NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

AN EXPERT SYSTEM FOR REWARD SYSTEMS DESIGN

by

Alper Erturk

September 2000

Thesis Co-Advisors: Erik Jansen
Mark E. Nissen

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY	**2. REPORT DATE** September 2000	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: An Expert System for Reward Systems Design		**5. FUNDING NUMBERS**	
6. AUTHOR(S) Alper Erturk			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		**8.PERFORMING ORGANIZATION EPORT NUMBER**	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		**10.SPONSORING/MONITORING AGENCY REPORT NUMBER**	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release, distribution is unlimited.		**12b. DISTRIBUTION CODE**	

13. ABSTRACT ***(maximum 200 words)***

Today's business environment is a highly competitive marketplace. In this competition, organizations distribute numerous rewards to motivate, attract and retain employees, such as pay, fringe benefits and promotions. However, not all managers have the necessary knowledge and expertise to effectively decide and structure reward systems.

This thesis presents an expert system to assist managers with designing the most appropriate reward system in their organizations. The system queries the user about the organization's goals, structure, culture, technology and its management's vision. This information is then filtered through decision matrixes in the knowledge base to generate the results along with an explanation and an estimated accuracy factor. The system was designed and programmed using Microsoft Visual Basic 6.0. The decision tables in the knowledge base were designed and structured using a Microsoft Access database.

The results show that similar knowledge base expert systems could be designed and programmed to assist managers for other purposes in organizations.

14. SUBJECT TERMS Artificial Intelligence, Expert Systems, Visual Basic, Organizations, Reward Systems			**15. NUMBER OF PAGES** 116
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

AN EXPERT SYSTEM FOR REWARD SYSTEMS DESIGN

Alper Erturk
Lieutenant Junior Grade, Turkish Navy
B.S., Turkish Naval Academy, 1994

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

and

MASTER OF SCIENCE IN SYSTEMS MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September 2000

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Today's business environment is a highly competitive marketplace. In this competition, organizations distribute numerous rewards to motivate, attract and retain employees, such as pay, fringe benefits and promotions. However, not all managers have the necessary knowledge and expertise to effectively decide and structure reward systems.

This thesis presents an expert system to assist managers with designing the most appropriate reward system in their organizations. The system queries the user about the organization's goals, structure, culture, technology and its management's vision. This information is then filtered through decision matrixes in the knowledge base to generate the results along with an explanation and an estimated accuracy factor. The system was designed and programmed using Microsoft Visual Basic 6.0. The decision tables in the knowledge base were designed and structured using a Microsoft Access database.

The results show that similar knowledge base expert systems could be designed and programmed to assist managers for other purposes in organizations.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

There are many people who have contributed to this thesis by providing technical and moral support. Without their support, this thesis would not have been possible. Because I might forget some of them, I would like to thank all those who have contributed in some way or another.

First of all, I would like to thank Prof. Erik Jansen, who has spent a lot of time helping me find related literature, structure necessary questions and implement the model, and for the conversations on the subject. He was the person who provided me with his sound advice and insight throughout this thesis.

Also, I would like to thank Prof. Mark E. Nissen for his technical support and recommendations.

I want to thank my peer and dear friend Alper Sinav, for his valuable ideas and assistance.

I am also grateful to my wife, Kaniye Yelda Erturk, who has encouraged me throughout my thesis study. I really appreciate all of her support, inspiration and perpetual patience during my studies over the past two years.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

This chapter presents the purpose of this thesis study, research questions, general scope, methodology and expected benefits.

A. PURPOSE

The purpose of this research study is to design and create a knowledge-based, management tool to recommend the most effective and efficient reward system for an organization. This tool could be used to analyze the effectiveness and efficiency of the current reward system as well as other possible systems. The thesis examines different kinds of reward systems for different types of organizations, reviews selected, relevant literature on Artificial Intelligence and Expert Systems, designs and builds a knowledge-based tool using Microsoft Visual Basic 6.0, and implements the application program.

B. RESEARCH QUESTIONS

The research questions addressed by this study are as follows:

- What kinds of reward systems are likely to be the most efficient and effective given the organization's goals, structure, culture, technology and management's vision?
- What are Artificial Intelligence, knowledge-based systems, expert systems, their applications, and their advantages and disadvantages?
- What would be the appropriate interview questions to reveal the most efficient and effective reward systems for an organization?
- How is it possible to design, create and implement a knowledge-based tool to assist managers in deciding the most appropriate reward system?

- How can the efficiency of such a system be measured?
- How can the results from this study be generalized?

C. SCOPE OF THE THESIS

Organizations distribute numerous rewards to motivate, attract and retain employees, such as pay, fringe benefits, and promotions. Reward systems are closely related to the quality of the work life and organizational effectiveness. Effective reward systems keep employees in the organization, meet the basic needs of employees, and are fair and equitable. Effective reward systems also fit the organization's culture, structure, goals, technology, and the vision of its leaders. However, the characteristics of reward systems are diverse, and not all managers possess the expertise necessary to effectively structure reward systems.

Knowledge-based systems and management support systems are important enablers of the information revolution, which intensely affects organizations. Although they are new systems in industry, they are widely used. In today's fast-changing business environment, this increased usage of knowledge-based systems has a significant impact on the speed and accuracy of decisions made by managers. Today, many highly skilled professionals, who have developed abilities through years of experience, broadly use these kind of systems to support their ideas and make faster decisions. These systems also are used to distribute power and responsibilities among managers, provide information for high-level decision making, and automate routine decisions in the decision-making process. Moreover, knowledge-based systems can be used to capture and distribute some expertise in the organizations. (Turban and Aronson, 1998)

This thesis designs and creates a knowledge-based management tool that asks several different questions about the organization's goals, structure, culture, technology and management's vision, and recommends the most effective and efficient reward systems according to the user's answers to the system. This can be used to assist managers in selecting the best reward system to implement according to the organization's properties and management's vision.

Unlike current commercial systems used for organizational diagnosis and design (Burton and Obel, 1998), this knowledge-based tool only focuses on the reward systems of organizations. This tool may reduce decision time and increase the accuracy and consistency of decisions regarding reward system design.

D. METHODOLOGY

The research methodology includes review and examination of different reward systems used in different types of organizations. Next, questions and heuristics were constructed based on the literature review. A requirement analysis was then used to determine the expected functionality of the knowledge-based application program, followed by a conceptual design and plan, and a small prototype. I used Microsoft Access for the knowledge base, and Visual Basic 6.0 as the shell for developing the rules, coding and graphical user interface. Upon completion of the prototype, I tested and evaluated the use of the knowledge-based tool from a technical point of view.

E. EXPECTED BENEFITS OF THIS THESIS

This thesis provides a knowledge-based application program to assist managers in analyzing their current reward system and deciding what reward system types are most appropriate for their organizations. Furthermore, this program also provides information for high-level decision making and helps managers make faster, more accurate decisions.

II. BACKGROUND

A. REWARD SYSTEMS

In today's fast changing business environment, organizations frequently reposition themselves to be able to compete. They seek new methods of organizing work and motivating employees, of satisfying the interests of employees and stakeholders, and providing greater workforce ability and flexibility. In the U.S., a growing number of companies are adopting new systems for rewarding employees. Profit sharing, employee-share ownership schemes and gainsharing are some examples of these new systems, which are becoming the standard in an increasing number of companies and organizations doing business in the U.S.

Today's competitive business environment forces companies to attract and retain talent in order to effectively compete. In tight labor markets, organizations compete against each other to obtain the talent they need. They may pay tremendous salaries, offer stock options and bonus and expensive benefit packages, and give non-monetary incentives. Consequently, organizations should see their employees as assets and talents rather than as the cost of doing business.

Reward systems are essential in order to motivate, attract and retain employees. Yet, what is a reward system? There are several definitions. One is:

> Something that increases the frequency of an employee action is a reward. Whether something is a reward or not depends *entirely* on its effect on employee behavior. If an employee's performance is followed by something, *and* the performance happens more frequently in the future, that something is a reward. If the performance happens as frequently or less often, that something is *not* a reward. Rewards increase the chances that a performance will be repeated. (Zigon, 2000)

Another broad definition of reward systems is:

> The reward system comprises the related set of structures and processes through which actions are directed and motivated to achieve individual and collaborative action outcomes. The set of processes comprise goal setting, measuring and assessing action outcomes, judging contributions (i.e.discrimination), distributing rewards, and giving feedback. (Jansen, 1986)

Companies should implement reward systems that are positively reinforcing to be able to make the employees change and improve their performance. Therefore, reward systems include both financial and non-financial issues, and also include strategies, plans and processes to implement and maintain the reward systems.

For these reasons, pay is one of the most important elements that make an organization competitive. It is very important to design effective compensation systems that will provide improvements in output, create positive changes in employees' behavior, and provide a good and effective communication inside the organization. These improvements will also help employees focus on key issues, such as the quality of the products and services, efficiency and customer satisfaction.

A new reward system implementation also requires managing change in an organization. Nevertheless, reactions to these changes can be initially positive or

negative. Therefore, it is important to understand and follow carefully what is happening and what the reactions are at each step of the change process. A good design and good communication are two important factors for the implementation of an effective reward system.

1. Learning Organizations

Senge defined learning organizations as "Organizations where people continually expand their capacity to create the results they truly desire, where new and expansive patterns of thinking are nurtured, where collective aspiration is set free, and where people are continually learning how to learn together." (Senge, 1990)

As stated before, the change process of implementing a new reward system in the organization is very important. Likewise, becoming a learning organization is as important and vital as the change process and is a requirement to survive in a dynamic and competitive environment.

Reward systems and measurement systems for effectiveness and efficiency are two important elements of learning organizations. Both expectancy theory and learning theory emphasize that people tend to demonstrate behaviors or do things for which they are rewarded. However, some organizations use reward systems that do not fit their strategic goals and current structure. Therefore, it is hard to build a competitive learning organization for them. They should realize that their old and inefficient reward system is ineffective, and should replace them with the right mix of rewards.

2. Personal Satisfaction From Reward Systems

A lot of research has been done about what kinds of rewards satisfy employees. Also, one research study shows that individual satisfaction is a complex reaction that can change from one person to another. (Lawler, 1975) The most important conclusions of this research are summarized in the following paragraphs.

First, reward satisfaction is related to what is expected and what is received. People feel dissatisfied when they receive less than expected, and they feel uncomfortable when they receive more than expected. The rewards they expect are those they believe they should receive. Uncomfortable feelings from receiving more than expected is easily rationalized by individuals. However, the feeling of receiving less than expected generates dissatisfaction that can be typically eased only by offering a promotion or a more valuable reward.

Second, people usually feel satisfaction or dissatisfaction by comparing what they receive with what other people receive. These comparisons are made by referring to others, who are both inside and outside the organization. Moreover, people develop their ideas of what they should receive from these comparisons. Although people generally think about several inputs, such as training, seniority and education, they usually think that the areas they excel in should count more heavily than the others.

Third, employees' satisfaction with their extrinsic and intrinsic rewards affects their overall job satisfaction. Employees who receive both extrinsic and intrinsic rewards would feel the most satisfied.

Fourth, different rewards vary in importance to different employees. People differ broadly in what is more important to them. Also, the amount of the reward an employee receives strongly affects the importance of that reward. However, most of the things that affect the importance of rewards are beyond the control of the organization, such as economic welfare, education and family background.

Finally, most extrinsic rewards are important because they lead to other important rewards. For instance, money is important, because it leads to other important and valuable things, such as food and status. Therefore, even when conditions change, extrinsic rewards remain important.

3. Necessary Reward System Properties

Organizational rewards are key factors in motivating employees. In order to increase the quality of the work life and organizational effectiveness, reward systems have to contain some important properties (Hachman and Suttle, 1977). Properties necessary to provide the quality of the work life are as follows:

- Reward Availability: Reward systems should provide enough available rewards to be able to satisfy the employees' basic needs.
- External Equity: Level of rewards in the organization should be high enough to be compared with those in other organizations.
- Internal Equity: Available rewards in the organization should be distributed equally among the employees.
- Individuality: Reward systems in the organization should deal with people as individuals, which means awarding them with rewards that they think they desire.

The following four properties are necessary to provide organizational effectiveness.

- Membership: Rewards should provide external equity, high overall job satisfaction and a higher reward level for better performance.
- Absenteeism: Important rewards should create a sense of willingness to go to work. This is also related to high job satisfaction.
- Performance Motivation: Important rewards in the organization should be perceived as relating to the performance of the employees. Higher level rewards should be received by the employees who show better performance.
- Organization Structure: Reward systems implemented in the organization should be fit into the organization's structure and management style. This property will be analyzed more extensively in the congruence theory of reward systems.

4. Congruence or Fit Theory of Reward Systems

Congruence or fit can be defined as consistency among different areas or organizational components. Thus, congruence is a measure of how well pairs of organizational components fit together (Nadler and Tushman, 1980). In this thesis, I define a framework for organizing more specific variables. This framework includes the organization's goal, structure, technology, culture, and management's vision.

The measure of congruence between the reward system and the components in this framework shows the degree to which reward systems are consistent with the organizational goals, structure, technology, culture and management's vision.

In the organizations, reward systems have a relatively high or low degree of congruence with each one of those components. According to the basic definition of congruence, the greater degree of congruence between the reward systems and

components of the organizational framework, the more effective and more efficient the reward systems will be.

It is also important to know why these components are important and what the fit between these components and reward system mean.

a. Organization's Goals

The goals of an organization are defined as "...negotiated or assigned targets or standards that serve to constrain actions and action outcomes within a given time frame" (Jansen, 1986). Moreover, organizational strategy is defined by Bryson as "A pattern of purposes, policies, programs, actions, decisions or resource allocations that define what an organization is, what it does, and why it does it" (Bryson, 1995).

The goals of an organization are the most important components in selecting strategic objectives. They are also important in determining more specific variables, about which the organization has to collect the data. Furthermore, strategy and goals of an organization determine the outputs, products, markets and customers.

Therefore, the goals of the reward systems should be consistent with the goals of the organization. Additionally, reward systems should be adequate to meet the demands of these goals, should be able to help improve the outputs and products. Thus, one purpose of reward systems is goal congruence.

b. Organization's Structure

Organizational structure includes the choices that are division of labor, configuration, distribution of power or organizational chain of command and

departmentalization. Division of labor determines how the labor process is divided, and includes the horizontal and vertical division of labor; configuration determines how the managerial work is divided, such as hierarchical configuration; distribution of power determines the chain of command in the organization, such as centralized or decentralized; and finally departmentalization determines the groupings of departments by their purpose, resources and processes. (Galbraith, 1977)

The reward systems implemented in the organization should be able to improve the organization's performance in the current structure. Also, the structure of reward systems should be consistent with the structure of the organization.

c. Organization's Technology

Technology in the organization includes the computer technology used in the management and decision process, such as computers, simulation tools and network systems. It also includes the technology used in production processes, such as integrated computer systems and required hardware and machinery. (Von Glinow, 1988)

Since technology affects the availability of reward systems, reward systems should be able to provide the requirements of high technology in the organization. Furthermore, because some technologies allow organizations to decrease the number of employees and human work force, reward systems should be able to cover these kinds of effects.

d. Organization's Culture

The culture of an organization is "...the shared and relatively enduring pattern of basic values, beliefs and assumptions in an organization" (Sethia and Von Glinow, 1985). Organizational culture is created by managerial styles, philosophies, systems and procedures. Four basic types of cultures used in this thesis are Apathetic culture that is risk averse, Integrative culture that values people and attempts to challenge, Caring culture that is people-oriented, and Exacting culture that is performance and success driven. (Von Glinow, 1988)

Reward systems should support and provide for the needs of the organizational culture. Also, reward systems can be used to create and develop a desired culture.

e. Vision of Management

Vision is a description of what the organization would look like and includes the organization's missions, values, basic strategies and philosophy (Bryson, 1995). Developing a clear vision is also important as the first step in leading to change and is critical in ensuring that the organization is moving towards the desired direction (Conger, Spreitzer and Lawler, 1999).

