INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

PROVIDING INTEGRATED QUANTITATIVE AND QUALITATIVE INFORMATION SUPPORT: DESIGN AND DEVELOPMENT OF A DECISION SUPPORT SYSTEM INCORPORATING A TEXT BASE FOR CORPORATE PERFORMANCE ANALYSIS

A Dissertation

Presented for the

Doctor of Philosophy

Degree

The University of Mississippi

Chi Hwang

December 1995

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

UMI Number: 9615659

UMI Microform 9615659
Copyright 1996, by UMI Company. All rights reserved.

This microform edition is protected against unauthorized
copying under Title 17, United States Code.

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

To the Graduate Council:

I am submitting herewith a dissertation written by Chi Hwang entitled "Providing Integrated Quantitative and Qualitative Information Support: Design and Development of a Decision Support System Incorporating a Text Base for Corporate Performance Analysis." I have examined the final copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Business Administration.

John D. Johnson, Major Professor

We have read this dissertation
and recommend its acceptance:

Accepted for the Council:

Dean of the Graduate School

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

DEDICATION

This dissertation is dedicated to my parents

Mr. Su and Mrs. Fan-Zun Hwang

黃甦　黃凡容

and my wife

Mrs. Jade Wang

王良美

who have faithfully supported my academic career.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

ACKNOWLEDGEMENTS

I would like to thank my major professor, Dr. John Johnson, for his trust, guidance and support. I would also like to thank the other committee members, Dr. Sumali Conlon, Dr. Brian Reithel, and Dr. Morris Stocks, for their valuable comments and assistance.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

ABSTRACT

Quantitative and qualitative information are often complimentary in decision making. While quantitative information processing for decision support has been increasingly automated, the provision of computer integrated qualitative information, mostly represented in texts, seems to lag behind. The information base used for computer decision support can be expanded from the traditional data base, model base, and knowledge base to include a text base. Thus, decision support systems (DSS) can be designed to allow large collections of text to be organized, stored, retrieved, and integrated with quantitative information to provide better decision support. The purpose of this dissertation is to formulate a conceptual DSS design that incorporates a text base, and based on such a design, to develop a workable and testable prototype DSS in a selected application domain, corporate performance analysis. A graphic-oriented, window-operated DSS prototype was developed. This prototype allows a decision process consisting of such tasks as ratio selection, quantitative data retrieval for the ratio, ratio execution, and textual information retrieval to support the ratio analysis to be integrated and supported. Using an intelligent agent (IA) as an interface, the system is also capable of flexibly interacting with users in model construction, automatically determining the location of the financial data needed for ratio execution, and intelligently deriving search keys to retrieve relevant textual information.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

TABLE OF CONTENTS

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

LIST OF TABLES

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

LIST OF FIGURES

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

FIGURE PAGE

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

CHAPTER I

INTRODUCTION

Problem Statement

Decision making has long been recognized as the central activity in organizations (Huber and McDaniel, 1984; Simon, 1973). The provision of information support for decision makers, therefore, has been one of the major concerns for research and development of decision support systems (DSS). Unfortunately, traditional computer decision support tends to focus on quantitative information processing, and thus often neglects the processing of text that nevertheless contains essential qualitative information needed in many decision tasks.

Quantitative and qualitative information are often complimentary in decision making. Evidence shows that the integration of quantitative and qualitative information can increase the efficiency and effectiveness of many decision tasks (Chugh and Meador, 1984; Covaleski et al., 1991; Rappaport, 1987). The human decision maker often bases his/her judgements on qualitative information that is generally expressed in natural language and recorded semantically in texts. In making an investment decision, for example, a successful financial analyst not only observes numbers shown on financial statements, cash flow schedules, or economic index summaries, but also, examines related textual information contained in corporate reports, trade journals, government bulletins, and business newspapers that are mainly descriptive and qualitative in nature.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

While quantitative information support has been successfully and increasingly automated for many years, the provision of computer integrated textual information to support decision making has tended to lag behind. As a result, the unbalanced information support would create a serious bottleneck in organizational decision making. With the advances of text processing technologies, this problem is compounded when the production of text, particularly in electronic forms, continues to grow.

A major challenge for management scientists and information systems researchers currently is to seek for alternative DSS design that can provide integrated quantitative and qualitative information support. Traditional DSS design uses an information base consisted of a data base, a model base, and in some cases, a knowledge base. The study proposed in this dissertation is motivated by the belief that such an information base used for computer decision support can be expanded to include a text base, and DSS can be designed in a way that allows textual information to be organized, stored, retrieved, and integrated with quantitative information to provide better decision support. Thus, the objective of this dissertation is to address two research questions:

(1) How can a text base be integrated into the traditional DSS design?

(2) How can this integrated DSS be built?

The first question can be answered by building a conceptual DSS design framework to integrate a text base. The second question can be answered by developing a workable and testable DSS prototype based on the conceptual design framework.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Significance of the Research

The primary significance of this dissertation research is that it explores the possibility of incorporating a text base with the traditional information base that only utilizes a data base, a model base, and in some cases, a knowledge base. By using a system development life cycle as the methodology, this dissertation also offers an opportunity to substantiate such an envisioned DSS design.

In the filed of Management Information Systems (MIS), innovation is the concept that leads its researchers into a wonderland that is full of amazement and achievements. The past three decades have witnessed just that. Although providing automated and integrated textual information in decision making is a difficult task that requires substantial works in design consolidation and technology integration, this dissertation intends to take the initial step toward such an adventure.

Structure of the Dissertation

The structure of the dissertation mirrors the stages of the system development life cycle as the chosen research methodology. This chapter identifies the research questions and formulates the research objectives. Chapter II reviews literature related to the design and development of a DSS that incorporates a text base. Subjects in the review includes the importance of integrating quantitative and qualitative information in decision making, the traditional information design paradigms, the evolution of the information base for computer decision support, the nature and structure of text base, and the current research and development of text-

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

based information systems. In Chapter III, the system development life cycle as a methodology in the MIS field is discussed.

Chapter IV presents a conceptual design framework for DSS that intends to incorporate a text base. Based on a definition of the decision process involving the use of integrated quantitative and qualitative information, the design framework consists of an integrated information base containing a data base, a model base, a knowledge base, and a text base, and an intelligent agent as the interface for integrated information processing and user interfacing. Since the proposed DSS is designed to be used in the domain of corporate performance analysis, Chapter V reviews important issues related to information support in such a domain. The chapter also explores the concepts and theoretical foundations for the construction of the domain knowledge base to be used in intelligent text retrieval.

The conceptual design framework proposed in Chapter IV and the domain analysis in Chapter V pave the way for the realization of the envisioned DSS, which can be accomplished in three steps. Chapter VI takes the first step to develop a conceptual and logical DSS design to be used as a blueprint for actual system implementation. Chapter VII take one step further to prototype the system by developing databases and programming. In Chapter VIII, the system is completed by conducting formal testing.

Chapter IX evaluates the prototype DSS. Valuable insights, propositions, and suggestions obtained from this system development life cycle are also documented. Finally, Chapter X concludes the dissertation with a summary.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

CHAPTER II

LITERATURE REVIEW

This Chapter reviews literature related to the design and development of a DSS that incorporates a text base. The first section discusses the importance of integrating quantitative and qualitative information in decision making. The second section examines traditional information design paradigms that ignore text processing. In the third section, the evolution of the information base for computer decision support is reviewed and its movement toward an integration of text base is discussed. Finally, the fouth section analyzes the nature and structure of text base and review current research and development of text-based information systems.

Quantitative and Qualitative Decision Making

Simon (1960) classified executive decisions into a continuum, with highly programmed (*structured*) decisions at one end and highly unprogrammed (*unstructured*) decisions at the other. Decisions are programmed if they are well-structured, repetitive and routine, and can be solved by standard procedures. Unprogrammed decisions are ill-structured, novel, and cannot be solved by cut-and-dried methods. As Simon contends, this problem classification is important because different techniques must be developed to solve different types of decision problems.

Quantitative methods with their quantitative information support have been successfully used as aids to solve structured problems for several decades (see Figure 2.1). Two major characteristics underlie the use of these methods. First, the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

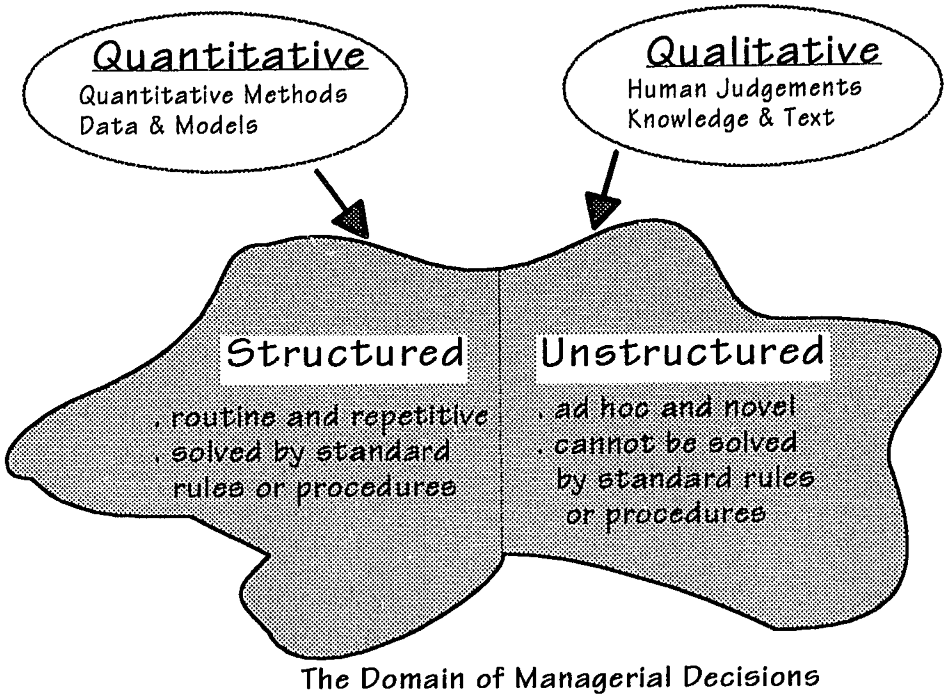


Figure 2.1. The Classification of Managerial Decisions, Information Support, and Decision Aids

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

problems being solved can be formalized and structured through certain mathematical or statistical models. Second, since these models are prespecified and quantitative, the data sources are mostly quantitative and from predictable sources. However, these prespecified quantitative methods alone are not sufficient to solve problems that are unstructured and cannot be modelled in purely quantitative terms. In the real world, the bulk of decision problems faced by managers are usually at least partly unstructured (Mintzberg et al., 1976; Remus and Kottemann, 1987). Although quantitative methods can be applied to model and structure the quantitative aspects of these problems, the ability to solve the problems as a whole depends strongly upon the decision maker's effectiveness in solving the remaining unstructured and non-quantifiable parts of these problems, which often requires support of qualitative information.

