. Sensemaking is rooted in the environment of the sensemaker and the more complex the environment, the more difficult making sense of it will be. Thus, environment does co-produce the measure of performance of sensemaking.

Who is the decision maker? – Churchman (1971), defines the decision maker as an entity that can affect the measures of performance via its resources, and the designer produces actions in the decision maker. Given the purpose of the sensemaking inquirer is to generate environmental accounts to rationalize complex environments to allow for better organizational actions, this role is played by the organization. In other words, in order for the organizational managers (decision maker) to take action, the environment must be rationalized by a sensemaker, a group of sensemakers such as a department or other organizational unit, or the organization itself (designer).

Does it have a designer? – One of the principals of sensemaking is that it enacts sensible environments, meaning that the sensemaker has some slight influence over his or her environment (Weick 1995). Given that this dissertation has already made the argument that the organizational managersare the decision makers, it can also argue that the sensemaker is the

designer in that they can potentially produce actions in the decision maker and thus, affect the performance of the system.

What are the designer's intentions? – Given that the goal of the sensemaking process is to enable action by reducing uncertainty, and that the client is the one affected by those actions, the designer is attempting to maximize the value to the client by attempting to achieve the goal of the sensemaking process.

Is there a built in guarantee that the designer's intention is realizable? – One of the principles of sensemaking is that it seeks plausibility over accuracy (Weick 1995). If the goal of the process was accuracy, then this requirement would be violated as no human can have a complete comprehension of his/her environment. However, since it is only plausibility that we seek in sensemaking, the guarantee is there that it is realizable. However, is there a guarantor for the sensemaking process to ensure that the sense that is being made is plausible? This proposal posits that the guarantor is a majority consensus, a notion that Weick supports when he writes "Sense may be in the eye of the beholder, but beholders vote and the majority rules" (Weick 1995). As you may recall from the earlier section on the inquiring systems, consensus was also the guarantor for the Lockean inquiring system. Consensus in the case of the Weickian inquirer

however is a majority consensus, not the absolute consensus that is required in the Lockean system.

Although the Weickian inquiring system shares the same style ofguarantor with the Lockean inquirer, it is still an independent system because of the differences in epistemology. In fact, overlapping of characteristics in the inquiring systems is not a new concept. For example, the Kantian inquirer is often seen as a combination of the Liebnizian and Lockean inquirers (Courtney, Croasdell et al. 1998; Courtney 2001; Linden, Kuhn et al. 2008; Parrish Jr and Courtney Forthcoming in 2008). The epistemological stance of the Weickian inquirer is called the evolutionary epistemology and is based upon comparing the process of theorizing with sensemaking. The evolutionary epistemology basically states that we know that the world is not static and that we deal with its dynamic nature by creating accounts based on the ongoing data that we collect from our environments (Weick 2004; Antoft and Salomonsen 2007).

Because sensemaking did not violate any of the requirements that Churchman put forth for systems, the first question posed in this dissertation can be answered in the affirmative. Sensemaking can be viewed as an appropriate foundation for the construction of an inquiring system.

This however, only answers the first question. To answer the other two questions that are asked in this dissertation we must cross into the realm of design. In the upcoming sections of this chapter, the kernel theory developed here is used in conjunction with other theories as the foundation on which the initial design principles for a system to support sensemaking in organizations were developed. The next section details the generation of the design requirements that then guided the development of the initial design principles for the system.

Generating the Design Requirements

Walls, et al. (1992) state that action research in addition to iterative hypothesis development is an appropriate way to construct new IS design theory. This process involves generating system requirements from the chosen kernel theory or theories and then developing hypothesized design and development principles that reflect these requirements. In order to generate the initial design requirements for the Weickian inquiring system, we really needed to look no further than the principles of sensemaking process. In this section, each principle will be examined in the context of information systems design and design requirements were derived from these principles. The principles are summarized in Table 6 and then their derivation is discussed in the following paragraphs. For each principle, we will first revisit the meaning of the

sensemaking principle and then discuss the implications that it had on the design requirements of the Weickian inquirer.

Table 6: Design Requirements of the Weickian Inquiring System.

1.	The system must support multiple identities and perspectives.
2.	The system must provide a means for capturing the sensemaking products of its users.
3.	The system must make historical environmental information and sensemaking products available to users.
4.	The system must be able to display information composed in the present at some future date and time chosen by the user.
5.	The system must allow for users to interact with one another.
6.	The system must be constructed so that it is always available to the users.
7.	The system must present the users with a mixture of information that they choose as well as information that they did not choose to serve as environmental cues for the sensemaking process.
8.	The system must protect the anonymity of the users.

Grounded in identity construction – A sensemaker is needed and the identity of that sensemaker is the product of many selves. In other words, we must reconcile the many different identities and perspectives that make up the individual to create the sense that we make. The identity of the sensemaker also affects the sensemaking process and the results of the process are based on the perspective of the sensemaker (Weick 1995). From a design standpoint, this means that many different types of people with varying perspectives could be using the system and their interactions with the system could produce very different results. Because of this, we can derive our first design requirement.

- **Design Requirement 1**: the Weickian inquirer must be able to support many different organizational identities and perspectives.

Retrospective – it is by reflecting on past experiences that we are able to construct accounts that allow us to make sense of the present and to make plausible predictions about the future (Weick 1995). With regards to systems, this means that the system must be able to provide retrospective information to its users. From this, we can derive our second and third design requirements.

- **Design Requirement 2**: the Weickian inquirer must provide a means for capturing the sensemaking products of its users.

- **Design Requirement 3:** the Weickian inquirer must make historical environmental information and sensemaking products available to users.

Enacts Sensible Environments – This principle states that by attempting to make sense of our current environments we can subsequently exert some small amount of influence over our future

environments (Weick 1995). With regards to system design, this principle seems to imply that some sort of feedforward mechanism should be included in the system that could allow others to provide suggestions to other users or reminders to themselves about actions that might be taken in the future based on the sense that they attained in the present.

- **Design Requirement 4:** The Weickian inquirer must be able to display information composed in the present at some future date and time when the context dictates that information be made available.

Social – The interactions that we have with other sensemakers influences our own sensemaking products and the sense that we produce can affect the sensemaking products of others (Weick 1995). As far as design implications go, this requirement calls for the inclusion of the capability for users to communicate and interact with each other so that they are made aware of the perspectives and sensemaking of others.

- **Design Requirement 5:** The Weickian inquirer must allow for users to interact with one another.

Ongoing - There is no stopping rule for sensemaking, the reality of the present is constantly revealing itself to the sensemakers and providing not only instances for sensemaking but data on which to base retrospective accounts (Weick 1995). From a design standpoint, this means that the system will have to account for the fact that sensemaking does not just occur during business hours.

- **Design Requirement 6:** The Weickian inquirer must be constructed so that it is always accessible to the users; active use must be encouraged.

Based on Extracted Cues – Even though there are many potential cues in our environment on which to base our sensemaking, sensemakers will tend to focus on certain ones (Weick 1995). This tends to lead one to believe that different users of the system will look for different types of information on which to base their sensemaking. However, as our environments become more complex, our tendency is to look toward the cues with which we are familiar, which can lead to our being mislead (Weick 1995). So the information that is presented must not become routine.

- **Design Requirement 7:** The Weickian inquirer must present the users with a mixture of information that they choose as well as information that they did not choose to serve as environmental cues for the sensemaking process.

Focused on plausibility rather than accuracy – The goal is not to create accounts that exactly portray the environments that we are attempting to make sense of, all we need is something plausible. From a design standpoint, getting users to give a "best guess" and to share those guesses with others could be difficult because of a fear of negative consequences that may occur if their guesses are erroneous or unpopular with other users. Therefore, users will have to be assured that if they share their sensemaking product, they will be protected from the negative consequences that may be brought about by their participation.

- **Design Requirement 8**: The Weickian inquirer must protect the anonymity of the users.

As we discussed in the earlier chapter, inquiring systems can also be viewed from an object oriented perspective (Parrish Jr and Courtney Forthcoming in 2008). In the case of the Weickian inquirer, the system subclass would have attributes of environmental cues, user preferences, and perceived complexity. The methods that the system would support would be (1) capture sense, (2) display information, (3) store message, (4) deliver message, (5) store preferences, and (6) the validate function shared by all inquiring systems. Regardless of how they are viewed, the purposes of these design requirements are to guide the development of design principles for the system in question. The upcoming section details the design process that took place that resulted

in the development of the design requirements for systems to support organizational sensemaking beginning with the theoretical justification for the process that was used, and then moving on to a discussion on the derivation of the initial design principles and how those design principles were refined in the design process.

Constructing the Design Principles

As stated earlier, the design process for the artifact was guided by the work on emergent knowledge process design by Markus and her colleagues (2002). This work was deemed especially applicable since the three characteristics (process structure, user types, and user information needs) that are supported by the design of the artifact exhibit the qualities of emergent knowledge processes. Since sensemaking is often linked to strategy (Gioia and Chittipeddi 1991; Thomas, Clark et al. 1993; Gioia and Thomas 1996; Schneider 1997) , which is an emergent process. Since the process of sensemaking can have no real structure, it seems that sensemaking may be deemed an emergent process as well. Additionally, it is impossible to predict the types of users that will actually undergo sensemaking and whether or not they will use the tools provided to assist with it. Finally, in describing the information needs of users in emergent knowledge processes, Markus, et al. cite Weick's (1995) assertion that much of the

knowledge in sensemaking is tacit and hard to transfer to others in a codified sense. Much of it is expert knowledge that must be transferred as conditional rules such as "if – then" statements (Markus, Majchrzak et al. 2002). Now that the case has been made for sensemaking as an emergent knowledge process, thus providing the guidelines for how the system will be developed, we can turn to actually looking at what will be developed.

The overall system design principles were originally derived from our kernel theory and later enhanced by the work of Weick and Meader (1993) in group support systems (GSS) and sensemaking. To review, their work argued that typical GSS design was focused on the decision and not the actual process of supporting the group and its sensemaking. While some GSS do support sensemaking indirectly (especially the strategies of action and deliberation), the notion of a system designed to support sensemaking had not yet been asked by research (Weick and Meader 1993). Because of this, it was decided that the final artifact would be a GSS that supports all of the sensemaking strategies elaborated on by Weick and Meader as well as one that supports the design requirements generated from our kernel theory.

The following paragraphs will describe the design process and how the design principles evolved through the process. After a discussion on the design process, the actual artifact will be presented and there will be a discussion on how the final design supports the strategies of

sensemaking from Weick and Meade as well as how it reflects the qualities of the design process for EKP. Finally, the actual artifact will be presented and there will be discussion on how it is reflective of the final design principles.

The Initial Design Principles

One of the purposes of an IS design theory is to provide guidance to practitioners (Hevner, March et al. 2004). As this proposal moves from theory building to theory testing and refinement, it is appropriate that we use our theory to guide us in the construction of our artifact. In this section, we will look at how our design specifications satisfy the design principles for the Weickian inquirer. The specifications will be summarized in Table 7 and then discussed in the following paragraphs.

Table 7: Design Principles.

	Design Requirement	Design Principle
1.	The Weickian inquirer must be able to support many different organizational identities and perspectives.	User preferences will be stored in order to customize the inquirer to the various identities of the users.
2.	The Weickian inquirer must provide a means for capturing the sensemaking products of its users.	The system will accept user input and will be connected to a database that will provide storage.
3.	The Weickian inquirer must make historical environmental information and sensemaking products available to users.	The system will query the database using SQL to present the stored information to the users.
4.	The Weickian inquirer must be able to display information composed in the present when the context dictates that information be made available.	Alert functionality will be built into the system to notify users of combinations of environmental cues that have sensemaking accounts attached to them, or when the assumptions that are critical to the sensemaking activity have changed.
5.	The Weickian inquirer must allow for users to interact with one another.	The system will support interaction between users.
6.	The Weickian inquirer must be constructed so that it is always available to the users.	The system will be constructed on a web platform.
7.	The system must present the users with a mixture of information that they choose as well as information that they did not choose to serve as environmental cues for the sensemaking process.	Users can choose cue streams that will be stored in their preferences. Additionally, random cue streams will also be presented.
8.	The system must protect the anonymity of the users.	User information will not be presented or stored with the user inputs.

The system must support multiple identities and perspectives – The system will have to account for the fact that different people with potentially very different styles of sensemaking will be using the system. While identities can be maintained through the use of profiles, perspectives are

a more difficult matter. It is helpful to us that perspective can be roughly equated to the sense that we make of a given situation or environment, and that the sense that we make is grounded in the self and socially constructed identity that we take on. So, our identity in part determines our perspective, which is then indicated in the sense that we make. In order to account for this, the system will allow for some user customization in the form of preferences that will allow them to tailor the system to their specific individual identities. The perspectives that are partially attributable to these identities, will be maintained through storage and review of the sensemaking products that are reflective of the perspectives.

The system must be able to capture the sensemaking products of its users – The system will be designed so that users will have the opportunity to provide accounts and to associate them with various environmental cues so that the system can store the products of their sensemaking. Additionally, system events can be captured and associated with pre-defined environmental cues in an automated manner. The system will be constructed using a three-tier architecture with the data layer consisting of a Microsoft SQL Server 2005 database server that will enable the storage of the captured sensemaking product.

The system must make historical environmental information and sensemaking products available to users – The system can be designed so that the database is queried using Transact

SQL to provide relevant sensemaking products to users when current environmental conditions match sensemaking accounts stored in the system. For example, let's imagine that the system was being used in a hospital emergency room in Orlando, FL and one day the temperature rose to 98 degrees the same day a marathon took place. The nurses that worked in the emergency room stored an account that the high temperature, coupled with the outdoor event caused an increase in the number of patients needing treatment for dehydration. The next year, when an abnormal amount of patients are needing treatment for dehydration, the system could make that account available to users, prompting them to make sense of the current situation.

The system must be able to display information composed in the present at some future date and time when the context dictates that it be displayed – there may be some times when a user may want to manually prompt other users to engage in sensemaking based on some environmental cues that may be present at some future time. The system will support this by allowing the users to store alerts in the system and having those alerts fire at a pre-determined time.

The system must allow for users to interact with one another – since sensemaking is a social process (Weick 1995; Maitlis 2005), the system has to have some means to allow for the social interaction of its users. Some methods by which this can be accomplished are enabling eMail, IM, or chat-room functionality within the system.

The system must be constructed so that it is always available to the users – sensemaking is an ongoing process (Weick 1995) and there is no way to know when an environmental cue may be noticed that inspires an account that would be of value to making sense of an organization's environment. Because of this, the system will be web based and ASP.NET will be used as the programming language. It will be up to the host organization(s) whether or not the system will reside outside of the organizational firewall and be accessible to all those that have access to the URL or if it will be hosted as an extranet application. An extranet application is a web application that resides behind the firewall, but is accessible to external users through such technologies as virtual private networks (VPN).

The system must present the users with a mixture of information that they choose as well as information that they did not choose to serve as environmental cues for the sensemaking process – because sensemakers tend to go with the familiar in times of complexity (Weick 1995) or to disregard evidence that disconfirms their expectations when they feel that they have a good grasp of their environments (Weick and Sutcliffe 2001), environmental cues should be injected into the system that are not chosen by the sensemakers. However, they will be allowed to choose some of the cues that they see in keeping with the first design principle, support for multiple individuals and perspectives.

The system must protect the anonymity of the users – user information will not be associated with the sensemaking accounts stored in the system.

The preceding design specifications were the initial foundation on which the Weickian inquiring system was constructed in its artifactual form. The following paragraphs describe the process that took place that took these initial hypothetical design principles from the conceptual realm across the bridge that leads to practice in as an instantiation of a functional information system.

The initial design requirements were derived from the kernel theory of the Weickian inquiring system described earlier in this work. Over a period of four months these principles were refined through the development of different prototype systems that were displayed to different user groups and then amended based on their input. The organization that agreed to assist with the development of the artifact was looking to implement a change management initiative and felt that the proposal for the ChangeMan system (as it was then named, later changed to SenseMan) would help them with this endeavor, especially in the area of software updates. The organization that agreed to assist us was the Clerk of the Circuit and County Court Office of Lake County, FL. To provide better context for the remainder for this discussion on the process of developing the

artifact, it would be prudent to provide an overview of the organizational environment and its information systems.

The Design Setting

The design artifact was implemented in the office of the Clerk of the Circuit and County Courts of Lake County, Florida. The organization is headed by the Clerk of Courts, who is an elected official, and has approximately 300 employees. Its major responsibility is providing many different record-keeping and financial services to the public as well as other government agencies on state and local levels. The duties of the Clerk of Courts as mandated by the Florida Constitution are:

- Clerk of the Circuit Court
- Clerk of the County Court
- County Comptroller/Treasurer
- County Auditor
- County Recorder

- Secretary/Ex-officio Clerk to the Board of County Commissioners

Additionally, the Clerk of Courts of Lake County, Florida, is also responsible for the following duties outside of the constitutionally mandated duties:

- Secretary/Treasurer for the Lake County Water Authority
- Secretary/Treasurer for the Lake County Law Library
- Treasurer for the Lake County Historical Society
- Treasurer for Lake/Sumter Emergency Medical Services, Inc.
- Agent for Passport and Documentary Stamps

As you can see, the Clerk of Courts has many responsibilities. In order to fulfill these varied responsibilities the Clerk's office is divided into several departments. These departments are:

- Administrative Services
- County Finance
- Courts Management
- Executive

- Information Resources

The Administrative Services department consists of the divisions of Records Management, Administrative Support, Recording and Indexing, and Support Services. The Records Management Division performs several functions for court and official records including annual auditing, microfilming, and destruction of inactive records in accordance with retention requirements. Administrative Support houses such functions as human resources and purchasing. The Recording and Indexing division is an important division with the responsibility of processing all documents that will be recorded into the public record such as deeds and satisfactions of mortgages. The Recording and Indexing division also provides research assistance to those who are searching for information within the public record. The Support Services division has diverse responsibilities including the processing of passport and marriage license applications and maintenance of the physical facilities.

The County Finance department is divided into three divisions. Board Accounting is responsible for providing financial and accounting services for the Board of County Commissioners and the County Manager. In addition to providing financial services to the Board of County Commissioners, the Clerk also processes the minutes of the meetings of the Board of County Commissioners and any of its committees. The Board Support division is responsible for

making sure that these minutes are accurate and concise. The Clerk Finance division provides all accounting and budgetary functions for the Clerk of Courts.

Courts Management is the largest of the departments and provides a variety of services to assist the public, attorneys, and judges with court related functions. Because the functions related to managing these functions are so varied, the department is broken into four divisions that are each comprised of multiple areas that provide specific functions in each area. For example, the Family Law division has areas dedicated to child support, probate, guardianship issues, mental health cases, domestic relations, injunctions, and juvenile dependency and delinquency. Other divisions include: Civil Law, responsible for filing documents with the court and for civil court cases in the county and the judicial circuit; Criminal Law which is responsible for criminal felony and misdemeanor cases and managing the jury; the Traffic division that is responsible for traffic related court cases; and the Law Library division that provides legal resources to attorneys and the public.

The final two departments are the Executive department and Information Resources. The Executive department ensures that the statutory obligations of the office are being met also provides the general management function for the office. The Information Resources department supports the information systems of not just the Clerk of Courts, but also the Lake County

Sheriff's department, the Board of County Commissioners, and the Judiciary of Lake County. They are also responsible for emergency and disaster preparedness. The Information Resources department is comprised of two major divisions. The Application Development and Support division is responsible for supporting the various user groups with their PC and peripherals, the development of applications to support the mission of the Clerk (in-house development and outsourced development), and managing the Clerk's information systems projects from a project management perspective. The Network Support and Security manages the technological infrastructure of the Clerk's office including the servers and network. They also maintain the databases and various server applications that are utilized by the Clerk's Office such as mail server, and the web server. Finally, the Network Support and Security department is responsible for the overall security of the Clerk's information systems. The overall organizational structure of the Clerk's office is depicted in Figure 3 below.

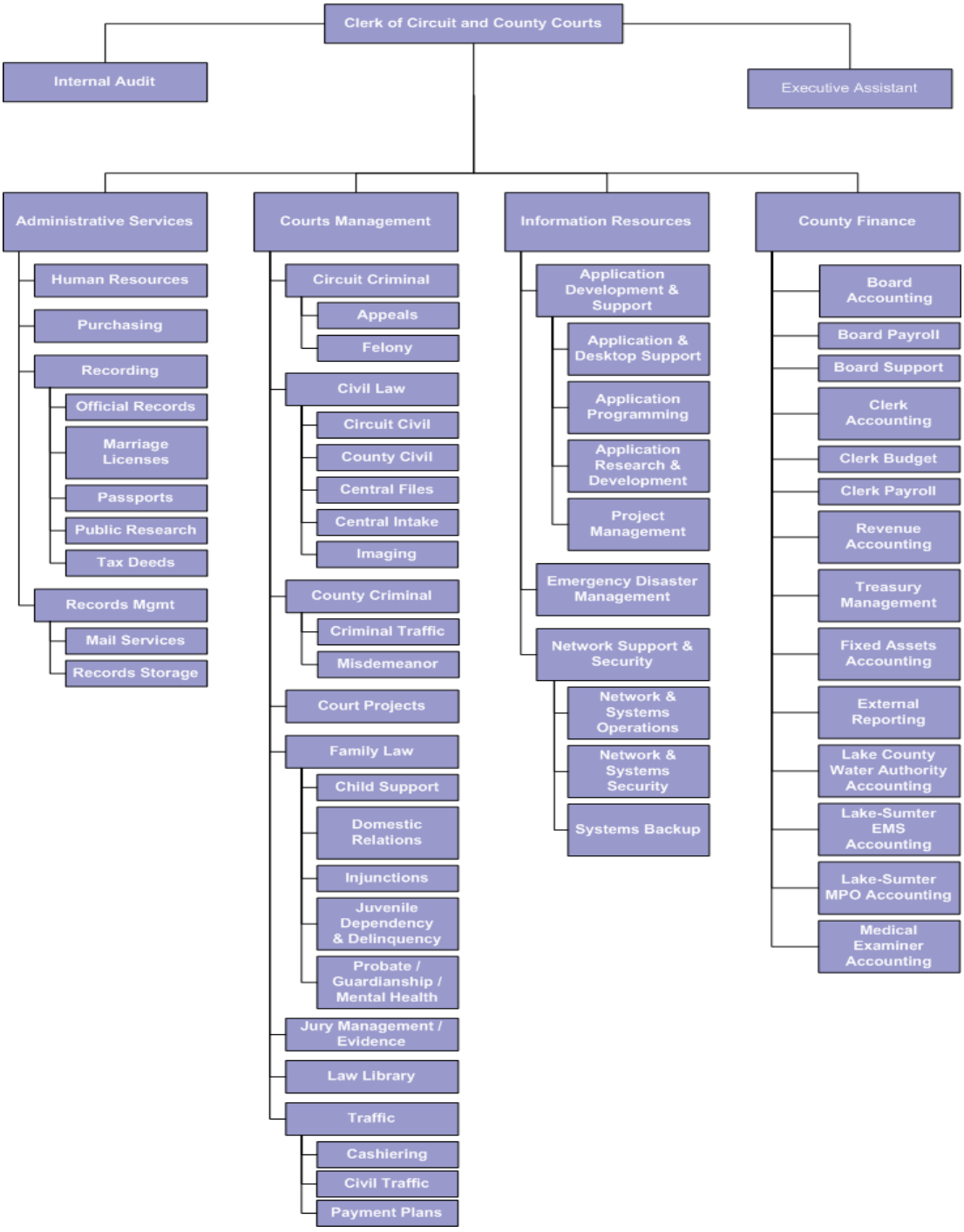

Figure 3: Lake County Clerk of Courts Organizational Chart.

Although the Clerk's Office employs several computer technologies and software to carry out its mission, this dissertation will focus on four applications. These applications are:

- CourtView
- New Vision
- Munis
- TrakMan

All of these applications were purchased from vendors and any system changes generally are processed through the vendor and then tested in the Clerk's test computing environment before going live. The following paragraphs discuss these applications in more detail.

CourtView is a courts management software solution. It is used primarily by the Courts Management department to add and edit case information for all of the different kinds of court cases such as information about the parties on the case, docket entries, and collections of fines and fees. In addition to Courts Management using CourtView, it is also used by the Support Services division of the Administrative Services department to process marriage license requests. It is also used by other departments and the public to query information on court cases via a web interface.

New Vision is the application that is used to record and index official records information. When documents are submitted to the public record, Clerk employees use the program to scan the documents, input the index information, as well as to verify that the index information and documents are accurate.

Munis is an enterprise resource planning (ERP) class system that is used by the County Finance department to assist with the general financial management and accounting needs of the organization. It is also, however, used for functions housed in the Administrative Support division of the Administrative Services department. Human Resources uses the application's HR module to assist with the human resources needs of the office, and Purchasing uses the application to support the purchasing function.