Management's vision is also very important for organizations because it affects what the organization can achieve. Management's vision also determines the values, mission statements and strategic direction of the organization.

Reward systems are tools to help managers achieve the requirements of these values and mission statements. Furthermore, reward systems help managers accomplish the organization's strategic goals.

In this thesis, the knowledge-based tool I design tries to find the reward system that best fits those organizational decision variables by asking several questions related to each area. The program gives six choices, which include an "I don't know" option, and recommends the best reward system for a particular organization. The questions that the program asks the user are presented in Appendix A, and the decision table that shows the relations between the questions and the reward system types is presented in Appendix B. This table is created from the literature review.

5. Reward System Types

The reward systems included in this program cover a broad area of rewards. They are selected among the new systems that have been implemented recently by big corporations in the U.S. and Europe. Most of them actually have different types of compensation systems. (Lawler III, 1975) They are:

a. Gainsharing

Gainsharing focuses on sharing gains produced by productivity and performance. In gainsharing, employees share financial gains as a result of improved performance. Unlike profit sharing, employees share the gains in areas that are directly under their control The main goal of gainsharing is to increase performance and

productivity by allowing employees who produce the outputs to share in the benefits of their innovations.

There are several different types of gansharing. Some examples of these types are the Scanlon Plan that is based on total sales and labor costs, the Rucker Plan that is based on labor as a percentage of sales less the cost of bought-in goods and services, the Improshare Plan that is based on labor productivity measured on the basis of work study standards, and the Value Added Plan that is based on the calculation of sales less the cost of bought-in goods and services. (Lawler III, 1975)

b. Employee-Share Ownership Schemes

Employee-share ownership schemes are tax efficient and cost effective reward systems that increase participation, commitment and efficiency of the employees by allowing them to buy shares of the company and to be partners.

There are several different types of ownership schemes. Some examples of these schemes are Approved Profit Sharing Schemes that offer employees the shares of the company under certain conditions and let them become a partner in the company, Employee Share Ownership Plans (ESOPs) that produce a structured arrangement, which allows all employees to have actual shares in the company, and Save As You Earn (SAYE) that allows employees to commit to saving a fixed amount of after-tax income. (Lawler III, 1975)

c. Skill-Based Plan

A skill-based plan provides employees a direct link between their pay and skills they can learn and use efficiently. In other words, the more employees learn and use a wide range of skills; the more they can earn. A skill-based plan is a type of people based reward system rather than a job based reward system. It focuses on the employees' ability to apply a wider range or higher level of skills to different jobs or tasks. The main goal of a skill-based pay plan is to improve the efficiency, performance and competitiveness by increasing employee effectiveness and enhancing the efficiency of work arrangements. Skill-based plans are best for organizations that have good communication, feedback and a high range of tasks. Since a skill-based pay plan is a people based system, it encourages learning and creative input from all employees. (Lawler III, 1975)

d. Competency Models

Competence can be described as the ability to meet performance expectations in a role and produce the required results. It focuses on the knowledge and skills, performance delivery, behavior and attributes which employees bring to the job. Competency based plans pay employees for their increasing organizational competence, current and future contribution potential. It is different from other reward systems because its focus is actually on the improvement of the employees to increase organizational competence and future contribution. By using competences as factors,

organizations can promote role flexibility, continuous development and support organizational change. (Lawler III, 1975)

e. Cafeteria Packages

Cafeteria packages can be described as the benefit packages awarded to employees that do not include salaries and direct payments. These non-monetary benefits include extra leave, more emphasis on health insurance, childcare vouchers and travel concessions. Generally, employees have the freedom to select the benefits they want in a cafeteria package system. This system also gives employees the chance to select the benefits, which they currently need, or consider valuable. Thus, organizations avoid offering benefits which employees do not value. Cafeteria package systems fit very well with a culture of empowerment. (Lawler III, 1975)

f. Annual Hours

The annual hours payment scheme is a different type of contract between the organization and its employees. It is based on working an agreed number of hours in the planning period ahead. In this scheme, instead of working basic and overtime hours, employees agree on the total number of hours they will work for the next period in advance. Additionally, this contract often includes payment for an agreed number of "reserved hours", which will be worked in exceptional situations. It provides more predictable budget planning and eliminates overtime costs. The main goal of the annual hours payment system is to achieve gains through improving utilization by having people

at work only when there is work to do and having the right number of people at work at any given time. (Lawler III, 1975)

g. Basic Payment System

The basic payment system or base pay is based on the formal job descriptions in the organization. Base pay is a secure and stable income that often functions to motivate employees to show up for work and do just enough to get by. Most basic payment systems are designed to distribute payments according to employees' seniority and job responsibilities. Furthermore, base pay should reflect the values of the organization and recognize the particular role of each employee in the organization. (Lawler III, 1975)

h. Team-Based Payment System

Team based pay schemes provide monetary rewards to employees who work within formally established teams. Team based payments are given based on team performance. These payments can be shared equally among the team members or a distribution of payment can vary among employees. Organizations that use less hierarchical structures use team based payment systems more effectively.

Team rewards depend on both task completion or results and building relationships in the team structure. Therefore, teamworking is often linked to the requirement of employees to be more flexible in the tasks they perform. (Lawler III, 1975)

B. ARTIFICIAL INTELLIGENCE (AI) AND EXPERT SYSTEMS (ES)

1. Basic Definitions of Artificial Intelligence

Artificial Intelligence is a term that has many different definitions. One of the best definitions is, "AI is behavior by a machine that, if performed by a human being, would be called intelligent" (Turban and Aronson, 1998). A different definition provided by Rich and Knight is, "AI is the study of how to make computers do things at which at the moment, people are better" (Rich and Knight, 1991).

AI deals with two main concepts. First, it includes studies and research on the thought process of humans in order to understand what intelligence is. Second, it studies how to represent these ideas by using machines, such as computers and robots.

Furthermore, AI can be viewed from many different perspectives. For instance, from the intelligence point of view, it is making machines appear intelligent; from the business viewpoint, it includes tools to solve business problems; from the entrepreneurial viewpoint, it is making machines more useful and effective; and from the programming perspective, it includes symbolic programming to solve problems. (Brown and O'Leary, 1995)

An interesting test has been designed by Alan Turing and is called the "Turing Test." This test has been used to determine if a computer demonstrates intelligent behavior. According to the Turing Test, a computer can be considered as intelligent when a human interacting with an unseen human and an unseen computer cannot determine which is which.

2. Knowledge in Artificial Intelligence

Data, information and knowledge can be classified by their degrees of abstraction and quantity (Figure 2.1). Knowledge has the highest degree of abstraction and least degree of quantity. (Turban and Aronson, 1998)

Data is just a meaningless point in space and time. It is like a letter or a word out of context. Since it is out of context, it does not have a meaningful relation to anything else. Hence, a collection of data is not information without any relation between the pieces of data. Information is an understanding of the relations between the pieces of data, or between the pieces of data and other information. It totally depends on the context. Information relates to description, definition, or perspective and provides answers for questions such as what, who, when and where. Beyond relations, there is pattern. Pattern is more than a relation of relations. Pattern represents both consistency and completeness of relations. To an extent, it creates its own context. When one realizes and understands the patterns and their insinuations, information becomes knowledge. Knowledge has completeness, which information does not contain. Knowledge includes strategy, practice, method, or approach. It provides answers for questions such as how. Then, wisdom arises when one understands the fundamental principles of patterns. Wisdom embodies principles, insight, moral, or archetype. It provides answers for questions such as why. (Bellinger, 1997)

Even though machines and computers cannot have experience and cannot learn as people, they can use the knowledge given to them by people. This knowledge can include theories, concepts, heuristics, procedures, methods and relations. The collection of this

knowledge used in an AI system is called "Knowledge Base." Knowledge bases are usually focused on some specific and narrow subject areas. (Turban and Aronson, 1998)

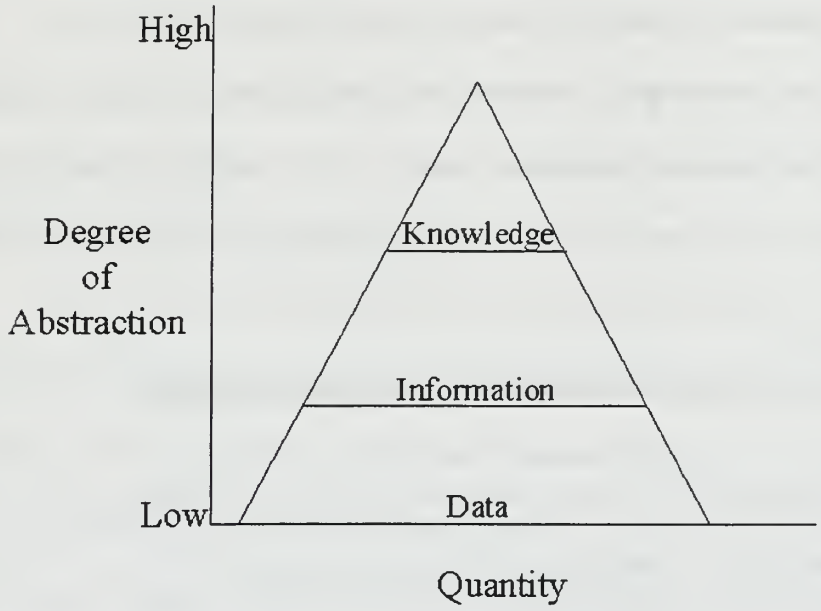

Figure 2.1. Abstraction and quantity of Data, Information and Knowledge. (Turban and Aronson, 1998)

3. Differences Between AI and Conventional Computing

Conventional computer programs are programmed based on predefined, step-by-step procedures that are called algorithms. These algorithms usually use numbers and solve problems by following a sequential problem solving procedure that consists of mathematical formulas.

On the other hand, AI systems use symbolic manipulation in processes. They use the techniques of searching and pattern matching. The AI software searches the knowledge base and looks for matchups and specific patterns, which satisfy the

conditions to solve the problem. Even though algorithms are not directly used in AI, they are sometimes used for pattern searching or the last step in solving problems.

In this thesis, I follow the same procedure. I provide a knowledge base in Microsoft Access database focused on reward systems, which is created by a broad literature review. Then, the program searches this knowledge base, finds the patterns, which the question matches up the appropriate reward system. Next, it solves the problem of finding the best reward system by using some computational algorithms at the final step.

4. Advantages and Disadvantages of Artificial Intelligence

According to Kaplan (1984), when compared to natural intelligence, AI has some advantages and disadvantages. Advantages of AI can be stated as:

- AI is more permanent as long as computer systems and programs remain unchanged
- AI can be cheaper than natural intelligence in most cases
- AI is consistent and thorough
- AI can be documented when needed
- AI can do some tasks much faster than a human can

Disadvantages of AI can be stated as:

- Natural intelligence is creative, but AI depends on the knowledge constructed to the system by people
- Natural intelligence benefits from sensory experience directly, while most AI programs work with symbolic inputs

- Humans can use a wide context of experience, and bring that experience to the solution of the problem by focusing it down

5. Expert Systems

There are a lot of different types of AI. Examples of these are Expert Systems, Neural Networks, Genetic Algorithms, Automatic Programming, Robotics, Fuzzy Logic, Game Playing, Intelligent Tutor, Natural Language Processing, Speech Understanding, Machine Learning and Computer Vision.

Expert systems are currently the most common types of AI. They are actually computerized advisory systems that imitate the reasoning processes and knowledge of experts to solve the problems. (Turban and Aronson, 1998)

Knowledge is a major resource for everyone and usually only a few experts have it. However, those experts may not be available when they are needed. Thus, expert systems can provide the expertise needed, regardless of where the human expert may be located. Nevertheless, expert systems are not designed to replace those experts, but to make their knowledge and expertise widely available whenever needed.

Human experts solve problems quickly and fairly accurately. They can also explain how they think, what they do to reach that solution, and they can judge the reliability of their judgments. They can communicate with other experts, change their view, and use tools, mathematical models and rules of thumb to support their ideas.

A list of basic concepts of expert systems follows (Turban and Aronson, 1998):

- Expertise: Expertise is a comprehensive, task-specific and focused knowledge obtained from experience, training and reading

- Experts: Although the definition of experts is based on the degree or level of the expertise they have, experts can be defined as human beings, which have developed a high-level proficiency in making judgments on a specific narrow domain

- Transferring Expertise: This means to transfer expertise from one person to another or from one computer to another. This transfer involves a four-step process, which includes knowledge acquisition from sources, knowledge representation, knowledge inferencing and transfer of knowledge to the user

- Inferencing: Inferencing can be defined as the process of making a conclusion from given evidence by reasoning. In other words, it is the ability of expert systems to reason

- Explanation Capability: Explanation capability is the ability of an expert system to explain how it reaches its recommendations or solution

6. Types of Reasoning

Computer models of expert systems are computerized based on the different models of human reasoning. The most important and most common reasoning models are:

a. Rule-Based Reasoning

Rule-based reasoning is the most common type of model used in expert systems. The rules are in the form of "IF Condition, THEN Action." The action part of the rules can include actions that affect the result, testing another rule, or adding a new fact to the database. Most of the rules are designed from the heuristics and rules of thumb. They can also be specific rules, such as laws. (Brown and O'Leary, 1995)

b. Case-Based Reasoning

Case based reasoning uses past experiences and cases, which include information about the situation, the solution, the results of using that solution and the key attributes that can be used for a quick search of similar patterns. If no appropriate similar prior case is found, then the human-created solution can be added to the case base and it allows the system to learn from them. (Turban and Aronson, 1998)

c. Neural Networks

Neural networks are the types of reasoning based on pattern recognition. They look for patterns and are able to recognize those patterns even if the data are ambiguous or distorted. Neural networks are organized into layers. They have the processing elements called "Neurons" and their relations with each other. Neural network computing uses procedures that are conceptually similar to those that function in biological systems. (Brown and O'Leary, 1995)

d. Frame-Based Systems

In frame-based systems, knowledge is represented in frames, which are a type of representation of the object oriented programming approach. (Brown and O'Leary, 1995)

e. Real-Time Systems

Expert systems usually have a limit on the response time to problems. Real-time systems are designed to decrease the response time and to produce the required response by the time it is needed. (Turban and Aronson, 1998)

f. Ready-Made Systems

Ready-made systems are the expert systems developed for the particular needs of the users and they can be purchased as a software package, such as project management in operations research. (Turban and Aronson, 1998)

7. Important Properties of Expert Systems

As stated before, expert systems are designed to provide and make the knowledge and expertise widely available. In order to be able to achieve this, expert systems have to have some important properties:

- Like human experts, expert systems should be able to explain why specific information is needed for the solution process. This is an important part of the system's explanation ability.
- Expert systems should be able to explain the steps of the process used to reach the solution. In other words, it should be able to explain how that recommendation is derived and be able to show the rules used.
- Expert systems should be able to work with fuzzy or incomplete information. As in the program of this thesis, choices, such as "I don't know" or "I don't have any idea", should be offered to the user to answer the question asked by the program. Therefore, even if the user has no idea what is being asked, the program should continue and reach a conclusion.
- Expert systems shouldn't give the "Duh..." answer to the user. They should be able to warn the user when a question was not answered or the user skips a step.