The processing of text to provide qualitative information support in growing in importance. Today, American businesses generate nearly 400 billion pages of documents a year, and the number is growing by 70 billion annually (Blair and Gordon, 1991; Mann, 1991). Because of advanced technologies, such as telecommunication, optical scanning, optical character recognition (OCR), voice recognition (VR), multimedia, and CD-ROM, textual information is growing not only in written form, but also in electronic forms such as e-mail, digital library, and on-line databases (see Figure 2.2.). This trend, for example, is reflected in the number of on-line databases in the electronic information services market, which has grown from 300 in 1979 to nearly 5,200 in 1993 (Hawkins, 1993). As Dewire

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

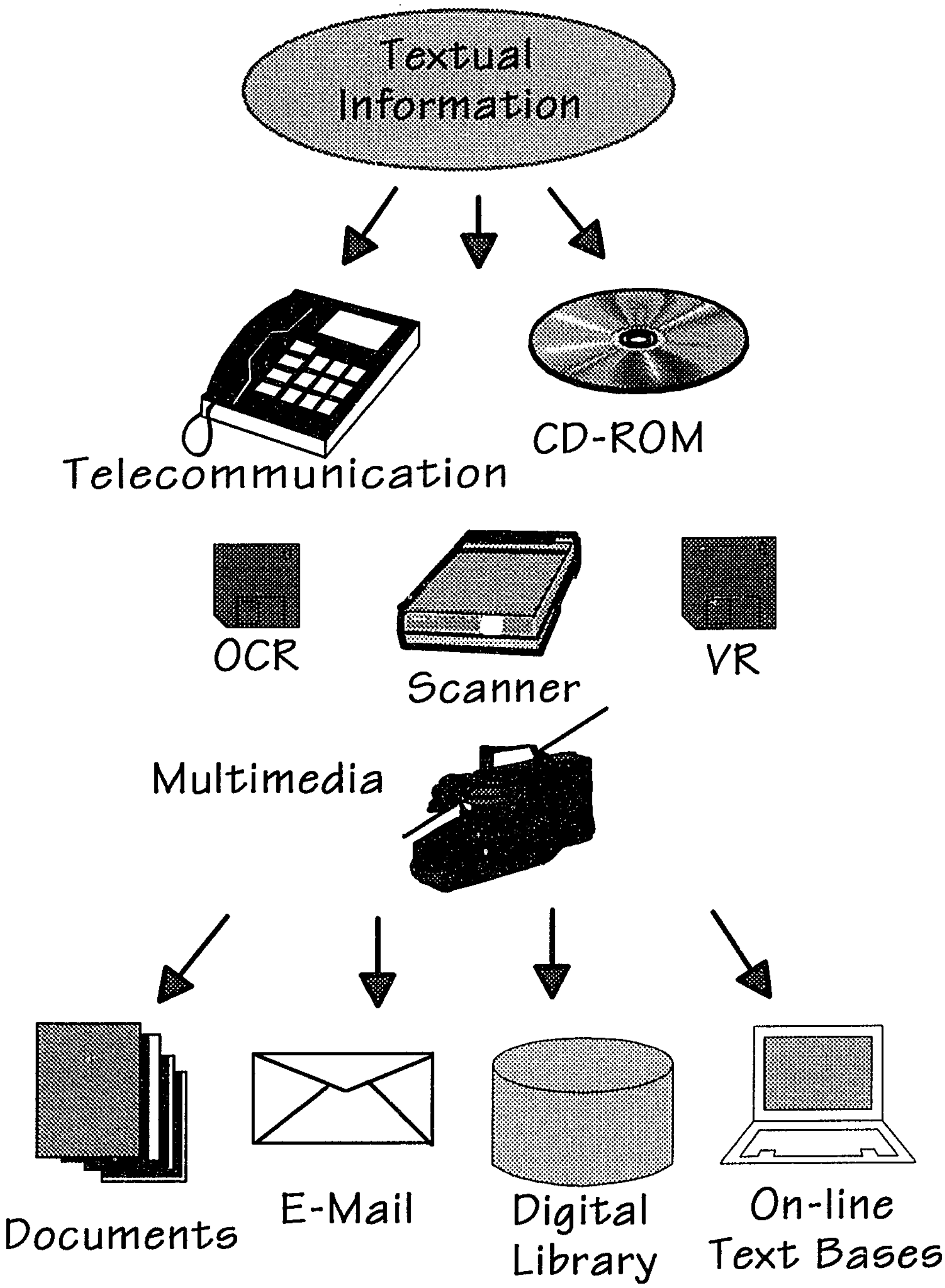


Figure 2.2. The Growth of Textual Information

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(1994, p. xxi) argues:

> "In the 1990s, the ability to efficiently store, index, categorize, search, and retrieve textual information ... will become as critical a success factor as the ability to efficiently manage data in 1980s."

Traditional Design Paradigms of Information Systems

With the advent of information technology, organizations in the post-industrial era have experienced significant restructuring of their business processes and decision activities. The "new" organizations have often been characterized as "information processing" (Daft and Lengel, 1986; Galbraith, 1974; Huber, 1984; Simon, 1973) or "decision making" institutions (Huber 1984; Huber and McDaniel, 1986; Simon, 1973). Business organizations have generally used two approaches to systems analysis and design in their development of Management Information Systems (MIS): the information processing approach and the decision-making approach. The *information processing* approach is concerned with the impact of the flow of information on the performance and behavior of organizational information processing. Using this approach, analysts map tasks into the blueprint of information system design in accordance with the analysis of adequate information flow between users, organizations, and computers. The focus of the *decision-making* approach shifts from information itself to the process of problem solving and decision making, on the belief that an information system is actually embedded with a decision making system. Using such an approach, analysts model information systems based on a

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

decision analysis in which flows of organizational decision making activities are identified and integrated.

Although these two information system design paradigms have been accepted and adopted for years, they tend to emphasize more formal information requirements (Barlow, 1988; Jackson, 1986), focus primarily on quantitative data (Agarwal et al., 1992; Huber, 1984; Mintzberg, 1975), and limit their usefulness as models of computer applications in organizational communication (Daft and Lengel, 1984; Jones and McLeod, 1986).

The information needed to support decision making in organizations is highly diversified. For example, according to *EDP Analyzer* (1979) a summary of corporate information needs includes:

(1) *Comfort information*: about the state of the business

(2) *Problem information*: about a crisis or an important project

(3) *Information for outside dissemination*: information disseminated to outsiders, like stockholders and auditors

(4) *External intelligence*: reports on the business environment and competition

(5) *Internal operations data*: figures indicating how things are going

(6) *Trigger information*: warning or alerting data

Obviously, much of the above information is informal and qualitative, and it often either exists outside the normal data processing sphere, or is of a transitory nature, and so, is difficult to provide with conventional MIS. By focusing on approaches defined in terms of formal information requirements and quantitative data, MIS

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

"operate in a secondary domain" and "often rest on a relatively well-established and formalizable problem space" (Flores et al., 1988, p. 155).

Since most informal and qualitative information is in a textual form, the social and linguistic aspects of the information requirements cannot be ignored. Some researchers (Flores et al., 1988; Lyytinen, 1985; Medina-Mora et al., 1993; Winograd, 1987/1988) argue that information systems can be thought of as technical systems with social or linguistic implications. Thus, information systems can be designed so as to allow text to be organized and stored in text bases, and to facilitate the retrieval of textual information to support organizational activities.

The Evolution of the Information Base for Computer Decision Support

The evolution of the information base for computer decision support has gone through three major phases: the data base phase, the model base phase, and the knowledge base phase. In the first phase, management information systems (MIS) built in the 1960s used a *data base* as their primary information resource. A data base is a repository of mainly numeric, fix-length, and transactional types of data, generated from the basic operations of the business. Through *data base management systems* (DBMS) and their query facilities, data can be retrieved, processed, and reported to aid decision making. Computer decision support in this phrase concentrated on providing quantitative information in solving structured problems.

In the 1970s, researchers began to consider the possibility of models as a valuable addition to computer decision support to supplement the data base (Gorry

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

and Scott-Morton, 1971; Keen and Scott-Morton, 1978). Consequently, in the second phase of the evolution of the information base, the data base was integrated with a *model base*, and decision support systems (DSS) were created (Blanning 1993; Bonczek et al., 1982; Keen and Scott-Morton 1978; Sprague,1982). Models are quantitative and can be viewed as algorithms, procedures, subroutines, programs and so on (Chang et al., 1992). In analogy with a DBMS, the model base is created, maintained, and manipulated by a *model base management system* (MBMS), in which sophisticated models can be constructed flexibly to allow users to perform a variety of simulation experiments, optimization tasks, etc., with different variables under different configurations.