TrakMan is an elaborate document management system that stores the location of documents and items within the organization, administers check-in and check-out functionality of documents, and maintains information about the documents such as destroy dates once the retention requirements have expired. TrakMan is used by the Information Resources department to log the location of assets. It is also used by the Courts Management department to request case files in preparation for trial and to scan and redact court documents. Finally, the application is used

extensively by the Records Management division of the Administrative Services department to support the various functions related to the management of court and official records.

Just as these applications are depended on by their users to perform their job tasks, they are also dependent on each other. CourtView interfaces with the TrakMan system to receive images of court documents, to request that case files be pulled to take to trial, and to automate the recording of civil judgments that will eventually become part of the official records. These civil judgments are just one of many documents that serve as a point of interaction between TrakMan and NewVision. The eRecording function of TrakMan takes documents from the web and automatically indexes and records them in the New Vision system just as if they were brought to a Clerk employee. The images from New Vision of recorded documents are also sent to TrakMan to redact sensitive information from them before being allowed to be released to public view on the Clerk's Internet site. Finally, both CourtView and NewVision generate files from their prospective financial modules and send them to the Munis application for insertion into its general ledger module.

With regards to updating the software programs, many of the users are currently involved with the evaluation of software updates, have been in the past, or will be in the near future. This is because the Clerk's office does not have a static group of personnel that evaluates the updates.

Instead, users are often brought into the evaluation process on a basis of their schedule availability, or the applicability of the update to their specific job function. Because of the interrelatedness of the applications, there are times when an update to one application may involve testing of one or more of the other applications. Additionally, as a result of their interdependence on each other, the applications act more like modules within a larger software application than they do stand alone, independent software applications. Marakas (2006) defines a system to be a set of interrelated elements, with an identifiable boundary, that work together towards some common goal. We have already seen that the applications are interrelated. The system is bounded because it can be defined within the context of other systems as the hardware, software, and users that interact with the Clerk's office, and the applications all work together to serve the mission of the Clerk's office. From this description, it is easy to see why the organization was looking to get control of change management issues with regards to updating their software and why an application that is designed to enhance sensemaking would be valuable in this context.

The Design Process

The development of the SenseMan system began as a meeting with the senior staff of the Clerk's office where the initial system concepts as well as the outline for the research project were

presented. At this meeting, some of the members voiced their concerns about the development process being too invasive in the face of budget cuts in Florida that were affecting all government agencies. However, that being said, they could see some potential value in the proposal and with the blessing of the elected official that heads the agency they agreed to become partners on the research project.

Since the primary researcher was a former employee of the organization, he had prior knowledge of how the software update process worked and utilized that in the initial design. However, much had changed since his departure and the applications had become much more interdependent than when he was employed there. So it was necessary to spend some time analyzing the update process and the applications involved. Once this analysis was complete, the initial prototype was conceptually constructed and an initial instantiation was created.

The initial prototype was shown to several Clerk staff and refined based on their input. As more input was gathered, the system that had initially been conceptualized as a group decision support system (GDSS) was deemed to be not necessarily what was needed to enhance sensemaking organization. This realization came about for two reasons that presented themselves almost simultaneously. One reason was that after demonstrating one prototype to the Chief Deputy of Information Resources, he felt that the system might conflict with the change management

initiatives that they had begun. At that time, the ChangeMan system's (as it was then referred to) design principles, dictated that the prototype design should collect user input that would be stored and utilized in conjunction with alert functionality that would provide the impetus for immediate decision making (Design principle 4). This was in conflict with the organization's plan to have the decisions for updates to be made by the change management board, in accordance with the ITIL standards that they had recently adopted

The second reason was made evident after a continued literature review uncovered the work of Weick and Meader (1993) on GSS. Weick and Meader stated that the systems that were designed for group support were preoccupied with the decision, and not on the sensemaking process. Due to this preoccupation, the problem requiring a decision is not fully defined meaning that decision makers are often making decisions on problems that makes much less sense to them than it would to others in the organization. Additionally, the information that they have to make their decisions is not as rich as if it would have been the product of sensemaking. Weick and Meader (1993) identified five strategies for sensemaking that could be supported by information systems, but are often neglected. These strategies, described earlier in this dissertation are summarized as follows in Table 8.

Table 8: Strategies for Sensemaking.

Strategy	Description
Action	Trial and error experimentation to learn about the environment and how it behaves.
Triangulation	Using data from a variety of sources to overcome deficiencies in the single perspective presented by a single data source.
Affiliation	Resolving confusion and learning about the environment by reconciling others views of the environment with their own.
Deliberation	A slow process that involves processing feedback and using more recent events to reinterpret earlier ones. The process eventually leads to a clear picture of an environment that was once quite confusing.
Contextualization	Relating organizational events or environments that are better understood to those that are not as well understood to provide meaning. Labels, metaphors and platitudes are often the tools used to achieve this process.

At this point, the focus of the system was modified not to focus on sensemaking as it relates to change management decisions, but rather to focus on collecting the sensemaking for change management and making that sensemaking product available to the actual decision makers. The difference, albeit subtle, required a shift from looking at the system from the perspective of a GDSS and to looking at it from the perspective of a GSS. The change, made because of the

aforementioned two reasons, was a better fit with the literature on sensemaking and IS, the goals of the proposed system, and the needs of the organization that was hosting the system. Additionally, although the change did not require the total reconceptualization of the design requirements because they were rooted in the process of sensemaking, the work of Weick and Meader did require us to reexamine our design principles in the context of GSS and consider each of the five sensemaking strategies in our design principles and determine how they would be satisfied by them. This reexamination lead to the removal of design principle number 4 because the alert functionality was deemed to be more related to making decisions than it was to supporting sensemaking. The effects of considering the five sensemaking strategies are discussed in the following paragraphs.

Simulations are often used as a tool to enhance the action sensemaking strategy. In this process, multiple action and reaction scenarios are able to be constructed and the results of those scenarios are used to clarify the environment. Weick and Meader (1993) state that this is one of the strategies for sensemaking where IS support has actually made an impact. As it relates to the design principles presented in this dissertation, it was decided that users should have the ability to construct scenarios in some manner and there should be a mechanism for capturing the predicted outcomes of those actions and a way to validate them as well.

Triangulation was another sensemaking strategy that Weick and Meader (1993) felt had that IS had historically supported in GSS. This strategy was also represented by our initial design principles in that the information that the system provides to the users is from various data sources. However, it is important to heed the warning issued by Weick and Meader that providing too many different sources of information too fast may hinder the sensemaking process (Weick and Meader 1993). Because of this, our design principle number 7 was amended to include that the number of streams presented to users of the system should be limited.

Affiliation was represented partially by design principle number 5 that called for interaction between the users. However, it does not fully satisfy the needs of this strategy just because the users are allowed to interact, it does not mean that they will necessarily interact in the context of their interpretations. Due to this, design principle number 5 was amended to state that "the system will support interaction between users in such a way that allows them to view the sensemaking products of others and to provide their feedback on those products." Additionally, it should be noted that design principle number 8 that calls for user input not to be presented to others in the system to preserve anonymity was shown to help support the affiliation sensemaking strategy (Weick and Meader 1993).

Deliberation is supported in part by the same principles that support triangulation, and it is also subject to the same warning. Providing more inputs into the deliberative process is beneficial to it, but too much input can cause users to abandon the process and turn to stereotypes to interpret events (Weick and Meader 1993). That being said, it was felt that the adjustment made to the design principles in the triangulation section of this section would be almost sufficient to fully support this strategy. Taking into account that deliberation is a slow process, it was also deemed necessary to create a new design principle that users should be made aware of events in the future as soon as they are available. Doing this would allow the users to take their time to evaluate the situations and not circumvent the process.

Contextualization was supported by design principles 2 and 3 that called for the use of some kind of knowledge base that would allow users to store and recall sensemaking accounts and information. Having the ability to store and recall this information provides access to prior events that have better understanding attached to them that can be used to reinterpret current, more confusing events. These changes were incorporated into the final design principles to generate the list presented in Table 9 and used to guide the construction of the artifact detailed in the next section.

Table 9: Final Design Principles.

1. The system should be able to support different user identities and provide customized information based on that identity.
2. The system should be able to store the sensemaking input of the users.
3. The system should be able to recall and present the sensemaking products of users.
4. The system will support interaction between users in such a way that allows them to view the sensemaking products of others and to provide their feedback on those products.
5. The system should be constructed on a platform that is almost always available to users such as a web platform.
6. The system should present information to users from both internal and external sources that are relevant to their identities as well as information that is relevant to other identities in a cautious manner.
7. The system should not present user information to help encourage user participation.
8. The system should allow users to construct cause-effect scenarios and to receive feedback on those scenarios.

The SenseMan System

The final prototype that was implemented in the organization was named the SenseMan system. The name was changed from ChangeMan because it was more illustrative of what the system actually did and because it helped to quell some fears from management that it would interfere

and not enhance the current change management efforts of the organization. The system was constructed on a web platform using ASP.NET (VB) and was hosted on the organization's intranet site. Data storage was implemented using a SQL Server 2005 database housed on another server in the organization.

The SenseMan system was componentized to have each function be served by a separate web page. The Intro page (Intro.aspx) collected the user ids of the users and passed them along to the main page of the application. The user id was not tied to any user information and served only three purposes:

1. To determine the profile of the user so that the appropriate data streams would be displayed.

2. To determine whether the user is a system user or a system administrator so that the functions that are applicable to that role are shown on the Main page.

3. To serve to limit the amount of times that a user could review a piece of sensemaking output, which will be discussed in more detail later.

The Intro page is shown in Figure 4 below.

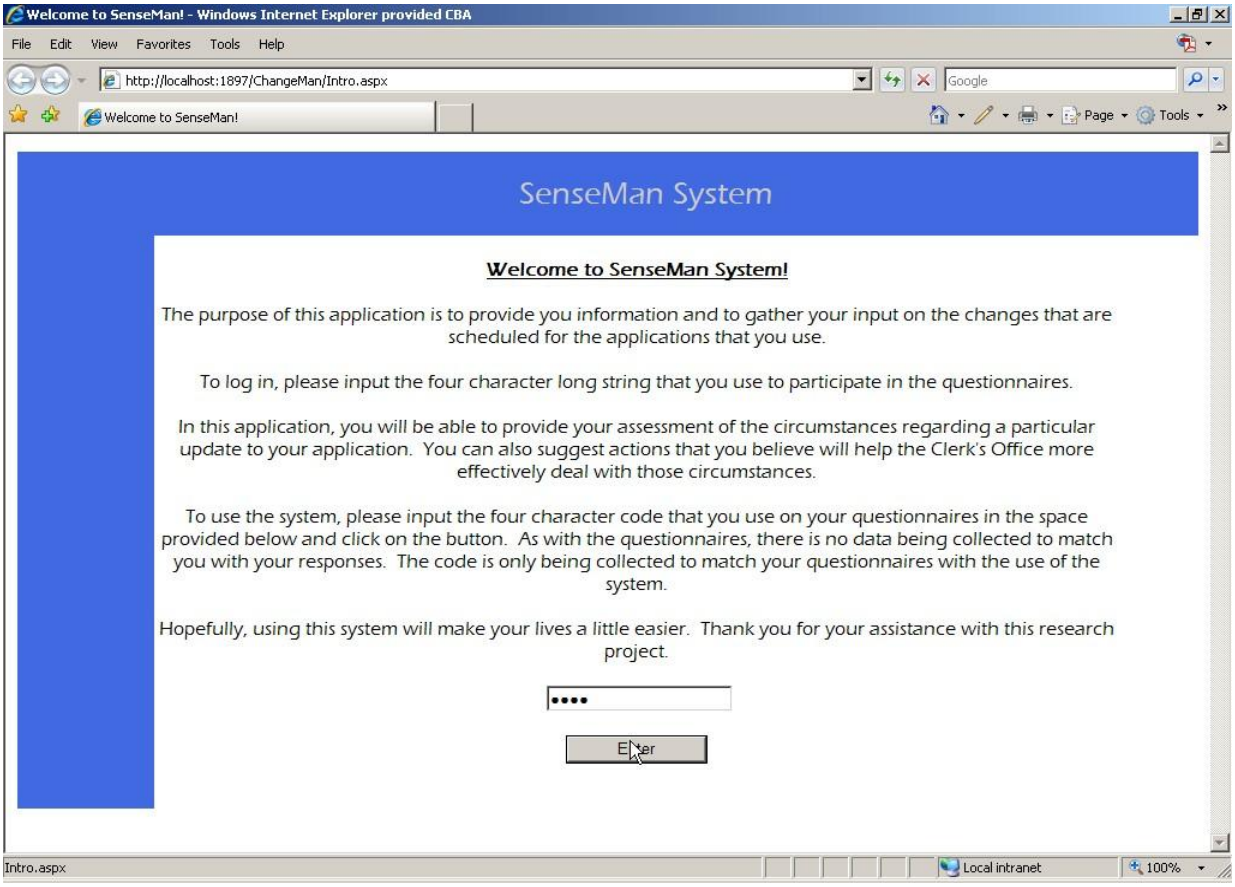

Figure 4: The Intro Page of the SenseMan System.

After the user inputs a valid user code (validated only on the basis of correct format) they are directed to the system's Main page (Main.aspx). From this page the functions that the user can perform are listed on the screen based on their role. System users have the options of Create and Review. System administrators have the options of Report and Administer. The system users view of the Main screen in shown in Figure 5 below.

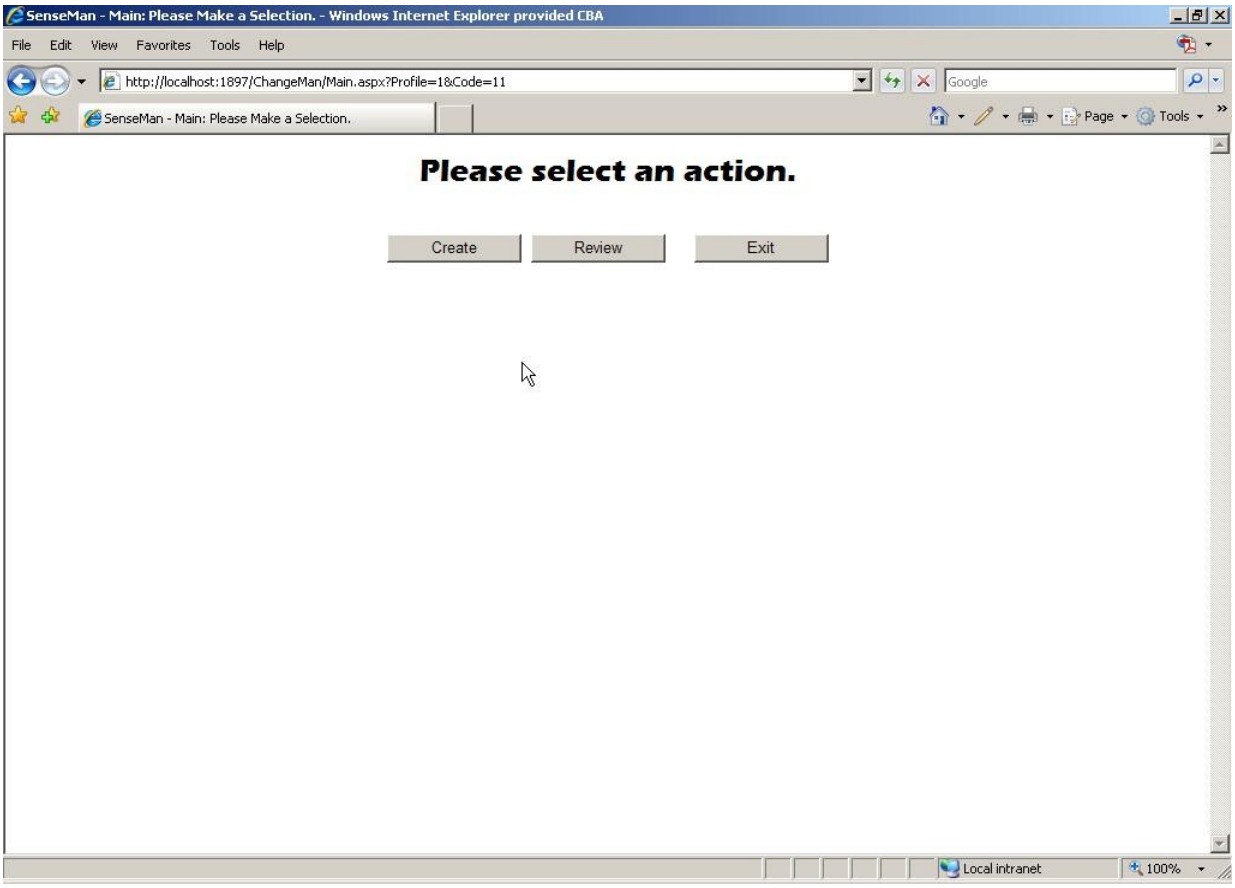

Figure 5: The Main Screen (System User View).

Each option that the user chooses takes them to a different module in the system. These modules are presented in Table 10 below and then each one of them is presented in the paragraphs that follow.

Table 10: The SenseMan System Modules

Module	Function
Admin	The Admin module allows the administrator to add/delete/update events and data streams.
Create	The Create module presents environmental information to the system user in the form of data streams that display information relevant to their profile as well as information relevant to other profiles and the environment in general. The Create module also allows users to input their sensemaking product by asking them to input an action and the reasoning behind the suggested action.
Review	The Review module presents the sensemaking inputs collected in the Create module and allows system users to assign a level of agreement to the input.
Report	The Report module aggregates the data collected in the Create and Review modules and presents it in a form that can then be taken to decision makers.

The Admin Module

The Admin module exists to provide the users an interface to update the system. From the admin module, system administrators can add new events, delete outdated events, or update current events. The Admin module also allows system administrators to add, edit, or delete the data streams that the system presents to the users. The Admin module, while not directly involved in the process of enhancing sensemaking, serves the very important purpose of allowing the system administrators a way to keep current the events that provide part of the context for sensemaking as well as the data feeds from which sensemaking cues are extracted. For example, if the area around the organization was going through a lot of construction, the administrators might want to add a feed about the construction activities that are going on in the area since these types of activities can cause issues for updating software such as power outages or the possibility of cut fiber lines. By providing the capability to add this feed, the users will be able to consider this in their sensemaking activities. Additionally, adding events allows the administrators to determine exactly what events they would like to collect the sensemaking product of the organization. The Admin module is shown in Figure 6 below.

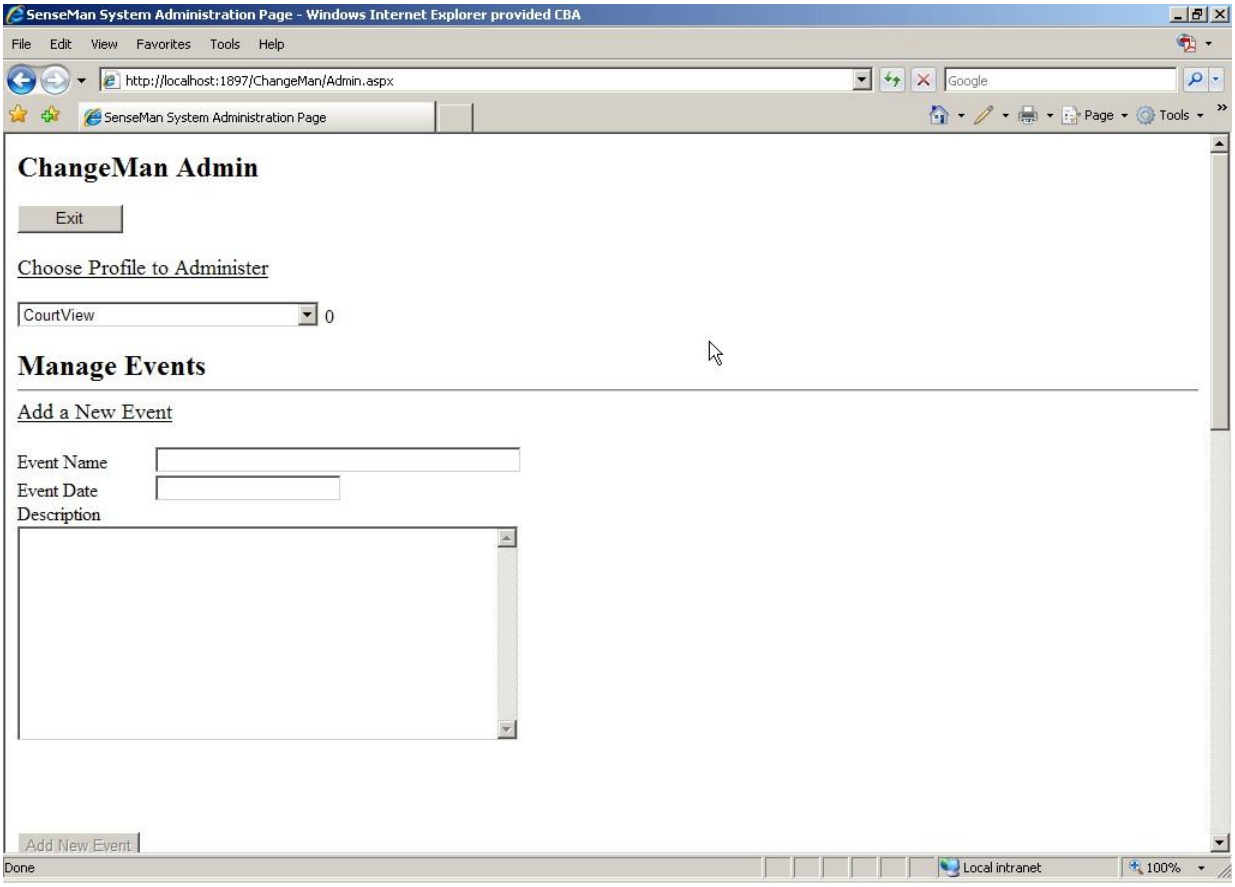

Figure 6: The SenseMan Admin Module.

The Create Module

The purpose of the Create module is to stimulate sensemaking in complex and uncertain environments, such as software change management. Often times, when faced with complexity and uncertainty, humans tend to reduce the amount of environmental scanning that they do and

try to deal with the complexity by going to the same sources of information that they feel comfortable with or with which they have had positive results in the past (Weick 1995; Boyd and Fulk 1996). The problem with this approach is that in complex and uncertain environments those sources of information may eventually provide unreliable guidance (Weick 1995). The Create module is designed to combat this tendency by presenting information to system users from a variety of data sources that are chosen somewhat at random. I state that the selection is somewhat random because there are some items in the environment that the users will need to be aware of based on their role in that environment. Because of this, the first data stream is always one that is directly applicable to them in the context of the environment. However, the other data streams are more randomized in nature. The second and third data streams are chosen at random from streams that are applicable to not only their profile, but to other profiles as well. The final data stream is chosen from a list of data streams about the environment external to the organization. This list was originally populated with local news sites and weather feeds from external Internet sites by the researcher, and then was then amended and maintained by the system administrator at the Clerk's office. So, given the variety of data streams that the system houses, when users visit the create module it is quite possible that they will not see all of the same data streams that they saw the last time they visited. The Create.aspx page from the SenseMan system is shown in Figure 7 below.

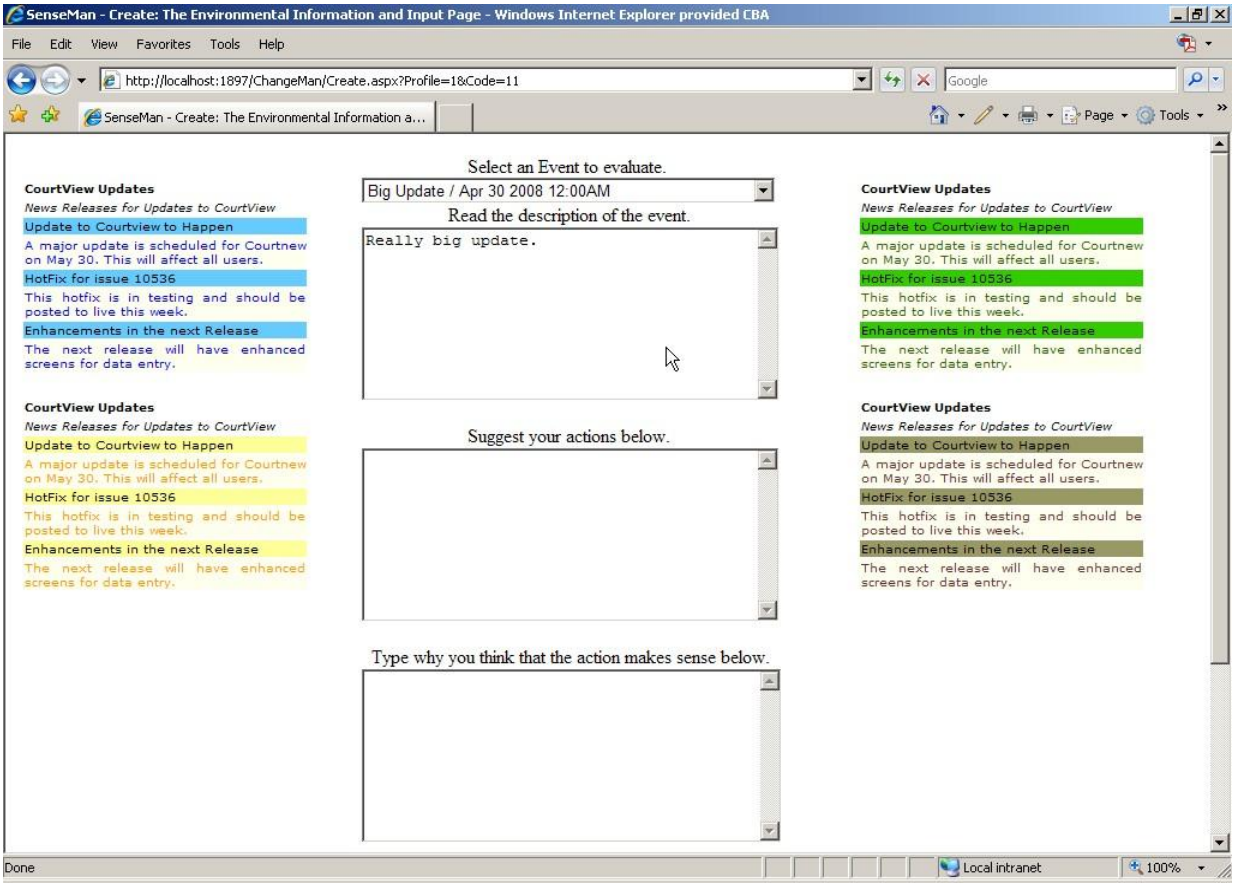

Figure 7: The SenseMan Create Module.