8. Benefits and Limitations of Expert Systems

As in most systems, expert systems have many benefits, and also some limitations (Brown and O'Leary, 1995). The important benefits of expert systems can be stated as:

- Since expert systems can work faster than human experts, they give better service and increase output and productivity
- They help in reducing the decision time
- They offer more process and product quality, not to mention flexibility in the production process
- They can transfer the knowledge to remote locations
- They make knowledge widely available and provide a broad usage of scarce expertise. They also provide easier accessibility to knowledge
- They eliminate the need for expensive equipment
- They have the ability to work with fuzzy and incomplete information
- They have the ability to collect and integrate several expertise opinions
- They provide improvement in decision making and problem solving
- They can provide expert training

The most important problems and limitations of expert systems can be stated as:

- Knowledge required for the expert systems might not be always available
- Sometimes expertise is very hard to extract from experts
- Expert systems work very well only in a narrow domain of knowledge
- The complicated vocabulary that experts use for expressing their judgments is often hard to understand by regular users

- The approach of each expert to the problem might be different, yet still correct
- Most experts have no independent means of checking whether their conclusions are reasonable
- Sometimes there is a lack of trust on the part of end users

C. GENERAL OVERVIEW OF VISUAL BASIC

Microsoft Visual Basic 6.0 is the latest and most advanced version of the old Basic language. It gives the user a complete Windows application development system in one package. Thus, Visual Basic is a Windows application itself. The user can load and execute the Visual Basic system just as other Windows applications. Furthermore, the user can use the running Visual Basic program to create other programs. Visual Basic lets the user write, edit, test and debug the Windows applications. Additionally, it has a lot of tools, such as ActiveX controls. Currently, there are four different editions of Visual Basic 6.0: Learning edition, Professional edition, Enterprise edition and Visual Basic 6.0 Working Model. The differences between the editions occur in functionality and the limitations imposed. However, all editions of Visual Basic require a Win 95/98/NT operating system on which to run. (Bradley and Millspaugh, 1999)

Visual Basic is actually an event-driven programming language, which contains some of the elements of an object-oriented programming language.

1. Object-Oriented Programming (OOP)

Newer Fourth Generation Programming Languages, such as C++ and Java, are true object-oriented programming languages. As stated before, Visual Basic also has

some elements of object-oriented programming. However, Visual Basic is eventually getting closer to becoming a true object oriented language with each release.

In developing an object oriented programming application, a programmer deals with the object model. The object model includes three important elements (Bradley and Millspaugh, 1999):

a. Objects

Objects are the Graphical User Interface (GUI) elements. In Visual Basic, examples of objects can be Labels, Text Boxes, Command Buttons, and List Boxes.

b. Attributes

Attributes of the object are the graphical or dimensional properties of the objects. In Visual Basic, examples of attributes for a label can be Caption, Height, Length, and Visibility.

c. Methods

Methods are the actions that an object performs in response to a GUI event. In Visual Basic, an example of a GUI event can be clicking a command button, and an example of a method of a command button object can be opening a form or displaying text.

2. Event-Driven Programming

Unlike the traditional procedure-driven programming models, the lines of the program in the event-driven programming model are not written and executed in a

sequential logic. When the event occurs, it causes the program to jump to the procedure you have written to handle that event, and execute it. In the event-driven programming, the user is in control rather than the program.

An event is a thing that happens or takes place. For instance, clicking the mouse generates a mouse event, and pressing a key generates a keystroke event. Furthermore, almost everything the user does generates an event. Events may be generated by GUI components or other external elements, such as a mouse or a keyboard. Additionally, a programmer can also define custom events.

As an illustration of an event-driven procedure, suppose that the event is a user clicking on a button. Also suppose that the action of clicking that button is a calculation of a variable. When the user clicks that button, the program jumps to the procedure you have programmed to calculate results. The full program is not executed in a linear fashion or a logic sequence, but only user-action event-triggered procedures are executed. If a particular event does not take place, the corresponding procedure is not going to be executed in the event-driven program.

3. Program Development in Visual Basic

In Visual Basic, the code development process is very important and easy to follow. Code development consists of two basic phases (Bradley and Millspaugh, 1999):

a. Planning (Design)

The planning or designing phase in Visual Basic includes four basic steps: designing a graphical user interface (GUI) and objects used in the GUI, planning the

properties of the objects, and finally planning, identifying and designing the associated methods.

b. Programming (Implementation)

The programming or implementation phase in Visual Basic includes three basic steps: defining and implementing the GUI using Visual Basic forms and controls, setting the planned properties, and finally writing the code for each method.

4. Visual Basic Environment

Visual Basic provides the programmer with a simple and easy-to-control GUI. In this GUI, the programmer controls the operations of the computer by moving a pointer and selecting icons. Since it is easy to create a user interface only by selecting and clicking on the icons, Visual Basic can be easily used in rapid application development. (Perry, 1997)

Visual Basic programs show a window type screen, which is called a "form". Command buttons, labels, text boxes, and these kinds of objects are found on those forms and are called "controls." The Visual Basic environment consists of eight basic elements for the implementation of programs (Figure 2.2). Those basic elements are:

- The Main Window which holds menu bar, tool bar, size and location information
- The Form Window which is used to design the form
- The Project Explorer Window, which contains project files
- The Properties Window which contains property names and values for the selected object

- The Form Layout Window which shows the position of the form on the desktop screen
- The Toolbox which contains the controls to select and add to the form
- The Toolbar, which contains shortcuts for frequently used operations
- Menu bar which contains all possible operations

5. Why Visual Basic As A Shell For Expert System

Although there are a lot of shells particularly designed for rule-based expert system implementation, such as CLIPS and JESS, I chose Visual Basic for the implementation of my application. There are several reasons for selecting Visual Basic as a shell for my knowledge-base expert system application. The important reasons are discussed below:

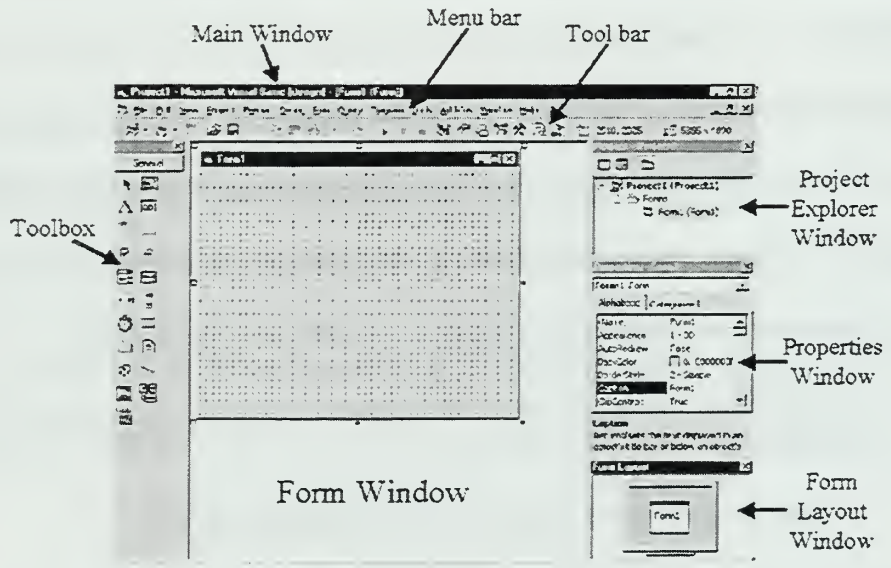

Figure 2.2. Visual Basic Environment.

First of all, it is easy to create a good graphical user interface with Visual Basic. Moreover, it is easy to add, remove, or modify the sizes, places and properties of the controls on the form according to the user's changing requirements.

Second, since Visual Basic has the ability to create, connect and work with a database, it is easy to link the program to the knowledge base that I created with a Microsoft Access database. Also, modifications on the knowledge base do not affect the code and implementation of the Visual Basic program.

Third, by using Visual Basic, it is easy to handle "IF Condition THEN Action" type of rules, no matter how many rules there are.

Fourth, because Visual Basic is an event-driven language, the program does not run sequentially. The user is in control. For instance, in my program, the user can decide which group of questions to answer first.

Fifth, if it is necessary to use a spreadsheet or graph, it is easy to link Visual Basic to spreadsheets, graphs or word editors. Also, it is easy to create reports by using its ability to link to other programs.

Sixth, since Visual Basic has ActiveX controls to extend the functionality of the program, and has the ability to include procedures from Dynamic Link Libraries (DLLs), it is easy to add a new functionality according to the user's requirements.

Seventh, because Visual Basic can also be used for rapid application development, it might be easier to develop a prototype in a relatively short time.

Eighth, one of the most important goals of expert systems is to make the knowledge widely available. Consequently, Visual Basic has the ability to be connected to the Internet and be used online by making some changes in the coding.

Finally, Visual Basic has the ability to be compiled and distributed as one project. Also, the user does not need the Visual Basic development environment to run the program. The program will install, set up and run easily on another computer. This ability also serves the "distribution of the knowledge to remote areas" goal of expert systems. Furthermore, because the program is compiled, the application will load and run faster, as well as be more secure.

III. METHODOLOGY

A. SYSTEM DEVELOPMENT LIFE CYCLE

Since an expert system is actually a kind of computer software, it is developed through a software development process. The main goal of a software development process is to increase the chance of developing high-performance, maintainable software on time and without exceeding budgetary limitations.

Many different models have been suggested by researchers. I chose the "System Development Life Cycle (SDLC)" model by Boehm (1981), which is a kind of waterfall model. In the adaptation of a system development life cycle to expert systems, the specific nature of expert systems determines which tasks should be completed, which order should be followed and in what depth those tasks should be performed.

Different tasks in developing an expert system are combined into six main phases parallel to SDLC. Nevertheless, it should be known that the development process is not a linear process. Some of the tasks in the phases can be performed at the same time and returning to previous tasks or even a previous phase is possible. Explanations and the tasks provided in each phase are as follows:

1. Project Initialization

Project Initialization is the first phase of the development cycle of expert systems. Tasks included in this phase are inter-related and can be performed in any order or simultaneously. The first task is to clearly define the problem that will make it easier to

understand and will help to produce an effective program. The second task is to assess and justify the need for the program. The third task is to evaluate the alternative solutions, which include the availability of experts, education and training, using packaged knowledge and conventional software. The fourth task is to verify the expert system approach that includes requirement analysis and justification of appropriateness. The fifth task is to consider the managerial and organizational issues that include resources, financing, user support, legal and other constraints. (Turban and Aronson, 1998)

2. System Analysis and Design

System Analysis and Design is the second phase of the development process of expert systems. Several tasks are completed in this phase. The first task is to complete the conceptual design. The conceptual design helps show the general capabilities of the program, interfaces and relations with other programs and systems. It also shows necessary resources and any other information and requirements for detailed design. After the completion of the conceptual design the second task is to determine the development strategy and methodology. There are different types of development strategies including developing the program oneself, hiring a developer or entering into a joint venture. Once the development strategy is selected, the development methodology, or how the system is to be built, has to be decided. The third task is to define the sources of knowledge. Defining sources of knowledge includes selecting the expert and finding other knowledge resources, such as books and web sites. The fourth task is to decide on the computing resources. This task includes deciding on the programming language,

program shell, support tools, construction aids and hardware support. Upon completion of the fourth task, the fifth task is to make a feasibility study including the study of financial, technical and operational feasibility. The ending milestone of this phase is the approved complete project plan. (Turban and Aronson, 1998)

3. Rapid Prototyping

Rapid prototyping is the third phase of the development process. The first task in this phase is to build a prototype. Then, the second task is to test the prototype. Once testing is completed, expert and user feedback is evaluated and analyzed. After evaluation, if any improvements and modifications are needed, the system is modified. The next task is to demonstrate the system and complete the design. (Turban and Aronson, 1998)

4. System Development

After the prototype is completed, the next phase is system development. The first task in this phase is to complete the knowledge base including defining the potential solutions and recommendations, defining the input facts, developing an outline, mapping the decision matrix, and completing the knowledge base. The second task is to evaluate and improve the knowledge base. The next task includes determining if the system does what it is supposed to do. This task also includes finding out if the knowledge base is built correctly. (Turban and Aronson, 1998)

5. Implementation

The implementation phase of an expert system can be a long, complex and hard process. The first task is to get the acceptance of the user. The acceptance of the user depends on the quality, capacity and ease of use. The second task is to determine the installation approaches. The next task is to demonstrate a fully working system. The following and last task includes selecting the mode of delivery, providing the security, preparing documentation, integration and field-testing. (Turban and Aronson, 1998)

6. Post Implementation

Post implementation is the final phase of the development process. It includes maintenance, required upgrading and operation. (Turban and Aronson, 1998)

B. DEVELOPMENT OF "REWARD CONSULTANT" EXPERT SYSTEM

The choice of developing a system to help the managerial decision making process in organizations was based on extensive literature review, finding an expert on reward systems and not having an expert system that deals with the reward systems in organizations. Thus, in this thesis, the managerial decision process was modeled within the specified parameters and variables, and it was evaluated using an expert system prototype.

Although I used a different conceptual design and programming language, my methodology of development of "Reward Consultant" expert system followed the logic of Burton and Obel's "Organizational Consultant" expert system (Burton and Obel, 1998). Burton and Obel proposed a model and developed an expert system for analyzing

the contrasts and fits between an organization's strategy and its structure. They comprise factors such as organizational size, complexity, formalization and centralization; then they determine the fit and contrast between organization's strategy and its structure based on those factors. Nevertheless, Burton and Obel's "Organizational Consultant" does not include reward systems and their effects on employees. Conversely, I proposed a model and developed an expert system for analyzing the reward system in an organization based on the organization's goals, structure, culture, technology and its management's vision.

Methodology and development of the "Reward Consultant" expert system paralleled the first four phases of system development life cycle outlined and described in the first section of this chapter. Subsequently, I explain the methodology of "Reward Consultant."

1. Problem Definition

First, a clear definition of the problem has been done in designing an expert system to help managers decide on the most effective and efficient reward system for their organizations given the goals, culture, technology, structure and management's vision of the organization. Many different systems could be used to create a system for reward selection. Nevertheless, my real intention was to build a system that follows a decision process similar to that an expert uses and makes a selection that an expert would make. An expert system was justified as needed in this area. There are some reasons for that decision. One of them was that the expert systems used in organizational purposes in the market do not include reward systems. Another reason was the necessity of making

decisions faster and more accurately. The third reason for that justification was the difficulty in finding an expert on reward systems when needed.

Then, I evaluated the availability of resources. Prof. Jansen, co-advisor of this thesis, is an expert on reward systems in organizations. I built the knowledge base based on the literature review that I did and the help of Prof. Jansen.

2. Selection of Reward Systems

After defining the problem, the next task was to select the reward systems that I used in this expert system. In today's business environment, organizations use numerous reward systems to attract and retain employees. Nevertheless, it would have been too difficult to handle all kinds of reward systems used in business in an expert system prototype. After a long literature review and consultation with an expert, the following reward system types were decided upon to be used in "Reward Consultant" expert system. These reward systems are Basic Payment, Skill-Based Payment, Competency Models, Cafeteria Packages, Annual Hours, Team Based Payment, Gainsharing and Employee Share Ownership Schemes. These reward systems are the most used systems by today's highly-competitive organizations. Although most of them are direct payment systems, some of them cover non-payment options as well.