Researchers in the field of Artificial Intelligence (AI) have been trying to fill the gap by incorporating the qualitative dimension of decision making and expert intelligence into information systems. These systems are created to deal with tasks that require knowledge, perceptions, reasoning, learning, understanding and other similar cognitive abilities, and thus are distinguished as "a qualitative expansion of computer capabilities" (Duda et al., 1979, p. 729), or simply as "intelligent systems" (Zahedi, 1993, p.12). Systems employing hand-crafted rules or other symbolic manipulations are ideal for providing qualitative information support; however, their primary emphasis is more on "thinking" or "judging" for decision makers instead of providing them with a more complete information base for decision making.

Beginning in the late 1980s, many DSS researchers started to think that DSS should act as a more knowledgeable or intelligent aid in human decision processes

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(e.g., Agarwal et al. 1992; Blanning 1987; Courtney et al. 1987; Dalah and Yadav 1992; Goul et al. 1986; Liu et al., 1990; Young 1990). They argued that, through the artificial intelligence techniques, various aspects of the computer decision processes, such as data retrieval, problem modelling, and user interfacing can be more effectively and intelligently facilitated. Therefore, in the third phase of the evolution of the information base, an expert system (ES) incorporating a *knowledge base* has been proposed as a an additional component of the DSS.

This may be done in two ways (Turban and Watkins, 1986). The ES can be incorporated as a separate component of the DSS, or the ES can be used to facilitate the user's interaction with the other components of the DSS, such as data base, model base, or the user interface. In the discussion below, the second arrangement will be of primary interest. That is, the dissertation will focus primarily on a DSS design in which the knowledge base is used only to facilitate the user's interaction with the other DSS components. This new DSS type, incorporating a knowledge base, has been labeled the "knowledge-based DSS," "intelligent DSS," or "expert DSS."

A knowledge base consists of facts, concepts, theories, heuristics, and other qualitative and symbolic knowledge organized and analyzed to make them useful in problem solving. Through hand-crafted rules or other symbolic manipulations, the KB is able to support certain qualitative aspects of human decision making, such as perception, reasoning, learning, and understanding. However, knowledge-based systems are usually designed to support decision making in very specific, stable

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

domains, and their "intelligence" is often limited by the bottleneck of knowledge engineering and the abilities of the knowledge engineer (Coat, 1988). The knowledge base, therefore, is not adequate for storing very large quantities of qualitative information that can be easily created, maintained, and retrieved by the decision makers.

Until now, the evolution of computer decision support has primarily involved an expansion of the computer's capability to process quantitative data through increasingly sophisticated quantitative modeling. Although the intelligent DSS has incorporated a knowledge to capture and process the qualitative expert judgments and insights of human decision processes, computer decision support still relies heavily upon quantitative modelling. As discussed in precious section, quantitative models, no matter how sophisticated, are not sufficient to completely represent or solve decision problems that have unstructured and non-quantifiable components. This unbalanced evolution of the information base therefore has serious implications for the changing nature of organizational decision making. As sophisticated quantitative information processing becomes increasingly automated, the organization is able to take on more problems and/or problems of greater complexity. To the extent that these additional problems have qualitative as well as quantitative components, the organization will find itself facing the dilemma that the rapid growth of uncontrolled and unprocessed texts can easily create a major bottleneck in solving problems that have unstructured and non-quantifiable components and require both quantitative and qualitative information support.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

This dissertation, therefore, suggests that the information base for computer decision support should be expanded to integrate a *text base* as it evolves into the fourth phase of the integrated and intelligent DSS (see Figure 2.3.). Each of the four data resources (i.e., data base, model base, knowledge, text base) will then play a distinct, but complimentary role in supporting decision problems which need both quantitative and qualitative information support. As illustrated in Figure 2.4., the data base provides quantitative data needed as the input for quantitative models, and the model base provides quantitative models needed for sophisticated quantitative information processing. Together, the data base and model base provide quantitative information support through the quantitative modelling and structuring of the decision problems. The knowledge base, as used in the intelligent DSS, assists the user in the qualitative reasoning process, though only over a very restricted domain. The text base, on the other hand, is able to provide qualitative information support over much wider domains, precisely because it does not attempt to substitute for the user's own reasoning process, but instead provides the user with efficient and effective access to large bodies of textual information which allow the user to do his or her own thinking more effectively.

Finally, the incorporation of a text base into the DSS increases the complexity of the DSS, and so increases the importance of an intelligent DSS interface. Thus, the new role for the knowledge base in running the intelligent DSS becomes more important after a text base is introduced.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

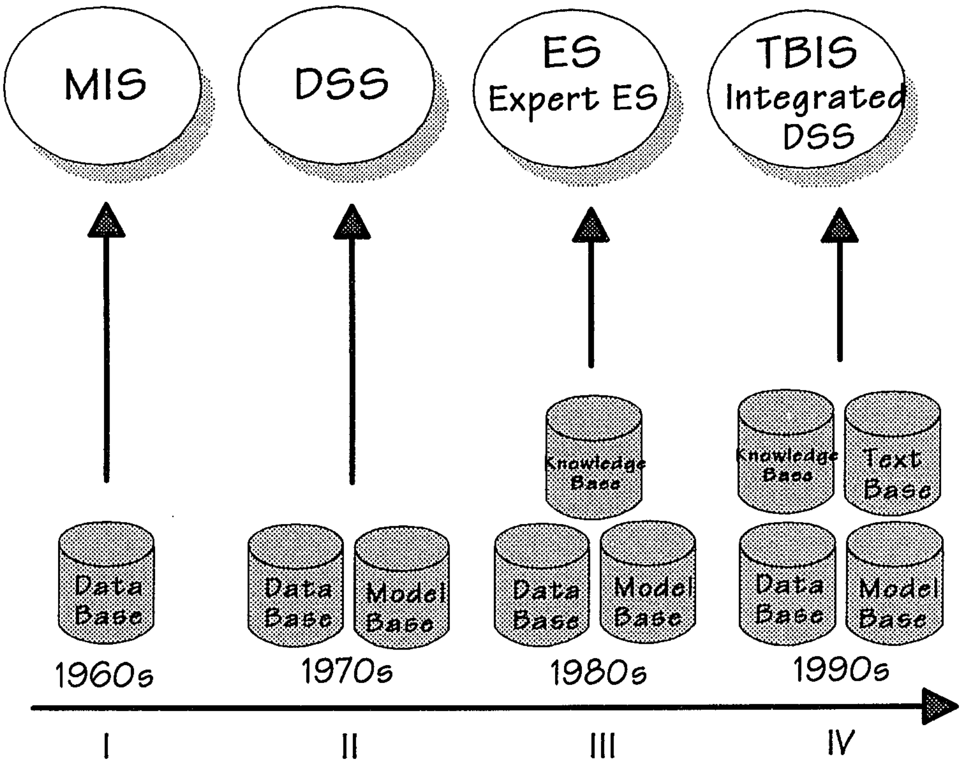


Figure 2.3. The Evolution of the Information Base for Computer Information Support

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Information Type / Type of Information Processing	Quantitative	Qualitative
Storage and Retrieval of Large Quantities of Information	Data Base	Text Base
Sophisticated Processing of Information	Model Base	Knowledge Base

Figure 2.4. The Role of the Four Data Resources in Computer Information Support

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Text Base and Text-based Intelligent Systems (TBIS)

A *text base* is a collection of text that has been converted into a "structure," as dictated by the software being used (Dewire, 1994). Numeric or alpha-numeric data may be contained in text, but this data is treated as character strings for text processing purpose. Based on its source of origin, a text base can be internal or external. A text base is *internal* if it is created by the organization itself. Documents such as business plans and manuals, corporate memos and correspondence, management proposals and reports, as well as meeting minutes and notes are all valid sources for the internal text base. The primary use of an internal TB in organizational decision making is to provide textual information support in the areas of internal communications and management control. The *external* text base, on the other hand, is created by outside entities like Dow Jones, McGraw Hill, Compuserve, and so on. Since the external text base contains textual information concerning entities and conditions outside of an organization, it is mainly used to support corporate planning and control at a strategic level.

There are two main types of text base structure. A *bibliographic* text base is the type most similar to the card catalogues used in libraries. A set of bibliographic records is usually created to provide a surrogate for the documents in the collection. These records include a bibliographic citations and abstracts describing the contents of the documents. A *full* text base, in contrast, contains the bibliographic information and the complete texts of the documents, along with descriptors selected from the documents by some indexing method. The documents in the full text base,

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

though largely unstructured, constitute most of the textual information available today.

The rapid growth of textual information has caused two problems for the development and use of text base. The first is to efficiently organize and process large volumes of uncontrolled and unstructured text; the second is to retrieve textual information from such a large repository of text effectively. These problems can potentially be reduced with the help of advanced text processing systems. The development of such systems has made significant strides in the last few years.