The data streams that are being displayed are actually RSS readers that have been coded into the page and populated with the contents of an XML file. The XML file can originate from almost any website on the Internet that publishes information via that format, or can be created and customized by the organization and stored on local web servers. For example, in this implementation of the SenseMan system, the Clerk's office decided to create custom XML files

that detailed the upcoming software updates for each of their applications with their descriptions, other technological updates that were scheduled, and current events within the organization. They also, however, included XML data from the local county government website that contained information about news and events in the county, XML data from the state news agencies, and XML data about weather events.

Displaying information is only half of the purpose of the Create module. The Create module also provides a place for users to take that information and record it in the form of an action and a reason. The action refers to an action that the user believes should be taken in the context of some event and the environment. The reason is the rationale that the user has for taking that action, or essentially the sensemaking that went on to cause them to come to the conclusion that the action is necessary. The action and reason serve as the mechanism by which the system captures the sensemaking products of the user, which can then be reviewed and evaluated by other users in the Review module. The inputs are placed on the same page as the environmental data feeds to provide a convenient place for the user to record this information without having to move to another module.

Report Module

The report module provides a mechanism for aggregating and displaying what sensemaking is going on in the organization as well as the general level of consensus about the sensemaking products amongst organizational members for use by the organization in decisions or other activities. This level of consensus is determined by taking the average of the scores that were assigned to actions and reasons in the Review module. This is important because even though sensemaking is the product of an individual mind, it is still subject to the criticism of other sensemakers (Weick 1995). The idea that sensemaking is a process that is subject to majority rule is a good transition to our discussion on the Review module. The report module is shown in Figure 8below.

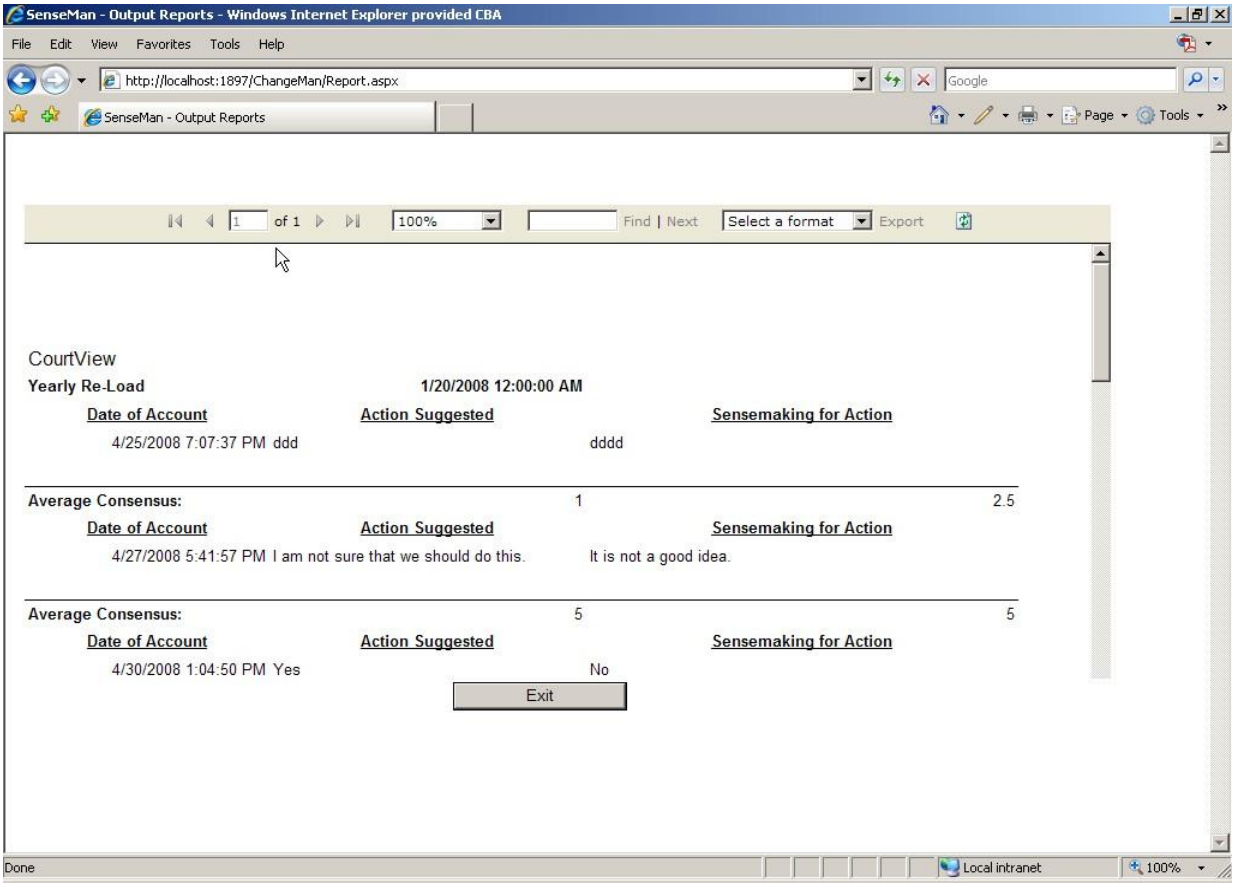

Figure 8: The SenseMan Report Module.

Review Module

The final module that makes up the SenseMan system is the Review module. The review module allows users to express their level of agreement with the sensemaking products (actions and reasons) of the other system users. For each action-reason pair that the system has for a

given event, the system gives them the ability to rate their level of agreement with each component of the pair on a five point Likert scale that goes from 1, meaning poor, to 5 which stands for excellent. A five point Likert scale was chosen over a seven point Likert scale because it was more parsimonious and there is no difference in the mean scores if one were rescaled to the other (Dawes 2008). Users can evaluate each pair for an event once and the system indicates when they have evaluated all of the sensemaking products for a given event.

Allowing the users to evaluate the sensemaking products of others confidentially allows for users to interact with each other by means of their sensemaking without the constraints of society, organizational level, and so forth because both the sensemaker and the evaluator are protected by anonymity. Additionally, the system is designed to not show the evaluators the aggregate scores of the actions and reasons. The goal of this design feature is to get an evaluation of the sensemaking product that is unaffected by what others think. Besides allowing the organization a mechanism to aggregate the overall level of consensus on the sensemaking associated with a particular event, the Review module also has the purpose of stimulating the evaluator to reexamine his own sensemaking in the context of the sensemaking products of others. This reexamination may allow sensemakers to find deficiencies in their sensemaking, or may validate their sensemaking products. The Review module is shown in Figure 9 below.

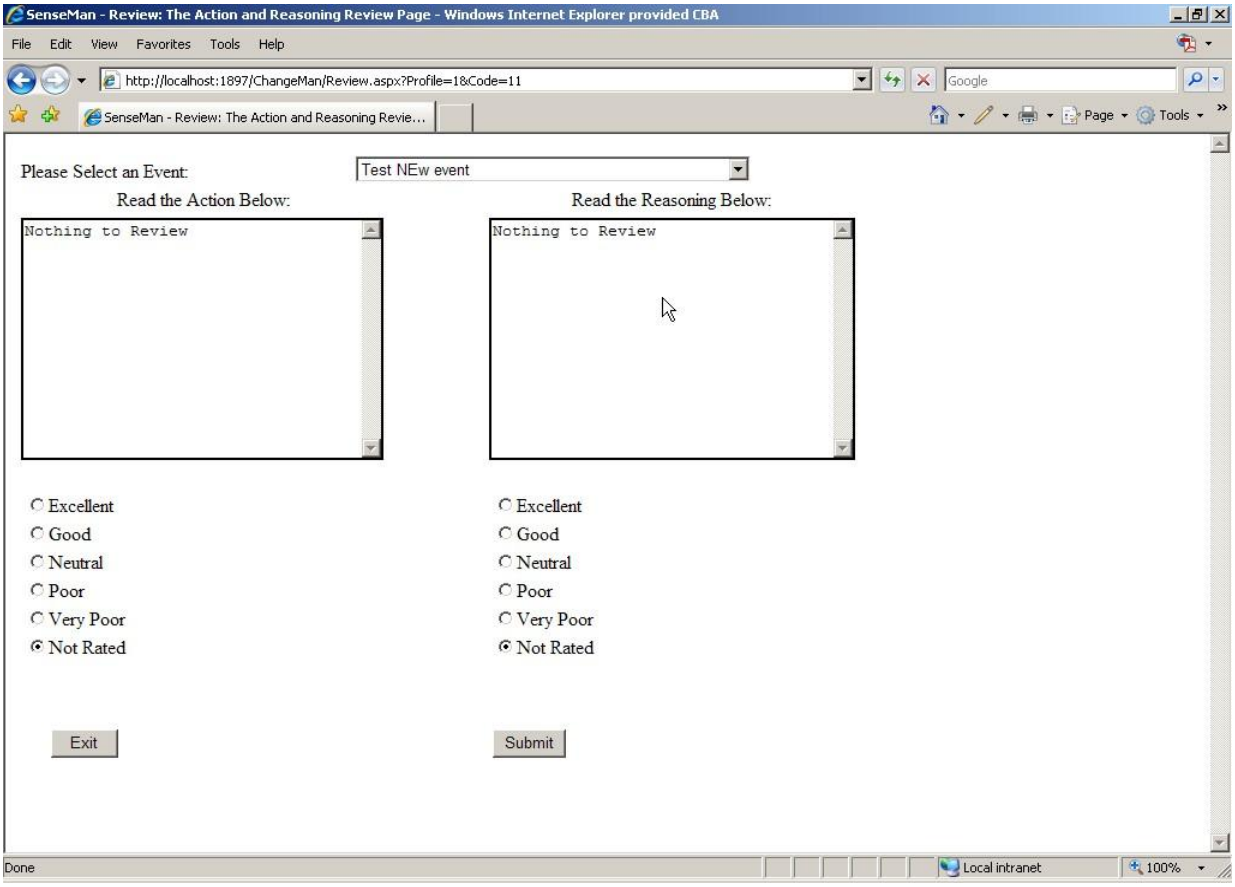

Figure 9: The SenseMan Review Module.

Now that the design process for the system and the system itself have been presented, it is necessary to demonstrate how the design process follows the guidelines set for designing systems for EKPs (Markus, Majchrzak et al. 2002) and how the design product is reflective of the final design principles formulated earlier in this dissertation.

Consistency of the Design Process and Product with the Kernel Theories

The final component of both the design product and the design process as explicated by Walls, Widmeyer, et al. (1992; 2004) are testable hypotheses. In the case of the design process, these hypotheses are driven by the choice of development method. The design principles influence the selection of the hypotheses that will evaluate the design product. The following sections discuss the hypotheses that were selected for both the design process and product.

The work of Markus and her colleagues on EKP design theory defined six principles for EKP design. The six principles were: (1) recruiting naïve users into the design process, (2) knowledge transfer and refinement through iterative prototypes, (3) designing for offline, as well as online use, (4) integrating the knowledge of systems experts with local experts, (5) use of a dialectic design methodology, and (6) componentizing everything including the knowledge base. These principles also drive the creation of the testable hypotheses for the design process. In the case of this design process, the testable hypotheses are whether or not the system design process adhered to the principles for EKP design just mentioned. The design process for the SenseMan system is reflective of the process that was used to generate designs for EKP's in that:

- Several users were involved in the design process that were unaware of the system before its existence and before their involvement (recruiting naïve users into the design process).
- Multiple prototype systems were developed and amended based on the input from the users (refinement through iterative prototyping).
- Since the purpose of the design is to enhance sensemaking, which is an ongoing process, people will use the system and continue sensemaking even when they are not in contact with the system (design for offline use).
- Allowing the users to choose and construct their own custom data feeds to integrate into the system (integrate systems experts' knowledge with local experts).
- Overcoming hurdles to the design such as the issue described earlier where the Chief Deputy of Information Systems felt that the system would conflict with other change management systems through the use of creativity and a dialectic process (using a dialectic design process).
- All of the major functions of the system were designed in separate web pages, allowing for one page to be critiqued and amended without having to touch the other pages. Additionally, the database was designed as a relational database so all of the major entities were separated into their constituent tables. In future development of the system, the knowledge base that will store the historical sensemaking products will be stored will have a separate database structure (componentizing everything).

Having shown that our design process is reflective of the principles governing the design of systems to support EKP, it can be said that the testable hypotheses of the design process have all held.

Separate from the design process, the design product portion of the theory must also be evaluated via testable hypotheses. These hypotheses are derived from the design principles of the system. In the case of this system design theory, the hypotheses are actually twofold. First of all, the design theory must be reflective of the design principles. Next, the ability of the system to achieve its goal of enhancing sensemaking will have to be evaluated against the hypotheses presented earlier in this dissertation. The first part of the hypothesis is answered here by comparing the system to its design principles to determine if it is reflective of them. The second portion of the hypothesis is evaluated using quantitative statistical methods that are detailed in the next chapter.

The SenseMan system is reflective of the design principles of the Weickian inquirer in the following ways:

- Supporting user identities and providing customized information based on that identity is realized through the use of profiles in the system that customize the primary data feeds

and determine what functions users have access to in the system. Additionally, the system supports the different perspectives of the users that are found in the sensemaking products that are maintained in the system and viewed by other users.

- The ability to store the sensemaking input of users is demonstrated in the Create module. The sensemaking input is captured in this module and then stored in the database.

- The recall and presentation of the sensemaking products of users is demonstrated in the Review module that recalls the sensemaking products for a given event and then displays them to the system users for their evaluation.

- Interaction and feedback is also achieved using the Review module to present the sensemaking products of others and allowing for them to provide their feedback. Feedback is also seen in the Report module that aggregates all of the individual sensemaking feedback in order to provide it on an organizational level.

- Constructing the system so that it is almost always available to users is seen in the fact that the system is constructed on a web platform that most users can access using a web browser and many of the users can access it away from the office.

- The cautious presentation of internal and external information is seen in the data feeds that are customized by the employees of the Clerk's office and also taken from sites on the Internet, but limited to only four data feeds.

- Anonymity to encourage user participation is seen in the fact that the user id does not have any identifying information about the individual using the id stored in the database. Users do not know who authored individual sensemaking products nor do they know who has reviewed the sensemaking products that they have authored.

- The construction of cause-effect type scenarios is seen in the Create module that requires users to not only provide actions, but also to provide the reasoning for those actions. Feedback on those scenarios is accomplished by the Review module.

Now that the design process and product have been demonstrated to be reflective of the theories that guided them, it is necessary to evaluate the design's effectiveness with regards to supporting sensemaking. This will be done by utilizing the IS artifact in the organizational setting for which it was created. In addition to the testable hypotheses advocated by Walls, Widmeyer, et al., (1992) the design science principles of Hevner et al (2004) call for any product of design science research to be evaluated using rigorous methods. In an effort to test our hypotheses in accordance with this guideline, this study will use a multi-method approach that involves the use of multiple data collection methods that will mix quantitative and qualitative methods. The results of this type of analysis will be dissimilar data sets that provide different perspectives on a single phenomena (Mingers 2001; Dube and Pare 2003; Hesse-Biber and Leavy 2006).

The quantitative data collection was accomplished through survey research and analyzed through the statistical procedure of repeated measures ANOVA. This will tell us if there is any difference in the groups with regards to their perceived complexity about software updates. The qualitative data was obtained through semi-structured interviews and then subjected to a qualitative analysis. This method will give us deeper knowledge about the actual experiences of the participants that used the system. This type of study is illustrative of the parallel type of multi-methodological research design as described by Mingers (2001). In this type of study data collections are carried out simultaneously and the results feed into each other, as opposed to a sequential design where one method provides results that then drives the second data collection (Mingers 2001).

This type of multi-methodological design is not new to IS research. Trauth and Jessup (2000) used this parallel multi-methodological approach to study the use of GSS in gender equality discussions. In their study, the quantitative data provided useful information. However the qualitative data provided deeper information and helped to reveal a very different picture of the discussions (Trauth and Jessup 2000; Mingers 2001). The point here is that this dissertation seeks to achieve the rigor required by Hevner et al. (2004) by basing its findings on the system's effectiveness on multiple sources of information analyzed through different lenses. The result of triangulating these different data sources should provide a much more convincing and accurate picture of how effective the system really is (Dube and Pare 2003).

CHAPTER FOUR: QUALITATIVE EVALUATION OF THE EFFECTIVENESS OF THE IS ARTIFACT

Sensemaking is designed to assist with tasks that are filled with uncertainty and complexity (Ashmos and Nathan 2002). Additionally, in describing sensemaking, Maltis (2005, p. 21) states that "…sensemaking allows people to deal with uncertainty and ambiguity by creating accounts that enable action. So despite the fact that there is no measure for sensemaking, we can use the reduction in complexity as a proxy measure to evaluate the ability of the IS artifact to support sensemaking. This chapter discusses the quasi-experiment that was used to test the hypotheses relating to the system's effectiveness in supporting sensemaking. First there is a discussion about the design of the experiment including the measures used, the quasi-experimental design, and the characteristics of the population and sample. Next, the survey methodology used to collect the data from the sample population is presented as well as the data analysis method that was chosen to evaluate the collected data. Finally, the results of the quantitative study are presented.

Experimental Design

The measure that was used in our study is actually a measure of perceived uncertainty from Boyd and Fulk (1996). Boyd and Fulk used their measure to build a model to evaluate the effects of perceived uncertainty on environmental scanning, which has been noted as a sensemaking process (Thomas, Clark et al. 1993; Ashmos and Nathan 2002). The measure separates the construct of perceived uncertainty along two independent dimensions. The first dimension is perceived variability which is concerned with the rate and strength of change in the context being studied. The second dimension is perceived complexity which is a complex construct comprised of the following items discussed earlier in the introductory section and presented here again:

Perceived Adequacy of Information – Do the participants believe that they have access to enough information about the environment to reduce complexity?

Perceived Analyzability – This measures the participant's perceptions on their ability to understand the cause and effect relationships that take place in an environment as a result of taking some action.

Perceived Predictability – How do the participants feel about their abilities to identify environmental forces that may affect the organization and about their ability to know what those effects may be?

To assess the reduction in perceived complexity, the users of the system were evaluated via survey research involving a pretest to assess levels of the variables being measured before treatment and then a posttest after a period of 6 weeks to assess levels of the measured variables after the treatment. Participants in the study were allowed to self-select participation in the experiment. Since the sample is not based on full random assignment, the study is classified as a quasi-experiment. Although these types of experiments usually create less support for counterfactual inferences, they still have many of the same structural details and purpose of the randomized experiments and the experimenters can wield considerable control over how the measures are selected and measured, how the treatment is scheduled, and how nonrandom assignment is done (Shadish, Cook et al. 2002). For example, the organization has only about 190 application users that will constitute the population of the study. Because of this it was decided to conduct the experiment without a control group which can still provide us with strong information on causality and reduce other possible explanations for the effect of the treatment depending on the design of the experiment (Shadish, Cook et al. 2002).

One way to increase the ability of the experiment to provide us with strong causal inference is to add a nonequivalent dependent variable in the analysis to test for interaction effects between

variables in a one-group pretest-posttest design. This design has been shown to be much more interpretable than the one-group pretest-posttest design if the constructs are similar and if they are exposed to the same set of environmental circumstances to a plausible degree (Shadish, Cook et al. 2002). In this experiment, the constructs were all similar because they were all measures of perception of different phenomena. It is similar to the example of the awareness study listed in Shadish, Cook et al. (2002) of the experiment of McKillip and Baldwin (1990) that used different measures of awareness for sexually transmitted diseases, alcohol abuse, and exercise to see if the actual effects for the construct targeted by the treatment were greater than those that the treatment was not hypothesized to have an effect on.

In addition to providing the measure of perceived uncertainty used in this study, the results of the Boyd and Fulk (1996) study greatly assisted in the formulation of the hypotheses that will be tested. In their study, Boyd and Fulk found that the sensemaking process of scanning decreased when complexity increased. This is consistent with the writings of Weick (1993; Weick 1995) who found that in complex environments sensemakers tended to turn to familiar data sources that would eventually mislead them. Additionally, as Boyd and Fulk deconstructed the measure of perceived uncertainty into the components of perceived variability and perceived complexity, this study has deconstructed the perceived complexity measure into its constituent parts to examine the effect that our system design has on perceived adequacy of information, perceived analyzability, and perceived predictability individually in this measure. This being the case, the

hypotheses for the study that were stated earlier were amended to more accurately reflect the experimental design:

- H1: System use will cause a posttest increase in perceived adequacy of information for understanding with regards to software updates over the pretest levels.
- H2: System use will cause a posttest increase in perceived analyzability of events with regards to software updates over the pretest levels.
- H3: System use will cause a posttest increase in perceived predictability of events with regards to software updates over the pretest levels.

Since Boyd and Fulk found no connection between perceived variability and environmental scanning, it can be inferred that there should be no connection between sensemaking and scanning as well. The fact that there should be no change in the measure of perceived variability allows us to use it as the non-equivalent dependent variable in the analysis and also leads us to our final hypothesis.

- H4: There will be no difference between the pretest and posttest levels of perceived variability of the software updates.

Sampling

The population for the study was all users of the four software applications described in the previous section. The population was determined to be all users for two reasons. First, many of the users are currently involved with the evaluation of software updates, have been involved in these updates in the past, or will be involved in them in the near future. This is because the Clerk's office does not have a static group of personnel that evaluate the updates. The second reason is that as a result of their interdependence on each other, the applications act more like modules within a larger software application than they do as stand alone, independent software applications. If we return to the definition of a system by Marakas (2006) as a set of interrelated elements, with an identifiable boundary, and a common goal, we can see how this is the case. We have already seen that the applications are interrelated. The system is bounded because it can be defined within the context of other systems as the hardware, software, and users that interact with the Clerk's office, and the applications all work together to serve the mission of the Clerk's office. Because of this, this study assesses all users as if they were part of a single software application and based on this definition the population size is approximately 190.

Sampling Procedure

Having defined the population, the sampling procedure will now be discussed. First an email was sent to all potential participants informing them of the study. A few days later, survey packets were distributed in manila envelopes to all potential survey participants. The survey packet contained the survey instrument, a waiver of informed consent, and a code that was used to match the respondent to their responses on successive measures. All potential participants were invited to review the informed consent document. If they chose to participate in the research study, they were asked to complete the survey instrument, remove the code from the package and keep it for use on future data collections, and return the survey to a marked collection bin in their area. If they chose not to participate, they simply had to return the manila envelope and its contents to the bin. Since everyone is following the same procedure, this helps to protect the anonymity of the respondents.

Sample Description

The initial sample size of the population was 106 respondents, resulting in an initial participation rate of 55.7%. Since the participants were allowed to self-select whether or not they wanted to participate, the makeup of the population was of some concern. To get a better picture of the population the respondents were asked questions on the survey to assess their length of experience with the primary software application that they utilize as a part of their job function. The results are reported in Table 11 below.

Table 11: Distribution of the Sample Population Based on Experience.

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	0-1 YEARS	12	11.3	11.5	11.5
	1-3 YEARS	41	38.7	39.4	51.0
	3-5 YEARS	12	11.3	11.5	62.5
	5-8 YEARS	25	23.6	24.0	86.5
	8+ YEARS	14	13.2	13.5	100.0
	Total	104	98.1	100.0	
Missing	System	2	1.9		
Total		106	100.0		

As can be seen from the table, the respondents were not evenly distributed within the application. However, the fact that the majority of respondents have had greater than one year of experience using the application tells us that we can be fairly confident that most of our sample have experienced an update and more than likely have been involved with one in some manner.