3. Knowledge Acquisition

Knowledge acquisition is the central part of an expert system (Hart, 1986). An expert collects information on an organization, its service area, its products, its goals and missions. This information is used by the expert to have certain ideas about the

organization before the other areas are even researched. An experienced organizational consultant can have an idea about which reward system should be applied without even asking several questions about it.

The process of selecting and deciding on a reward system is complicated and difficult, especially because organizations have many different characteristics. In order to be able to simplify this complex managerial decision process, questions that address the specific organizational characteristics were arranged into five main groups: organizational goals, culture, structure, technology and its management's vision. The importance of these five groups is explained in Chapter II.

4. Construction of Question Sets

To generate the questions essential to extract the information about the organization from the user, an examination was done about the logical process that is used by experts to analyze an organization. The managerial decision process involves several steps. The first is the collection of information about the organization's characteristics and its management's vision. Then, the expert makes a logical comparison of this information from the user with his or her experience. This comparison helps experts in making the most appropriate decision and reduces the risk of error.

As explained in Section 3, those questions were put into five main groups that examined the organization's characteristics. Next decision tables that show the relationship between the questions and reward systems were constructed. These decision tables were adapted to the Microsoft Access database to construct the knowledge base. An example of these decision tables is shown in Table 3.1. As seen in Table 3.1, the

columns of the table include the number of the question, the text of the question, help title, help context and the type of eight reward systems. The Help title and help context were used to explain the unfamiliar terms in the question to the user. The relationship between the question and the reward systems is presented as ones and zeros. For example, in Table 3.1, question number 103 ("Rewarding team contribution.") has relations with competency models, gainsharing, employee share ownership schemes and team based reward systems, so ones appear in the corresponding columns.

Another table is constructed to sort the reward systems after the calculations based on the mathematical model constructed in Visual Basic are run. An example of this table is presented in Table 3.2. As seen in Table 3.2, the columns of this table include reward name, reward code, reward explanation and weighted results. After the calculations in Visual Basic, the value of each reward system's variable is recorded to weighted results field. Next, a query of this table in descending order is made in Access by using Visual Basic codes. According to this query, the best reward system with the highest value is selected as the most appropriate reward system.

Number	Question	RelatedArea	HelpT	HelpE	BasicPayment	SkillBased	Competency	Gainsharing	Cafeteria	AnnualHours	EmpShare	TeamBased
101	Implementing a	Goal			1	0	0	0	0	0	0	0
102	Rewarding und	Goal			0	1	1	1	0	0	1	0
103	Rewarding tear	Goal			0	0	1	1	0	0	1	1
104	Promoting emp	Goal			0	1	1	0	0	0	0	0
105	Flexibility in pr	Goal			0	1	1	0	0	0	0	1
106	Rewarding loya	Goal			1	0	0	1	0	0	1	0
107	Helping to crea	Goal			0	0	0	1	0	0	1	0
108	Focusing empl	Goal			0	0	0	1	0	0	0	0
109	Linking team ri	Goal			0	0	0	1	0	0	0	1
110	Supporting a c	Goal			0	0	0	1	0	0	1	0
111	Linking employ	Goal			0	0	0	1	0	0	0	0
112	Cost effectiven	Goal			0	0	0	0	0	0	1	0
113	Tax efficiency.	Goal			0	0	0	0	0	0	1	0
114	High commitm	Goal			0	0	0	0	0	0	1	0
115	Supporting an	Goal			0	0	1	0	0	0	0	0
116	Supporting role	Goal			0	0	1	0	0	0	0	0
117	Support differe	Goal			0	0	1	0	0	0	0	0
118	Creating a worl	Goal			0	0	0	0	0	1	0	0
119	Creating a worl	Goal			0	0	0	0	0	1	0	0
120	Constant unit p	Goal			0	0	0	0	0	1	0	0
121	Reducing overt	Goal			0	0	0	0	0	1	0	0
122	Reducing the r	Goal			0	0	0	0	0	1	0	0
123	More predictab	Goal			0	0	0	0	0	1	0	0
124	Reduce workin	Goal			0	0	0	0	0	1	0	0

Table 3.1. Example of Decision Tables in Microsoft Access.

RewardName	RewardCode	RewardExplanation	WeightedResults
Annual Hours	AH	Annual hours payment scheme is a different type	0
Basic Payment System	BP	Basic payment system or Base pay is based on tl	2.5
Cafeteria Packages or Status	CP	Cafeteria packages can be described as the benef	2.2
Competency Models	CM	Competence can be described as the ability to me	0
Employee Share Ownership S	ES	Employee-share ownership schemes are tax effici	2
Gainsharing	GS	Gainsharing focuses on sharing gains produced by	2
Skill Based Payment System	SB	Skill-based plan provides employees a direct link t	2
Team Based Payment Systen	TB	Team based pay schemes provide monetary rewar	2.5

Table 3.2. Example of Sort Table in Microsoft Access.

5. Heuristics

Once the questions had been structured, the next step was the development of heuristics. These heuristics were based on the broad literature review that I conducted, and were used to develop the decision tables explained in the previous section. The knowledge used for these heuristics is based on the general organizational and reward system characteristics. These rules were first developed with words. Below is an example of the list of the rules used:

- Reducing overtime costs is related to annual hours system
- Tax efficiency is related to employee share ownership schemes
- Integrative culture is related to skill based, gainsharing, employee share ownership and team based reward systems
- The adjustable payroll package is related to the cafeteria package system
- An effective communication system is related to skill based, competency models, gainsharing, employee share ownership and team based reward systems
- Seniority among employees is related to basic payment, competency models and cafeteria package systems

Next, these rules in words were used to generate decision tables. These decision tables were then used to generate heuristics in the IF-THEN format. After that, the mathematical model based on these heuristics was constructed. A more detailed explanation of this mathematical model is presented in Chapter IV.

6. Dealing with Uncertainty

Uncertainty needed to be considered when the questions and decision tables were constructed. Throughout the question process, there are some instances of uncertain data and uncertain rules. This uncertainty is due partly to the subjectivity of some questions and the lack of incomplete information in the answers provided by the user. For example, one of the questions related to the organization's technology asks, "What is the degree of high technology you use in your organization?" While there is a question of what high technology includes, the difference between high technology and standard technology could be small and exact observation and measurements could be needed.

In addition to subjective uncertainty about the questions, an uncertainty factor could occur according to the user's knowledge about the question. In order to decrease that kind of uncertainty, "I don't know" was added to the answers as an option. It means that the user has no idea about the question or is not sure of the answer.

In the program, an accuracy factor is calculated by dividing the total number of questions answered other than "I don't know" by the overall number of questions answered. This number is presented as a percentage in the result statistics. The calculation of the accuracy factor is explained in more detail in Chapter IV.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. ANALYSIS AND DESIGN

A. SYSTEM ANALYSIS AND DESIGN

1. Conceptual Design

This expert system relies on the answers provided by the user in order to be able to make a selection from a number of reward systems. Therefore, the first start is made by the user. Figure 4.1 is the data flow diagram of the "Reward Consultant" expert system and shows the conceptual design and how the system works.

According to this conceptual design, the user starts the consultation by answering the questions and providing the required information to the user interface. Choices for the answer for each question are provided by the system. Once the user answers a question, the answer is integrated with both the mathematical formulas based on heuristics and the reward system data read in from the knowledge base. Then, the value of each reward system variable is calculated by the program. This process is repeated for every question and answer. Upon completion of answering the questions, calculated data and observed data from the user are passed through the search engine. Next, the search engine compares the data with the read in reward systems data from the knowledge base, does a pattern matching, decides which reward system is most appropriate and shows the selected reward system as output in text or graphic format.

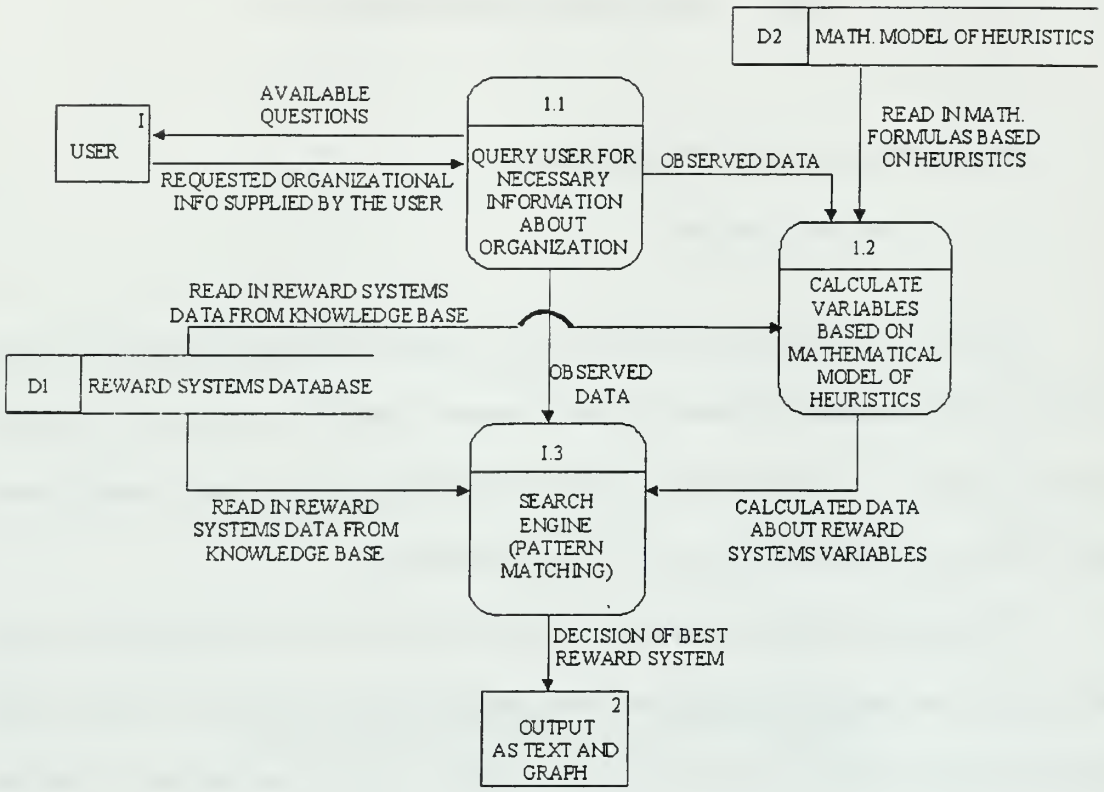

Figure 4.1. Data Flow Diagram of the "Reward Consultant" Expert System.

2. Construction of Heuristics and Mathematical Model

The model is designed to compare the average values of reward systems variables, and reveal the reward system that has the highest value as the most appropriate reward system for the organization.

For each question, five choices are given to the user, and each answer has a value to be added to the related reward system variable. The answers are "To a very great extent," "To a great extent," "To a normal extent," "To a little extent," "To a very little extent," and "I don't know." The values of these answers are five, four, three, two, one

and zero, respectively. Because each reward system may intersect with a different number of questions, an average value is calculated for each reward system type. Average value for a reward system type is calculated by dividing the total value of reward system variable by the total number of questions answered related to the same reward system type. At the end of the program, these average values are compared to each other and the reward system type with the highest average value is recommended as the most appropriate reward system type.

Table 4.1 is a downscaled example of a table from the knowledge base. As seen in this table, every question is related to one or more reward systems. Each one of those reward systems has a variable in the mathematical model. When the user makes a selection from the answers and clicks the button, the value of the selected choice is added to the variable of the related reward system for that question. For example, according to Table 4.1, when a user selects the choice "To a great extent" for question number 105, the variables of skill based pay (Skill), competency models (Competency) and team based pay (Team) are summed by four.

SkillTotal = SkillTotal + 4

CompetencyTotal = CompetencyTotal + 4

TeamTotal = TeamTotal + 4

After the questions are answered, the average values for all reward system types are calculated for comparison. Examples of this calculation are as follows:

SkillAverage = SkillTotal / SkillAnsweredQuestions

CompetencyAverage = CompetencyTotal / CompetencyAnsweredQuestions

Number	Question	RelatedArea	HelpTi	HelpE	BasicPay	SkillBased	Competency	Gainsharing	Cafeteria	Annual	EmpShare	TeamB
101	Implementi	Goal			1	0	0	0	0	0	0	0
102	Rewarding	Goal			0	1	1	1	0	0	1	0
103	Rewarding	Goal			0	0	1	1	0	0	1	1
104	Promoting	Goal			0	1	1	0	0	0	0	0
105	Flexibility ir	Goal			0	1	1	0	0	0	0	1
106	Rewarding	Goal			1	0	0	1	0	0	1	0
107	Helping to	Goal			0	0	0	1	0	0	1	0
108	Focusing e	Goal			0	0	0	1	0	0	0	0
109	Linking tea	Goal			0	0	0	1	0	0	0	1
110	Supporting	Goal			0	0	0	1	0	0	1	0
111	Linking em	Goal			0	0	0	1	0	0	0	0
112	Cost effect	Goal			0	0	0	0	0	0	1	0
113	Tax efficier	Goal			0	0	0	0	0	0	1	0
114	High comm	Goal			0	0	0	0	0	0	1	0
115	Supporting	Goal			0	0	1	0	0	0	0	0

Table 4.1. Downscaled Example Table of Reward Systems Knowledge Base.

The fields in the decision tables in the knowledge base are filled in according to the literature review. The heuristic rules allow the program to put those relations into a mathematical model. These rules were used in an IF-THEN format to generate the mathematical model. For example, the heuristic used in our example above is:

IF Question 105 is related to Basic Payment system

THEN Add the value of answer to Basic Payment variable

IF Question 105 is related to Skill Based Pay system

THEN Add the value of answer to Skill Based Pay variable

IF Question 105 is related to Competency Models

THEN Add the value of answer to Competency Model variable

IF Question 105 is related to Gainsharing system

THEN Add the value of answer to Gainsharing variable

IF Question 105 is related to Cafeteria Packages system

THEN Add the value of answer to Cafeteria Package variable

IF Question 105 is related to Annual Hours Payment system

THEN Add the value of answer to Annual Hours variable

IF Question 105 is related to Employee Share Ownership schemes

THEN Add the value of answer to Employee Share Ownership variable

IF Question 105 is related to Team Based Payment system

THEN Add the value of answer to Team Based Payment variable

First, the inference engine checks if the question was answered before. If the question was already answered, then the inference engine does subtractions in order to be able to make the calculations again. Next, the inference engine checks which choice is selected by the user. If the selection is any other than "I don't know," then it does the calculations explained above. If the answer is "I don't know," then it adds zero to the variables and increases the number of unanswered questions by one. At the end, the system also calculates what percentage of questions is answered with meaningful answers.

Once the questions are answered and all calculations are completed, the program saves the values of variables to another table in the database, which is presented in Figure 3.2. Then, that table is sorted in ascending order by an SQL statement in Visual Basic. Therefore, the program gets the sorted results from the database and reveals the reward system with the highest value as the most appropriate reward system for the organization.

B. TESTING AND ERROR HANDLING

The "Reward Consultant" expert system was tested in several different ways. For the first test, all extreme answers were tried for each question. First, all questions were answered as "I don't know", then all were answered as "To a very great extent." Also, all different answers are given to different questions at the same time. Subsequently, it was verified that the algorithm was built correctly and the program does the calculations and mathematical comparisons accurately. Accordingly, the selection of the reward system type with the highest average value is handled by the program properly.

The second test was done to see how the program responds if the user changes an answer by going back to the previous question or by going back to a category already answered. If the user gives a different answer from the previous answer, the program warns the user and makes the change. Thus, it was verified that the program made all corrections and changes to the calculations and databases.