Systems which combine advanced information retrieval (IR) and natural language processing (NLP) techniques are called text-based intelligent systems (TBIS) (Jacobs, 1992). These systems derive their power from processing large quantities of raw, unstructured text, in an intelligent manner. Generally, there are four types of TBIS: automated indexing and hypertext systems, intelligent IR systems, text extraction systems, and summarization and abstracting systems (Jacobs 1992). In this section, the dissertation briefly review recent research and development efforts devoted to each of these systems, and discuss how they can applied to create and maintain a text base and to retrieve text in the integrated and intelligent DSS design. Although the systems reviewed in this dissertation represent only a portion of the most advanced technologies in the field of text processing, they are significant because they are either functioning, commercial available systems, or systems which are believed to be reliable indicator of ongoing research trends and emerging commercial applications.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Automated indexing systems

In the last three decades, considerable effort have been devoted to the development of methods for computer indexing of text. The goal of this research is to develop ways to provide inexpensive, efficient, but consistent indexing of large TB. These efforts have led to a number of sophisticated automatic indexing systems, which support robust text structuring in a variety of application areas. Some of the outstanding systems developed include Burgin and Dillion's (1992) FASIT, based on recurssive transition networks; Turtle and Croft's (1990) Inquery, employing an inference network; Salton's (1971) Smart Retrieval System, using vector processing; and Ginsberg's (1993) WorldViews, utilizing a knowledge representation scheme. These systems are able to determine key terms and topics efficiently and effectively, thus making the organization of large collections of text a manageable and affordable task for the corporate management of ever-growing collections of document. Through effective indexing, unstructured text can be better organized to help decision makers find specific textual information for decision problems that have unstructured and qualitative components.

Hypertext Systems

Hypertext is a collection of text connected through ordered nodes and links. Because it allows access to text through a variety of nonlinear arrangement, hypertext provides a mechanism to organize and retrieve both structured and unstructured text efficiently and effectively. Hypertext systems are often developed as front-end

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

browsing systems or on-line help for faster information retrieval and better cognitive guidance (Campagnoni and Ehrlich 1989). Major hypertext systems include The New Oxford English Dictionary (Raymond and Tompa, 1988), gIBIS (Conklin and Bengman, 1989), KMS (Akscyn et al., 1988), NoteCards (Halasz, 1988), and WALT (Frisse and Cousins, 1992). In addition, research has shown the possibility of building hypertext automatically (Salton et al., 1994) and intelligently (Carmel et al., 1989) for very large collections of texts.

The integration of hypertext with DSS has generated increasing interest among researchers and practitioners in recent years. For example, Minch (1990) developed a comprehensive framework for integrating hypertext into DSS, Wolfe (1994) developed a hypertext-based group DSS for total quality control, and Bieber and Kimbrough (1992) demonstrated an implementation of a hypertext DSS for the U.S. Coast Guard. Hypertext technology can be best applied to organize such documents as minutes, memos, manuals, and procedures, which have structures that are basically hierarchically or chronologically oriented. For highly unstructured text, hypertext is a very useful tool to organize the text, and thereafter, allow the decision maker to browse relevant textual information efficiently and effectively. Furthermore, hypertext provides a method to link quantitative data with qualitative information. Thus, quantitative and transactional data can be indexed by date and linked by hypertext technology with textual documents that contain relevant historical information about important events.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Intelligent IR systems

This class of systems augment ordinary information retrieval capabilities through robust query processing, user modeling, reasoning, and inferencing. It uses advanced linguistic analyses and tools (e.g., electronic thesauri, knowledge bases, lexical semantic relations, etc.) to map user's conceptualizations and information needs. Some of the existing systems in this category include CODER (Fox, 1987) for the analysis and representation of heterogeneous documents, CANSEARCH (Pollitt, 1987) for cancer-therapy-related document retrieval from the MEDLINE database, EP-X (Smith et al., 1989) for retrieval from the environmental pollution literature, GRANT (Cohen and Kjeldsen, 1987) for search of funding agencies, I^3R (Croft & Thompson 1987) for search of information on computer science, and WIN (Pritchard-Schoch, 1993) for on-line retrieval of legal documents. Intelligent IR systems can be incorporated into the DSS design as a vehicle for efficient and effective retrieval of textual information.

Text extraction systems

These are natural language understanding systems that analyze unstructured text, extract key information, and transform it into intermediate forms such as fixed-field data structures in traditional data bases and frames or semantic networks in knowledge bases. The input text can be e-mail messages, memos, book chapters, or articles from the Wall Street Journal, and the output will be a summary of key information stored in a pre-arranged template. SCISOR (Jacob and Rau, 1990), for

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

example, is a system developed at the GE Research and Development Center to analyze and extract key descriptors from financial news stories concerning corporate mergers and acquisitions (e.g., merger target, merger type, deal amount, etc.). This system is attached to the Dow Jones on-line financial service and achieves 80-90 percent accuracy (combined recall and precision) and has been ported to other subject areas. Other text extraction systems in development include KERNEL (Palmer et al., 1993), PROTEUS (Grishman, 1989), and TACITUS (Hobbs et al., 1992).

The major use of this class of subsystems is to extract pre-specified types of information from unstructured text and convert this information into structured text. The extracted textual information can then be stored in a traditional data base, as a "textual" data base, for faster retrieval. Text extraction systems, however, are very subject-specific. Application domains with multiple subjects would require extensive work in terms of knowledge acquisition and engineering.

Summarization and abstracting systems

This class of systems generally select key sentences from the text, combine them, and ultimately summarize them into normal language in order to provide a summary of the subject matter of the document (see Paice, 1990 for more details). These systems would potentially allow large amounts of input text to be summarized in shorter forms (many pages from an article to a page or so), and so, reduce the time decision makers have to spend in reading irrelevant text. However, the technologies for automatic summary or abstraction construction have not reached a

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

level at which they would be of practical use (Paice 1990).

In summary, if we consider the above system types from a different perspective, we can see that there are a number of basic text-processing functions of TBIS technologies that can be incorporated in various combinations into the integrated and intelligent DSS design: *text organization*, *text retrieval, user interfacing*, and *text transformation* (see Figure 2.5.). Among these, the functions of text organization and text transformation support off-line text preparation and maintenance, and the functions of user interface and information retrieval support on-line text retrieval. Through TBIS technologies, text can be organized into a more "retrievable" form and can also be retrieved more "intelligently." A perfect DSS design will presumably employ all four types of TBIS technologies. However, at the present time, automatic indexing, hypertext, and intelligent information retrieval are the only types of technologies that are currently commercially available.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Text Processing Function / TBIS Technology	On-line Text Retrieval		Off-line Text Organization	
	User Interface	Text Retrieval	Text Structuring	Text Conversion
Automatic Indexing			X	
Hypertext	X	X	X	
Intelligent Information Retrieval	X	X	X	
Text Extraction				X
Text Summarization & Abstracting				X

Figure 2.5. TBIS for Text Processing

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

CHAPTER III

RESEARCH METHODOLOGY

Management Information Systems (MIS) is a newly-emerged and highly-interdisciplinary academic field. Just as it continues to struggle for identification, research methodologies in MIS have been subject to intensive investigations, modifications, and improvements in the last two decades (e.g., Benbasat, 1984, Dickson et al., 1979; Galliers and Land, 1987; Hamilton and Blake, 1982; Ives et al., 1980; Lucas, 1978; Nunmaker et al., 1991; Van Horn, 1973). Yet, there in no agreement on the legitimacy of the research aspect of MIS as an academic discipline (Nunamaker et. al., 1991).

Research is a "systematic, intensive study directed toward fuller scientific knowledge of the subject studied" (Blake, 1978) or a "systematic inquiry aimed at providing information to solve problems" (Emory and Cooper, 1991). The definition for research may vary in terminologies, but one thing is common: without a clear understanding of the research domain where problems need to be solved or subjects need to be studied, research questions may be inaccurately formulated, research methodologies may be inappropriate, and eventually, research conclusions may be misleading.

Based on one of the most widely-accepted definition, MIS is:

> "an integrated, user-machine system for providing information to support operations, management, analysis and decision-making functions in an organization. The system utilizes computer hardware and software; manual

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

procedures; models for analysis, planning, control and decision making; and a database" (Davis and Olson, 1985, p.6).

This definition implies that the existence of MIS is manifested through substantial extension of two bodies of concepts: social science (e.g., psychology, sociology, business administration, etc.) and engineering science (e.g., information science, computer science, system engineering, etc.). This interdisciplinary nature, therefore, suggests that research in the MIS domain should embrace a broad range of methodologies in both academic fields.

The goal of this dissertation is to examine how a text base can conceptually be integrated into traditional DSS design, and how this integrated DSS can be built. To accomplish these goals, a life cycle of systems development is adopted as the research methodology.

System development is a common research methodology found in engineering science, but unique to MIS within the business school environment. However, in the domain of MIS, systems development is a key research methodology that interacts with other methodologies, such as theory building, experimentation, and observation, to form an integrated and dynamic research program as shown in Figure 3.1. The advancement of MIS research and practice often comes from new systems concepts, but systems must be developed in order to test and measure the underlying concepts. While studies using theory building, experimentation, and observation serve as the impetus for the improvement of existing systems, these empirical studies become possible only when workable and testable production systems are available.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