However, this sample suffers (as most samples do) from the various threats to external validity such as the interaction of any causal relationships found between different units, over variations in treatment, with settings, with outcomes, or with any explanatory mediators that are context-dependent.

The Survey Instrument

Survey research was chosen as the quantitative methodology to collect the data to test the hypotheses. The survey design was governed by the tailored design method (TDM) of Dillman (2007). To place the survey construction in the proper context, or for those not familiar with TDM, it would be prudent to go over some of the points associated with the method.

Dillman's (2007) tailored design method seeks to increase response rates and information quality in self-administered surveys. It is a method that is built upon social exchange theory and seeks to "…create respondent trust and perceptions of increased rewards and decreased costs for being a respondent, that take into account features of the survey situation, and that have as their goal the overall reduction of survey error ." Dillman addresses every aspect of survey creation from question construction, to the survey process, to design principles for web-based surveys. The design principles that Dillman presents with regards to question design, survey design, and web and email survey design are located in Appendices A and B. The survey instrument that resulted from the TDM principles is located in Appendix E.

Validating the Survey Instrument

The survey instrument was evaluated for both reliability and validity. As a preliminary test, the survey instrument was sent to academics and real estate agents in the Orlando, FL area and assessed complexity with regards to software changes to Microsoft Office. The Reliability was assessed by calculating the Cronbach's Alpha coefficient for each of the items being assessed by the survey instrument. All of the items being assessed on the survey pilot test met the criteria for

the Cronbach's Alpha test of .7 or higher. Once it had been established that the instrument was at least reliable, we felt comfortable moving forward with the instrument to conduct the pretest.

The reliability of the instrument was assessed again for the participants that self selected to participate in the pretest assessment. Additionally, the validity of the data collected via the survey instrument in the pretest was assessed by conducting a confirmatory factor analysis (CFA). The results of both tests are discussed in the following sections.

Reliability Results

Reliability coefficients were calculated for the variables, perceived adequacy of information (PAI), perceived analyzability (PA), perceived predictability (PP), and perceived variability (PV) assessed by the survey instrument using SPSS 15.0 for Windows. Data was collected for each variable by 5 items on the survey instrument. Perceived adequacy of information is measured by variables PAI_1 – PAI_5, perceived analyzability is measured by variables PA_1 – PA_5, perceived predictability is measured by variables PP_1 – PP_5 and perceived variability is represented by the measures for variables PV_1 – PV_5. The initial reliability results are listed in Table 12 below.

Table 12: Initial Reliability Coefficients for Measured Variables.

Measured Item	Cronbach's Alpha	N of Items
PAI	.890	5
PA	.934	5
PP	.857	5
PV	.608	5

Respondent ratings of perceived adequacy of information and perceived analyzability obtained from the survey were judged to be very reliable for the participants to whom it was given, with reliability coefficients of .890 and .934, respectively. The respondent ratings for perceived predictability were judged to have good reliability given their reliability coefficient of .857. The respondent ratings for perceived variability however scored a reliability coefficient of .608, which is less than the acceptable level of .650.

Because the items from perceived variability did not meet the standards for modest reliability, a review of the corrected item-total is warranted. Upon review of the corrected item-totals, it was

discovered that item PV_4 did not correlate with the corrected total very well and its removal would increase the reliability coefficient to .638, as well as providing for a more parsimonious model. For this reason, the variable was removed from the analysis and the reliability coefficient was recalculated resulting in a value of .638, as expected.

However, since this value is still below the acceptable value of .650 another review of the corrected item-totals was undertaken. This review revealed that the variable PV_5 did not correlate very well with the corrected item-totals and that its removal would provide for an estimated reliability coefficient of .698, which is within the acceptable range. After removing PV_5, the reliability analysis was conducted again and resulted in a value of .702 which is well within acceptable values. Additionally, reviewing the corrected item-totals indicated that additional removal of variables was not warranted. The final reliability coefficients are listed in Table 13 below.

Table 13: Final Reliability Coefficients for Measured Variables

Measured Item	Cronbach's Alpha	N of Items
PAI	.890	5
PA	.934	5
PP	.857	5
PV	.702	3

Validity Results

Even though the constructs are being assessed individually, it is still prudent to see if the constructs that are being measured are actually reflective of the latent construct of perceived complexity. Due to this the validity of the survey instrument must be assessed using confirmatory factor analysis (CFA). The individual survey item responses were aggregated to provide a pretest assessment for each of the constructs being measured. A factor analysis was then performed using maximum likelihood as the extraction method with promax rotation, as we were being guided by theory in this instance

The identity of each factor was determined after a review of which items correlate the highest with that factor. Items that correlate the highest with a factor provide for the meaning of the factor and the meaning is the concept by which they are tied together. In the case of this

analysis, we have the results of prior research from Boyd and Fulk (1996) to say that the items of pretest PAI, PA, and PP should represent the concept of perceived complexity. A successful result is one in which those items load on to a single factor and explain a large portion of the total variability.

In the context of this study, such success was attained allowing us to say that we have validity evidence supporting the conclusion that the scores from this instrument are a valid assessment of a person's perceived complexity. The descriptive statistics for the items are presented in Table 14, below.

Table 14: Descriptive Statistics for CFA Items.

	Mean	Std. Deviation	Analysis N
Pretest_PAI	2.9419	.86417	106
Pretest_PA	2.6931	.97315	106
Pretest_PP	2.5857	.78426	106
Pretest_PV	2.5898	.70203	106

The descriptive statistics are presented to demonstrate the no standard deviation was larger than the respective means and also that no standard deviation was extremely out of line with the others. Additionally, the one factor that was extracted explains 54.2% of all the variance of the

items. The factor loadings are shown in Table 15 below. Loading coefficients were suppressed if they were less than .01.

Table 15: Factor Loading Matrix for CFA items

	Factor 1
Pretest_PA	.984
Pretest_PAI	.859
Pretest_PP	.679
Pretest_PV	>.10

Analysis Method

Since the same instrument and sample population were used to assess both pretest and posttest levels of the measured variables, repeated measures ANOVA was deemed to be the appropriate statistical methodology to analyze this type of panel data. Much like the regular ANOVA procedure, repeated measures tests the equality of means. However, because the standard ANOVA procedure fails to measure the correlation between the repeated measures because it violates the assumption of independence of the procedure. Therefore, repeated measures ANOVA is used because it does not have this assumption. Moreover, repeated measures was deemed appropriate because it is less sensitive to large error variances when there variation in

the sample population and it is more efficient when participants are difficult to recruit (UCLA: Academic Technology Services 2007).

Now that the design of the experiment, the procedures and qualities of the initial sample and the population as well as the characteristics of the survey instrument have been discussed, we can move forward to an analysis of the data using the repeated measures ANOVA procedure. The next section will detail the results of our analysis for each of the measured variables and also any interaction effects with the nonequivalent dependant variable.

Results

As stated earlier, the pre-test survey was delivered to all users in the manner described in the section on the sampling procedure used in the study. For the post-test, the sampling process changed a bit as we were only interested in getting responses from those participants that filled out the first surveys so that we could use the paired responses in the repeated measures ANOVA procedure. That being the case, the surveys were placed in manila envelopes as in the first administration, but they were placed in various areas within the organization for the participants to pick up. The collection process for the surveys was the same as in the pre-test. Collection bins were placed in the organization and respondents were asked to return their surveys to the bins.

Final Sample Population

One risk that you take when you oversee an experiment that assesses participants in multiple time periods is the attrition of participants. This study was no exception as only forty participants submitted surveys in the first data collection. Because of the low response rate, participants were given a second opportunity to complete the survey if they had not already done so. This second data collection garnered twenty-six additional surveys for a total of sixty-six. Although you can never be completely sure of all the reasons for participant attrition, the primary researcher was made aware of two reasons for the loss of participants in this study. The first reason was that some of the participants had lost their codes from the first survey. Since there was no information that connected them personally to their codes, they could not be informed what their code was and, therefore, could not match them to their earlier responses. The second reason was passed on to the researcher by the management. It seems that at the time of the post-test, the organization was dealing with some changes that had to be implemented by a certain date. This caused some of the participants to not fill out the survey because they felt that they just could not spare the time.

Perceived Variability

To effectively carry out our experimental design with the nonequivalent dependent variable, we first examined pre-test and post-test levels of perceived variability as reported by the survey participants to see if it was not affected by the treatment. Despite our expectations, the participants did have an change in their perceptions of variability regarding software updates to a statistically significant degree ($F = 6.716$, $p = .012$) as reported in Table 16 below.

Table 16: Tests of Within-Subject Effects for Perceived Variability

Source	TIME	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared	Noncent. Parameter	Observed Power(a)
TIME	Linear	1.417	1	1.417	6.716	.012	.095	6.716	.723
Error(TIME)	Linear	13.506	64	.211					

a Computed using alpha = .05

Now that we have seen that the participant's perceptions on the variability of software updates have changed, we can now refer to the descriptive statistics in Table 17 below to determine how they changed.

Table 17: Descriptive Statistics for Perceived Variability

	Mean	Std. Deviation	N
Pretest_PV	2.6040	.71549	66
Posttest_PV	2.8128	.74507	66

An examination of the pre-test versus the post-test means of the participant responses regarding their perceptions on variability show that despite the fact that they still somewhat disagree that there is not much variability with their software updates, they are moved more towards having a neutral feeling about it after their experiences with the system. That being said, H4 is not supported by the data that we collected.

These results, while interesting, proved to be problematic with the analysis because we can no longer view perceived variability as a nonequivalent dependent in the analysis. That being the case, the ability of this analysis to reduce the plausibility of counterfactuals is somewhat reduced. It was decided, however to go ahead with the analysis because the short timeframe between the participants experiences with the system and the post-test could help to balance out any

maturation or learning effects that may be plausible explanations for our results (Shadish, Cook et al. 2002).

Perceived Adequacy of Information

The analysis for perceived adequacy of information was not nearly as interesting as it much more closely followed our expectations. The repeated measures analysis did reveal a statistically significant difference between the participant's perceptions on the adequacy of their information regarding software updates after using the SenseMan system (F = 11.011, p = .001). The within-subject effects are displayed in Table 18 below.

Table 18: Tests of Within-Subject Effects for Perceived Adequacy of Information.

Source	TIME	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared	Noncent. Parameter	Observed Power(a)
TIME	Linear	3.381	1	3.381	11.011	.001	.145	11.011	.905
Error(TIME)	Linear	19.961	65	.307					

a Computed using alpha = .05

An examination of the descriptive statistics for this construct supports the hypothesis that the users of the system would perceive and increase of their adequacy of information as evidenced in Table 19 below.

Table 19: Descriptive Statistics for Perceived Adequacy of Information.

	Mean	Std. Deviation	N
Pretest_PAI	3.0367	.85376	66
Posttest_PAI	3.3568	.85951	66

According to the change in the mean for PAI, it seems that the participants are more likely to feel that the information that they have regarding software updates is at least acceptable, whereas they were almost neutral on the subject when they responded to the pretest assessment.

Perceived Analyzability

The analysis results for perceived analyzability also were in alignment with our expectations. The participants in the study reported a statistically significant change in their perceptions about their ability to detect cause and effect relationships regarding their software updates ($F = 10.990$, $p = .002$). The results of the analysis are reported in Table 20 below.

Table 20: Tests of Within-Subject Effects for Perceived Analyzability.

Source	TIME	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared	Noncent. Parameter	Observed Power(a)
TIME	Linear	2.484	1	2.484	10.990	.002	.145	10.990	.904
Error(TIME)	Linear	14.694	65	.226					

a Computed using alpha = .05

To determine the direction of the change, the descriptive statistics were consulted in Table 21 below. An examination of the means shows that the participants feel better about their ability to detect cause and effect relationships in software updates after using the system. In fact, participants feel fairly neutral about their ability to detect these relationships whereas they had a more negative feeling about the same ability on the pre-test. Because there was a significant change in in perceived analyzability and because the change was in the hypothesized direction, we can say that there is evidence to support the second hypothesis, H2.

Table 21: Descriptive Statistics for Perceived Analyzability.

	Mean	Std. Deviation	N
Pretest_PA	2.8226	.97009	66
Posttest_PA	3.0970	.83408	66

Perceived Predictability

As described earlier in the dissertation, perceived predictability is a measure of the participant's feelings about their ability to detect items in their environment that may affect a software update. In the context of this study, the participants did report a statistically significant change in their ability to detect these items in the environment ($F = 13.493$, $p < .001$). The results of the analysis for this construct are located in Table 22 below.

Table 22: Tests of Within-Subject Effects for Perceived Predictability.

Source	TIME	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared	Noncent. Parameter	Observed Power(a)
TIME	Linear	2.667	1	2.667	13.493	< .001	.172	13.493	.951
Error(TIME)	Linear	12.851	65	.198					

a Computed using alpha = .05

A review of the pre-test and post-test means for the construct reveals that the change that the participant's reported was in the hypothesized direction, thus providing evidentiary support for H3. The descriptive statistics for perceived predictability are recorded in Table 23 below.

Table 23: Descriptive Statistics for Perceived Predictability.

	Mean	Std. Deviation	N
Pretest_PP	2.6248	.66997	66
Posttest_PP	2.9091	.74436	66

This quantitative analysis is only one lens through which to interpret the effectiveness of the system. In the next chapter, we turn from this positivist type analysis to a more interpretive one that involves analyzing data collected through semi-structured interviews to determine the participant's feelings with regards to the constructs being used to evaluate the effectiveness of the system.

CHAPTER FIVE: QUALITATIVE EVALUATION OF THE EFFECTIVENESS OF THE ARTIFACT

As stated earlier, this study seeks to achieve rigor in its analysis through the use of a multi-methodological design. The purpose of conducting a qualitative research study was twofold. First, the data collected from the study can be used to validate the results of the quantitative study conducted earlier. Second, the use of a qualitative analytical technique can provide us with richer information on the user's feelings on the effects of the system. In this chapter, the processes of conducting the interviews is presented as well as a description of the process of coding the qualitative data including the coding scheme and the method for ensuring that the coding is reliable. The chapter concludes with a discussion of the findings of the qualitative study.

Qualitative Analysis

Some researchers believe that reality is not something that can be separated from the context in which it exists and that it is the subjective construction of social groups. This is evidenced in the call for information systems research to be more attuned to the fact that phenomena should be

viewed both subjectively and objectively when studying information systems in organizations (Orlikowski and Robey 1991) and that users can be viewed as social actors that interact with their environments (Lamb and Kling 2003). Sensemaking is also influenced by the social interactions that the sensemaker has and the reality that they construct for themselves (Weick 1995). All of this points to the fact that in order to understand the effects of the system on sensemaking we need to study the interaction between the system's users with not only the system itself, but their interaction with the system in the context of their environment. Because of this, a qualitative research methodology using data collected in semi-structured interviews was chosen to analyze the qualitative data using a coding scheme based on the same constructs evaluated in the quantitative study.

The Qualitative Sample

The sample used for this study was a criteria based convenience sample with the criteria being the level the participant had in the organization In increasing order of level within the organization, interviews were conducted with: Clerk Employees, Deputy Clerks, Sr. Deputy Clerks, Supervising Deputy Clerks and , Chief Deputy Clerks. Clerk employees are generally new-hire employees or employees that cannot be deputized as clerks. Deputy Clerks are generally longer term employees that have met the requirements for deputization, meaning that

they can act with the Clerk's authority. Sr. Clerks supervise small groups of Deputy Clerks and report to Supervising Deputy Clerks whom are responsible for entire departments. Chief Deputy Clerks are responsible for managing the four divisions represented in the organizational chart depicted in Figure 3.

The participants were chosen based on their response to recruitment emails that were sent to pre-existing distribution groups within the organization based on the various levels. Participants were ensured that the utmost effort would be taken to ensure their confidential participation in the study. For example, if the participants did not want to respond to the recruitment email via their Clerk email account, they were given the primary researcher's email and telephone as means to indicate their intent to participate either by using their personal email address or by telephone. Once the participant's for the study had been identified, the process began.

The Participants

There were a total of eight participants for the qualitative study. The participants came from different areas within the organization, had different job orientations (technical vs. non-technical) and had all been with the organization for varying lengths of time. As stated earlier, the participants were recruited via email, but were asked to respond to the request using their

personal email or by telephone if they wanted to protect their identity as a participant, so that a record of their participation was not left on the organization's mail server. If they did want to protect their identity, precautions were taken to ensure that it was protected. Each interviewee was given the option of being interviewed outside the research setting, an option that some of the participant's took advantage of. The other researchers were not given access to any personal data about the participants.

The Procedure

The qualitative analysis began with six semi-structured interviews that were all conducted by the primary researcher. The interviews are semi-structured in that some questions are asked to all participants and other questions are asked based on their responses to those common questions. The interviews were conducted after the system was in place for at least four weeks. Each of the participants in the interview reported having used the system to varying degrees. Each interviewee was asked three questions in the interview that were pre-determined, and follow up questions were asked based on their responses to those questions. The interviews lasted an average of about thirty minutes each. The three questions that every participant was asked were:

1. What can you tell me about your experiences using the SenseMan system?

2. Did using the system change your feelings about software updates?

3. What aspects of the system did you find most/least beneficial?

The interviews were transcribed within twenty-four hours of the time that they took place by the primary researcher and provided to the other researchers that were assisting with the coding and data analysis via email. The primary researcher was also the only person to have access to the audio files of the interviews and they were kept in a locked cabinet before transcription and were not taken to the research setting. The audio files were destroyed after they had been transcribed.

Once the transcripts were provided to the other researchers, the coding process began. Researchers were instructed to evaluate the participant's responses to the interview questions and to identify statements in those responses that indicate their feelings about the constructs of perceived complexity described in the prior chapter dealing with the quantitative study and to indicate whether their participant's feelings on the construct were positive, negative, or neutral. Each researcher recorded their results in a tabular format in a manner consistent with the example below in Table 24.

Table 24: Sample of the Qualitative Coding Scheme.

Int #	Position	Tech?	Quesiton	Response	Construct	-1/0/1
3	Chief DeputyClerk	Yes	Can you tell me a little bit about your experiences using the senseman system? What functions did you use, how did you interact with it?	The benefit there was that it gave me several logins and then I looked thorough the information and read through the information.	PInfo	1

The other researchers that assisted had limited knowledge of the SenseMan system and the results of the quantitative analysis in an effort to limit any potential biases in the analysis. Once the three researchers had coded out the interview transcripts, they were synthesized into a single document. The synthesis of the codes was achieved by allowing the researchers to resolve any disagreements about the coding of particular statements. This was done by first identifying any statements with differing codes and then having a discussion between the researchers to debate the particular merits of their positions. It then went to a vote between the three researchers and the code supported by the majority of the researchers was deemed to be the one that was included. In the unlikely event that the three researchers had different positions and there was no majority rule, then the matter would be referred to the members of the primary researcher's dissertation committee for a decision. Once the document had been synthesized and the results

tabulated as to their relationship to the constructs, the transcripts were then re-examined to see if themes not related to the constructs could be identified.

Results

The Experimental Constructs

After the coding was completed and synthesized, the coded responses were aggregated so that we could get a general impression of which of the experimental constructs the participants experienced through their use of the system and their general feelings towards those constructs. The level of experience of the experimental constructs to each participant was measured by the number of times that the participant made a statement in the interviews that related to one of the constructs. These results were tabulated and presented in Table 25 below.

Table 25: Summary of Participant Experience with the Experimental Constructs.

	Code				
Interview Number#	PAnal	PInfo	PPred	PVar	Grand Total
1	3	11	3	0	17
2	2	4	1	0	7
3	8	16	0	1	25
4	5	7	3	0	15
5	4	5	2	0	11
6	2	3	1	1	7
Grand Total	24	46	10	2	82

From the data gathered from our participants, it seems that they believe that adequacy of information was the most experienced construct as it was mentioned the most by every participant in the interview process and almost twice as much as the next closest construct overall. The interviewee's perceptions on their ability to discern cause and effect relationships (perceived analyzability) was the next most experienced construct. Statements regarding perceived analyzability were also fairly prevalent in the analysis because statements relating to it were made by all of the participants in the interview process. Perceived predictability was not mentioned as much as the other statements and was not mentioned at all by some participants. Finally, the system seemed to have very little influence on the perceptions of variability held by the participants.

As important as determining the level of influence of the constructs reported by the participants after their experience with the system however, is the feelings that they had toward those constructs. As stated earlier, for each instance that an interviewee mentioned one of the constructs, a score was assigned to the statement to indicate whether the statement was negative (-1), neutral (0), or positive (1). We can get a general feel for the overall feelings of the participant towards each experimental construct by taking the mean of the scores that were assigned to each construct. These scores are reported below in Table 26.

Table 26: Summary of Participant Feelings on the Experimental Constructs.

	Code			
Interview Number#	PAnal	PInfo	PPred	PVar
1	1.00	0.27	1.00	--
2	1.00	0.75	1.00	--
3	0.50	0.63	--	1.00
4	1.00	0.43	0.67	--
5	0.50	0.60	0.00	--
6	1.00	0.67	0.00	1.00
Grand Total	0.75	0.52	0.60	1.00

As Table 26 shows, the experiences that each participant had with the constructs being studies were positive, on average, albeit to varying degrees. An interesting aspect of this analysis is that even though the participants had the most to say about their perceptions on the adequacy of their information, as a group they had the least intense positive feelings about them.

Other Findings

It is also important to mention that some other themes were made evident in the analysis of the qualiitative data. One of these themes is the concept of information overload. Both of the employees interviewed with a technical orientation made statements that referred to this in one way or another whether it was referenced directly as in the following statement…

> *"Maybe least beneficial would be because we are so dynamic and that there is so much information the amount of information may cause of a little bit of an information overload that in some cases might negate the benefits of the information being there."*

…or if it was referenced indirectly as in this statement:

"[Depth of information] Meaning that if the information that a user sees is not related to the software application, they might be confused as to why they are being shown that information."

Regardless of how it was presented, this emerged as a theme from the qualitative data and will be discussed more in the next section.

Another theme that emerged from the qualitative data is the communication of information about updates. Although the communication of information is related to the experimental construct of adequacy of information, this dissertation takes the position that it is different enough to be considered separately because in this context, it deals more with the organizational culture than the information itself. More than half of the participants in the qualitative study made comments that relate directly to this point, as evidenced by the following statements:

"And by keeping that limited to just one application knowing about it you could have some problems with consistency issues in the office where this department is implementing credit card [processing in the system] and the other department that has capability and is looking into it, does not know that the other department has already worked out all of the issues with the system and that they could leverage that information on their side. "

"Providing not only more adequate information, but in some cases providing information where it once did not exist in the past, to put it more bluntly…or maybe where the information was held more tightly in the past."

"But I am not a super user and I am not on the committee that meets that knows all this and makes these types of decisions. I am not one of the chiefs, I am one of the Indians and the Indian was lost when the change went through because the information was not handed down. "

"… But I think that we may not have done a great job of telling the general population about a new release. "

The preceding statements almost lead one to wonder if the issue is not that the information itself that the organization possesses about its software updates is lacking in adequacy, but rather lacking in distribution. This could be related to the issue of a lack intra-organizational communication was also identified as a theme from the qualitative analysis by multiple coders. In the next chapter, we will look at both studies and attempt to triangulate the results to better tell the whole story of the effectiveness of the SenseMan system in this organizational context.

CHAPTER SIX: DISCUSSION

This dissertation began with three interrelated questions and the preceding chapters have been dedicated to answering them. The questions that drove this research were:

1. Is sensemaking an appropriate epistemology for an inquiring system?
2. If sensemaking is an appropriate basis for an inquiring system, can design principles for such a system be derived to guide the construction of an IT artifact supporting sensemaking?
3. If the design principles can be derived to guide the construction of an artifact, how effective will the artifact be in enhancing sensemaking in individuals and organizations?

In this chapter, we will take a look at our findings and discuss them in the context of these three questions. However, before we can start this discussion there is one other question that must be addressed. Since this dissertation is essentially one of design science, how well did this dissertation meet the design science principles of Hevner, et al (2004) presented in Chapter two? Some scholars may say that because the process of generating an IS design theory from Walls, Widmeyer, et al. (1992; Walls, Widmeyer et al. 2004) was used, that this research does not fall

into that category. However, while re-examining the effectiveness of the 1992 paper, Walls, Widmeyer, et al. (2004) state that their process for IS design theories can be complementary to the design science principles of Hevner, et al. (2004). This dissertation takes that same position and the upcoming section is our attempt to justify that statement.

Is This Design Science Research?

Much has been made throughout this dissertation about the design science principles espoused by Hevner and his colleagues. Now that the evaluation of the artifact has been completed, it is appropriate to address this study in the context of their seven guidelines for design science research. To briefly review, the seven requirements of design science research that Hevner et al (2004) proposed were:

1. Design as an Artifact
2. Problem Relevance
3. Design Evaluation
4. Research Contributions
5. Research Rigor

6. Design as a Search Process

7. Communication of Research

The following paragraphs will describe how this dissertation meets these requirements.