Another test was done to see how the system works with incomplete information. The program does not allow the user to leave a question blank and "I don't know" is given to the user as default choice. However, the user may answer some of the questions in a category and leave the rest unanswered without going any further. In such cases, inadequate information is obtained from the user. Also, the number of questions answered would not be equal to the total number of questions. Similar situations were tested and it was confirmed that the program does the necessary modifications and calculations accurately.

The fourth test was done to see if the program communicates with the knowledge base in every case. In order to do that, all the questions of each category were answered and the answers were changed more than once before and after getting the results. Then, it was also verified that the interconnection between the program and the knowledge base was running correctly.

The last test was done to see if the explanations and reports are presented correctly. In order to do this, the answers of the questions were changed more than once and report and explanation screens were checked for every situation. Therefore, it was confirmed that the reports and explanations were displayed properly.

Ideally, Visual Basic programs would not need error handling codes at all. In reality, however, some hardware problems or unexpected actions by the user can cause run-time errors that halt the program. There is usually nothing the user can do to resume the running application. Other errors might not interrupt code, but they can cause it to act unpredictably. (Bradley and Millspaugh, 1999) In this expert system prototype, I specifically tried to handle the errors caused by unanticipated user actions and the validation of the data entered.

The first thing done was to prevent the user from changing the text of the question presented in text boxes. Second, when the user reaches the last or first record of the database, the program displays a warning message and prevents the user from going any further. Third, validation codes were added to the program in case the user entered any data unexpectedly. Finally, codes were added to give "I don't know" as the default

answer at the beginning of each category to prevent the user from leaving any question blank.

In conclusion, different tests were concluded and different error handling codes were built for different situations and processes. Consequently, it was verified that the mathematical model was constructed and running accurately, the program was making the necessary changes to the program and the calculations in case of incomplete information or extreme answers, interconnection between the program and the knowledge base was running correctly and the errors that might be caused by the user were handled as much as possible.

Nevertheless, correctness of a recommended reward system type for a particular organization was not confirmed. Therefore, correctness of reward system recommendations by "Reward Consultant" expert system should be validated and corroborated by experts on organizational reward systems. Validation of "Reward Consultant" expert system is recommended for future study and is explained thoroughly in Chapter V.

C. GRAPHICAL USER INTERFACE

The user interface for this program serves two important purposes. First, it is the media that the user uses to communicate with the system. Second, it is the interface used to extract the required information about the organization from the user. In order to be able to achieve those purposes, the user interface should be robust, and display all possible responses, results, reports and explanations for the user.

The graphical user interface for "Reward Consultant" can be divided into three groups: Query, Explanation and Output screens.

1. Query Screens

The "Reward Consultant" expert system asks the user questions about the organization with a query screen. First, the user selects a category from the opening screen, and then begins to answer the questions. Figure 4.2 shows the opening screen and Figure 4.3 shows an example of said query screen.

The questions are asked in five different categories: goals, structure, culture, technology and management's vision. Questions are taken from the database and appear to the user in a text box. The choices of answers for each question are listed in a list box containing six possible answers. This prevents user from giving an unpredicted answer and causing an error. When the user begins the consultation, the first answer defaults to "I don't know." This also prevents the user from leaving a question blank.

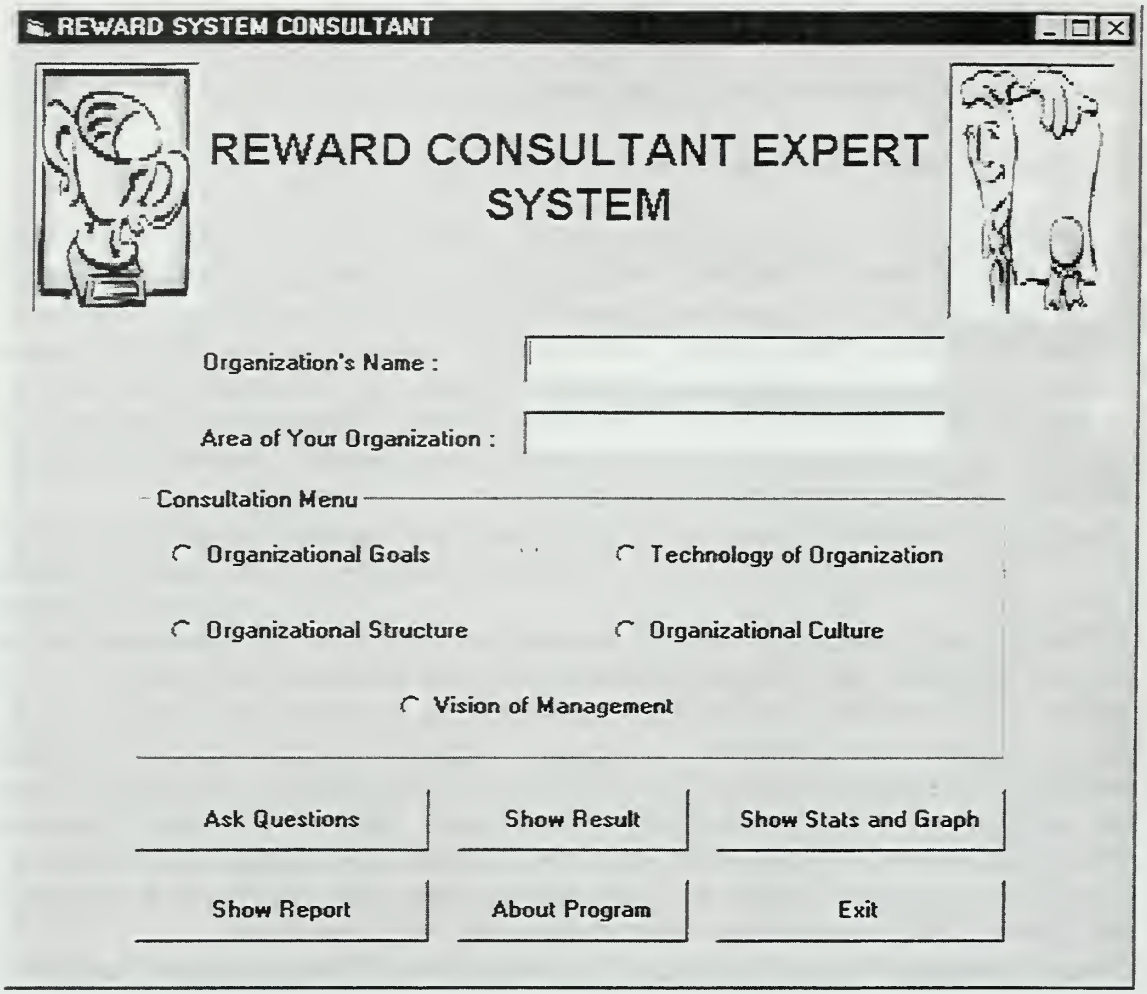

Figure 4.2. Opening Screen of the "Reward Consultant" Expert System.

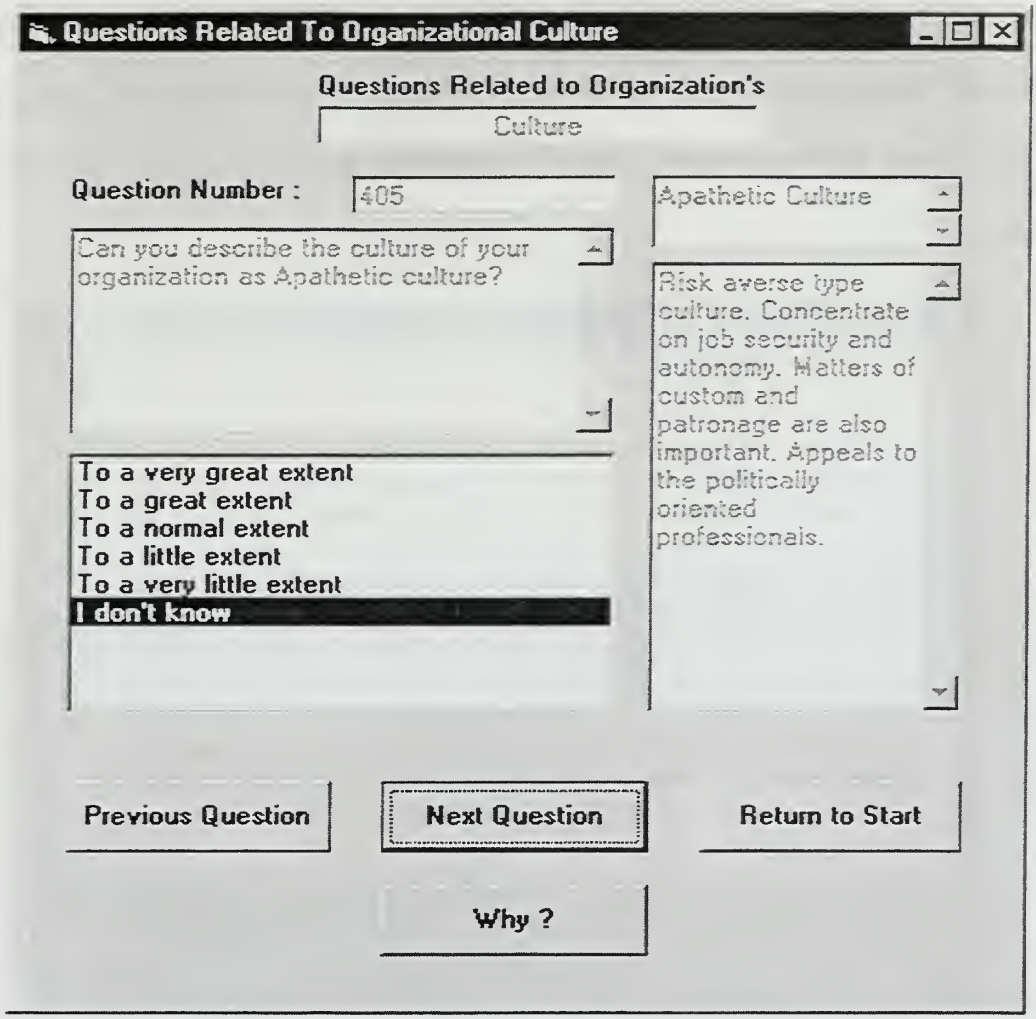

Figure 4.3. A Sample of Query Screen Used for Organizational Culture Category. (Notice that two text boxes are used to explain the term "Apathetic Culture" to the user.)

2. Explanation Screens

As an important part of the expert system's explanation ability, expert systems are supposed to explain the steps of the process to arrive at the solution and how that recommendation is derived. Also, expert systems should be able to explain why specific information is needed for the solution process. By pressing the "Why?" button shown in

Figure 4.3, and "How?" button in output screen, the user can link to an explanation screen for more information. Figure 4.4 shows an example of an explanation screen of the "Why?" button and Figure 4.5 shows an example of an explanation screen of the "How?" button.

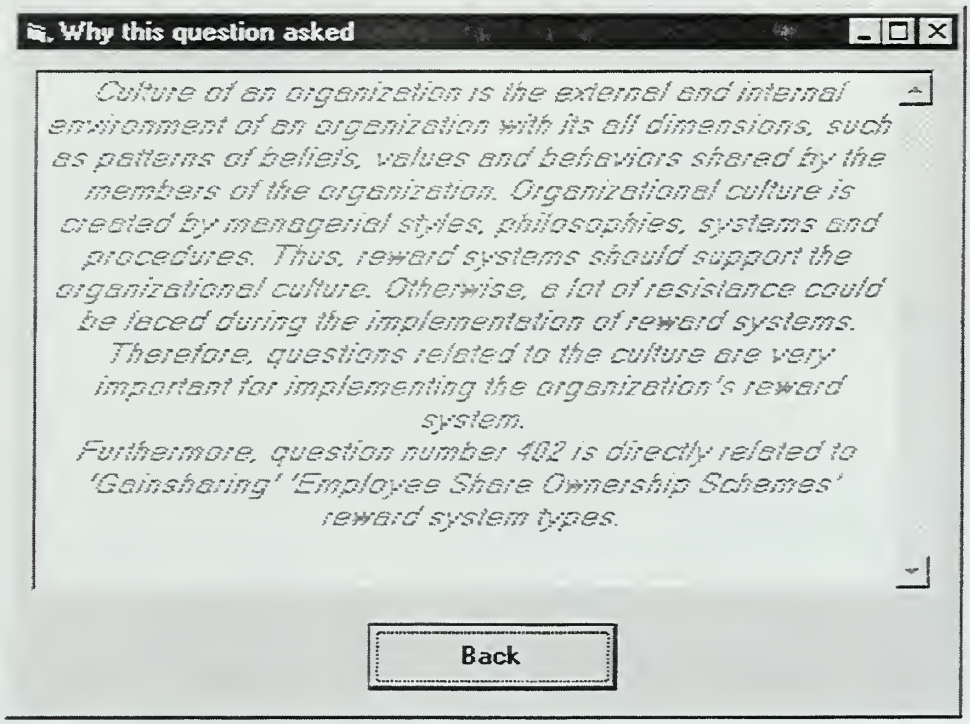

Figure 4.4. A Sample Screen for the Explanation of "Why?"

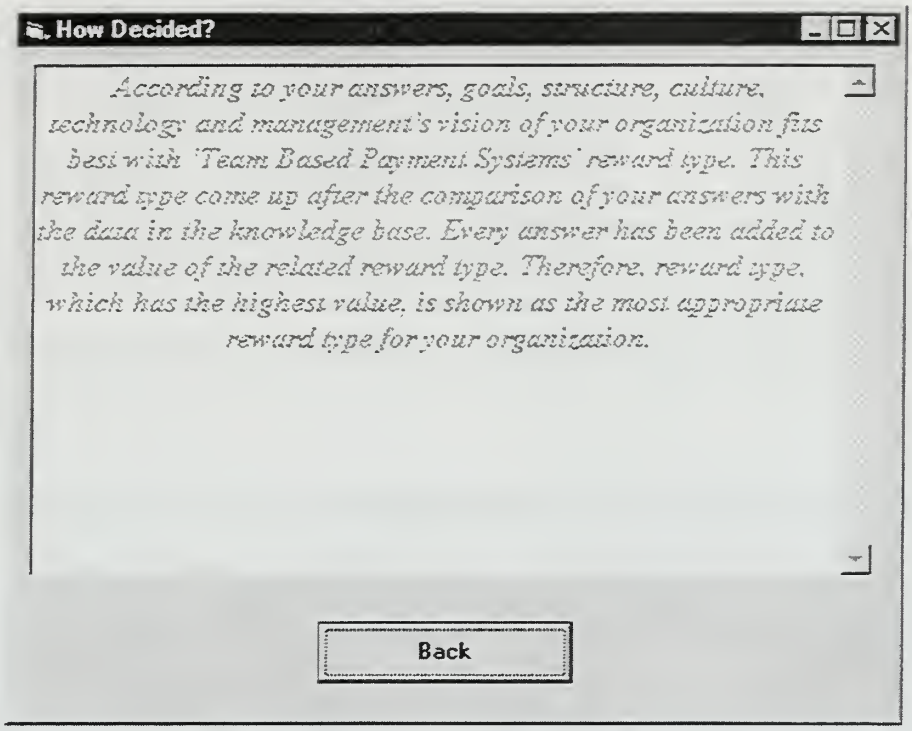

Figure 4.5. A Sample Screen for the Explanation of "How?"