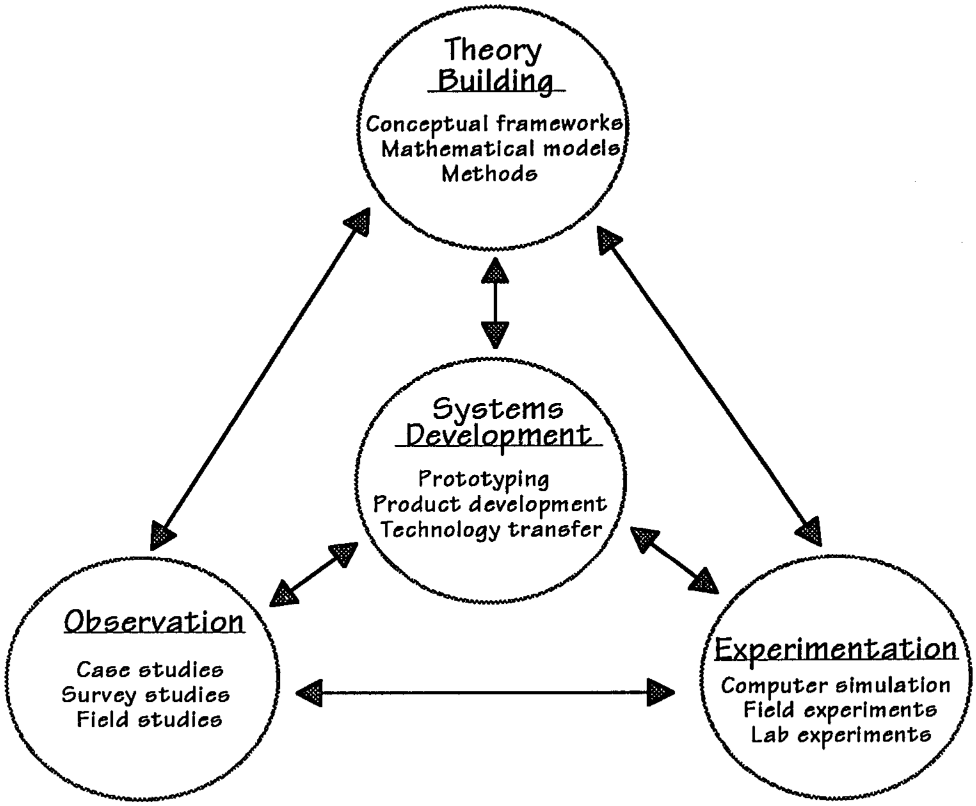


Figure 3.1. A Multimethodological Approach to IS Research

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The use of systems development as a research methodology is rooted in software engineering in which the progress of research development "is achieved primarily by posing problems and systematically following the design process to construct systems that solve them" (Denning, 1989, p.10). The developed system then serves as "an artifact that becomes the focus of expanded and continuing research" (Nunamaker et al., 1991). This research methodology is gaining acceptance in MIS community in two aspects. First, numerous doctoral dissertations in the MIS and MIS-related fields, such as artificial intelligence, expert system, and computer science, have used system development methodology to build prototypes to substantiate their conceptual works (for example, the work of Paradice (1986), Baldwin (1989), and Chang (1993). Secondly, many prestigious MIS journals, such as Communications of the ACM, Journal of Management Information Systems, Decision Support Systems, and so on, have published many articles using prototyping as a key component in their research.

The idea of the system development life cycle is that there is a well-defined process by which a system is designed, developed, and implemented. Stages in the system development life cycle are usually described differently in different implementation projects, or by different researchers. In this dissertation, the process of system development includes the following stages:

(1) formulate the problems;

(2) propose a conceptual design framework;

(3) analyze the application domain and knowledge construction;

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(3) develop a conceptual system design;

(4) prototype the system;

(5) test the system; and

(6) evaluate the system.

In fact, to a large degree, this process parallels the one employed in the social science which can be summarized as below:

(1) formulating the problems;

(2) state the hypotheses;

(3) build the research model;

(4) conduct the experimentation; and

(5) analyze and interpret the results.

Figure 3.2. graphically compares these two research processes.

Problem Formulation

A logical first step in any research is the formulation of one or more research problems to establish the research objectives. As stated at the end of the first section, this dissertation addresses two research issues: (1) how can a text base be integrated into the traditional DSS design? (2) how can this integrated DSS be built? The significance of these two research questions has been further supported by a review of related literature.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

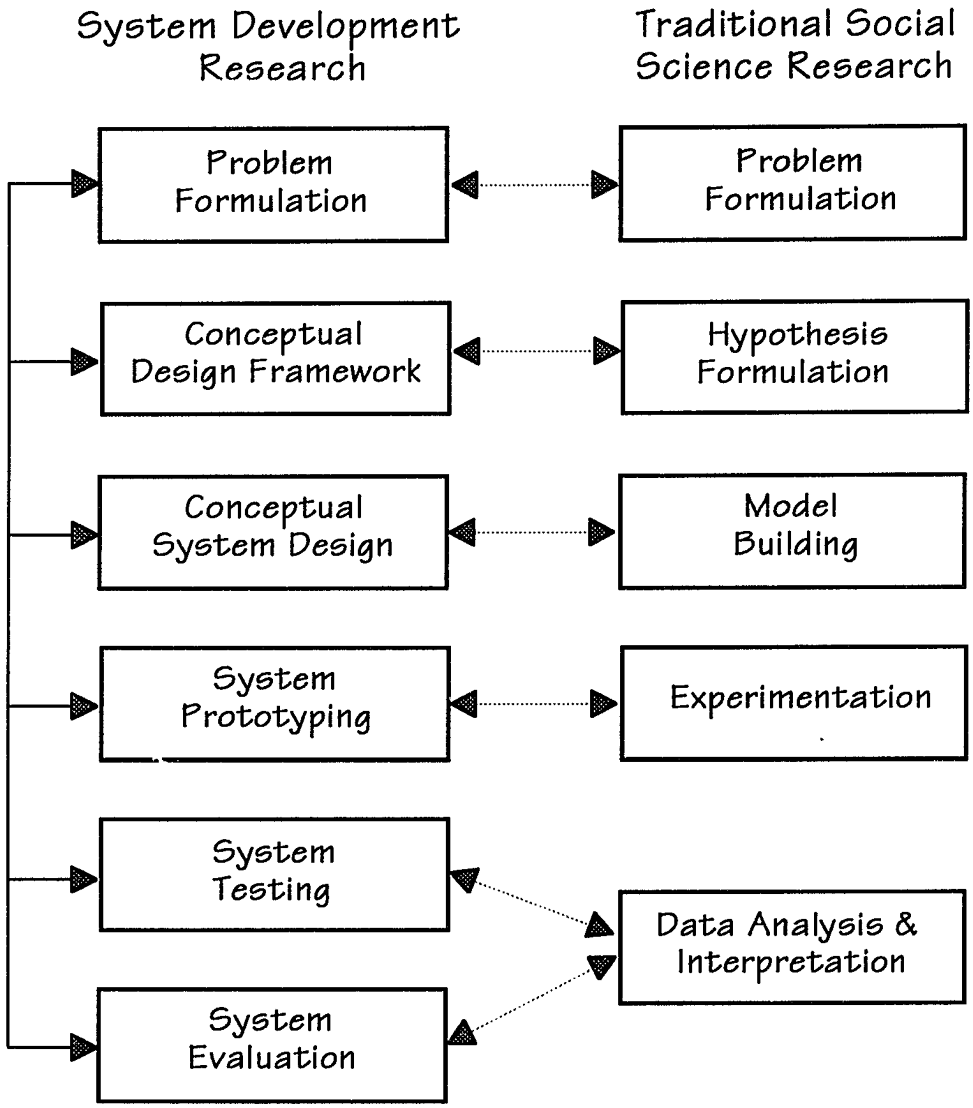


Figure 3.2. A Comparison of Research Process between System Development Methodology and Traditional Social Science Research Methodology

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Conceptual Design Framework

Following the formulation of research problems, a conceptual design framework needs to be developed to provide a high-level actualization of the system initially pictured in the research. Similar to the hypotheses formulation in social science research, this is a process of organizing ideas and suggesting actions.

At this stage, the dissertation will develop a unique architectural framework for the development of the proposed DSS. The architecture will be developed with flexibility, extensibility, and modularity. For flexibility and extensibility, alternatives to the actual implementation (or operationalization) of the DSS design will be explored with clear statements of their advantages and disadvantages. For modularity, the architecture will be presented in a high level of abstraction with definitions for the functionalities of system components and their interactions. The framework will then serves as a roadmap for the system building process.

Domain Analysis and Knowledge Construction

This step is unique to the development of systems that has knowledge processing capabilities. Unless a general problem solver is to be developed, a specific application domain needs to be selected and analyzed. General problem solvers, such as the ones proposed and developed by Simon (1960) and by Bonczek et al. (1979), are designed to assist human in solving domain-dependent problems, while most of the knowledge-based systems are designed to support decision making in a particular domain.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The selected domain in this research is corporate performance analysis. The goal of the domain analysis is to offer the developer an opportunity to understand the nature of the domain. Since a knowledge base is needed in the functioning of the proposed DSS, the understanding of the domain will support the works of knowledge construction, namely, knowledge representation, knowledge acquisition, and knowledge processing.

Conceptual Design

Generally, information system analysis and design can be carried out on three levels: the system requirements definition level, the functional specification level, and the programming level (Ross and Schoman, 1977; Senn, 1984). Based on the system requirements formulated in the previous steps, the proposed DSS can be carried out by formulating a conceptual design (functional specification) in this step, and prototyping the system (programming) in the next step. This step is analogous to the model building process in social science research where research models are built based on the formulated hypotheses.

Based on the proposed architectural framework and the analysis of the domain, the conceptual design of the proposed DSS is formulated by choosing ideal and applicable design alternatives. Structures and functionalities are then established accordingly. This design will serve as a blueprint with logical constructs that will be transformed into operational constructs and physical components by prototyping.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

System Prototyping

This step is concerned with the realization of the conceptual design suggested in the previous step. The goal is to develop a prototype system that "capture the essential features of a larger system," is intentionally incomplete," and "is subject to be modified, expanded, supplemented, or supplanted" (Naumann and Jenkins, 1982). Prototyping as one of the alternatives in the system development has been widely accepted and applied because of its explicit learning process for developers, strong involvement of end users, low cost, and fast developing time. Prototyping is particularly useful in testing the feasibility of an innovative system design. In essence, prototyping is similar to the experimentation process in a social science research where research models are operationalized and experimented.

Tasks in the prototyping include establishing a system development platform (hardware and software), preparing data, building databases, and developing programs. The process is actually a mini system development life cycle within the system development of the proposed system as a whole. In order to develop a workable and testable prototype, the process goes through the following four steps:

(1) identify design requirements;

(2) develop a working prototype;

(3) test the prototype; and

(4) revise and enhance the prototype.