Design as an artifact - The product of design science research, according to Hevner, March et al. (2004) should come in the form of a construct, a model, a method, or an instantiation. This dissertation satisfies this guideline by producing an instantiation in the form of the SenseMan system. Additionally, the design principles that have been derived through this research can be used to guide other instantiations of systems that are designed to support sensemaking.

Problem relevance - The problem of complex business environments was discussed in great length in the opening chapter of this dissertation. This is a problem most organizations face regardless of the industry. One of the major impacts on organizations of complexity and uncertainty is its effect on decisions. These effects range from a reduction in scanning or searching for new information sources (Weick 1995; Boyd and Fulk 1996) to the use of electronic aids that do not account for the sensemaking needed to reduce complexity (Weick and Meader 1993). Regardless of the level of the effect on decisions and other organizational working that complexity has, it is a relevant problem that is only growing (Courtney 2001) and is worthy of study.

Design evaluation – Hevner and his colleagues (2004) state that the usefulness and effectiveness of the design must be evaluated using rigorous methods. As we have seen, the SenseMan system was evaluated in a quasi-experimental setting using survey research as the data collection method and performing a statistical analysis of the collected data. The data collection instrument followed the guidelines of the Tailored Design Method of Dillman (2007) that are found in Appendix A and Appendix B. The instrument was pilot tested and the validity and reliability of the instrument were also tested and deemed to be acceptable. The statistical analysis was performed using a repeated measures ANOVA procedure which is deemed appropriate for analyzing this type of quasi-experimental design.

In addition to the quantitative study, a qualitative analysis of data gathered in semi-structured interviews was also performed to provide another perspective on the experiences that the users of the system had in not only the context of our experimental constructs, but in general. Later in this chapter, the results of these two studies will be triangulated to provide more compelling support for the results obtained in this dissertation.

Research contributions – Karl Weick and David Meader wondered what a sensemaking support system would look like but noted that no one had ever asked the question (Weick and Meader

1993). In this dissertation we have attempted to answer this question by developing a kernel theory and designing just such a system. While the design principles and the instantiation itself are the most obvious contributions of this dissertation, the purely theoretical contribution should not be overlooked. It has been more than 36 years since Churchman released the Design of Inquiring Systems, and even though others have examined different epistemologies on which to expand them, this is to my knowledge the first time one of these extensions has been developed into an instantiation of an actual system.

Research rigor – The research of this dissertation is deemed rigorous in that it built upon and utilized the work of many scholars in the realm of design science. The extension of inquiring systems theory was critically examined against the systems criteria of Churchman himself. The systems requirements and design principles were deeply rooted in this extension of inquiring systems theory as well as the work of GSS from Weick and Meader (1993). Sensemaking was deemed to be an emergent knowledge process, so the design process adhered to the principles for those types of systems from Markus and her colleagues (2001). Finally, the work of Hevener, et al. (2004) was used to evaluate the process in general to assess its adherence to their principles of design science research.

Design as a search process – Several different employees were consulted in the problem environment to refine our design principles and our design artifact. The development of the aforementioned principles and artifact were also refined through the use of iterative prototypes. However, the organization had placed some constraints that we had to work within, so the optimal design specifications were probably not achieved.

Communication of research – Communication about the results from this research can easily be tailored to fit technical, academic, or managerial audiences. The technical audiences would benefit from the design principles and processes that went into developing the system as well as having the opportunity to improve on the system itself. Communication to managerial audience would focus on the benefits of using information systems to enhance sensemaking and the decision process instead of focusing on the actual decision itself. Finally, I am sure that academic audiences would have much to say (positive and negative) about my extension of inquiring systems to include the evolutionary epistemology.

Having determined that the research presented in this dissertation adheres to the principles for design science research of Hevner. et al. (2004), it is now appropriate to discuss the results of the evaluation of the design artifact. As presented earlier, the final question asked in this dissertation asks how effective a system generated with the design principles formulated through this

research would be in supporting sensemaking as measured by reducing complexity. The following section will discuss our results as well as consider alternative explanations for the effects observed.

The Effectiveness of the Artifact

Triangulating the Results

Looking back over both the quantitative and the qualitative studies, we can see that they both were generally asking the same two questions.

1. Did the users of the system experience some change in their perceptions related to the experimental constructs as a result of using the system?

2. What was the direction of that change, positive or negative?

Since both studies generally asked the same questions, albeit from different methodological perspectives, this dissertation will triangulate the results by comparing the answers to the questions from both studies.

As opposed to the quantitative study, where statistical significance serves as the litmus test for determining whether or not there was a change, the qualitative study does not have such a hard and fast measure. Therefore, it was decided that a change would be reported in the qualitative study if the majority of the participants in the study had multiple instances of the construct in the analysis of the interview transcripts. We did have a hard and fast measure for direction of the participant's perceptions on the experimental constructs for both studies, however. In the qualitative study, the direction of the change in the mean from pre-test to post-test levels let us know whether the change was positive or negative. In the qualitative study, the average of the coded values on the context of the statement was the measure of change. Given these measures, the comparative answers to the aforementioned questions for both studies are listed in Table 27 below.

Table 27: Triangulation Results.

Experimental Construct	Hypothesized Change	Quantitative Change?	Direction of Quantitative Change	Qualitative Change?	Direction of Qualitative Change	Triangulation Achieved?
Perceived Adequacy of Information	Positive	Yes	Positive	Yes	Positive	Yes
Perceived Analyzability	Positive	Yes	Positive	Yes	Positive	Yes
Perceived Predictability	Positive	Yes	Positive	Yes	Positive	Yes
Perceived Variability	No Change	Yes	Positive	No	N/A	No

As Table 27 shows, the data shows that the two studies found consistent results for the experimental constructs that make up the dimension of perceived complexity from Boyd and Fulk (1996) the system's effect on perceived variability. As stated earlier, the quantitative study did find that a statistically significant change did occur in the survey population. The construct was barely mentioned, however in the qualitative study with only two of the participants having made statements regarding it albeit both of those statements were indicated a positive or increased perception of the rate and intensity of software updates.

Possible Reasons for Differing Results

The difference between the results in the two studies could possibly be due to the fact that the quantitative study was done on an anonymous basis and the participants felt more comfortable revealing that information as opposed to the in the interview process where their identity was known to the primary researcher. More interesting, however, is looking at possible reasons why the participants experienced an increase in their perceptions of variability of software updates and its affect on overall system effectiveness.

From our qualitative study, we have already seen that information about updates was not always communicated to all employees. In fact, some of those employees are only informed of the updates that they will be testing and are told to test the functionality for their specific job function. By communicating information out about other updates that are going on within the same application or other applications, it may be that the information sharing that SenseMan promotes could be a possible reason for the increase. If increased perceptions of variability increase the complexity of an environment, could it be that the system's overall effectiveness as well? While this may certainly be the case, this paper posits that this effect, will be short term and as the participants gain experience viewing software updates outside of the one update that affects them, this effect will subside and we will see that the participant's perceptions of variability will subside. This seems to be supported by the qualitative data where some of the interview subjects addressed the ability of the employees to more comfortably think outside of

their own "box", so to say. This statement was made by a manager in response to whether or not there are considering updates outside of the one that they are specifically working on.

> *"I believe that we have a pool of general clerks that are like that and would be able to determine things that would affect an update…that are forward thinking. We also have group that are in their tunnel and think only about putting this peg in this hole. We have about two hundred something in the office and 138 in courts and now many of those 138 are people that have grown up with computers and are capable of thinking more broadly. So I would say that, yes, they are able to."*

The view that the employees would be able to adapt to the increased information and use it to their advantage was also held by others. This comment was made by an interview participant with a technical orientation as a response to a follow-up question based on their feelings about the employees ability to deal with any information overload that they may have perceived as coming from the system.

> *"Oh, definitely. I think that as people use the system more, and become more comfortable they will be able to become a more intelligent user."*

Whether or not the participants are able to cope with the information that they are getting and how they adapt to that information would be interesting topics for a future study.

Other Discussion

It seems that the communication of information may be one possible reason that the participant's perceptions of variability increased. What about the possibility of information overload that the technical staff felt inclined to mention as identified in the qualitative study? From the interview transcripts, it seems that the technical staff was worried about the source of the information overload stemming from the breadth of the information that was included in the SenseMan feeds. The results from both studies as well as the responses from the non-technical staff; however seem to show that was not the case.

First, let's look at the results of both studies. The experimental construct of perceived predictability was included to determine the participant's feelings on their ability to determine external factors that might affect an update. This would seem to link this construct to the external data feeds that they are receiving in the system. The data from both studies seem to indicate that the participant's felt more positively about their ability to do this after using the system.

Additionally, the interview transcripts from the non-technical employees seem to point to the fact that they appreciated the data being there, and did use it to some degree, but it was not the most impactful data in the system. To illustrate this, examine some statements made by participants in reference to the external data and then the statements that they made about the data that is internal to the organization such as the data about their own application or related applications. First let's look at the statements about the external data.

> *"[The information was useful] Probably not for my application as far as the software application goes, but it was interesting to see what is going on in the county. "*

> *"I looked at the system. There was information in there that didn't necessarily pertain to me, but it was still interesting to read."*

> *"Is [the news about the county] interesting? Yes. But it doesn't really affect our actual operations on a daily basis, not yet. But that is not to say that something won't come up in the future that may."*

"[The external feeds] were OK. They really didn't make me think about the updates though."

Now some statements about the data that is internal to the clerk's office.

"I saw [software updates] differently based on the information that I saw and read about what others were doing and what was going on in my application."

"So when I logged into the Senseman system, I immediately noticed the two columns in the different colors. The left hand column notified me as to what was being changed and why. Prior to the senseman module, I never knew that. "

"Well, the part about what is going on with the programs is probably the most beneficial."

"It was nice to see what else is going on in the Clerk's office and what the other areas are doing and how it relates to us."

So it seems that the employees were fairly proficient in filtering out the information that was external to the organization unless they really saw some benefit in it. What is interesting here is that it seems that if any information had the ability to cause information overload, it would be the information that is just one degree of separation from their job function. If this is true, then it lends a little more credence to our explanation about why we saw an increase in the participant's perceptions of variability on the software updates. However, without actually doing a study on information overload itself, these statements cannot be verified.

So, it seems that in answering the questions that we have presented in the opening chapter of this study, we have uncovered more questions. These questions will serve as part of the foundation for the future research projects that will stem from this dissertation. These future research projects as well as the contributions of this dissertation will be presented in the final chapter.

CHAPTER SIX: CONCLUSION

This dissertation has taken us on a journey that began with three questions. From those three questions we created the concept of a new type of inquiring system, used that inquiring system to build a design theory, built a system based on the design theory, and evaluated the effectiveness of the system. This chapter will detail the contributions that this study has made to the field of information systems and will provide the readers an idea of the future direction that this research will take.

Research Contributions

This dissertation makes contributions to both academia and to practice. Since this dissertation is primarily written for evaluation by an academic audience, the contributions to academia will be discussed first, followed by the contributions of the research to practice. That being said, the following are the contributions that this dissertation makes to academia:

1. *It addresses an underrepresented research area and extends the notion of inquiring sytems* - To academics, the extension of inquiring systems theory is something that has

not been attempted by many scholars. The extension proposed in this dissertation, is different from other attempts to expand the epistemological foundations from the original five inquirers in that it is the only one that has been fully developed in the sense that it meets the system criteria that Churchman (1971) himself proposed. The fact that the Weickian inquiring system is fully developed is also exhibited by the fact that it has been shown to have the ability to be applied to a system design.

2. *It presents a new IS design theory* – The design theory for the Weickian inquiring system adheres to the process espoused by Walls, Widmeyer, et al (1992; Walls, Widmeyer et al. 2004) and satisfies all of their criteria for a design theory.

3. *It supports the notion that IS design theories can be complementary to design science research* – The fact that we have used the process of Walls, Widmeyer, et al (1992; Walls, Widmeyer et al. 2004) to build a design theory that is used as a part of a design science research project that meets the principles set forth by Hevner, March et al (2004) supports the notion that the two processes can be used in conjunction with each other as stated by Walls, Widmeyer, et al, (2004).

Being that this is a design science project that adheres to the principles of Hevner and his colleagues (2004), we must have relevance to practice as well. That being said, the contributions to practitioners made by this dissertation are:

1. *The design artifact* - the design artifact is a contribution to technical practice in that it is an example of a system that is truly a GSS designed to support sensemaking, not a decision support system under the guise of a sensemaking support system. Technical practitioners may also benefit from the design principles that have been proposed in this paper to guide their creation of IS artifacts with similar purpose.

2. *An example of IS supporting problem identification, not decision making* – While Weick and Meader (1993) have already made this point, they also state that a system that does this had not been created. The SenseMan system is a system that provides raw, unfiltered information to decision makers for them to make sense of it themselves, thus providing the rich information that Weick and Meader (1993) claim is missing from most GSS systems focused on decisions. This is exemplified in this dissertation by the use of the system as sort of a "bolt-on" type enhancement of the ITIL change management initiative that the organization was already undertaking. Thus, the IS design here supported the sensemaking that would then be used in the decision processes dictated by the ITIL standards.

Having discussed these contributions, it is also important to recognize that this study has its limitations, as all studies do. So it is prudent to discuss the limitations of this study because tt is only by being open about these limitations, that the contributions can be truly judged.

Limitations of the Research

Many of the limitations of this research stem from the research methodologies and the subject of the research. The first set of limitations can be attributed to the criticisms that some scholars have about survey research and about qualitative research in general. One of these criticisms is the fact that it is difficult to get a sample that can be deemed representative of a population (especially when the population is allowed to self-select as in the case of this study). This study certainly dealt with this issue given the amount of participant attrition from our pre-test to our post-test assessments. While other measures were taken to attempt to reduce the plausibility of any counterfactual explanations such as limiting the timeframe between pre-test and post-test assessments and triangulating the results of the study with the results of the qualitative study, the fact remains that the sample size for both studies can be viewed as a limitation on the study.

There are also the views that survey research is too statistical and it reduces interesting and important questions to mere numbers, the criticism that the method cannot adequately establish causal connections between variables and constructs and, that some things just aren't measureable by survey or otherwise (de Vaus 1986; Krosnick 1999). This set of limitations is common to all survey research and possibly more so to this study than others. The reason that the generalizability of this study is so difficult to defend is that the phenomenon being studied is

rooted in context and identity construction. While triangulation can assist us in getting a true picture of the participants in this study, it does not give us a picture of people in general. So, while we may be able to make generalizations about the effectiveness of the system to the population of the organization in which it was tested, we certainly cannot extend generalizations beyond that organizational boundary. It may be that other people may have identities or that there may be other contexts for sensemaking that this system does not work particularly well in or where it may excel. That being said, more research is required to assess the effectiveness of the system design on sensemaking in general and not just sensemaking in the organizational and sensemaking contexts represented in this dissertation.

Future Research

Some of the future research directions for the research stream that this paper has entered into have been mentioned earlier. One of these studies is how the inclusion of a system of this type in an organization affects the information levels of their employees. The notion of information overload is especially interesting in this context. The organization featured in this dissertation, while having some issues with communication of information was still open to the notion of communicating the information about software updates to its employees and that seemed to increase their perceptions of the variability of software updates. Did the employees experience

any information overload? If so, what type of information caused them to experience the overload? Did the information overload negate the effects of the system on sensemaking? All of these are good questions that deserve investigation.

Other directions for this research stem from the limitations of the study. One limitation has to do with the organizational identity that the sensemakers using the systems construct for themselves. This would theoretically have an effect on sensemaking since it is grounded in this identity (Weick 1995). For example, one would wonder if the system design principles would have a different effect if the design artifact were implemented in a private sector organization instead of a public sector organization as studied here. The question of setting however only provides one pathway that can be explored.

Another, and perhaps more interesting, pathway would be to evaluate the design principles in different contexts. It would be of particular interest to use the design principles here to construct a system artifact that would support sensemaking in contexts where sensemaking is difficult or where it breaks down. This tends to happen in events where what the sensemaker is experiencing is extremely different from what their expectations of what they would experience were, such as the experiences of those responding to disaster situations. One might wonder (as I often do) what the experiences of the smokejumpers in Weick's (1993) account of the breakdown of sensemaking in the Mann Gulch disaster would have been if they had some support with their sensemaking during their response. Research such as this would be beneficial to practice in that

it may help to improve the response of the brave persons that put themselves in harm's way to assist others. It would also be beneficial in situations such as battlefield support for our troops. Academics would benefit from the results of the design evaluation in that if we can target what treatments actually most effectively assist with sensemaking, we can better understand the phenomenon in this context and in general.

One final path for future research deals with looking at this study inversely. If this study looks at design to support sensemaking, why not look at sensemaking to support design activities. This stream of research could help to alleviate some of the techno-centric bias in our information systems as well as providing for new design methodologies and design principles. Regardless of which (if any) of these paths are chosen, the concept of sensemaking as it relates to information systems is a fertile ground for research and as our environments become more complex and uncertain it will only become more so.

APPENDIX A: DILLMAN'S DESIGN ELEMENTS FOR QUESTIONS

1. Choose simple over specialized words. Use vocabulary that can be understood by the respondents.

2. Choose as few words as possible to pose the question. Statements should be short, rarely exceeding 20 words.

3. Use complete sentences to ask questions. Each statement should be a proper grammatical sentence.

4. Avoid vague quantifiers when more precise estimates can be obtained.

5. Avoid specificity that exceeds the respondent's potential for having an accurate, ready-made answer.

6. Use equal numbers or positive and negative categories for scalar questions. In other words, try to have an almost equal number of statements expressing positive and negative feelings.

7. Distinguish undecided from neutral by placement at the end of the scale.

8. Avoid bias from unequal comparisons.

9. State both sides of attitude scales in the question stems.

10. Eliminate check-all-that-apply question formats to reduce primacy effects.

11. Develop response categories that are mutually exclusive.

12. Use cognitive design techniques to improve recall.

13. Provide appropriate time referents.

14. Be sure each question is technically accurate.

15. Choose wordings that allow essential comparisons to be made with previously collected data.

16. Avoid asking respondents to say yes in order to mean no.

17. Avoid double-barreled questions.

18. Soften the impact of potentially objectionable questions.

19. Avoid asking respondents to make unnecessary calculations.

20. Whenever possible, statements should be in simple sentences, rather than complex or compound sentences.

21. Do not use statements that are factual or capable of being interpreted as factual.

22. Avoid statements that can have more than one interpretation.

23. Avoid statements that are likely to be endorsed by almost everyone or almost no one.

24. Avoid statements containing universals such as all, always, none and never because they often introduce ambiguity.

25. Avoid using indefinite qualifiers such as only, just, merely, many, few, or seldom.

26. Avoid statements that contain “if” or “because” clauses.

27. Avoid use of negatives (e.g., not, none, never)

28. List taken from: Dillman, D. A. (2007). *Mail and Internet Surveys: The Tailored Design Method*: Wiley.

APPENDIX B: DILLMAN'S DESIGN ELEMENTS FOR SURVEYS

1. Write each question in a way that minimizes the need to re-read portions in order to comprehend the response task.
2. Place instructions exactly where that information is needed and not at the beginning of the questionnaire.
3. Place items with the same response categories into an item-in-a-series format, but do it carefully.
4. Ask one question at a time.
5. Minimize the use of matrices.
6. Begin by asking questions in the upper left quadrant; place any information not needed by the respondent in the lower right quadrant.
7. Use the largest and/or brightest measure symbols to identify the starting point on each page.
8. Identify the beginning of each succeeding question in a consistent way.
9. Number questions consecutively and simply, from beginning to end.
10. Use a consistent figure/ground format to encourage the reading of all words.
11. Limit the use of reverse print to section headings and/or question numbers.
12. Place more blank space between questions than between the subcomponents of questions.

13. Use dark print for questions and light print for answer choices.

14. Place special instructions inside of question numbers and not as freestanding entities.

15. Optional or occasionally needed instructions should be separated from the question's statement by font or symbol variations.

16. Do not place instructions in a separate instruction book or in a separate section of the questionnaire.

17. Use of lightly shaded colors as background fields on which to write all questions provides an effective navigational guide to respondents.

18. When shaded background fields are used identification of all answer spaces in white helps reduce item nonresponse.

19. List answer categories vertically instead of horizontally.

20. Place answer spaces consistently to either the left or right of the category labels.

21. Use numbers or simple answer boxes for recording of answers.

22. Vertical alignment of question subcomponents among consecutive questions eases the response task.

23. Avoid double or triple banking of answer choices.

24. Maintain spacing between answer choices that is consistent with measurement intent.

25. Maintain consistency throughout a questionnaire in the direction the scales are displayed.

26. Use shorter lines to prevent words from being skipped.

27. Major Visual changes are essential for gaining compliance with skip patterns.

28. Words and phrases that introduce important, but easy to miss, changes in respondent expectations should be visually emphasized consistently, but sparingly.

List taken from: Dillman, D. A. (2007). *Mail and Internet Surveys: The Tailored Design Method*: Wiley.

APPENDIX C: APPLICATION SOURCE CODE

This appendix contains the application code for the major components of the SenseMan system. The connection strings and database locations have been removed from the code to protect the security of the organization that it is implemented in.