3. Output Screens

Output screens can also be divided into three subgroups as Text Display, Chart and Statistical Display and Reports.

a. Text Display

Once the user answered the questions and entered the information, the most appropriate reward system is determined and displayed on the result screen in text format. Figure 4.6 is a sample output screen in text format.

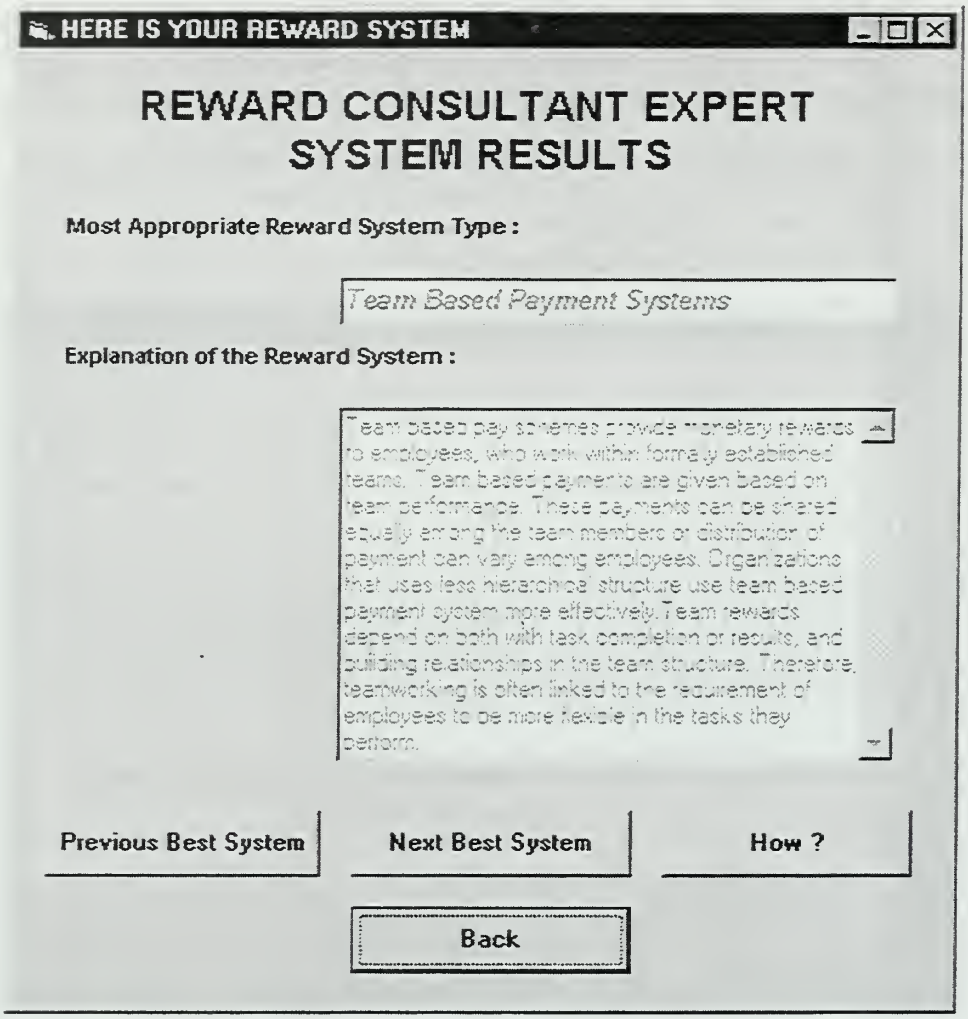

Figure 4.6. A Sample Screen for Output in Text Format.

Key features of this screen are that the user can see the next best reward system for the organization and that the user can also see how this recommendation is derived by pressing the "How?" button.

b. Chart and Statistical Display

Once the most appropriate reward system is determined for the organization and the result is displayed in text format, the user can also see the results in a bar chart by pressing the "Show Stats and Graph" button in the opening screen shown in Figure 4.2. Figure 4.7 is a sample screen for chart and statistical outputs.

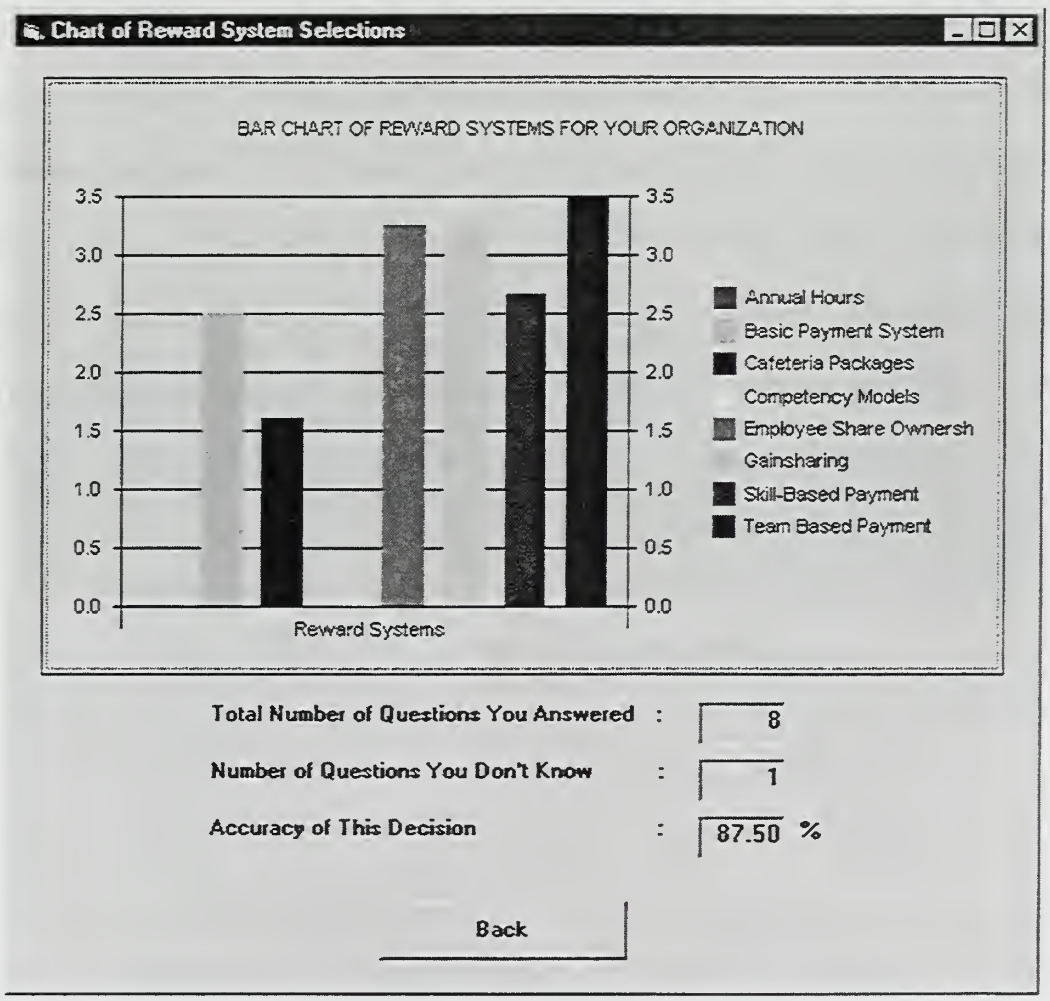

Figure 4.7. A Sample Screen for Chart and Stats Output.

Key features of this screen are that the user can see the relative values of the reward system variables in bar charts and can have a chance to compare them visually. The user also can see the total number of questions answered as well as the total number of questions answered as "I don't know." Thus, the user can see the accuracy of that recommendation as a percentage.

c. Reports

The user can see the reports that show the questions and related answers in text format by pressing the "Show Report" button in Figure 4.2. The user can see all the questions and answers in a report at once or he or she can see those reports separately for each question category. Moreover, the user can also see the total number of questions answered and the number of questions answered as "I don't know" for the selected category. The user can select the format of the report on a report selection screen. Figure 4.8 shows the report selection screen and Figure 4.9 shows an example of a report.

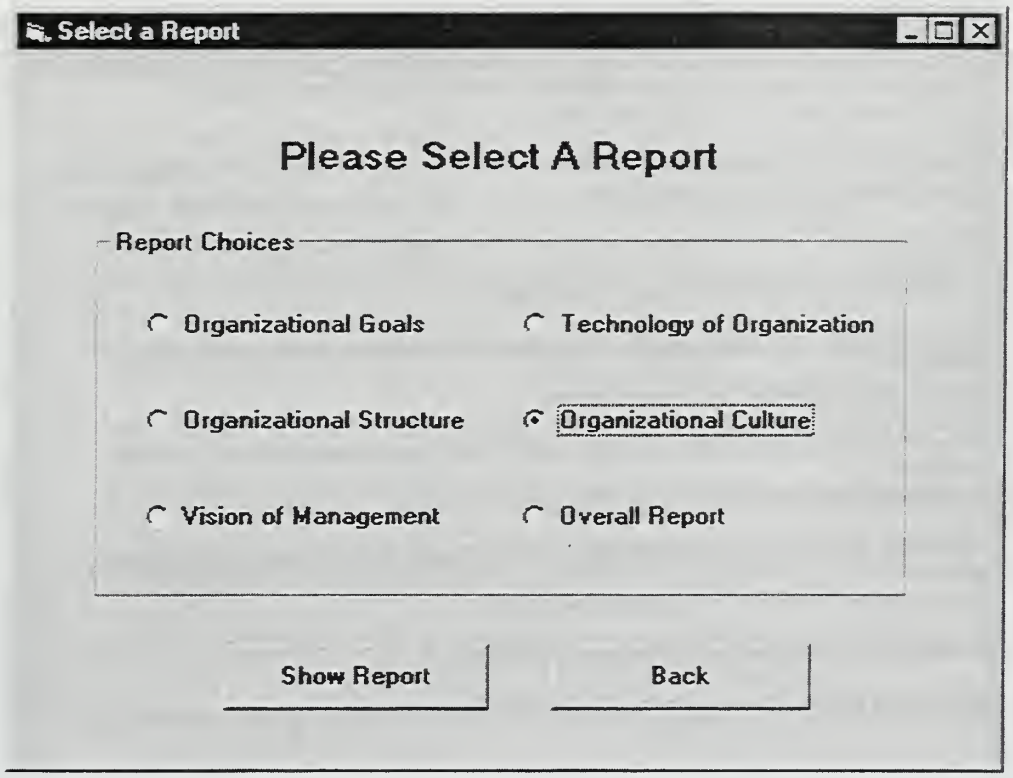

Figure 4.8. Report Selection Screen.

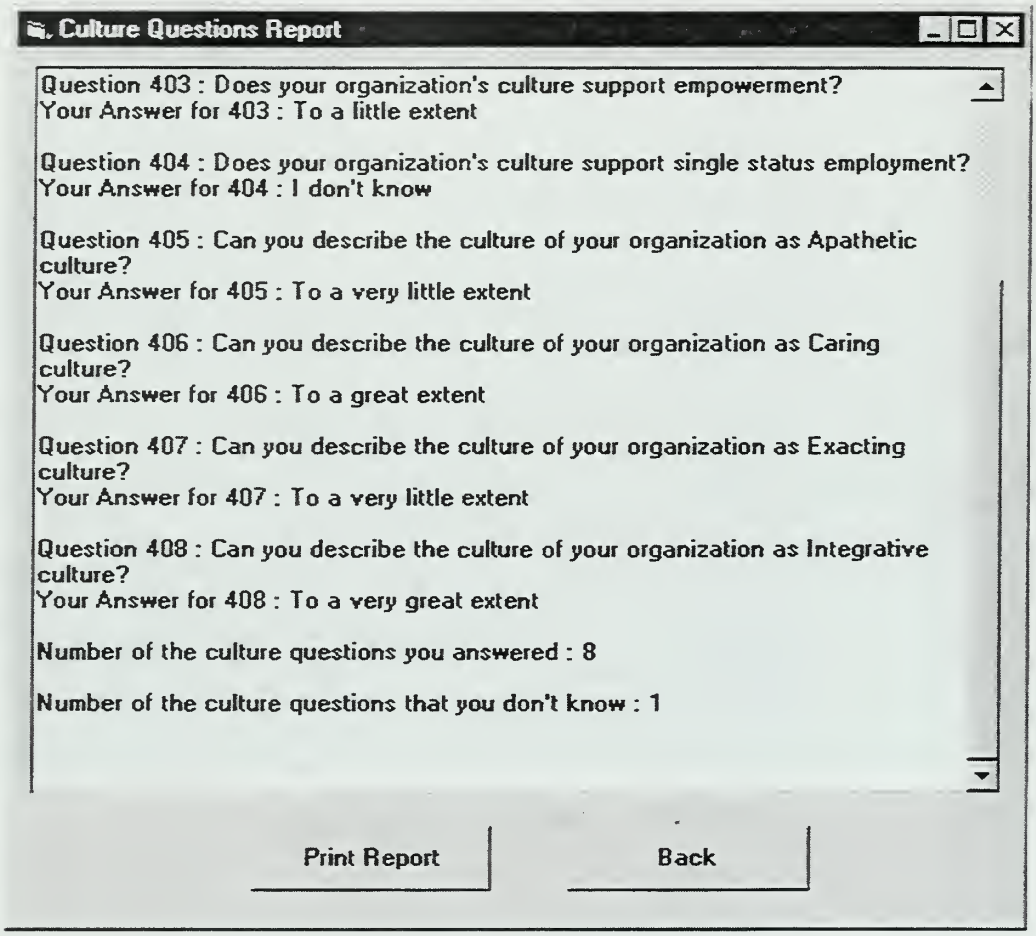

Figure 4.9. A Sample Screen for Reports.

V. SUMMARY

A. SUMMARY OF WORK

In today's business environment, rewarding employees is one of the most difficult and complicated decisions in the organization. In order to handle the reward decision without causing crisis situation, managers have to decide accurately and quickly. The existing systems used in organizations do not include the applications and tools for reward systems. Therefore, currently used methods for deciding reward systems critically depend on the expertise and experience of the organizational reward system experts.

This thesis presented the use of an expert system for organizational reward system selection. The system included eight reward systems: Gainsharing, Employee Share Ownership Schemes, Skill Based Payment, Competency Models, Cafeteria Packages, Annual Hours, Basic Payment and Team Based Payment systems. The methodology involved first creating questions based on a selected, relevant literature review. Heuristics were then developed to describe general characteristics of the reward systems. Next, a mathematical model was constructed based on the heuristics. Finally, the knowledge base was constructed in a Microsoft Access database based on the decision tables.

The system queries the user about the unknown information about the organizational goals, structure, culture, technology and its management's vision. Then, the answers of the user are applied to heuristics and the mathematical model to calculate other information about the reward systems. This collection is then filtered through

decision tables to determine the most appropriate reward system. Then, the result is displayed in text and graphical format.

The system was designed and programmed by using Microsoft Visual Basic 6.0 and the knowledge base was designed and created using Microsoft Access. The completed system was based on 77 questions and 7 database tables. The system can also calculate and display the accuracy of the decision made.

The results of this study show that the knowledge based expert system technique has potential for all types of organizational decisions. A similar system could be designed and implemented for different types of organizational decision processes.

B. RESEARCH QUESTIONS

The following questions were addressed in this thesis:

1. What kind of reward systems are likely to be the most efficient and effective given the organization's goals, structure, culture, technology and management's vision?