The prototype will be subject to formal testing and evaluation once it reaches a satisfactory state.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

System Testing

The prototype is tested to find discrepancies between the system and the conceptual designs. This is done in two aspects: functionality validation and performance benchmarking. Functionality validation is a walkthrough of the functioning of the system through the use of examples. The goal is to validate whether individual functions of the original system design are logically interconnected and reliably implemented. The system is also tested to provide performance benchmarks that depict the capabilities of the system and provide advice for further improvement. Performance benchmarks used in this research include overall speed of the system and the precision and recall rate of text retrieval. Details will be discussed in Chapter 8.

The results of the system testing will provide valuable information for system evaluation of the next step. Together, these two steps are analogue to data analysis and interpretation in the social science research.

System Evaluation

System testing in the previous step paves the way for evaluating the effectiveness of the theorized concepts and the prototype system. Based on the functionality test results and system performance benchmarks, the evaluation will provide an overall appraisal of the system by using Shannon's (1975) six criteria for total system evaluation:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(1) The prototype should be goal or purpose directed.

(2) The prototype should be robust, in that it does not give absurd answers.

(3) The internal representation of the concepts should have "face validity" based on a priori knowledge, past research, and existing theory;

(4) The user interface should be appropriate. It should be easy for the user to control and manipulate;

(5) The prototype should be adaptive and flexible, having an easy procedure to modify and update representations; and

(6) The prototype must be complete on important issues.

These six criteria will also serve to validate whether the research questions in this research have been answered. Valuable insights, propositions, and suggestions will also be documented with the purpose to enrich knowledge and understanding of the subject under study.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

CHAPTER IV

THE CONCEPTUAL DESIGN FRAMEWORK

Decision Process

The purpose of a DSS is to support a decision maker in his or her decision-making process by providing relevant information at the appropriate time. In designing a DSS, therefore, one should first study and model the decision process to provide a basis for the implementation of the design.

A number of models describing human decision process have been developed in the literature (e.g., Huber, 1989; Mintzberg, 1976; Simon, 1960). The process-oriented model of decision support has also been studied by many DSS researchers (e.g., Adams et al., 1990; Keen and Scott Morton, 1978; Parker and Al-Utaibi, 1986; Sprague and Carlson, 1982; Weber, 1986). These models suggest that the decision process can be analyzed conceptually into a series of concise and separate phases. For example, Simon (1960) divides the decision process into three phases: intelligence, design, and choice. Although there is no general agreement on what phases should constitute a decision process, many DSS researchers believe that the analysis of the decision process into well-defined phases provides the DSS designer a better understanding of the nature of decision making, and allows DSS to be designed according to a model that represents the important abstractions of the decision process (e.g., Ariav and Ginzberg, 1985; Ayati, 1987; Courtney et al., 1987; Sprague and Carlson, 1980).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

In this paper, the goal is to design a DSS to support decisions requiring both quantitative and qualitative (textual) information. We begin by examining a computer decision process involved in this type of decision making. In general, the decision process in this situation might consist of four phases:

(1) *model construction or selection*: the process of constructing new models or selecting an appropriate, predefined and quantitative model for solving the decision problem;

(2) *quantitative data retrieval*: the retrieval of quantitative data needed for the model constructed or selected by the decision maker,

(3) *model execution*: the processing of the model after the quantitative data is retrieved,

(4) *qualitative (textual) information retrieval*: the retrieval of text that is relevant to the decision making but cannot be represented in the quantitative model being processed.

For instance, when a loan officer processes a loan application, he/she usually needs to evaluate the applicant's credit report to make an appropriate decision. A loan assistant then prepares such a report by choosing or building a credit rating model (quantitative), obtaining the applicant's credit history (quantitative) for the model, calculating the rating (quantitative), and listing the result with the applicant's other salient attributes (qualitative and textual) that cannot be represented in quantitative terms but are relevant to the decision making. Thus, in this type of decision making, the quantitative aspect of the decision problem can be formulated in

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

terms of some mathematical decision models specifying certain variables and their relationships by the DSS designer or the decision maker, and the decision maker is to solve the problem by building or choosing the right model(s), retrieving data (quantitative) needed for the model(s), executing the model(s), and retrieving relevant textual (qualitative) information to be integrated with the quantitative result from the model execution. The aim of the DSS design proposed in this dissertation is to support such a decision process.

System Architecture and Design Principles

The above decision process model suggests two essential design principles for the structure of the proposed DSS. First, the DSS should operate on an integrated information base that contains models (and/or modelling capabilities) representing the problem domain, quantitative data needed for the model execution, texts stored for retrieval, and knowledge required to retrieve relevant text for integrated qualitative information support. Second, an intelligent interface should be incorporated into the design to coordinate the decision supports from multiple data resources and the interactions between the system and the users. A conceptual design framework of an integrated DSS is illustrated in figure 4.1.

The architecture of this DSS design is composed of three groups of functional components: (1) four data resources (i.e., data base, model base, knowledge base, and text base) as the comprehensive information base, (2) three data management systems (DBMS, MBMS, TBIS) as the local control systems, and (3) a knowledge-

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

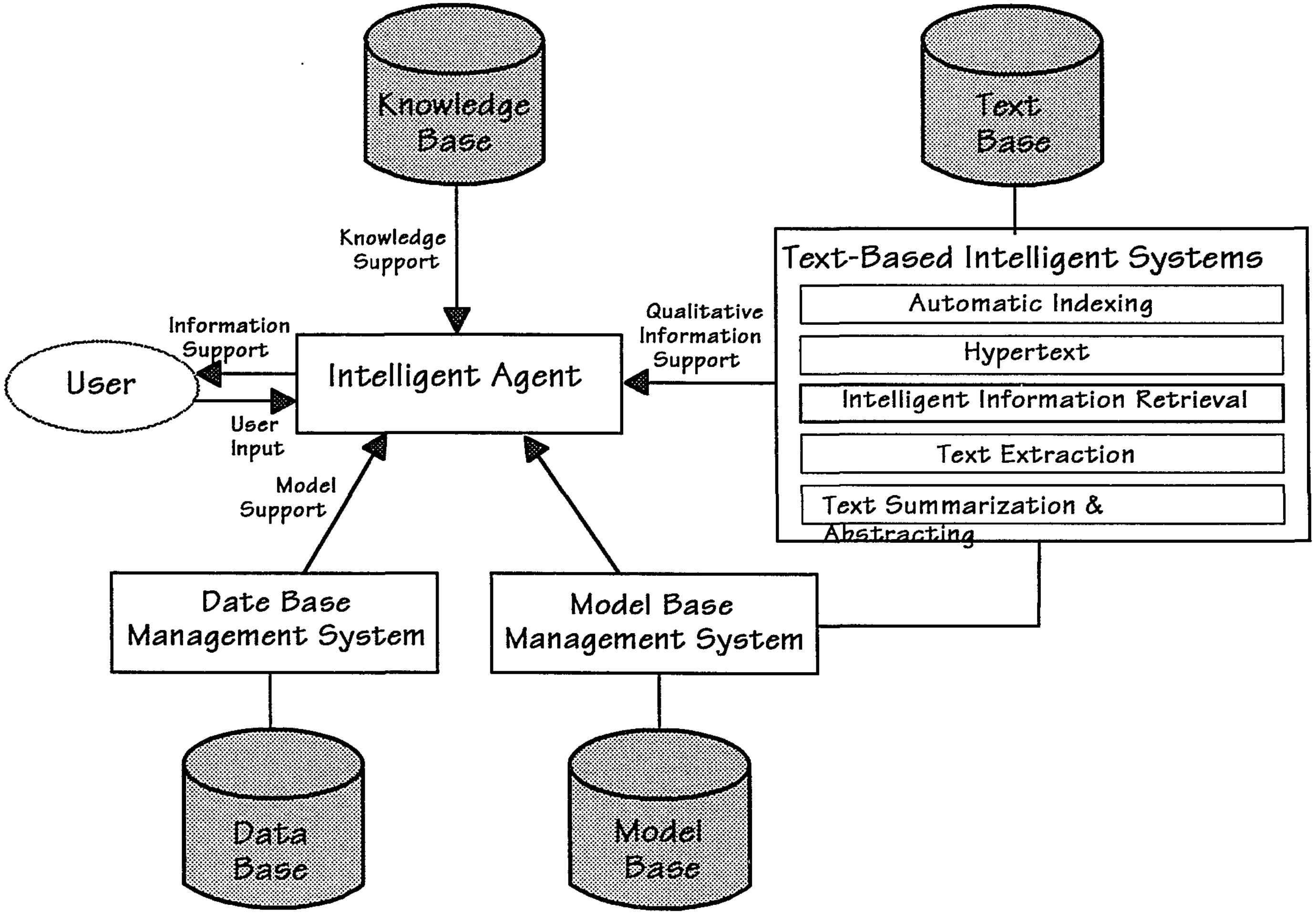


Figure 4.1. A Conceptual Design Framework for the Proposed DSS

based intelligent agent (IA) as the interface for integrated information processing and user interfacing. From a different perspective, the design is also composed of four modular components: (1) a data base module (data base and DBMS), (2) a model base module (model base and MBMS), (3) a text base module (text base and TBIS), and (4) an interface module (knowledge base and IA). These DSS components are structured and interconnected in a way that allows the system to provide quantitative data support from the data base module, model support from the model base module, qualitative information support from the text base module, and knowledge support from the interface module. In other words, through the functioning of IA, the system allows the tasks of model construction or selection, quantitative data retrieval, model execution, and qualitative information retrieval to be integrated and intelligently supported.