Admin.aspx

```
Imports System.Data.SqlClient
Imports CMClass
''*********************************************************
'The admin page is a page that allows users to add/edit/del
'data streams and events in the ChangeMan system.

'Author: James Parrish
'Created: April 24,2008
'Revised: April 28, 2008
'Revision:  (4/28) Added field to display the event description.
'*********************************************************
Partial Class _Default
    Inherits System.Web.UI.Page

    '* Declare form-level variables and objects
    Dim sqlConNW As New SqlConnection(ConStr)
    Dim dsNW As New Data.DataSet
    Dim dsnw1 As New Data.DataSet
    Dim dsNW2 As New Data.DataSet
    Dim sqlDANW As New SqlDataAdapter
    Dim dvEvents As Data.DataView
    Dim dvStreams As Data.DataView
    '* Declare form-level variable to indicate whether combo boxes
    '*  are in the process of being populated.  Used to control
    '*  execution of SelectedIndexChanged events for the controls.
    Dim blnIsLoading As Boolean = True

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        'The page load event fills the data controls of the page and sets the
        '        buttons to disabled to prevent accidental data updates

        If Not IsPostBack Then
            Try

                Fill_Profile()
                '* Load the text boxes and drop downs based on the value in
the profile.
                Label2.Text = cboProfile.SelectedIndex.ToString
```

```
            If blnIsLoading = False Then
                '*Load the combos.
                Call Fill_Events()
                Call Fill_Stream()

                '*Load the text boxes.
                Call Fill_Event_Info()
                Call Fill_Stream_Info()

            Else : Exit Sub

            End If

            '* Disable the controls until applicable action is taken.
            btnNewEvt.Enabled = False
            btnDelEvt.Enabled = False
            btnUpdateEvt.Enabled = False
            btnAddStream.Enabled = False
            btnDelSt.Enabled = False
            btnUpSt.Enabled = False

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End If


End Sub

Public Sub Fill_Profile()
    Try

        Dim sqlComProfiles As New SqlCommand

        '* Open the connection to be used for several operations during
        '*  form load
        If Not sqlConNW.State = Data.ConnectionState.Open Then
            sqlConNW.Open()
        End If

        '* Configure command object
        With sqlComProfiles
            .Connection = sqlConNW
            .CommandType = Data.CommandType.StoredProcedure
            .CommandText = "up_Fill_Profile_Combo"
        End With

        '* Clear existing records if present and loads dataset table
        If dsNW.Tables.Contains("Profile") Then
```

```
                dsNW.Tables("Profile").Clear()
            End If
            sqlDANW.SelectCommand = sqlComProfiles
            sqlDANW.Fill(dsNW, "Profile")

            '* Configure combo box
            cboProfile.DataSource = dsNW.Tables("Profile")
            cboProfile.DataValueField = "ProfileID"
            cboProfile.DataTextField = "ProfileName"

            '*Bind the data and set the loading variable to false.
            cboProfile.DataBind()
            blnIsLoading = False

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub

    Public Sub Fill_Events()

        Try

            'This function fills the event combo box and is used by several
            '   different procedures in the form.

            '* set boolean test variable to true.
            blnIsLoading = True

            'Declare SQL command object
            Dim sqlComEvents As New SqlCommand


            '* Open the connection if not already open.
            If Not sqlConNW.State = Data.ConnectionState.Open Then
                sqlConNW.Open()
            End If

            '
            '* Configure command object
            With sqlComEvents
                .Connection = sqlConNW
                .CommandType = Data.CommandType.StoredProcedure
                .CommandText = "up_Load_Event_Combo"
                .Parameters.Add("@ProfileID",    Data.SqlDbType.Int).Value    =
cboProfile.SelectedValue
            End With

            '* Clear existing records if present and loads dataset table
```

```
            If dsnw1.Tables.Contains("Event") Then
                dsnw1.Tables("Event").Clear()
            End If
            sqlDANW.SelectCommand = sqlComEvents
            sqlDANW.Fill(dsnw1, "Event")

            '* Configure combo box
            cboEvent.DataSource = dsnw1.Tables("Event")
            cboEvent.DataValueField = "EventID"
            cboEvent.DataTextField = "EventName"
            cboEvent.DataBind()

            '* Set the boolean test variable to false.
            blnIsLoading = False

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub

    Public Sub Fill_Event_Info()
        Try

            '* Configure connection object connection string to be used
            '*  thrhoughout the form.
            Dim sqlComEvent As New SqlCommand
            Dim dsNW2 As New Data.DataSet



            '* Open the connection to be used for several operations during
            '*  form load
            If Not sqlConNW.State = Data.ConnectionState.Open Then
                sqlConNW.Open()
            End If

            '*
            '* Load various combo boxes used throughout the form
            With sqlComEvent
                .Connection = sqlConNW
                .CommandType = Data.CommandType.StoredProcedure
                .CommandText = "up_Load_Event_Info"
                .Parameters.Add("@EventID",     Data.SqlDbType.Int).Value     =
cboEvent.SelectedValue
            End With

            '* Clear existing records if present and loads dataset table
            If dsNW2.Tables.Contains("Event") Then
                dsNW2.Tables("Event").Clear()
```

```
            End If
            sqlDANW.SelectCommand = sqlComEvent
            sqlDANW.Fill(dsNW2, "Event")

            '* Configure text boxes
            txtUpEvtName.Text = dsNW2.Tables(0).Rows(0).Item(1).ToString
            txtUpEvtDt.Text = dsNW2.Tables(0).Rows(0).Item(2).ToString

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub

    Public Sub Fill_Stream()
        Try

            '* set the test variable status to true.
            blnIsLoading = True

            '* Configure connection object connection string to be used
            '*  thrhoughout the form.
            Dim sqlComStreams As New SqlCommand
            Dim dsNW2 As New Data.DataSet


            '* Open the connection to be used for several operations during
            '*  form load
            If Not sqlConNW.State = Data.ConnectionState.Open Then
                sqlConNW.Open()
            End If

            '*
            '* Load various combo boxes used throughout the form
            With sqlComStreams
                .Connection = sqlConNW
                .CommandType = Data.CommandType.StoredProcedure
                .CommandText = "up_Load_Stream_Combo"
                .Parameters.Add("@ProfileID",    Data.SqlDbType.Int).Value    =
cboProfile.SelectedValue
            End With

            '* Clear existing records if present and loads dataset table
            If dsNW2.Tables.Contains("Stream") Then
                dsNW2.Tables("Stream").Clear()
            End If
            sqlDANW.SelectCommand = sqlComStreams
            sqlDANW.Fill(dsNW2, "Stream")

            '* Configure combo box
```

```
            cboStream.DataSource = dsNW2.Tables("Stream")
            cboStream.DataValueField = "StreamID"
            cboStream.DataTextField = "StreamName"

            cboStream.DataBind()
            '* set the test variable equal to false indicating that the combo
is finished loading.
            blnIsLoading = False

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try

    End Sub

    Public Sub Fill_Stream_Info()
        Try

            '* Configure connection object connection string to be used
            '*  thrhoughout the form.
            Dim sqlComStream As New SqlCommand
            Dim dsNW2 As New Data.DataSet



            '* Open the connection to be used for several operations during
            '*  form load
            If Not sqlConNW.State = Data.ConnectionState.Open Then
                sqlConNW.Open()
            End If

            '*
            '* Load various combo boxes used throughout the form
            With sqlComStream
                .Connection = sqlConNW
                .CommandType = Data.CommandType.StoredProcedure
                .CommandText = "up_Load_Stream_Info"
                .Parameters.Add("@StreamID",    Data.SqlDbType.Int).Value    =
cboStream.SelectedValue
            End With

            '* Clear existing records if present and loads dataset table
            If dsNW2.Tables.Contains("Stream") Then
                dsNW2.Tables("Stream").Clear()
            End If
            sqlDANW.SelectCommand = sqlComStream
            sqlDANW.Fill(dsNW2, "Stream")

            '* Configure text boxes
```

```
            txtUpStName.Text = dsNW2.Tables(0).Rows(0).Item(1).ToString
            txtUpStLoc.Text = dsNW2.Tables(0).Rows(0).Item(2).ToString

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub

    Protected Sub cboProfile_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles cboProfile.SelectedIndexChanged
        'This updates the form variables based on a change in profile.
        Try

            'Call the fill procedures.
            Label2.Text = cboProfile.SelectedIndex.ToString
            Call Fill_Events()
            Call Fill_Stream()
            Call Fill_Event_Info()


            '* Clear any previous status update labels.
            lblUpdate2.Text = ""
            lblUpdate3.Text = ""
            lblUpdate1.Text = ""
            lblUpdate4.Text = ""

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub

    Protected Sub cboEvent_SelectedIndexChanged(ByVal sender As Object, ByVal
e As System.EventArgs) Handles cboEvent.SelectedIndexChanged

        Try

            '* fill the text input boxes to reflect the new selection.
            Call Fill_Event_Info()
            btnDelEvt.Enabled = True
            '* Clear any previous status update labels.
            lblUpdate2.Text = ""
            lblUpdate3.Text = ""
            lblUpdate1.Text = ""
            lblUpdate4.Text = ""

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
```

```
        End Try
    End Sub

    Protected Sub cboStream_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles cboStream.SelectedIndexChanged
        Try

            '* fill the text input boxes to reflect the new selection.
            Call Fill_Stream_Info()
            btnDelSt.Enabled = True

            '* Clear any previous status update labels.
            lblUpdate2.Text = ""
            lblUpdate3.Text = ""
            lblUpdate1.Text = ""
            lblUpdate4.Text = ""

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub

    Protected Sub btnNewEvt_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnNewEvt.Click
        Try

            '* Open the connection to be used for several operations during
            '*  form load
            If Not sqlConNW.State = Data.ConnectionState.Open Then
                sqlConNW.Open()
            End If

            '* Declare a SQL command object
            Dim sqlComAddStream As New SqlCommand

            '* Set the properties of the SQL Command Object
            With sqlComAddStream
                .Connection = sqlConNW
                .CommandType = Data.CommandType.StoredProcedure
                .CommandText = "usp_Event_INS"
                .Parameters.Add("@ProfileID",    Data.SqlDbType.Int).Value    =
cboProfile.SelectedValue
                .Parameters.Add("@EventName",  Data.SqlDbType.VarChar).Value  =
txtEvnt1.Text
                .Parameters.Add("@EventDate",    Data.SqlDbType.DateTime).Value
= CDate(txtDate.Text)
                .Parameters.Add("@Dscr",    Data.SqlDbType.VarChar).Value    =
(txtDscr.Text)
            End With
```

```
            '* Execute the non-query.
            sqlComAddStream.ExecuteNonQuery()

            '* Set the update label to reflect that the update went through.
            lblUpdate1.Text = "Your event " + txtEvnt1.Text.ToString + " on "
+ txtDate.Text.ToString + " was added."
            lblUpdate2.Text = ""
            lblUpdate3.Text = ""
            lblUpdate4.Text = ""

            '* Repopulate the combo box
            Call Fill_Events()

            '* Clear the input fields
            txtEvnt1.Text = ""
            txtDate.Text = ""
            txtDscr.Text = String.Empty


            '* Set the focus to the status message and disable the buttons.
            lblUpdate1.Focus()
            btnNewEvt.Enabled = False

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub

    Protected Sub btnUpdateEvt_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnUpdateEvt.Click
        Try

            '* Open the connection to be used for several operations during
            '*  form load
            If Not sqlConNW.State = Data.ConnectionState.Open Then
                sqlConNW.Open()
            End If

            '* Declare a SQL command object
            Dim sqlComAddStream As New SqlCommand

            '* Set the properties of the SQL Command Object
            With sqlComAddStream
                .Connection = sqlConNW
                .CommandType = Data.CommandType.StoredProcedure
                .CommandText = "up_Event_UPD"
                .Parameters.Add("@EventID",     Data.SqlDbType.Int).Value     =
cboEvent.SelectedValue
```

```
                .Parameters.Add("@EventName", Data.SqlDbType.VarChar).Value =
txtUpEvtName.Text
                .Parameters.Add("@EventDate",   Data.SqlDbType.DateTime).Value
= CDate(txtUpEvtDt.Text)
            End With

            '* Execute the non-query.
            sqlComAddStream.ExecuteNonQuery()

            '* Set the update label to reflect that the update went through.
            lblUpdate2.Text = "Your event " + txtUpEvtName.Text.ToString + "
on " + txtUpEvtDt.Text.ToString + " was updated."
            lblUpdate3.Text = ""
            lblUpdate1.Text = ""
            lblUpdate4.Text = ""

            '* Repopulate the combo box and the input fields.

            Call Fill_Events()
            If blnIsLoading = True Then
                Exit Sub
            Else
                Call Fill_Event_Info()
            End If

            '* Set the focus to the status message and disable the buttons.
            lblUpdate2.Focus()
            btnDelEvt.Enabled = False
            btnUpdateEvt.Enabled = False

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub

    Protected Sub btnDelEvt_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnDelEvt.Click

        Try

            '* Open the connection to be used for several operations during
            '*  form load
            If Not sqlConNW.State = Data.ConnectionState.Open Then
                sqlConNW.Open()
            End If

            '* Declare a SQL command object
            Dim sqlComAddStream As New SqlCommand
```

```
            '* Set the properties of the SQL Command Object
            With sqlComAddStream
                .Connection = sqlConNW
                .CommandType = Data.CommandType.StoredProcedure
                .CommandText = "up_Event_Del"
                .Parameters.Add("@EventID",     Data.SqlDbType.Int).Value     =
cboEvent.SelectedValue
            End With

            '* Execute the non-query.
            sqlComAddStream.ExecuteNonQuery()

            '* Set the update label to reflect that the update went through.
            lblUpdate2.Text = "Your event " + txtUpEvtName.Text.ToString + "
on " + txtUpEvtDt.Text.ToString + " was deleted."
            lblUpdate3.Text = ""
            lblUpdate1.Text = ""
            lblUpdate4.Text = ""

            '* Repopulate the combo box and the input fields.
            Call Fill_Events()
            If blnIsLoading = True Then
                Exit Sub
            Else
                Call Fill_Event_Info()
            End If

            '* Set the focus to the status message and disable the buttons.
            lblUpdate2.Focus()
            btnDelEvt.Enabled = False
            btnUpdateEvt.Enabled = False

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub

    Protected  Sub  btnAddStream_Click(ByVal  sender  As  Object,  ByVal  e  As
System.EventArgs) Handles btnAddStream.Click
        Try

            '* Open the connection to be used for several operations during
            '*  form load
            If Not sqlConNW.State = Data.ConnectionState.Open Then
                sqlConNW.Open()
            End If

            '* Declare a SQL command object
            Dim sqlComAddStream As New SqlCommand
```

```
            '* Set the properties of the SQL Command Object
            With sqlComAddStream
                .Connection = sqlConNW
                .CommandType = Data.CommandType.StoredProcedure
                .CommandText = "usp_Stream_INS"
                .Parameters.Add("@ProfileID",    Data.SqlDbType.Int).Value    =
cboProfile.SelectedValue
                .Parameters.Add("@StreamName",   Data.SqlDbType.VarChar).Value
= txtNewStName.Text
                .Parameters.Add("@StreamLoc",  Data.SqlDbType.VarChar).Value  =
txtNewStLoc.Text

            End With

            '* Execute the non-query.
            sqlComAddStream.ExecuteNonQuery()

            '* Set the update label to reflect that the update went through.
            lblUpdate3.Text = "Your stream " + txtNewStName.Text.ToString + "
located at " + txtNewStLoc.Text.ToString + " was added."
            lblUpdate2.Text = ""
            lblUpdate1.Text = ""
            lblUpdate4.Text = ""

            '* Repopulate the combo box
            Call Fill_Stream()

            '* Clear the input fields
            txtNewStName.Text = ""
            txtNewStLoc.Text = ""

            '* Set the focus to the status message and disable the buttons.
            lblUpdate3.Focus()
            btnAddStream.Enabled = False

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub

    Protected  Sub  btnUpSt_Click(ByVal  sender  As  Object,  ByVal  e  As
System.EventArgs) Handles btnUpSt.Click
        Try

            '* Open the connection to be used for several operations during
            '*  form load
            If Not sqlConNW.State = Data.ConnectionState.Open Then
                sqlConNW.Open()
```

```
            End If

            '* Declare a SQL command object
            Dim sqlComAddStream As New SqlCommand

            '* Set the properties of the SQL Command Object
            With sqlComAddStream
                .Connection = sqlConNW
                .CommandType = Data.CommandType.StoredProcedure
                .CommandText = "up_Stream_UPD"
                .Parameters.Add("@StreamID",    Data.SqlDbType.Int).Value    =
cboStream.SelectedValue
                .Parameters.Add("@StreamName",   Data.SqlDbType.VarChar).Value
= txtUpStName.Text
                .Parameters.Add("@StreamLoc",  Data.SqlDbType.VarChar).Value  =
txtUpStLoc.Text

            End With

            '* Execute the non-query.
            sqlComAddStream.ExecuteNonQuery()

            '* Set the update label to reflect that the update went through.
            lblUpdate4.Text = "Your Stream " + txtUpStName.Text.ToString + "
located at " + txtUpStLoc.Text.ToString + " was updated."
            lblUpdate3.Text = ""
            lblUpdate1.Text = ""
            lblUpdate2.Text = ""

            '* Repopulate the combo box and the input fields.
            Call Fill_Stream()
            If blnIsLoading = True Then
                Exit Sub
            Else
                Call Fill_Stream_Info()
            End If

            '* Set the focus to the status message and disable the buttons.
            lblUpdate4.Focus()
            btnDelSt.Enabled = False
            btnUpSt.Enabled = False

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try

    End Sub
```

```
    Protected Sub btnDelSt_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnDelSt.Click
        Try


            '* Open the connection to be used for several operations during
            '*  form load
            If Not sqlConNW.State = Data.ConnectionState.Open Then
                sqlConNW.Open()
            End If

            '* Declare a SQL command object
            Dim sqlComAddStream As New SqlCommand

            '* Set the properties of the SQL Command Object
            With sqlComAddStream
                .Connection = sqlConNW
                .CommandType = Data.CommandType.StoredProcedure
                .CommandText = "up_Stream_DEL"
                .Parameters.Add("@StreamID",    Data.SqlDbType.Int).Value    =
cboStream.SelectedValue
            End With

            '* Execute the non-query.
            sqlComAddStream.ExecuteNonQuery()

            '* Set the update label to reflect that the update went through.
            lblUpdate2.Text = "Your Stream " + txtUpStName.Text.ToString + "
located at " + txtUpStLoc.Text.ToString + " was deleted."
            lblUpdate3.Text = ""
            lblUpdate1.Text = ""
            lblUpdate4.Text = ""

            '* Repopulate the combo box and the input fields.

            Call Fill_Stream()
            If blnIsLoading = True Then
                Exit Sub
            Else
                Call Fill_Stream_Info()
            End If

            '* Set the focus to the status message and disable the buttons.
            lblUpdate4.Focus()
            btnDelSt.Enabled = False
            btnUpSt.Enabled = False

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
```

```
        End Try
    End Sub

    Protected Sub txtEvnt1_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtEvnt1.TextChanged
        Try

            '* Enable the button if the text has been changed.
            btnNewEvt.Enabled = True
            txtDate.Focus()

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub


    Protected Sub txtUpEvtName_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtUpEvtName.TextChanged
        Try
            '* enable the controls if the text is changed.
            btnDelEvt.Enabled = False
            btnUpdateEvt.Enabled = True
            txtUpEvtDt.Focus()

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub

    Protected Sub txtUpEvtDt_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtUpEvtDt.TextChanged
        Try
            '* enable the controls if the text is changed.
            lblUpdate2.Focus()
        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try

    End Sub

    Protected Sub txtNewStName_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtNewStName.TextChanged
        Try

            'enable the add button and set focus
            btnAddStream.Enabled = True
```

```
            txtNewStLoc.Focus()

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub

    Protected Sub txtNewStLoc_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtNewStLoc.TextChanged
        Try

            'Set the focus
            lblUpdate3.Focus()

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub

    Protected Sub txtUpStName_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtUpStName.TextChanged
        Try

            'Enable the appropriate buttons and set focus.
            btnDelSt.Enabled = False
            btnUpSt.Enabled = True
            txtUpStLoc.Focus()

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub

    Protected Sub btnExit_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnExit.Click
        Try

            'Back to main.
            Response.Redirect("Main.aspx")

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub
```

```
    Protected Sub btnExit1_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnExit1.Click
        Try

            'Back to main.
            Response.Redirect("Main.aspx")

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub
End Class
```

Create.aspx

```
Imports System.Data.SqlClient
Imports System.Net
Imports System.Xml
Imports System.Data
Imports CMClass
'*********************************************************
'The create page allows users to create accounts and actions
'to justify their choice of actions for a given event.
'Author: James Parrish
'Created: April 24, 2008
'Revised: April 28, 2008
'Revision:  (4/28) Added field to display the event description.
'*********************************************************
Partial Class Create
    Inherits System.Web.UI.Page


    '* Declare the form level variables
    Dim blnIsLoading As Boolean
    Dim sqlConNW As New SqlConnection(ConStr)
    Dim dsEvent As New Data.DataSet
    Dim sqlDaCreate As New SqlDataAdapter
    Dim intIdent As Integer


    Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        If Not IsPostBack Then
```

```
            'Fill the event combo box.
            Call Fill_Events()

        End If
    End Sub

    Public Sub Fill_Events() 'Fill the events combo box at the top of the
page based on the login profile.
        Try



            'This function fills the event combo box and is used by several
            '   different procedures in the form.

            '* set boolean test variable to true.
            blnIsLoading = True

            'Declare SQL command object
            Dim sqlComEvents As New SqlCommand


            '* Open the connection if not already open.
            If Not sqlConNW.State = Data.ConnectionState.Open Then
                sqlConNW.Open()
            End If

            '
            '* Configure command object
            With sqlComEvents
                .Connection = sqlConNW
                .CommandType = Data.CommandType.StoredProcedure
                .CommandText = "up_Load_Event_Combo"
                .Parameters.Add("@ProfileID",    Data.SqlDbType.Int).Value    =
Request.QueryString("Profile")

            End With

            '* Clear existing records if present and loads dataset table
            If dsEvent.Tables.Contains("Event") Then
                dsEvent.Tables("Event").Clear()
            End If
            sqlDaCreate.SelectCommand = sqlComEvents
            sqlDaCreate.Fill(dsEvent, "Event")

            '* Configure combo box
            cboEvent.DataSource = dsEvent.Tables("Event")
            cboEvent.DataValueField = "EventID"
            cboEvent.DataTextField = "BoxInfo"
            cboEvent.DataBind()
```

```
            '* Set the boolean test variable to false.
            blnIsLoading = False
        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")

        End Try
    End Sub

    Protected Sub btnSubmit_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnSubmit.Click
        Try


            If txtAccount.Text <> String.Empty And txtAction.Text <>
String.Empty Then
                '* Call the subs to add the account and action.
                Call Add_Account()
                Call Add_Action()

                '* Set the update label to reflect that the update went
through.
                lblUpdate.Text = "Your action and reason for " +
cboEvent.SelectedItem.ToString + " was added."
                txtAccount.Text = ""
                txtAction.Text = ""
                lblUpdate.Focus()
            Else
                '* Let the user know that the forgot something.
                lblUpdate.Text = "Please enter an action and a reason for the
action."
                lblUpdate.Focus()
                Exit Sub
            End If
        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try


    End Sub
    Public Sub Add_Account()
        Try


            blnIsLoading = True
            '* Get the EventID from the Combo Box.
            Dim intEvent As Integer = cboEvent.SelectedValue
```

```
            '* Open the connection to be used for several operations during
            '*  form load
            If Not sqlConNW.State = Data.ConnectionState.Open Then
                sqlConNW.Open()
            End If

            '* Declare a SQL command object
            Dim sqlComAddAccount As New SqlCommand

            '* Set the properties of the SQL Command Object
            With sqlComAddAccount
                .Connection = sqlConNW
                .CommandType = Data.CommandType.StoredProcedure
                .CommandText = "up_Account_INS"
                .Parameters.Add("@EventID",     Data.SqlDbType.Int).Value     =
intEvent
                .Parameters.Add("@AccountText",  Data.SqlDbType.VarChar).Value
= txtAccount.Text
                .Parameters.Add("@AccountDate",   SqlDbType.DateTime).Value   =
Date.Now
                'Get a return parameter to accept the AccountID to be stored
in the intIdent variable.  This will be the value for
                'the AccountID parameter that will be needed in the SP to add
the Action.
                .Parameters.Add("@Identity",     SqlDbType.Int).Direction     =
ParameterDirection.ReturnValue

            End With

            '* Execute the non-query.
            sqlComAddAccount.ExecuteNonQuery()

            '* Accept the return value.
            intIdent = sqlComAddAccount.Parameters("@Identity").Value

            '* Set the text variable to false.
            blnIsLoading = False

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try

    End Sub
    Public Sub Add_Action()
        Try



            If blnIsLoading <> True Then
```

```
                '* Get the EventID from the Combo Box.
                Dim intEvent As Integer = cboEvent.SelectedValue

                '*  Open  the  connection  to  be  used  for  several  operations
during
                '*  form load
                If Not sqlConNW.State = Data.ConnectionState.Open Then
                    sqlConNW.Open()
                End If

                '* Declare a SQL command object
                Dim sqlComAddAccount As New SqlCommand

                '* Set the properties of the SQL Command Object
                With sqlComAddAccount
                    .Connection = sqlConNW
                    .CommandType = Data.CommandType.StoredProcedure
                    .CommandText = "up_Action_INS"
                    .Parameters.Add("@AccountID",  Data.SqlDbType.Int).Value  =
intIdent
                    .Parameters.Add("@ActionText",
Data.SqlDbType.VarChar).Value = txtAction.Text

                End With

                '* Execute the non-query.
                sqlComAddAccount.ExecuteNonQuery()

            Else
                Exit Sub
            End If
        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub

    Protected  Sub  Button2_Click(ByVal  sender  As  Object,  ByVal  e  As
System.EventArgs) Handles Button2.Click

        '* Retrieve and store the querystring variables.
        Try


            Dim intProfile As Integer = Request.QueryString("Profile")
            Dim intIdent As Integer = Request.QueryString("Code")
            Dim s As String = Server.UrlEncode(intProfile)
            Dim t As String = Server.UrlEncode(intIdent)
```

```
            'Redirect to the chosen form.
            Response.Redirect("Main.aspx?Profile=" & s & "&Code=" & t)

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try

    End Sub

    Protected Sub cboEvent_SelectedIndexChanged(ByVal sender As Object, ByVal
e As System.EventArgs) Handles cboEvent.SelectedIndexChanged
        Try


            If blnIsLoading = False Then
                '* Instantiate the SQL items for the form
                Dim sqlComEvents1 As New SqlCommand
                Dim sqlDaCreate1 As New SqlDataAdapter
                Dim dsEvent1 As New Data.DataSet

                '* Open the connection if not already open.
                If Not sqlConNW.State = Data.ConnectionState.Open Then
                    sqlConNW.Open()
                End If

                '
                '* Configure command object
                With sqlComEvents1
                    .Connection = sqlConNW
                    .CommandType = Data.CommandType.StoredProcedure
                    .CommandText = "up_Load_Event_Combo"
                    .Parameters.Add("@ProfileID",  Data.SqlDbType.Int).Value  =
Request.QueryString("Profile")

                End With

                '* Clear existing records if present and loads dataset table
                If dsEvent1.Tables.Contains("Event") Then
                    dsEvent1.Tables("Event").Clear()
                End If
                sqlDaCreate1.SelectCommand = sqlComEvents1
                sqlDaCreate1.Fill(dsEvent1, "Event")
                txtDscr.Text                                                    =
dsEvent1.Tables("Event").Rows(cboEvent.SelectedIndex.ToString).Item("Dscr").T
oString
            Else
                Exit Sub
            End If
        Catch ex As Exception
```

```
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try

    End Sub


End Class
```