As a result of a broad literature review, it was decided that eight types of reward systems including a basic payment system, gainsharing, cafeteria packages, annual hours, team based payment system, competency models, a skill based payment system and employee share ownership schemes were the most effective and efficient systems for the organizations. The "Reward Consultant" expert system shows one or more of these eight reward systems to be the most appropriate reward system or systems by comparing the answers of the user to the knowledge base. More detailed explanations about this question are presented more completely in Chapters II and IV.

2. What are Artificial Intelligence, Knowledge-based systems, expert systems, their applications, and their advantages and disadvantages?

The definition of Artificial Intelligence, Knowledge based systems, expert systems, their applications, advantages and disadvantages are presented in detail in Chapter II.

3. What would be the appropriate interview questions for revealing the most efficient and effective reward systems for an organization?

Including all question categories, 77 questions were constructed after a broad literature review and a study conducted with an expert. These questions were prepared and constructed to query a user in a more effective way and get more complete and consistent information from him or her. A complete list of those 77 questions are presented in Appendix A by category according to five different organizational characteristics including organizational goals, structure, culture, technology and its management's vision.

4. How is it possible to design, create and implement a knowledge-based tool to assist managers in deciding the most appropriate reward system?

The methodology used for this thesis involved creating questions used to extract the required information from the user, developing heuristics used to describe general characteristics of the reward systems, constructing a mathematical model based on heuristics and building the knowledge base by using a database application. After that, the application program of the system is designed and programmed by using Microsoft Visual Basic 6.0.

5. How can the efficiency of such a system be measured?

The efficiency of such a system can be measured by the validation of the program. Validation is a part of the computer programming activity. Since expert systems are computer programs and validation is a part of the computer programming cycle, the "Reward Consultant" expert system is to be validated. (Meseguer and Plaza, 1992) The types of validation methods for "Reward Consultant" are explained in more detail in Section C of this chapter.

6. How can the results from this study be generalized?

Although the rules and concepts based on the literature review are valid through the experts, the results of the program may not be generalized for all types of organizations. Different environmental and cultural issues of various communities and nations may affect the organizational characteristics in unconsidered ways. Therefore, the results of this expert system may not be applied to all organizations.

Nevertheless, after this research, it is clearly understood that the decision support systems and knowledge-based expert systems would be very useful for managers in organizations by reducing decision times and increasing accuracy and consistency of the decisions regarding related areas.

C. VALIDATION OF "REWARD CONSULTANT" EXPERT SYSTEM

Validation of a program can be defined as verifying if the program is fully operational and free of errors, performs its intended functions and satisfies user requirements. Several different methods and tools exist for program validation. Each method provides partial validation of the program. Therefore, a combination of methods

is usually required for a reasonable degree of validation. Since an expert system is a piece of software that performs various tasks intending to satisfy the needs of potential users, they should be validated by using several methods. (Meseguer and Plaza, 1992)

Nevertheless, expert systems perform tasks that have been done by humans until now. They usually have to deal with fuzzy, incomplete, uncertain and sometimes inconsistent information. Consequently, expert systems frequently do not have a single solution. Multiple solutions may be equally acceptable. Moreover, an expert systems evaluation has to rely on matching expert system performance against human expert capability. Hence, to defeat these problems, sound validation methods for expert systems are required. (Meseguer and Plaza, 1992)

Three different validations should be applied to the "Reward Consultant" expert system: validation of heuristics and decision matrices, validation of mathematical model and user or expert validation of results. Although heuristics and decision matrices were constructed based on the literature review, the IF-THEN sentences of decision matrices can be validated by an expert system validation tool. For instance, the KB-REDUCER system (Ginsberg, 1988) and the COVADIS system (Rousset, 1988) are two examples of expert system validation tools. These tools detect all inconsistencies and redundancies in forward chaining, proportional rule bases. Validation of the mathematical model was done by manual testing. During manual testing, several different combinations of scenarios were tried and the program gave accurate results for a variety of inputs. As final validation, user validation of results should be performed. This can be done by the application of different case studies or real life examples to the program and validating

the results with an expert. Final validation can also be done by using a kind of test similar to the Turing Test. (Turban and Aronson, 1998)

D. RECOMMENDATIONS FOR FUTURE STUDY

1. Adding More Reward Systems to the Knowledge Base

Currently, the "Reward Consultant" expert system covers only eight types of reward systems. Since the computation part of the program runs separately from the knowledge base, the reward system database could be expanded to cover more reward systems. A future study could be held to research new types of rewards systems in the organizations parallel to the new economic, social and cultural changes of continuously improving information technology. Then, those reward systems could be studied and added to the database of the system without having to make any changes to the coding of the system.

2. Adding New Question Categories

Today, organizations invent new structures, processes and business practices to keep up with the changes and inventions in the business environment. The "Reward Consultant" expert system involves five main categories that cover the organization's different perspectives. Consequently, new categories could be modeled and added to the system to capture those new structures and processes. Furthermore, the system's database could be expanded to include more questions, which would be used to extract more detailed and accurate information from the user.

3. Adding New Capabilities to the System

Although the "Reward Consultant" expert system runs correctly and does computations accurately, new modifications and abilities could be added to the system. For instance, the program currently cannot save a consultation scenario to a file. Therefore, capability of saving a consultation scenario, opening and running an old scenario could be programmed and added to the system. Likewise, some modifications could also be made to the graphical user interface, outputs and report formats.

4. Validation of the System

As explained in Section C of this chapter, user validation of the "Reward Consultant" expert system could be performed in the next step. This validation can be done in several ways. One is to apply a case study or a real life scenario to the system and validate the results by an expert. This would validate that the program could give logical results for different types of situations. Another form of validation could be done by using an expert system validation tool. This would validate that the program could give logical and expected results to several different inputs. Using validation tools would also validate that the heuristics and decision tables were constructed correctly and accurately.

E. LESSONS LEARNED

There are several things that I learned during the research and study of this thesis that could be beneficial to others working on similar projects. First, the decision tables or trees and knowledge base should be constructed carefully. When it is transferred to a database, data should be correctly formatted and arranged. Database tables and

relationships among those tables should be correctly and cautiously designed. Moreover, the database for the knowledge base should be carefully considered and selected. The selected database should allow an easy data exchange between the expert system and the database.

Second, the decision to use a reward system in organizations requires that the information about the organization be highly accurate. For this reason, a significant amount of time should be dedicated to the development of constructing questions and developing heuristics and the mathematical model of the system. Since most of the managerial decisions are subjective, it is impossible to find a 100% correct solution to the problem. However, every opportunity should be taken to increase the accuracy of the decision.

F. CONCLUSION

Knowledge based expert systems have enormous potential for managerial decision making processes in organizations. More important are the benefits that expert systems offer for the organizations. Managers must identify the characteristics of their organizations quickly and accurately with incomplete and vague information. This thesis presented ways that an expert system can improve accuracy, decision time and increase available information to make correct decisions. A wrong decision can cause dreadful results, and an expert system can help ensure that the analysis is correct.

APPENDIX A - LIST OF QUESTIONS ASKED BY THE SYSTEM

In this appendix, a list of all questions are presented and grouped by their categories. The numbers of questions were assigned according to the category of the question. Therefore, 100s are for organizational goals, 200s are for organizational structure, 300s are for organization's technology, 400s are for organizational culture and 500s are for management's vision.

The following questions are related to organizational goals:

How important or valuable is each of the following goals for your organization:

Q101. Implementing a clear, more easily understood reward system

Q102. Rewarding undervalued high individual contribution

Q103. Rewarding team contribution

Q104. Promoting employees according to the variety or number of their skills

Q105. Flexibility in promoting employees

Q106. Rewarding loyalty

Q107. Helping to create a focus on major improvement projects you are undertaking

Q108. Focusing employees' attention where they can most impact business achievement

Q109. Linking team reward to business performance

Q110. Supporting a culture of continuous improvement

Q111. Linking employee and organizational performance to one unified set of goals

Q112. Cost effectiveness

Q113. Tax efficiency

Q114. High commitment of the staff to the company in the long-term

Q115. Supporting an innovative and learning organization that can adapt to changes

Q116. Supporting role flexibility of employees in their work relationships

Q117. Support different carrier paths in your organization

Q118. Creating a work environment focused on the quality of outcomes

Q119. Creating a work environment focused on process efficiency

Q120. Constant unit production or service-delivery cost

Q121. Reducing overtime costs

Q122. Reducing the necessity of stockholding

Q123. More predictable payroll costs for budgeting purposes

Q124. Reduce working time and just meeting the requirements of weekly hours

The following questions are related to the organizational structure:

To what extent does your organization have?

Q201. Formal plans to achieve your long-term goals

Q202. An effective communication system

Q203. Regular informal feedback once or twice a year

Q204. Opportunities for employees to do a variety of tasks in their jobs

Q205. Opportunities for job rotation among your employees

Q206. Opportunities for employees to use a high level and range of skills

Q207. Flexible work arrangements

Q208. Capital intensive rather than labor intensive

Q209. An adjustable payroll package

Q210. What is the number of people in your organization? (More than 2000, 2000-1500, 1500-1000, 1000-500, Less than 500)

Q211. Is the overtime earnings account larger than average portion of the overall pay bill?

The following questions are related to the organization's technology:

Q301. Is your business environment continuously being changed by new technology?

Q302. Is the technology that you use a continuous process or manufacturing system?

Q303. What is the degree of the high technology you use in your organization?

The following questions are related to the organizational culture:

Q401. Is there a high involvement management culture in your organization, which encourages the participation of all concerned in the design and operation of the system?

Q402. Should the reward system support or promote a strong culture?

Q403. Does your organization's culture support empowerment?

Q404. Does your organization's culture support single status employment?

Q405. Can you describe the culture of your organization as Apathetic culture?

Q406. Can you describe the culture of your organization as Caring culture?

Q407. Can you describe the culture of your organization as Exacting culture?

Q408. Can you describe the culture of your organization as Integrative culture?

The following questions are related to management's vision:

Q501. Should the union -if there is one- be involved in the design or implementation process?

Q502. Do you think that rewards are related to employee's abilities to use a wide range of skills?

Q503. Do you think that rewards are related to employee's abilities to apply a higher level of skills to different jobs or tasks?

Q504. Do you think that rewards are related to employee's ability rather than the scope of the job?

Q505. Should the human resource development be integrated with reward policies?

Q506. Should the reward systems be skill-based or people-based rather than job-based?

Q507. Do you support non-pay benefit packages, such as extra holidays, dependant medical care insurance, childcare vouchers, rail travel concessions, increased emphasis on health insurance and pension?

Q508. Do you want to let the employees select their own rewards among a number of choices?

Q509. Do you want to offer a great range of rewards to the employees?

To what extent should the reward systems be related to each of the following?

Q510. Performance delivery

Q511. Individual behavior and attributes, which they bring to the job

Q512. Development of the individual as a path to increasing organizational competence

Q513. Training, development, performance and recruitment

Q514. Seniority (Amount of time in the organization)

Q515. Levels of ranks

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B - DECISION MATRIX

The following decision matrix shows the relationship between the questions and the reward system.

Number	RelatedArea	BasicPayment	SkillBased	Competency	Gainsharing	Cafeteria	AnnualHours	EmpShareOwn	TeamBased
101	Goal	1	0	0	0	0	0	0	0
102	Goal	0	1	1	1	0	0	1	0
103	Goal	0	0	1	1	0	0	1	1
104	Goal	0	1	1	0	0	0	0	0
105	Goal	0	1	1	0	0	0	0	1
106	Goal	1	0	0	1	0	0	1	0
107	Goal	0	0	0	1	0	0	1	0
108	Goal	0	0	0	1	0	0	0	0
109	Goal	0	0	0	1	0	0	0	1
110	Goal	0	0	0	1	0	0	1	1
111	Goal	0	0	0	1	0	0	0	1
112	Goal	0	0	0	0	0	0	1	1
113	Goal	0	0	0	0	0	0	1	1
114	Goal	0	0	0	0	0	0	1	1
115	Goal	0	0	1	0	0	0	0	0
116	Goal	0	0	1	0	0	0	0	0
117	Goal	0	0	1	0	0	0	0	0
118	Goal	0	0	0	0	0	1	0	0
119	Goal	0	0	0	0	0	1	0	0
120	Goal	0	0	0	0	0	1	0	0
121	Goal	0	0	0	0	0	1	0	0
122	Goal	0	0	0	0	0	1	0	0
123	Goal	0	0	0	0	0	1	0	0
124	Goal	0	0	0	0	0	1	0	0
201	Structure	0	0	0	1	0	0	0	0
202	Structure	0	1	1	1	0	0	1	1
203	Structure	0	1	0	0	0	0	0	0
204	Structure	0	1	0	0	0	0	0	0
205	Structure	0	1	0	0	0	0	0	0
206	Structure	0	1	0	0	0	0	0	0
207	Structure	0	1	0	0	0	0	0	1
208	Structure	1	1	0	0	0	0	0	0
209	Structure	0	0	0	0	1	0	0	0
210	Structure	1	0	0	1	0	0	1	0
211	Structure	1	0	0	0	0	1	0	0
301	Technology	0	0	0	1	0	0	1	0
302	Technology	0	1	0	0	0	0	0	0
303	Technology	0	0	0	1	0	0	1	0
401	Culture	0	1	0	0	0	0	0	1
402	Culture	0	0	0	1	0	0	1	0
403	Culture	0	0	0	0	1	0	0	0
404	Culture	0	0	0	0	1	0	0	0
405	Culture	1	0	0	0	1	0	0	0
406	Culture	1	0	0	1	1	0	1	0
407	Culture	0	1	0	1	1	0	1	1
408	Culture	0	1	0	1	0	0	1	1
501	Vision of Manag	0	0	0	1	0	0	0	0
502	Vision of Manag	0	1	0	0	0	0	0	0
503	Vision of Manag	0	1	0	0	0	0	0	0
504	Vision of Manag	0	1	0	0	0	0	0	0
505	Vision of Manag	0	0	0	1	0	0	1	0
506	Vision of Manag	0	1	1	0	0	0	0	0
507	Vision of Manag	0	0	0	0	1	0	0	0
508	Vision of Manag	0	0	0	0	1	0	0	0
509	Vision of Manag	0	0	0	0	1	0	0	0
510	Vision of Manag	0	1	1	0	0	0	0	0
511	Vision of Manag	0	1	1	0	0	0	0	0
512	Vision of Manag	0	0	1	0	0	0	0	0
513	Vision of Manag	0	0	1	0	0	0	0	0
514	Vision of Manag	1	0	1	0	1	0	0	0
515	Vision of Manag	1	0	1	0	1	0	0	0

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C - SAMPLE CODE OF THE SYSTEM