The Intelligent Agent

The interface is one of the most important elements in a DSS, since it manages the interactions between the users and the other system components. In the DSS design discussed here, the interface must be able to facilitate interactions involving the retrieval, processing, and integration of qualitative, as well as quantitative information. Three types of interactions can be identified in the decision process: the user-model interaction (model selection and execution), the model-data interaction (quantitative data retrieval), and the model-text interaction (qualitative information retrieval related to previous quantitative modelling). The interface in

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

this proposed DSS design framework should provide the specialized functionality necessary to handle these interactions efficiently, effectively, and intelligently.

The interface proposed in this framework is called an *Intelligent Agent* (IA). This is a knowledge-based interface system which (1) flexibly interacts with users during model selection (the user-model interaction), (2) automatically determines the location of the data needed for model execution (the model-data interaction), and (3) intelligently derives search keys for textual information relevant to the model (the model-text interaction).

There has been a somewhat diverse set of definitions of the "intelligent agent" concept in the artificial intelligence research community (Riecken, 1994). In this paper, however, an interface will be considered "intelligent" if it incorporates human reasoning capabilities and assumes a relatively large share of the burden of information retrieval, processing, and integration in the decision process. The structure of the IA is graphically illustrated in Figure 4.2, and its functions are discussed in the following subsections:

Flexible Model Construction

The type of dialogue which the interface permits between the user and the system depends on the sophistication of the interface and the level of services provided. In the most elementary dialogue form, the user must interact directly with the model base by guessing possible search terms and using the commands of the relevant artificial query languages. Consequently, the efficiency and effectiveness of

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

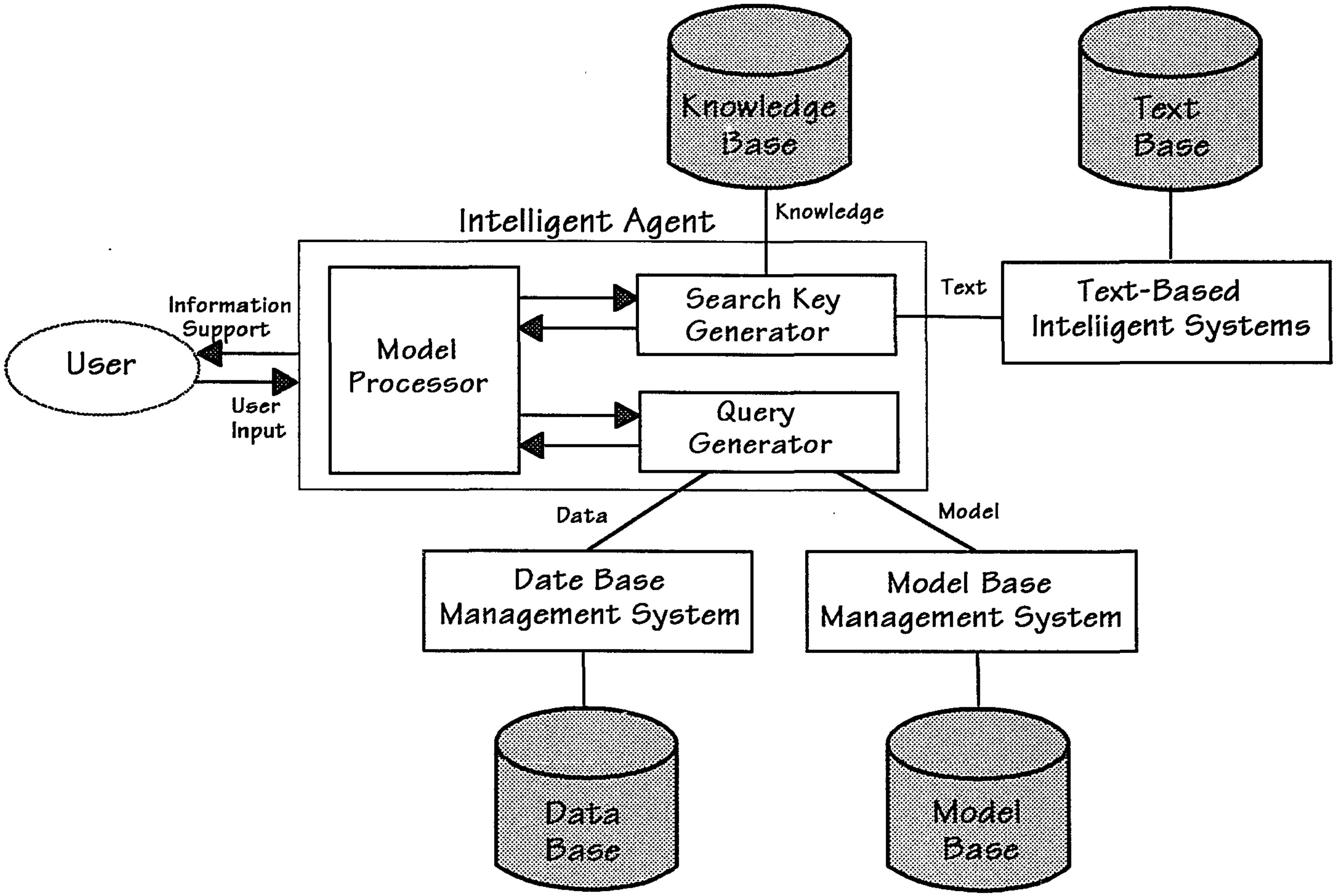


Figure 4.2. A Conceptual Design Framework for the Intelligent Agent

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

the model construction process is often hindered by the user' inability to derive valid search terms and to construct accurate queries. Dialogue forms of the next level would be menu-driven and operate in a semi-automatic fashion. Thus, model components can be pre-defined and stored in the model base, and models can then be constructed by choosing appropriate model components presented in a menu. Menu-driven interfaces are user-friendly and easy to implemented. So far, menu-driven interface is the most common interface type employed in today's information system design.

In a IA design, the most flexible dialogue form would be in natural language as facilitated by an natural language interface (NLI). Today, many NLI systems have been developed successfully as the front-end to retrieve data from data base or text base (e.g., Jacobs and Rau, 1990; Pritchard-Schoch, 1993; Samstag-Schnock and Meadow, 1993; Vickery and Brooks, 1987; Wu and Dilt, 1992). NLI systems to a data base generally incorporate a *parser*, a *lexicon*, a *semantic interpreter*, and a *domain knowledge base*. These systems work as follows. First, using the lexicon, the parser transforms the natural language input sentences into structured forms called parse trees. Using a knowledge base, the semantic interpreter then translates the parse trees into elements represented by formal logic or Prolog representation. Finally, the query generator converts these elements into internal data queries understood by the data base to retrieve data.

Models can also be retrieved based on natural language requests by analyzing user queries into model elements and data elements (Blanning, 1984). For example,

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

a query like

What is the break-even raw material price when the sale price is $7?

can be translated as a request for using the appropriate "break-even analysis" model in the MB, and inputing "$7" as the value of the "sale price" variable in the model. More generally, the model can be identified in a query statement by using a semantic model dictionary that contains the semantic attributes of models collected in the model base (Conlon et al., 1994). Consider, for example, a query like:

What would happen to labor costs if we raised wages by $.50 per hour?

By using the directory, the system will interpret the key terms "labor cost" and "wage" as directing the query to a financial model in the model base, if one exists. The query translation process can also be facilitated by storing many frequently-used parsable sentence structures in a dictionary to match the incoming queries.

Automatic data retrieval

The second function of the IA is to interface formulated models and desired data for subsequent model execution (model-data interaction). The purpose of this function is to reduce the burden on the user of retrieving data for model execution. There are a number of methods that can be used to provide the system with automated data access to multiple or heterogeneous data sources. For example, an universal query language can be employed to ease the query writing task (Kirshnamurthy et al., 1988), an intermediate language can be used for internal query translation (Templeton et al., 1987) and a metadatabase (Hsu et al., 1991) or a

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

domain knowledge base (Wu and Dilt, 1992) can be utilized for intelligent query decomposition and dispatching.

For DSS applications that have multiple data sources but employ an uniform data model, say, a relational model as advocated by Blanning (1985) and Suh and Hinomoto (1989), the complexity of the interfacing can be substantially reduced by simply storing information about data locations in the model base or using a data dictionary for an easy look-up based on the specific user requests.

Intelligent search key derivation

Finally, the IA should be able to intelligently construct a list of search keys for text retrieval (model-text interaction). This can be accomplished by developing a knowledge-based search intermediary system to intelligently aid users in information retrieval (Shoval, 1985; Smith et al., 1989; Vickery and Brooks, 1987). By making use of a knowledge base representing objects (entities or concepts) and their relationships in a problem domain, the intermediary system emulates the work of a human search expert in defining or refining an information searcher's search space and deducing appropriate search keys to retrieve relevant text.

Since the decision problems being solved in this DSS design are represented by quantitative models, the variables and their relationships (e.g., hierarchical connections, causal associations, etc.) in these models usually provide vital semantic implications for the derivation of search keys for retrieving relevant texts. For example, in the domain of batch manufacturing control, a semantic network can be

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

developed to store objects (or concepts) and their relationships such as "*products* RESULT_OF *production*," "*production* OPERATED_BY *schedules*," "*schedules* COMPOSED_OF *production batches*," "*production batches* USE *materials*." These objects are themselves quantitative constructs used in many batch control models, such as batch turnover ratios, raw material usage efficiency, and so on.

There are two processing elements involved in the derivation of search keys: a semantic network and a search key generator. The *semantic network* is a knowledge base containing objects (entities or concepts) and their relationships in a problem domain. For example, in the problem domain of financial ratio analysis, financial objects or concepts and their causal relationships can be identified and stored in a semantic network. The *search key generator* is the inference engine that draws inferences by tracing appropriate links in the semantic network. Using these two elements, the construction of a search key list can be accomplished by the following four major steps:

(1) identifying the value(s) of the qualitative determinant(s) associated with the quantitative model or model variable(s) and listing these value(s) as the primary search key(s) or triggers,

(2) constructing a search space (i.e., a contextual base) for the exploration of supplemental search keys by:

a. matching object(s) in the semantic network that semantically represent the qualitative determinant(s), and based on the matched object(s)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

b. navigating the network to identify other objects that are semantically related.