Intro.aspx

```
''*********************************************************
'The intro page validates the code input by the user and
'sets the profile that the user will use to interact with
'the system.
'Author: James Parrish
'Created: April 24,2008
''*********************************************************
Partial Class Intro
    Inherits System.Web.UI.Page

    Protected Sub btnEnter_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnEnter.Click
        Try

            Dim t As String = Mid(txtLogin.Text, 3)
            '* Test for incomplete or empty input.
            If Len(txtLogin.Text) < 4 Then
                lblUpdate.Text = "You must enter a four character code."
                Exit Sub
                '* test for incorrect input
            End If
            If txtLogin.Text = "ADMN" Then
                Dim s As String = Server.UrlEncode(0)
                Response.Redirect("Main.aspx?Profile=" & s & "&Code=" & t)
            ElseIf    IsNumeric(Right(txtLogin.Text,    2))    =    False    Or
IsNumeric(Left(txtLogin.Text, 2)) = True Then
                lblUpdate.Text = "Please enter the code that you used for
your survey.  It should consist of two letters and two numbers (AA##)."
                Exit Sub
                '* Take the login string and use the first two characters
                '* to set the profile for the users.

            ElseIf Left(txtLogin.Text, 2) = "CV" Then
                Dim s As String = Server.UrlEncode(1)
                Response.Redirect("Main.aspx?Profile=" & s & "&Code=" & t)
```

```
            ElseIf Left(txtLogin.Text, 2) = "NV" Then
                Dim s As String = Server.UrlEncode(2)
                Response.Redirect("Main.aspx?Profile=" & s & "&Code=" & t)
            ElseIf Left(txtLogin.Text, 2) = "MU" Then
                Dim s As String = Server.UrlEncode(3)
                Response.Redirect("Main.aspx?Profile=" & s & "&Code=" & t)
            ElseIf Left(txtLogin.Text, 2) = "TM" Then
                Dim s As String = Server.UrlEncode(4)
                Response.Redirect("Main.aspx?Profile=" & s & "&Code=" & t)
            Else
                '* They have not matched a system type, have them try again.
                lblUpdate.Text = "We do not have a login of that type on
record, please try again."
                lblUpdate.Focus()

            End If

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub

    Protected Sub txtLogin_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtLogin.TextChanged
        btnEnter.Focus()
    End Sub
End Class
```

Main.aspx

```
Partial Class Main
    Inherits System.Web.UI.Page


    Protected Sub btnExit_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnExit.Click
        'Go back to the beginning.
        Response.Redirect("Intro.aspx")
    End Sub

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        If Not IsPostBack Then
            Try
```

```
                'Enable/Disable the buttons based on the type of profile they
have.
                'Admins can Report and Administer, Users can Create and
Review.
                Dim intProfile As Integer = Request.QueryString("Profile")
                If intProfile <> 0 Then
                    'User
                    btnCreate.Visible = True
                    btnReview.Visible = True
                    btnReport.Visible = False
                    btnAdmin.Visible = False
                    btnExit.Visible = True
                Else
                    'Admin
                    btnCreate.Visible = False
                    btnReview.Visible = False
                    btnReport.Visible = True
                    btnAdmin.Visible = True
                    btnExit.Visible = True
                End If
            Catch ex As Exception
                Context.Items.Add("Error", ex.Message.ToString)
                Server.Transfer("ErrorPage.aspx")
            End Try
        End If

    End Sub

    Protected Sub btnCreate_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnCreate.Click
        Try


            'Pull the variables from the querystring.
            Dim intProfile As Integer = Request.QueryString("Profile")
            Dim intCode As Integer = Request.QueryString("Code")

            'Encode the new strings.
            Dim s As String = Server.UrlEncode(intProfile)
            Dim t As String = Server.UrlEncode(intCode)

            'Redirect to the chosen form.
            Response.Redirect("Create.aspx?Profile=" & s & "&Code=" & t)

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
```

```
    End Sub

    Protected Sub btnReview_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnReview.Click
        Try


            'Pull the variables from the querystring.
            Dim intProfile As Integer = Request.QueryString("Profile")
            Dim intCode As Integer = Request.QueryString("Code")

            'Encode the new strings.
            Dim s As String = Server.UrlEncode(intProfile)
            Dim t As String = Server.UrlEncode(intCode)

            'Redirect to the chosen form.
            Response.Redirect("Review.aspx?Profile=" & s & "&Code=" & t)

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub

    Protected Sub btnReport_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnReport.Click
        Try
            'Go to reports.
            Response.Redirect("Report.aspx")

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub

    Protected Sub btnAdmin_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnAdmin.Click

        Try
            'Go to admin.
            Response.Redirect("Admin.aspx")

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try

    End Sub
End Class
```

Report.aspx

```
Imports System.Data.SqlClient
Imports Microsoft.Reporting.WebForms
Imports CMClass
'*********************************************************************
'* The report page pulls in the data from the reviews and aggregates
'* it into a report (Summary.rdlc) for decision makers to use to
'* guide their CM.
'* Author: James Parrish
'* Create: April 24, 2008
'*********************************************************************
Partial Class Report
    Inherits System.Web.UI.Page

    '* Declare the form level variables.
    Dim sqlConNW As New SqlConnection(ConStr)
    Dim sqlCon As New SqlConnection
    Dim sqlCmd As New SqlCommand
    Dim myDS As New Data.DataSet
    Dim sqlDA As New SqlDataAdapter

    Protected Sub btnExit_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnExit.Click
        Try

            '* Go to the main page.
            Response.Redirect("Main.aspx")

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub
End Class
```

Review.aspx

```
Imports System.Data.SqlClient
Imports CMClass
'*****************************************************************
'* The review page pulls in action and account data and allows the
'* the users to rate how they feel about certain accounts and actions
'* that will be stored in the database.  Users can only review items
'* once.
'* Author: James Parrish
'* Create: April 24, 2008
'*****************************************************************

Partial Class Review
    Inherits System.Web.UI.Page
    '* Declare the form level variables
    Dim blnIsLoading As Boolean = True
    Dim sqlConNW As New SqlConnection(ConStr)
    Dim dsEvent As New Data.DataSet
    Dim sqlDaCreate As New SqlDataAdapter
    Dim intIdent As Integer
    Dim intProfile As Integer
    Dim intAccount As Integer
    Dim intAction As Integer


    Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        If Not IsPostBack Then
            '* Fill the controls and pull in variables from the querystring.
            Call Fill_Events()
            Call Fill_Review()
            cboEvent.SelectedIndex = 0
            intIdent = Request.QueryString("Code")
            intProfile = Request.QueryString("Profile")
        End If
    End Sub

    Public Sub Fill_Events()
        Try


            'Fill the events combo box at the top of the page based on the
login profile.
            'This function fills the event combo box and is used by several
            '   different procedures in the form.

            '* set boolean test variable to true.
            blnIsLoading = True
```

```
            'Declare SQL command object
            Dim sqlComEvents As New SqlCommand


            '* Open the connection if not already open.
            If Not sqlConNW.State = Data.ConnectionState.Open Then
                sqlConNW.Open()
            End If

            '
            '* Configure command object
            With sqlComEvents
                .Connection = sqlConNW
                .CommandType = Data.CommandType.StoredProcedure
                .CommandText = "up_Load_Event_Combo"
                .Parameters.Add("@ProfileID",    Data.SqlDbType.Int).Value    =
Request.QueryString("Profile")

            End With

            '* Clear existing records if present and loads dataset table
            If dsEvent.Tables.Contains("Event") Then
                dsEvent.Tables("Event").Clear()
            End If
            sqlDaCreate.SelectCommand = sqlComEvents
            sqlDaCreate.Fill(dsEvent, "Event")

            '* Configure combo box
            cboEvent.DataSource = dsEvent.Tables("Event")
            cboEvent.DataValueField = "EventID"
            cboEvent.DataTextField = "EventName"
            cboEvent.DataBind()


            '* Set the boolean test variable to false.
            blnIsLoading = False

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub
    Public Sub Fill_Review()

        Try

            'Declare SQL command object
            Dim sqlComReview As New SqlCommand
            Dim dsReview As New Data.DataSet
```

```
            Dim sqlDAReview As New SqlDataAdapter
            intIdent = Request.QueryString("Code")
            intProfile = Request.QueryString("Profile")

            '* Open the connection if not already open.
            If Not sqlConNW.State = Data.ConnectionState.Open Then
                sqlConNW.Open()
            End If

            '
            '* Configure command object
            With sqlComReview
                .Connection = sqlConNW
                .CommandType = Data.CommandType.StoredProcedure
                .CommandText = "up_PullReviewData"
                .Parameters.Add("@EventID",     Data.SqlDbType.Int).Value     =
cboEvent.SelectedValue
                .Parameters.Add("@UserID",    Data.SqlDbType.VarChar).Value    =
intProfile & intIdent

            End With

            '* Clear existing records if present and loads dataset table
            If dsReview.Tables.Contains("up_PullReviewData") Then
                dsReview.Tables("up_PullReviewData").Clear()
            End If

            '* Set the selectcommand and fill the data adapter.
            sqlDAReview.SelectCommand = sqlComReview
            sqlDAReview.Fill(dsReview, "up_PullReviewData")

            '* If there is no data.
            If dsReview.Tables(0).Rows.Count = 0 Then
                '* Then tell them that there is none.
                txtAction.Text = "Nothing to Review"
                txtAccount.Text = "Nothing to Review"
            Else
                '* Otherwise, show them the data that you have.
                txtAction.Text                                                  =
dsReview.Tables(0).Rows(0).Item("ActionText").ToString
                txtAccount.Text                                                 =
dsReview.Tables(0).Rows(0).Item("AccountText").ToString
                intAccount = dsReview.Tables(0).Rows(0).Item("AccountID")
                intAction = dsReview.Tables(0).Rows(0).Item("ActionID")
            End If


        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
```

```
        End Try

    End Sub

    Protected Sub btnSubmit_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnSubmit.Click
        Try


            '* Make sure that there is a value for both ratings before adding
them to the DB.
            If rblRate1.SelectedValue = 0 Or rblRate2.SelectedValue = 0 Then
                lblUpdate.Text = "Please select a rating for both items."
            Else
                '* Add the review and refresh the combo.
                Call Add_Review()
                Call Fill_Review()
            End If

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try

    End Sub
    Public Sub Add_Review()

        Try


            '* Get the EventID from the Combo Box.
            Dim intEvent As Integer = cboEvent.SelectedValue

            'Declare the objects and fill variable values.
            Dim sqlComReview As New SqlCommand
            Dim dsReview As New Data.DataSet
            Dim sqlDAReview As New SqlDataAdapter
            intIdent = Request.QueryString("Code")
            intProfile = Request.QueryString("Profile")

            '* Open the connection to be used for several operations during
            '*  form load
            If Not sqlConNW.State = Data.ConnectionState.Open Then
                sqlConNW.Open()
            End If

            'Set the command object parameters.
            With sqlComReview
                .Connection = sqlConNW
                .CommandType = Data.CommandType.StoredProcedure
```

```
                .CommandText = "up_PullReviewData"
                .Parameters.Add("@EventID",     Data.SqlDbType.Int).Value     =
cboEvent.SelectedValue
                .Parameters.Add("@UserID",    Data.SqlDbType.VarChar).Value    =
intProfile & intIdent

            End With

            '* Clear existing records if present and loads dataset table
            If dsReview.Tables.Contains("up_PullReviewData") Then
                dsReview.Tables("up_PullReviewData").Clear()
            End If

            '* Set the select command and fill the data adapter.
            sqlDAReview.SelectCommand = sqlComReview
            sqlDAReview.Fill(dsReview, "up_PullReviewData")

            '* Populate the variable values.
            intAccount = dsReview.Tables(0).Rows(0).Item("AccountID")
            intAction = dsReview.Tables(0).Rows(0).Item("ActionID")


            '* Declare a SQL command object
            Dim sqlComAddAccount As New SqlCommand

            '* Set the properties of the SQL Command Object
            With sqlComAddAccount
                .Connection = sqlConNW
                .CommandType = Data.CommandType.StoredProcedure
                .CommandText = "up_AddReviewData"
                .Parameters.Add("@AccountID",    Data.SqlDbType.Int).Value    =
intAccount
                .Parameters.Add("@ActionID",    Data.SqlDbType.Int).Value    =
intAction
                .Parameters.Add("@UserID",    Data.SqlDbType.VarChar).Value    =
intProfile + intIdent
                .Parameters.Add("@AccountRating",  Data.SqlDbType.Int).Value  =
rblRate1.SelectedValue
                .Parameters.Add("@ActionRating",  Data.SqlDbType.Int).Value  =
rblRate2.SelectedValue

            End With

            '* Execute the non-query.
            sqlComAddAccount.ExecuteNonQuery()
            lblUpdate.Text = "You have added your review"
            Call Fill_Review()
        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
```

```
        End Try
    End Sub

    Protected Sub cboEvent_SelectedIndexChanged(ByVal sender As Object, ByVal
e As System.EventArgs) Handles cboEvent.SelectedIndexChanged
        Try

            'When the event changes, pull in applicable review data.
            Call Fill_Review()

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub

    Protected Sub btnExit_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnExit.Click
        Try

            ' Pull values from the querystrings and use them to build new
ones.
            Dim intProfile As Integer = Request.QueryString("Profile")
            Dim intIdent As Integer = Request.QueryString("Code")
            Dim s As String = Server.UrlEncode(intProfile)
            Dim t As String = Server.UrlEncode(intIdent)

            'Redirect to the chosen form.
            Response.Redirect("Main.aspx?Profile=" & s & "&Code=" & t)

        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
    End Sub
End Class
```

Code for RSS Feeders

```
Imports System.Net
Imports System.Xml
Imports System.Data
Imports System.Data.SqlClient
Imports CMClass

Partial Class RSSFeed1
```

```
    'Declare form level variables.
    Inherits System.Web.UI.UserControl
    Public Title As String = String.Empty
    Public Description As String = String.Empty
    Dim blnisloading As Boolean = True
    Dim sqlConNW As New SqlConnection(ConStr)

    Public Function RandomNumber(ByVal MaxNumber As Integer, _
        Optional ByVal MinNumber As Integer = 0) As Integer
        '* This random number generator will generate the  number that will
        '* serve to determine what feed is displayed in the create page.

        'initialize random number generator
        Dim r As New Random(System.DateTime.Now.Millisecond)

        'if passed incorrect arguments, swap them
        'can also throw exception or return 0

        If MinNumber > MaxNumber Then
            Dim t As Integer = MinNumber
            MinNumber = MaxNumber
            MaxNumber = t
        End If

        Return r.Next(MinNumber, MaxNumber)

    End Function

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        Try


            blnisloading = True
            Dim intProfile As Integer = Request.QueryString("Profile")
            'Load the stream data.
            '*************************************************************
            '* Configure connection object connection string to be used
            '*  thrhoughout the form.
            Dim sqlDAStreams As New SqlDataAdapter
            Dim sqlComStreams As New SqlCommand
            Dim dsNW2 As New Data.DataSet


            '* Open the connection to be used for several operations during
            '*  form load
            If Not sqlConNW.State = Data.ConnectionState.Open Then
                sqlConNW.Open()
            End If
```

```
            '*
            '* Load various combo boxes used throughout the form

            With sqlComStreams
                .Connection = sqlConNW
                .CommandType = Data.CommandType.StoredProcedure
                .CommandText = "up_Load_Stream_Combo"
                .Parameters.Add("@ProfileID",    Data.SqlDbType.Int).Value    =
intProfile

            End With

            '* Clear existing records if present and loads dataset table
            If dsNW2.Tables.Contains("Stream") Then
                dsNW2.Tables("Stream").Clear()
            End If

            '* Set the select command and fill the data adapter.
            sqlDAStreams.SelectCommand = sqlComStreams
            sqlDAStreams.Fill(dsNW2, "Stream")

            '* Finished loading...set the test variable to false.
            blnisloading = False


            '* populate the variable for the feed location
            Dim Feed As String = dsNW2.Tables(0).Rows(0).Item("StreamLoc")

            '* Populate the dataset.
            Dim rssData As DataSet = RefreshFeed(Feed)

            '* Set the values for the repeater items.
            Dim          channelItems          As          Object()          =
rssData.Tables(1).Rows(0).ItemArray
            Dim           titleColumn          As           Integer          =
rssData.Tables(1).Columns("title").Ordinal
            Dim         descriptionColumn         As         Integer         =
rssData.Tables(1).Columns("description").Ordinal

            '* Set values for title and description.
            Title = channelItems.GetValue(titleColumn).ToString()
            Description = channelItems.GetValue(descriptionColumn).ToString()

            '* Set the datasource for the repeater and bind it.
            Repeater1.DataSource = rssData.Tables(2)
            Repeater1.DataBind()
        Catch ex As Exception
            Context.Items.Add("Error", ex.Message.ToString)
            Server.Transfer("ErrorPage.aspx")
        End Try
```

```
    End Sub

    Private Function RefreshFeed(ByVal feed As String) As DataSet

        '*Request the website of the feed.
        Dim rssFeed As HttpWebRequest = DirectCast(WebRequest.Create(feed),
HttpWebRequest)

        'Create the dataset and read in the XML/
        Dim rssData As DataSet = New DataSet()
        rssData.ReadXml(rssFeed.GetResponse().GetResponseStream())

        '* Return the dataset.
        Return rssData

    End Function

    Protected Sub Repeater1_Unload(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Repeater1.Unload
        sqlConNW.Close()
    End Sub
End Class
```

APPENDIX D: INTERVIEW CODINGS

Trans#	Title	Techie?	Question	Statement	Code	-1/0/1
1.	Clerk	Yes	In general, what was your experience using the SenseMan system?	Um, by keeping the feeds up to date for the users so that they had pertinent information available for analysis for their events.	PInfo	1
			Tell me a little more about the feeds. When you looked at the feeds, what exactly were you looking at them for?	My intent was to see if there was anything in there that pertained to my area.	PPred	1
				My work area or the applications that I use and see if there is anything that might require action on my part or more work or more investigation or even maybe bringing it up in a meeting with my team.	PAnal	1
				Especially the feeds that pertain to a specific application, the feeds that pertain to the clerk's office and the more helpful ones pertained specifically to an application.	PInfo	1
				the news feeds were probably the least helpful...	PInfo	-1
				However, some of the feeds for the county news had information that was beneficial to the organization, not necessarily to the update of an application, but that is all relative to what is going on at the time. They could be very beneficial.	PPred	1

Trans#	Title	Techie?	Question	Statement	Code	-1/0/1
			With regards to your experiences administering the system, what thought process did you go through to choose the particular feeds?	I think that we looked at feeds that would be relative. First of all the feeds for applications, anything that would be considered a change, an improvement, or an enhancement, we wanted to notify the users of that impending change. The feeds that were more general such as the external news feeds were chosen because they were relative to our county and our organization. The clerk feed would be the most beneficial of those feeds because it was tailored directly to the clerk's office. I did not have much input as to the content of that feed, so to speak. That was mainly done at a higher level…at a manager level.	PInfo	1
			Did using the system change your feelings about software updates?	I would say that prior to having to having the system in place, the users were not aware of many of the system updates that were taking place unless they were the ones approving the updates or verifying the updates, so I would say that using the system is beneficial for all users because you are able to now know what is affecting the system and even if it does not pertain to every user I think that it is good to publish that information for everyone to see.	PInfo	1
			So prior to using the system would you say that generally just the people involved with the update in some way knew about it?	I would say that it was sporadic depending on the level or severity of the update and if it was a normal update and there was sufficient time to review the information, …	PInfo	0

Trans#	Title	Techie?	Question	Statement	Code	-1/0/1
				They would not have had the opportunity to provide their input on the implementation [before the sytem].	PAnal	1
				If it was an emergency update or fix or something like that it would usually only be known by the people directly involved by the people that were directly involved researching the issue or making the change.	PInfo	-1
			Because you were more aware of what was going in the organization as a whole, did you see anything that was going on in other areas that might affect yours?	Oh, definitely. One example would be the implementation of the credit card processing in the courts area. I see that as having an effect on multiple business areas in the organization and on multiple software platforms in the organization	PAnal	1
				And by keeping that limited to just one application knowing about it you could have some problems with consistency issues in the office where this department is implementing credit card [processing in the system] and the other department that has capability and is looking into it, does not know that the other department has already worked out all of the issues with the system and that they could leverage that information on their side. By having that published and the fact that it affected multiple systems…multiple areas, I think that is a good idea.	PInfo	1
			What would you say were the most/least beneficial aspects system	The most beneficial , I would say, is seeing the actual schedule of events that is taking place with each system.	PInfo	1

	Trans#	Title	Techie?	Question	Statement	Code
				[Depth of information] Meaning that if the information that a user sees is not related to the software application, they might be confused as to why they are being shown that information. But I think that for myself, I could use that information [not related to an application] to enhance my decisions, but I am not sure other users could use it that way.	PInfo	-1
			Do you think that in the grand scheme of things, people may be able to pick out things [from their environment] that may affect their updates?	Oh, definitely. I think that as people use the system more, and become more comfortable they will be able to become a more intelligent user.	PPred	1
2.	Sr. Deputy Clerk	No	In general, what was your experience using the SenseMan system?	I looked at the system. There was information in there that didn't necessarily pertain to me, but it was still interesting to read.	PInfo	0
			How was it interesting? Did it make you think about your own software updates?	Yes. I did think about my own software updates when I read about the other updates that were going on.	PPred	1
			How would you say that your use of the system changed your perceptions of software updates, if it did at all?	Before having the system, I would only really know about updates if I had time to look at the change before it went live.	PInfo	1
				I also think that before the system there were a lot of people that didn't know what was going on unless they were the person that requested the change.	PInfo	1

Trans#	Title	Techie?	Question	Statement	Code	-1/0/1
			Are there every any updates for your system, which may affect other systems such as finance and recording? Would you say that your experiences using the system made you more aware of this?	Yes, definitely. I was not aware that all the programs were so interconnected before [using the system].	PAnal	1
			What part of your experiences with the system made you view [updates] differently?	I saw them differently based on the information that I saw and read about what others were doing and what was going on in my application.	PInfo	1
			Did it make you think at any time that you should suggest some action based on what you read in the system? For example, after reading some of the feeds, did you say to yourself "we should do this.."?	Oh yeah, just because sometimes you aren't thinking about an update, when you read about others doing things you might say "we could do that" or "why aren't we doing that?" I also saw the item on the clerk news about the website enhancements and thought of a few things that we could do based on that for our department.	PAnal	1
3	Chief Deputy Clerk	Yes	Can you tell me a little bit about your experiences using the senseman system. What functions did you use, how did you interact with it?	The benefit there was that it gave me several logins and then I looked thorough the information and read through the information.	PInfo	1

Trans#	Title	Techie?	Question	Statement	Code	-1/0/1
				Although I looked at the functionality to provide input, I did not actually provide any input to the system.	PAnal	0
				So I feel that I tried to at least look at the breadth of the functionality of the system and to gain information by reading the feeds.	PInfo	1
			In your position, you are probably more keenly aware of updates than most employees here. That being the case, did the system change the way that you viewed system updates?	Personally, well the way that I look towards them…not from my own personal perspective, but from the holistic view of everyone being able to know about it. I have long been adherent to the fact that everyone should be involved in updates and that everyone should know what is going on.	PInfo	1
				Now with this particular system, because I helped to develop some of what [the system's] feeds were going to be, I tried to look at it from the standpoint from our average employee, if there is such a thing as an average employee and say what do they want to know and what are they frustrated about because I do feel that there is that frustration.	PInfo	1
				So the intent was to really provide all the logical, high level information with regards to what was going on with change in the environment, not necessarily the just limited to the software systems and the benefit that I see from the see from the senseman system was that appreciation that not only would they see that info..	PInfo	1

Trans#	Title	Techie?	Question	Statement	Code	-1/0/1
				…but also that they would be able to provide feedback if they would like to regarding that information to the organization.	PAnal	1
			So the main benefit from the system as you see it, is that it provides more adequate information to the members of the organization?	Providing not only more adequate information, but in some cases providing information where it once did not exist in the past, to put it more bluntly…or maybe where the information was held more tightly in the past.	PInfo	1
			Having information and providing feedback or acting on it are two entirely different things. Some of the users have provided feedback in the past few weeks, but do you feel that the average employee will choose to act on this information now that they have it?	To be honest, it will take some time. It is one of those things were people might be hesitant and not want to jump right in to providing actions. The people that would jump right in would be our higher level people and we have been doing a better job of getting those people the information and we do get input.	PAnal	0
				When I think of the average employee, I am thinking of the front-line personnel that have been using the system. People who, as we have gone through this process and explained what are intentions are with the senseman system, people who have expressed their appreciation for wanting that type of system. But because they have never had it, they might not be as willing to provide that input until they see that the trust is there and that there is value in providing that feedback.	PAnal	0

Trans#	Title	Techie?	Question	Statement	Code	-1/0/1
			Do you feel that the system has provided any enhancement to your personal ability to pick out cause and effect relationships or to identify environmental factors that may affect updates? Did it stimulate any ideas about updates?	I can't say that it stimulated a paradigm shift for me from the aspect that I have always been one that has been more liberal with sharing information and to get that information out there. It was still a paradigm shift for me and our environment, to say that rather than work through the supervisors to get that input, to give people, unsolicited, a way to get that information not through a formal reporting channel, and not to provide feedback from a formal reporting channel, which is the way that I would have envisioned doing it before [the system].	PInfo	0
				Its allowing a feed of information going out broadly and independent of what the supervisor or representative for the area is doing and being able to read that independently and being able to provide independent, anonymous, and unsolicited information back.	PInfo	0
			Do you believe that there is a benefit to the decision makers in getting unsolicited anonymous feedback that has not gone through a formal reporting channel?	Yes. Because no matter how good you are or how good I believe our own environment is there are always those because of other factors, where they may have been in the past in particular, other companies where the attitudes may have been different.	PAnal	1