The program has 23 different forms programmed in Visual Basic. Following is the sample code from the program that shows the actual code of the mathematical model of the system.

```
Option Explicit

Private Sub cmdNext_Click()
  Dim index As Integer
  'Checking if the answer is given
  If (txtSecretQNumber.Text <> "") Then
    index = Val(txtSecretQNumber.Text) - 300
    'If the question is answered before then subtract all previous values about this
    question
    If (gaTechnoAnswers(index) <> "") Then
      giTechnoCount = giTechnoCount - 1
      If (gaTechnoAnswers(index) = "To a very great extent") Then
        gcBasicTotal = gcBasicTotal - (Val(txtBasic.Text) * 5)
        gcSkillTotal = gcSkillTotal - (Val(txtSkill.Text) * 5)
        gcCompetencyTotal = gcCompetencyTotal - (Val(txtCompetency.Text) * 5)
        gcGainsharingTotal = gcGainsharingTotal - (Val(txtGain.Text) * 5)
```

```
    gcCafeteriaTotal = gcCafeteriaTotal - (Val(txtCafe.Text) * 5)
    gcAnnualTotal = gcAnnualTotal - (Val(txtAnnual.Text) * 5)
    gcEmpshareTotal = gcEmpshareTotal - (Val(txtEmpshare.Text) * 5)
    gcTeambasedTotal = gcTeambasedTotal - (Val(txtTeam.Text) * 5)
  ElseIf (gaTechnoAnswers(index) = "To a great extent") Then
    gcBasicTotal = gcBasicTotal - (Val(txtBasic.Text) * 4)
    gcSkillTotal = gcSkillTotal - (Val(txtSkill.Text) * 4)
    gcCompetencyTotal = gcCompetencyTotal - (Val(txtCompetency.Text) * 4)
    gcGainsharingTotal = gcGainsharingTotal - (Val(txtGain.Text) * 4)
    gcCafeteriaTotal = gcCafeteriaTotal - (Val(txtCafe.Text) * 4)
    gcAnnualTotal = gcAnnualTotal - (Val(txtAnnual.Text) * 4)
    gcEmpshareTotal = gcEmpshareTotal - (Val(txtEmpshare.Text) * 4)
    gcTeambasedTotal = gcTeambasedTotal - (Val(txtTeam.Text) * 4)
  ElseIf (gaTechnoAnswers(index) = "To a normal extent") Then
    gcBasicTotal = gcBasicTotal - (Val(txtBasic.Text) * 3)
    gcSkillTotal = gcSkillTotal - (Val(txtSkill.Text) * 3)
    gcCompetencyTotal = gcCompetencyTotal - (Val(txtCompetency.Text) * 3)
    gcGainsharingTotal = gcGainsharingTotal - (Val(txtGain.Text) * 3)
    gcCafeteriaTotal = gcCafeteriaTotal - (Val(txtCafe.Text) * 3)
    gcAnnualTotal = gcAnnualTotal - (Val(txtAnnual.Text) * 3)
    gcEmpshareTotal = gcEmpshareTotal - (Val(txtEmpshare.Text) * 3)
    gcTeambasedTotal = gcTeambasedTotal - (Val(txtTeam.Text) * 3)
```

```
ElseIf (gaTechnoAnswers(index) = "To a little extent") Then
  gcBasicTotal = gcBasicTotal - (Val(txtBasic.Text) * 2)
  gcSkillTotal = gcSkillTotal - (Val(txtSkill.Text) * 2)
  gcCompetencyTotal = gcCompetencyTotal - (Val(txtCompetency.Text) * 2)
  gcGainsharingTotal = gcGainsharingTotal - (Val(txtGain.Text) * 2)
  gcCafeteriaTotal = gcCafeteriaTotal - (Val(txtCafe.Text) * 2)
  gcAnnualTotal = gcAnnualTotal - (Val(txtAnnual.Text) * 2)
  gcEmpshareTotal = gcEmpshareTotal - (Val(txtEmpshare.Text) * 2)
  gcTeambasedTotal = gcTeambasedTotal - (Val(txtTeam.Text) * 2)
ElseIf (gaTechnoAnswers(index) = "To a very little extent") Then
  gcBasicTotal = gcBasicTotal - (Val(txtBasic.Text) * 1)
  gcSkillTotal = gcSkillTotal - (Val(txtSkill.Text) * 1)
  gcCompetencyTotal = gcCompetencyTotal - (Val(txtCompetency.Text) * 1)
  gcGainsharingTotal = gcGainsharingTotal - (Val(txtGain.Text) * 1)
  gcCafeteriaTotal = gcCafeteriaTotal - (Val(txtCafe.Text) * 1)
  gcAnnualTotal = gcAnnualTotal - (Val(txtAnnual.Text) * 1)
  gcEmpshareTotal = gcEmpshareTotal - (Val(txtEmpshare.Text) * 1)
  gcTeambasedTotal = gcTeambasedTotal - (Val(txtTeam.Text) * 1)
End If
If Val(txtBasic.Text) = 1 Then
  giBasicCount = giBasicCount - 1
End If
```

```
If Val(txtSkill.Text) = 1 Then
  giSkillCount = giSkillCount - 1
End If
If Val(txtCompetency.Text) = 1 Then
  giCompetencyCount = giCompetencyCount - 1
End If
If Val(txtGain.Text) = 1 Then
  giGainsharingCount = giGainsharingCount - 1
End If
If Val(txtCafe.Text) = 1 Then
  giCafeteriaCount = giCafeteriaCount - 1
End If
If Val(txtAnnual.Text) = 1 Then
  giAnnualCount = giAnnualCount - 1
End If
If Val(txtEmpshare.Text) = 1 Then
  giEmpshareCount = giEmpshareCount - 1
End If
If Val(txtTeam.Text) = 1 Then
  giTeambasedCount = giTeambasedCount - 1
End If
If (gaTechnoAnswers(index) = "I don't know") Then
```

```
      giTechnoDontKnow = giTechnoDontKnow - 1
    End If
  End If

'Add 1 to total number of questions answered
giTechnoCount = giTechnoCount + 1
'Add values to each of related reward system variables
If Val(txtBasic.Text) = 1 Then
  giBasicCount = giBasicCount + 1
  gcBasicTotal = gcBasicTotal + (Val(txtBasic.Text) * Val(txtSecretAnswer.Text))
End If
If Val(txtSkill.Text) = 1 Then
  giSkillCount = giSkillCount + 1
  gcSkillTotal = gcSkillTotal + (Val(txtSkill.Text) * Val(txtSecretAnswer.Text))
End If
If Val(txtCompetency.Text) = 1 Then
  giCompetencyCount = giCompetencyCount + 1
  gcCompetencyTotal = gcCompetencyTotal + (Val(txtCompetency.Text) *
                      Val(txtSecretAnswer.Text))
End If
If Val(txtGain.Text) = 1 Then
  giGainsharingCount = giGainsharingCount + 1
```

```
    gcGainsharingTotal = gcGainsharingTotal + (Val(txtGain.Text) *
                         Val(txtSecretAnswer.Text))
  End If
  If Val(txtCafe.Text) = 1 Then
    giCafeteriaCount = giCafeteriaCount + 1
    gcCafeteriaTotal = gcCafeteriaTotal + (Val(txtCafe.Text) *
                       Val(txtSecretAnswer.Text))
  End If
  If Val(txtAnnual.Text) = 1 Then
    giAnnualCount = giAnnualCount + 1
    gcAnnualTotal = gcAnnualTotal + (Val(txtAnnual.Text) *
                    Val(txtSecretAnswer.Text))
  End If
  If Val(txtEmpshare.Text) = 1 Then
    giEmpshareCount = giEmpshareCount + 1
    gcEmpshareTotal = gcEmpshareTotal + (Val(txtEmpshare.Text) *
                      Val(txtSecretAnswer.Text))
  End If
  If Val(txtTeam.Text) = 1 Then
    giTeambasedCount = giTeambasedCount + 1
    gcTeambasedTotal = gcTeambasedTotal + (Val(txtTeam.Text) *
                       Val(txtSecretAnswer.Text))
```

```
    End If
    'If question is answered as I don't know then add 1 to total number of questions
    unknown
    If (lstAnswers.List(lstAnswers.ListIndex) = "I don't know") Then
      giTechnoDontKnow = giTechnoDontKnow + 1
    End If
    If (gaTechnoAnswers(index) = "") Or (gaTechnoAnswers(index) =
                                        lstAnswers.List(lstAnswers.ListIndex)) Then
      gaTechnoAnswers(index) = lstAnswers.List(lstAnswers.ListIndex)
    Else
      MsgBox "Be careful. You are changing your answer." & vbCrLf & "Your answer
            for this question was '" & gaTechnoAnswers(index) & "'.",
            vbExclamation, "Answer Misfit"
      gaTechnoAnswers(index) = lstAnswers.List(lstAnswers.ListIndex)
    End If
  End If
  If (datQuestions.Recordset.EOF) Or (Val(txtNumber.Text) = giTechnoLast) Then
    MsgBox "This was the last question. Thank you.", vbExclamation, "End of
          Questions."
  Else
    datQuestions.Recordset.MoveNext
  End If
```

```
End Sub

Private Sub cmdPrevious_Click()
    'Go to the previous question. If user is already at the first question then warn the user
    If (datQuestions.Recordset.BOF) Or (Val(txtNumber.Text) = 301) Then
      MsgBox "This was the first question.", vbExclamation, "Beginning of Questions."
    Else
      datQuestions.Recordset.MovePrevious
    End If
End Sub

Private Sub cmdStart_Click()
    Unload Me
    frmOpening.Show
End Sub

Private Sub cmdWhy_Click()
    'Determine the related reward systems of that question.
    gsGeneralText = ""
    gsBasicText = ""
    gsSkillText = ""
    gsCompetencyText = ""
```

```
gsGainsharingText = ""
gsCafeteriaText = ""
gsAnnualText = ""
gsEmpShareText = ""
gsTeamBasedText = ""
If Val(txtBasic.Text) = 1 Then
  gsBasicText = "'Basic Payment' "
End If
If Val(txtSkill.Text) = 1 Then
  gsSkillText = "'Skill Based Payment' "
End If
If Val(txtCompetency.Text) = 1 Then
  gsCompetencyText = "'Competency Models' "
End If
If Val(txtGain.Text) = 1 Then
  gsGainsharingText = "'Gainsharing' "
End If
If Val(txtCafe.Text) = 1 Then
  gsCafeteriaText = "'Cafeteria Packages' "
End If
If Val(txtAnnual.Text) = 1 Then
  gsAnnualText = "'Annual Hours Payment' "
```

```
End If
If Val(txtEmpshare.Text) = 1 Then
  gsEmpShareText = "'Employee Share Ownership Schemes' "
End If
If Val(txtTeam.Text) = 1 Then
  gsTeamBasedText = "'Team Based Payment' "
End If
gsGeneralText= gsGeneralText & gsBasicText & gsSkillText & gsCompetencyText &
        gsGainsharingText & gsCafeteriaText & gsAnnualText & gsEmpShareText &
        gsTeamBasedText
frmTechnoWhy!txtWhy.Text = "Technology in the organization includes the computer
            technology used in management and decision process, the technology
            used in production process, and required software, hardware and
            machinery. " & _
              "Reward systems should be able to provide the requirements of the
            high technology in the organizations, and should be able to cover side
            effects. " & _
              "Therefore, questions related to the technology affects the
            organization's " & _
              "reward systems." & vbCrLf & "Furthermore, question number " &
             txtNumber.Text & _
              " is directly related to " & gsGeneralText & " reward system types."
```

```
    'Display the explanation screen
    frmTechnoWhy.Show vbModal
End Sub

Private Sub Form_Load()
    lstAnswers.Selected(5) = True
End Sub

Private Sub lstAnswers_Click()
    txtSecretAnswer.Text = 5 - lstAnswers.ListIndex
End Sub
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Bellinger, Gene, *Knowledge Management - Emerging Perspectives*, [http:// www.outsights.com/systems/kmgmt/kmgmt.htm], 1997.

Bradley, Julia Case and Millspaugh Anita C., *Programming in Visual Basic 6.0*, Irwin/McGraw-Hill, New York, 1999.

Brown, Carol E. and O'Leary, Daniel E., *Introduction to Artificial Intelligence and Expert Systems*, [http://www.bus.orst.edu/faculty/brownc/es_tutor/es_tutor.htm], 1995.

Bryson, John M., *Strategic Planning for Public and Nonprofit Organizations: A Guide to Strengthening and Sustaining Organizational Achievement*, Jossey-Bass Publishers, 1995.

Burton, Richard M. and Obel, Borge, *Strategic Organizational Diagnosis and Design*, 2nd Edition, Kluwer Academic Publishers, Boston, 1998.

Conger, Jay A., Spreitzer, Gretchen M. and Lawler III, Edward E., *The Leader's Change Handbook: An Essential Guide to Setting Direction and Taking Action*, Jossey-Bass Publishers, 1999.

Galbraith, Jay R., *Organizatinal Design*, Reading Mass Addison- Wesley, 1977.

Ginsberg, A., *Knowledge Base Reduction: A New Approach to Checking Knowledge Bases for Inconsistency and Redundancy*, Proc.AAAI-88, 1988.

Hackman, J.Richard and Suttle, J.Lloyd, *Improving Life At Work: Behavioral Science Approaches to Organizational Change*, Goodyear Publishing Company, 1977.

Hart, A., *Knowledge Acquisition for Expert Systems*, McGraw-Hill, New York, 1986.

Jansen, Erik, *Toward A Strategic Reward Systems Perspective*, Ph.D. Dissertation, University of South California, December 1986.

Kaplan, S.J., *The Industrialization of Artificial Intelligence: From By-Line to Bottom-Line*, AI Magazine, Summer 1984.

Lawler III, Edward E., *Improving the Quality of Work Life: Reward Systems,* U.S. Department of Commerce National Technical Information Service, PB-255045, June 1975

Meseguer, Pedro, and Plaza, Enric, *Validation of KBS: The Valid Project*, IIIA Research Report 92-20, [http://www.iiia.csic.es/People/enric/valid-kbs.html], 1992.

Nadler, David A. and Tushman, Michael L., *Organizational Dynamics*, American Management Association, New York, 1980.

Perry, Greg, *Teach Yourself Visual Basic 5 in 24 Hours*, Macmillan Computer Publishing, 1997.

Rich, Elaine and Knight, Kevin, *Artificial Intelligence*, 2nd Edition, McGraw-Hill, New York, 1991.

Rousset, M.C., *On the Consistency of Knowledge Bases: The COVADIS System*, Proceedings of ECAI'88, Munchen, August 1988.

Senge, Peter M., *The Fifth Discipline: The Art & Practice of The Learning Organization*, Currency Doubleday, New York, 1990.

Sethia, Nirmal and Glinow, Mary Ann Von, *Gaining Control of the Corporate Culture*, Jossey-Bass Publishers, 1985.

Turban, Efraim and Aronson, Jay E., *Decision Support Systems and Intelligent Systems*, 5th Edition, Prantice Hall, 1998.

Von Glinow, Mary Ann, *The New Professionals: Managing Today's High-Tech Employees*, Ballinger Publishing Company, 1988.

Zigon, Jack, *Rewards and Performance Incentives*, [http://www.zigonperf.com/Articles/rewards.htm], 2000.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Road, Ste 0944
   Fort Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Road
   Monterey, California 93943-5101

3. Deniz Kuvvetleri Komutanligi 1
   Personel Daire Baskanligi
   Bakanliklar
   Ankara, Turkey

4. Deniz Kuvvetleri Komutanligi 1
   Kutuphanesi
   Bakanliklar
   Ankara, Turkey

5. Deniz Harp Okulu 2
   Kutuphanesi
   Tuzla
   Istanbul, Turkey

6. Chairman, Code SM 1
   Naval Postgraduate School
   Monterey, CA 93943

7. Chairman, Code IS 1
   Naval Postgraduate School
   Monterey, CA 93943

8. Professor Erik Jansen, Code SM/Ek 1
   Naval Postgraduate School
   Monterey, CA 93943

9. Professor Mark E. Nissen, Code SM/Ni 1
   Naval Postgraduate School
   Monterey, CA 93943

10. Yildizlar Egitim Merkezi Komutanligi.....1
Harp Oyunu Grup Baskanligi
Golcuk
Kocaeli, Turkey

11. LTJG. Alper Erturk.....2
Kavacik Kaptanlar mahallesi
Ciftlik caddesi No:17 Daire:4
Anadoluhisari, Beykoz
Istanbul, TURKEY