Trans#	Title	Techie?	Question	Statement	Code	-1/0/1
				There are always those that may not feel that comfort…going back to my other comment. That they might not naturally do it until they know that there is a comfort and acceptance of it and I feel that you are always going to get different feedback back when it is unsolicited and anonymous and when they feel that they aren't going to be judged in some…whatever they perceive the way to be way.	PAnal	0
			Do you believe that there is benefit in getting the information in its raw form as opposed to it being transported through a channel?	Yes. And that is another good point. In some cases when it goes through a channel, some of it would be totally filtered out. So, definitely there is more input because you are getting all of the raw data, not filtered data or summarized data or "this is what we want you to hear" data.	PInfo	1
			Do you feel that the system has affected your views on the amount of the updates that you are going through?	So yes, it broadened… it made me more pensive and made me think more about what was going on and to pay more attention to the updates as opposed to just going through the stream of all the bombardment of the changes going on.	PVar	1
			So would you say that it increased your mindfulness?	Of each individual change, as an individual change and not just as a process that was going on. And in that, how that change was affecting people and systems and what information that they would need to know.	PAnal	1

Trans#	Title	Techie?	Question	Statement	Code	-1/0/1
			So the cause and effect relationships [regarding updates] were made more evident?	Yes	PAnal	1
			Some of your employees perform some very repetitive, process oriented tasks. Do you believe that they would get any benefit from this system since it encourages them to think in a more broad perspective?	Actually, and I wish that I could remember the comment that was made, but just in as we were explaining what was going on with the system we had one of those front-line people come to me and make an observation about how because they mainly use one type of system, it was forgotten that they also use another system. The person that came to me was in Jury Management and despite the fact that she uses the Jury system, she also daily uses the CourtView system. And they said how great this was because they were always left out of the loop on CourtView and anything with respect to that because it was assumed that they weren't a user of it. Therefore they would hear all sorts of things and actually be affected by what was going on and now they felt more a part of that group or a part of that team.	PInfo	1

Trans#	Title	Techie?	Question	Statement	Code	-1/0/1
				[It will be a] Major change as to the employees to work from images as opposed to the paper that they were used to using. So I don't see any employee that will not be impacted in a major way by some of the technology changes that are coming. And those will be change things that will in many cases that will threaten not necessarily their job, but the way that they do their job and their security in their job and their day to day routine. Those types of changes require a lot of communication and I see in those situations in particular this will be a great vehicle to help support the change process of those people that in the past have been [treated like] well, this is the way we are going to do it and just get used to it and don't complain about it (chuckles) and it will really help open up that whole change process.	PInfo	1
			In your opinion, what would you say that you feel is the most beneficial aspect of the system, and what was the least beneficial aspect of the system?	Boy, most beneficial is just the existence of the information…that there is information there that was not available previously, that I think that people do want.	PInfo	1

Trans#	Title	Techie?	Question	Statement	Code	-1/0/1
				Maybe least beneficial would be because we are so dynamic and that there is so much information the amount of information may cause a little bit of an information overload that in some cases might negate the benefits of the information being there.	PInfo	-1
				So the potential [for employees] to be inundated when they are so busy and in some cases have little time to look through [the information] the thing that is the most beneficial might in some ways be the least beneficial as well.	PInfo	0
				The fact that just a small amount of information is presented in the feed with the ability to click on the link to get more information is well conceived.	PInfo	1
				. It is just that the strength can become the weakness when you are giving information where none had existed before. It's a balancing act and I am not sure how you are able to weigh that out.	PInfo	0

Trans#	Title	Techie?	Question	Statement	Code	-1/0/1
4.	Clerk Employee	No	Can you tell me a little bit about your experiences using SenseMan?	So any change that takes place can really throw me into left field because I am not aware of what is going on whereas other people are.	PInfo	-1
				So when I logged into the Senseman system, I immediately noticed the two columns in the different colors. The left hand column notified me as to what was being changed and why. Prior to the senseman module, I never knew that.	PInfo	1
				[Before the system] All of a sudden [Munis] was changed and I didn't know why and I didn't have anything to do with it.	PAnal	1
				The left hand column had information about the Clerk and how things would affect it. Although because this system [Munis] spreads over the county and the clerk, things that might affect finance might not affect someone on the clerk side using Munis.	PPred	1
				So that was the first thing that I noticed and it opened up a plethora of information that I was not privy to and it gave me a sense of understanding.	PInfo	1

Trans#	Title	Techie?	Question	Statement	Code	-1/0/1
			Did you feel that it was too much information?	No. I did not think that it was too much. Again, before Senseman, I would log on [to Munis] and the system would have been changed…the screens were changed. And now I am hunting through [Munis] because the way that it acted before, now it was not acting that way. But now I knew when things were going to change when the update was taking place, and when it was going live.	PInfo	1
			Did you find the fields about the other applications to be particularly useful?	I did not find it useful because I did not use it in that respect.	PInfo	-1
				But I am not a super user and I am not on the committee that meets that knows all this and makes these types of decisions. I am not one of the chiefs, I am one of the Indians and the Indian was lost when the change went through because the information was not handed down. Whereas if I as an Indian or a user that is now using senseman can keep up with that what is going on and educate myself.	PInfo	1
			What about the feeds that was about what was going on in the clerk's office and in the environment? Did those feeds cause you to think about the updates ?	They just gave me an idea of who was going to be affected and if on any given day I needed something from another department I could be sensitive to the fact that they were going through an update.	PAnal	1

Trans#	Title	Techie?	Question	Statement	Code	-1/0/1
				I am not really concerned about what happens in those other departments but I am now more sensitive to what is going on during updates because I know what I went through when Munis was updated and the whole screen had changed.	PPred	0
			So would you say that your feelings on updates have changed?	Because I can now educate myself on how the office is being affected and how I as an individual can be more sympathetic to others that may be going through changes when I need something to perform my job and they might not be able to give it to me because of a change that is being made in the software. Because I know how the change affects me, so therefore although this particular change may not affect me I can understand how it affects others.	PAnal	1
			So is it safe to say that you are more aware of the cause and effect relationships involved with updates?	Correct.	PAnal	1
			What do you feel is the most beneficial part of the system as far as using it?	I would say that again, the going down below the chiefs to the Indian level…that they are allowed to use it.	PInfo	1

Trans#	Title	Techie?	Question	Statement	Code	-1/0/1
				And that those people because they had access to senseman they may be able to see something that someone on the committee had overlooked…	PPred	1
				The system gives people a venue to say "has anyone thought of this."	PAnal	1
5.	Supervising Deputy Clerk	No	What were your experiences using the system?	My experiences were limited to seeing what is out there on the system and to proof the information that is being displayed.	PInfo	0
			Is it that because of your position, and because you are involved with so many of the updates with your area, that the data feeds were not as useful to you?	They were not for the areas that I am involved in, but I did find them very interesting for the other applications such as New Vision and applications in other areas. It is nice to know what is going on in my sister departments.	PInfo	1
			Did that information about what is going on in your sister departments cause you to view your own updates in a different way?	I don't recall that it did for me, although I can see the potential that it could do that.	PPred	0

Trans#	Title	Techie?	Question	Statement	Code	-1/0/1
			What about the external feeds about things going on in the county or things going on in the clerk's office not specific to any application, were you able to make any connections there with regards to updates to your application?	Probably not for my application as far as the software application goes, but it was interesting to see what is going on in the county.	PPred	0
				That information is on their websites, but who has the time to go through and read it all? This seemed like it was a more condensed version of that information that was of interest to us [Clerk's Office] so I felt that was useful.	PInfo	1
			Did you have the opportunity to look at any of the actions and the reasoning that people were putting on the site.	I think the system gives the clerks an opportunity to ask the clerks [what they think] and I think that is good for the clerks to be more involved and aware. I liked seeing that was there…that we are looking at this and do you have any ideas.	PAnal	0

Trans#	Title	Techie?	Question	Statement	Code	-1/0/1
			Did the system make you view updates differently?	Mine was not because I am involved with many of the updates anyway, but I can see how it would be helpful to the users that were out there. If I were a user sitting out there not knowing what is going on, I would see it as being useful. But I did not experience that myself.	PInfo	0
			What about the crossover updates, such as eRecording. Do you see any potential for this type of system to help with those updates?	Oh yeah.	PInfo	1
			Many of the clerks are testers now or will be in the future. From what I understand when they test updates, they test their specific job function. Do you see any benefits in them opening up their perspective to consider how updates may affect functions other than the one they are testing?	… But I think that we may not have done a great job of telling the general population about a new release. Usually they get an email telling them not to log on to the system or letting them know that an update has happened, but if they are not involved with the testing this type of system could absolutely help with getting the word out that this is the update and it is supposed to affect these things and to keep an eye out for general functionality. Yes, we have tested it…but you never know when it gets into the real world what scenario might break it.	PAnal	1

Trans#	Title	Techie?	Question	Statement	Code	-1/0/1
			I think that maybe what I am getting at is do you see benefit in having them consider updates in more of a cause and effect manner with other functionality or areas? Do you think that senseman can help with that experience?	That I don't know.	PAnal	0
				Yes, I think that the awareness would be gained [from using the system].	PAnal	1
			From your perspective, do you think that general clerk's would be attuned to their updates enough to say that something in their environment would affect an update?	I believe that we have a pool of general clerks that are like that and would be able to determine things that would affect an update…that are forward thinking. We also have group that are in their tunnel and think only about putting this peg in this hole. We have about two hundred something in the office and 138 in courts and now many of those 138 are people that have grown up with computers and are capable of thinking more broadly. So I would say that, yes, they are able to.	PPred	0
			What do you think is the most beneficial part of the system?	I think the awareness that the system provides about not what only is going on in your world but in the worlds around you is important.	PAnal	1
				However, I am really liking the ability to provide information to everyone such as the newsletters from our vendors.	PInfo	1

Trans#	Title	Techie?	Question	Statement	Code	-1/0/1
			So you would say that more adequate information is being provided to the users through the system.	Yes, in a more concise manner so that we don't have to deal with emailing things to distribution lists and such.	PInfo	1
			What do you think is the least beneficial component of the system?	As interesting as it is, I would have to say the items from the county news.	PInfo	-1
				Is it interesting? Yes. But it doesn't really affect our actual operations on a daily basis, not yet. But that is not to say that something won't come up in the future that may.	PPred	0
6	Deputy Clerk	No	Tell me a little about your experiences with the system.	I used it to view information about the updates to CourtView and TrakMan	PInfo	0
			What did you think of the information?	I thought that it was good.	PInfo	1
				It was nice to see what else is going on in the Clerk's office and what the other areas are doing and how it relates to us.	PAnal	1

Trans#	Title	Techie?	Question	Statement	Code	-1/0/1
			Did is change the way that you viewed updates?	Well, mostly because I was able to see more that just the stuff that I was working on and how other stuff relates to us.	PAnal	1
				It made me realize that there are more updates than just the things that I am testing.	PVar	1
			What about the other feeds for the County and the news, what did you think of those?	They were OK. They really didn't make me think about the updates though.	PPred	0
			OK, what would you think is the most beneficial part of the system and the least beneficial part of the system?	Well, the part about what is going on with the programs is probably the most beneficial.	PInfo	1
				.		

APPENDIX E: THE SURVEY INSTRUMENT

Change Management Questionnaire

Instructions: For each statement below, please circle the answer that best expresses your feelings about the statement as it relates to updates to the CourtView application that you use as a part of your job function. Thank you very much for your participation!

		Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree	Not Applicable
		SD	D	N/A/D	A	SA	N/A
	START HERE	⇩	⇩	⇩	⇩	⇩	⇩
1.	I have access to enough information about CourtView updates.	1	2	3	4	5	N/A
2.	CourtView changes at a rapid pace.	1	2	3	4	5	N/A
3.	I know what will be changed once CourtView is updated.	1	2	3	4	5	N/A
4.	I can tell what situations outside my department might affect a CourtView update.	1	2	3	4	5	N/A
5.	I know the reasons that CourtView is being updated.	1	2	3	4	5	N/A
6.	I am aware of how updates affect my use of CourtView.	1	2	3	4	5	N/A
7.	The way I use CourtView changes often because of an update.	1	2	3	4	5	N/A
8.	The information that I have about CourtView updates helps me to understand them.	1	2	3	4	5	N/A
9.	CourtView is always being updated.	1	2	3	4	5	N/A
10.	There is sufficient information available about CourtView updates.	1	2	3	4	5	N/A
11.	I can tell how events in the office will affect a CourtView update.	1	2	3	4	5	N/A
12.	I can tell what events around the office might affect an update to CourtView.	1	2	3	4	5	N/A
13.	I can find information about CourtView updates.	1	2	3	4	5	N/A
14.	I am aware of the effects that outside events may have on a CourtView update.	1	2	3	4	5	N/A
15.	I know what each CourtView update is supposed to do.	1	2	3	4	5	N/A
16.	I can tell when CourtView has been updated.	1	2	3	4	5	N/A
17.	The information that I have about CourtView updates is helpful.	1	2	3	4	5	N/A
18.	I know when it might not be a good time to update CourtView.	1	2	3	4	5	N/A
19.	I know what to expect when CourtView is updated.	1	2	3	4	5	N/A
20.	CourtView updates often involve adding new functions.	1	2	3	4	5	N/A

Please Continue on the Back

CONTINUE HERE

Instructions: Please answer the questions below by drawing a circle around the appropriate response.

		0-1 years	1-3years	3-5years	5-6 years	6+ years
21.	How long have you have you used CourtView?	1	2	3	4	5

		Technical	Non-Technical
22.	Are you in a technical (IR) or a non-technical position?	1	2

		Yes	No
23.	Did you use the SenseMan system?	1	2

		1-2 times	3-5 times	5-7 times	7-10 times	10+ times
24.	If you answered yes to #23, how many times would you say that you logged on to the system?	1	2	3	4	5

The following is very important!!

Write your survey code HERE: CV

Please remember to write your code on this form in the space above and to remove the sheet of paper with your code on it and store it in a safe place
Please return all other materials back to the manila envelope and turn it in to one of the collection bins in your area

** Thank you for your time in completing this questionnaire. **

Please share any additional comments you have in the box provided below.

REFERENCES

Adams, L. A. and J. F. Courtney (2004). "Achieving relevance in is research via the DAGS framework." System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on: 257-266.

Antoft, R. and H. H. Salomonsen (2007). Studying Organizations by a Pragmatic Research Design: the case of qualitative case study designs. Beyond Waltz - Dances of Individuals and Organizations, Vienna, Austria.

Ashmos, D. P. and M. L. Nathan (2002). "Team Sense-Making: A Mental Model For Navigating Uncharted Territories." Journal of Managerial Issues **14**(2): 198.

Asif, Z. and H. K. Klein (2007). THE IMPORTANCE OF DELIBERATIVE INQUIRY FOR IS RESEARCH. 2007 Americas Conference on Information Systems, Keystone, CO, USA.

Boyd, B. K. and J. Fulk (1996). "Executive Scanning and Perceived Uncertainty: A Multidimensional Model." Journal of Management **22**(1): 1.

Churchman, C. W. (1968). Challenge to Reason, McGraw-Hill.

Churchman, C. W. (1968). The Systems Approach, Dell.

Churchman, C. W. (1971). The Design of Inquiring Systems: Basic Concepts of Systems and Organization. New York, Basic Books.

Churchman, C. W. and B. G. Buchanan (1969). "On the Design of Inductive Systems: Some Philosophical Problems." The British Journal for the Philosophy of Science **20**(4): 311-323.

Courtney, J. F. (2001). "Decision making and knowledge management in inquiring organizations: toward a new decision-making paradigm for DSS." Decision Support Systems **31**(1): 17-38.

Courtney, J. F., D. Croasdell, et al. (1998). "Inquiring Organizations." Australian Journal of Information Systems **6**(1).

Dawes, J. (2008). "Do data characteristics change according to the number of scale points used?" International Journal of Market Research **50**(1): 61-77.

Dillman, D. A. (2007). Mail and Internet Surveys: The Tailored Design Method, Wiley.

Dube, L. and G. Pare (2003). "RIGOR IN INFORMATION SYSTEMS POSITIVIST CASE RESEARCH: CURRENT PRACTICES, TRENDS, AND RECOMMENDATIONS." MIS Quarterly **27**(4): 597-635.

Gioia, D. A. and K. Chittipeddi (1991). "Sensemaking and Sensegiving in Strategic Change Initiation." Strategic Management Journal **12**(6): 433-448.

Gioia, D. A. and J. B. Thomas (1996). "Identity, Image, and Issue Interpretation: Sensemaking During Strategic Change in Academia." Administrative Science Quarterly **41**(3): 370-403.

Gregg, D. G., U. R. Kulkarni, et al. (2001). "Understanding the Philosophical Underpinnings of Software Engineering Research in Information Systems." Information Systems Frontiers **3**(2): 169-183.

Guo, Z. and J. Sheffield (2006). Habermasian Inquiring System: Toward a General Framework for Knowledge Management Research. 39th Annual Hawaii International Conference on System Sciences (HICSS).

Hall, D. and D. B. Paradice (2005). "Philosophical foundations for a learning-oriented knowledge management system for decision support " 39 **3**: 445-461.

Herrscher, E. G. (1989). "Design of the modern inquiring system, Part VI, Hermann Hesse (1877-1962)." Systems Research **6**: 267-270.

Hesse-Biber, S. and P. Leavy (2006). The Practice of Qualitative Research. Thousand Oaks, Sage Publications.

Hevner, A. R., S. T. March, et al. (2004). "Design Science in Information Systems Research." MIS Quarterly **28**(1): 75-105.

Lamb, R. and R. Kling (2003). "RECONCEPTUALIZING USERS AS SOCIAL ACTORS IN INFORMATION SYSTEMS RESEARCH." MIS Quarterly **27**(2): 197-235.

Linden, L. P., J. R. Kuhn, et al. (2008). "CHURCHMAN'S INQUIRING SYSTEMS: KERNEL THEORIES FOR KNOWLEDGE MANAGEMENT." Communications of the AIS.

Maitlis, S. (2005). "THE SOCIAL PROCESSES OF ORGANIZATIONAL SENSEMAKING." The Academy of Management Journal **48**(1): 21-49.

Malhotra, Y. (1997). Knowledge Management in Inquiring Organziations. 3rd Americas Conference on Information Systems, Indianapolis, IN.

Malhotra, Y. (2001). "Expert systems for knowledge management: crossing the chasm between information processing and sense making." Expert Systems With Applications **20**(1): 7-16.

Malhotra, Y. (2002). "Why Knowledge Management Systems Fail? Enablers and Constraints of Knowledge Management in Human Enterprises." Handbook on Knowledge Management **1**: 577-599.

Malhotra, Y. (2004). Why Knowledge Management Systems Fail? Enablers and Constraints of Knowledge Management in Human Enterprises. Knowledge Management Lessons Learned: What Works and What Doesn't. M. E. D. K. T. K. Srikantaiah, Information Today Inc.**:** 87-112.

Marakas, G. M. (2006). Systems Analysis and Design, and active approach. New York, McGraw-Hill/Irwin.

Markus, M. L., A. Majchrzak, et al. (2002). "A design theory for systems that support emergent knowledge processes." MIS Quarterly **26**(3): 179-212.

McKillip, J. and K. Baldwin (1990). "Evaluation of an Std Education Media Campaign: A Control Construct Design." Eval Rev **14**(4): 331-346.

Mingers, J. (2001). "Combining IS Research Methods: Towards a Pluralist Methodology." Information Systems Research **12**(3): 240.

Nunamaker Jr, J. F. and M. Chen (1991). "Systems development in information systems research." System Sciences, 1990., Proceedings of the Twenty-Third Annual Hawaii International Conference on **3**.

Orlikowski, W. and D. Robey (1991). "Information Technology and the Structuring of Organizations." Information Systems Research **2**(2): 143-169.

Parrish Jr, J. L. and J. Courtney (Forthcoming in 2008). "Churchman's Inquirers as Design Templates for Knowledge Management Systems: An Object Oriented Approach." Communications of the ACM.

Parrish Jr, J. L. and J. F. Courtney (2007). A COLLABORATIVE MODEL FOR MAKING ELECTRONIC RECORD CONTROL PRACTICE DECISIONS. Americas Conference on Information Systems, Keystone, CO, AIS.

Parrish Jr, J. L. and J. F. Courtney (2007). "Electronic Records Management in Local Government Agencies: The Case of the Clerk of Courts Office in Lake County Florida." Information Systems Management **24**(3): 223-229.

Pavesi, P. E. and P. E. J. Pavesi (1989). "Design of the modern inquiring system, Part V, Etienne Bonnot de Condillac (1715-1780)." Systems Research **6**: 176-178.

Pavesi, P. E. and P. E. J. Pavesi (1991). "Design of the modern inquiring system, Part VII, Ramon Lull (1232-1315)." Systems Research **9**: 85-92.

Richardson, S. M., J. F. Courtney, et al. (2006). "Theoretical principles for knowledge management system design: Application to pediatric bipolar disorder star, open." Decision Support Systems **42**(3): 1321-1337.

Rittel, H. W. J. and M. Webber (1973). "Dilemmas in a general theory of planning." Policy Sciences **4**(2): 155-169.

Schneider, S. C. (1997). "Interpretation in Organizations: Sensemaking and Strategy." European Journal of Work and Organizational Psychology **6**: 93-101.

Shadish, W., T. Cook, et al. (2002). "Experimental and Quasi-Experimental Designs." Boston: Haoghton Mifflin Company.

Simon, H. A. (1977). The New Science of Management Decision, Prentice Hall PTR Upper Saddle River, NJ, USA.

Snell, J. L. (1988). "Design of the modern inquiring system, Part IV, Warren S. McCulloch (1898-1969)." Systems Research **5**: 359-361.

Swanson, E. B. and N. C. Ramiller (1997). "The Organizing Vision in Information Systems Innovation." Organization Science **8**(5): 458-474.

Systems, A. f. I. (2007, 2007). "The AIS LEO Award for Exceptional Lifetime Achievement in Information Systems " Retrieved September 26, 2007, from http://home.aisnet.org/displaycommon.cfm?an=1&subarticlenbr=151.

Thomas, J. B., S. M. Clark, et al. (1993). "Strategic Sensemaking and Organizational Performance: Linkages among Scanning, Interpretation, Action, and Outcomes." The Academy of Management Journal **36**(2): 239-270.

Trauth, E. M. and L. M. Jessup (2000). "Understanding Computer-Mediated Discussions: Positivist and Interpretive Analyses of Group Support System Use." MIS Quarterly **24**(1): 43-79.

UCLA: Academic Technology Services, S. C. G. (2007). "Repeated Measures Anova." Introduction to SAS, 2008, from http://www.ats.ucla.edu/STAT/sas/library/repeated_ut.htm.

van Gigch, J. P. (1988). "Design of the modern inquiring system, Part I, R. Descartes (1596-1650)." Systems Research **5**: 267-269.

van Gigch, J. P. (1988). "Design of the modern inquiring system, Part II, The contemporary computer." Systems Research **5**: 269-271.

van Gigch, J. P. (1988). "Design of the modern inquiring system, Part III, ALC Destutt de Tracy (1714-1836) and the idealists." Systems Research **5**: 357-359.

van Gigch, J. P. (1990). "Design of the modern inquiring system, Part VIII, Symbolism and Ernst Cassirer (1874-1945)." Systems Research **7**: 298-301.

van Gigch, J. P. (1993). "Design of the modern inquiring system, Part IX, Erwin Schrodinger (1887-1961) and the role of theory in discovery." Systems Research **10**: 81-85.

van Gigch, J. P. (1993). "Design of the modern inquiring system, Part X, The interplay of theory and experiment in science from Thales (600 BC) to Higgs (21st Century)." Systems Research **10**: 97-101.

van Gigch, J. P. (1994). "Design of the modern inquiring system, Part XI, Richard Feynman (1918-1988); The changing role of explanation in modern science." Systems Research **10**: 73-78.

Walls, J. G., G. R. Widmeyer, et al. (1992). "Building an Information System Design Theory for Vigilant EIS." Information Systems Research **3**(1): 36-59.

Walls, J. G., G. R. Widmeyer, et al. (2004). "Assessing Information System Design Theory in Perspective: How Useful was our 1992 Initial Rendition?" Journal of Information Technology Theory and Application **6**(2): 43-58.

Weick, K. and K. Sutcliffe (2001). Managing the Unexpected: Assuring high performance in an age of complexity. San Francisco, Jossey-Bass.

Weick, K. E. (1993). "The Collapse of Sensemaking in Organizations: The Mann Gulch Disaster." Administrative Science Quarterly **38**(4).

Weick, K. E. (1995). Sensemaking in Organizations, Sage Publications.

Weick, K. E. (2001). Making Sense of the Organization, Blackwell Publishing.

Weick, K. E. (2004). "Vita Contemplativa: Mundane Poetics: Searching for Wisdom in Organization Studies." Organization Studies **25**(4): 653.

Weick, K. E. and D. K. Meader (1993). Sensemaking and Group Support Systems. Group Support Systems: New Perspectives. L. Jessup and J. Valacich. New York, Macmillian.

Weick, K. E. and D. K. Meader, Eds. (1993). Sensemaking and group support systems. Group Support Systems. New York, Macmillian.