In order not to draw the semantic relationships out too far, special rule like "constrained spreading activation" (Cohen and Kjeldsen, 1987) can be applied. Finally, the search key generator forms text queries based on these search keys to retrieve relevant text from the text base.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

CHAPTER V

THE DOMAIN ANALYSIS AND KNOWLEDGE CONSTRUCTION

This dissertation intends to develop a prototype DSS to be used in the domain of corporate performance analysis. Similar to evaluating the health of a patient, corporate performance analysis involves a process of diagnosing the state of the health of the underlying company, and the effectiveness of such a diagnosis process relies on support of relevant and timely information. In this domain, the use of quantitative method such as financial ratio analysis is common. However, bank creditors, corporate mergers, management accountants, financial analysts, individual investors, and other organizational troubleshooters must open their analyses not only for quantitative constructs in financial statements, but also for many qualitative measures, such as marketing strategies, product development, and so on, contained in business surveys, industry reports, trade journals, and newspapers. This chapter addresses important issues related to information support in this domain. First, the chapter discusses the nature of ratio analysis and its limitations in providing relevant and timely information for corporate performance. Second, the chapter then examines the weaknesses of the traditional financial reporting system that provides a narrow range of relevant information, mostly quantitative, for ratio analysis. In light of the examination, the chapter reviews the event-driven accounting approach and the REA modeling technique that serve as an alternative to accommodate a broader spectrum of management information needs. Finally, the chapter presents a REA-based knowledge construction methodology for the development of a domain

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

knowledge base that can be used to provide intelligent text retrieval.

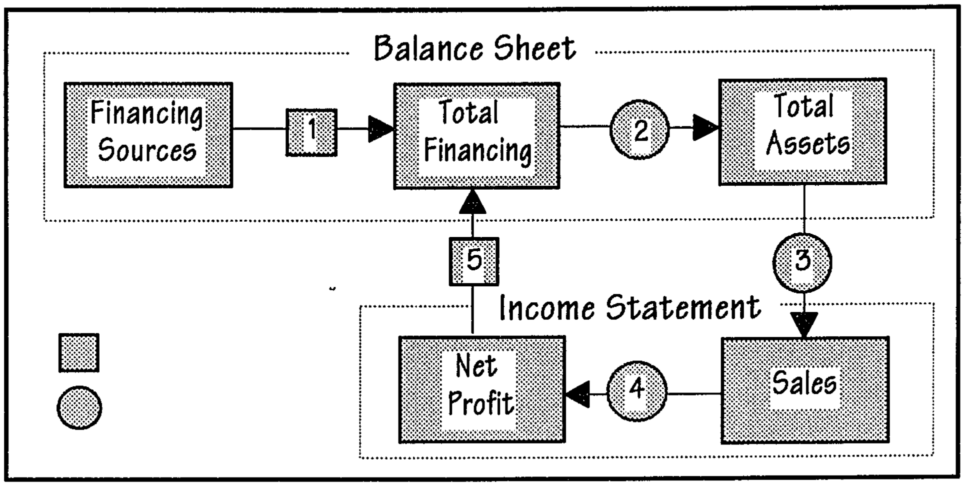
Ratio Analysis

Financial ratio analysis is frequently used as the starting point in the analysis of corporate performance. The objective of ratios analysis is to measure corporate performance in the past and determine what can be done to improve it in the future.

Ratios can be classified into different types or families based on their nature and purpose. For example, ratios can be classified into such types as liquidity, activity, leverage, profitability, and market value (Aragon, 1989), or liquidity, activity, profitability, and coverage (Kieso and Weygandt, 1983). However, a literature review on textbooks of financial management indicates that Morrow's (1991) families of financial ratios are the most comprehensive and structured. In this classification scheme, financial ratios are grouped according to their data source in the financial statements and place in the sequence of business functions that make up a complete business operating cycle. As shown in Figure 5.1, these five types of ratios can be used to evaluate corporate performance in different stages of the business operating cycle. Morrow (1991, p.410) defines these ratios as the following:

(1) *Financing structure ratios*. These ratios reveal the financing sources which were mobilized by a firm. The ratios show how the total funding of the firm is proportioned to the stockholders' investments of equity, bank loans, and trade or suppliers' credit. This proportioning or financing mix influences

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.



	Business Function	Type of Ratio	Type of Statement
1	Mobilization of financing	Financing structure	Balance sheet
2	Investment of capital	Asset allocation	Balance sheet
3	Generation of sales	Asset utilization	Interstatement
4	Control of operations	Sales profitability	Income statement
5	Determination of attractiveness of investment	Return on investment	Interstatement

Figure 5.1. Types of Ratios and Their Relationships

Source: Morrow, V. *Handbook of Financial Analysis for Corporate Managers*, Prentice Hall, Englewood Cliffs, NJ, 1991

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

as a whole.

(2) *Asset allocation ratios*. These ratios show how the total financing as mobilized is distributed to the different operating assets as inventories and receivables, and nonoperating assets as buildings under construction and unutilized factories.

(3) *Asset utilization ratios*. These indicate the extent to which the resources were used to generate and support sales.

(4) *Sales profitability ratios*. These reveal the degree of operating efficiency as indicated by the level of the costs of generating and servicing the sales in relation to the sales.

(5) *Return on investment ratios*. These summarize the overall efficiency of the operations and the forms of financing. These show how much the owners get from their investment.

Morrow's families of financial ratios will be used as a guide to ratio construction in the proposed DSS prototype.

A ratio is a numerical relationship between two figures derived from the financial statements. Because ratios can be precisely computed, they have been praised for their reliability and significance. However, ratio analysis has three major limitations when used in corporate performance analysis:

First, periodic financial statements do not reflect the most current corporate performance, as the traditional value-driven financial reporting system provides management information that emphasizes "short-term performance" (Covaleski et al.,

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

1991), and is "too delayed to be relevant" (Johnson and Kaplan, 1989). This argument will be further explained in the next section.

Secondly, based solely on financial statements, financial ratio analysis tends to ignore the importance of qualitative measures such as economic developments, market trends, corporate strategies, product evaluation, etc., Studies indicate that qualitative information can be used to improve the accuracy of bankruptcy prediction (Houghtton and Woodliff, 1987; Laitinen, 1993; Nagy and Obenberger, 1994; Peel et al., 1986; Tennyson et al., 1990), stock evaluation (Chugh and Meador, 1984; Rappaport, 1987), or corporate performance forecast (Barnes and Brown, 1981; Covaleski et al., 1991; Steele, 1982). For example, Tennyson et al. (1990) show that the prediction model consisted of both the financial ratios and the management narrative disclosures is obviously superior to the single dataset models.

Thirdly, based on quantitative modeling, ratio analysis tends to de-emphasize qualitative information processing aspect of human decision makers. As Newell and Simon (1972) argue, the efficiency and effectiveness of human information processing is largely restricted by the limitations of the human's short-term memory capacity. For this reason, Bouwman (1983) further indicates that in financial analysis decision makers tend to translate a series of figures into qualitative terms that generally require less memory capacity. Evidence shows that many investors evaluate stocks based heavily on qualitative criteria (Covaleski et al., 1991; Nagy and Obenberger, 1994).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Traditional Financial Reporting System

Ratio analysis is based on financial statements produced by traditional financial reporting systems that in turn is grounded on the value-driven accounting approach. The value-driven accounting approach starts with the design of a chart of accounts with which financial measurements of an organization's values of assets, liabilities, and equity can be identified, classified, recorded, and managed through times. In order to keep these accounts balanced, the double-entry bookkeeping system is used, based on the premise that in every financial transaction at least two of the occurrences must appear on opposite sides of the equation: assets = liabilities + equity (Sprague, 1972). The end result of the process is the creation of an organization's financial statements, such as balance sheet and income statement.

In this financial reporting system, accountants are responsible for identifying transactions that are needed as the system input by asking two questions:

(1) Will the transaction change the state of the financial statements?

(2) Can the effect of this transaction be measured financially?

As a result, an organization's financial transactions can only be recorded if the answers to the above questions are yes. This system, therefore, leads to a very rigid definition of a model of an organization's economics. The following summarizes the weaknesses associated with such a reporting system based on the arguments made by Denna et al. (1993), Lieberman and Whinston (1975), McCarthy (1979), Orman (1990), and Sorter (1972):

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(1) The organization's financial and nonfinancial data are largely set apart, because financial statements reduce most measurements to monetary terms.

(2) The semantic expressiveness of the recorded data is limited, because the data is only a subset of the descriptive characteristics about the business transactions.

(3) The chart of accounts does not provide a base broad enough to capture all necessary management information, because its classification scheme represents primarily the financial aspect of the organization.

(4) The financial reports cannot satisfy users with differing needs in the amount of details, because the aggregation level of stored data is arbitrary and too high.

(5) The financial reports do not provide timely support for decision makers, because accounting data is generally recorded subsequent to, rather than simultaneous with, the occurrence of the business transactions.

(6) This reporting system is largely separated from information systems operated by other business functional areas, because integration among these systems is costly and difficult.

Apparently, based on the conventional financial reporting system, ratio analysis does not provide sufficient, timely, and integrated quantitative and qualitative information support for corporate performance analysis. This dissertation, therefore, intends to develop a DSS prototype that will incorporate knowledge processing capability to integrate financial data (quantitative) and textual (qualitative)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

information to provide better information support for corporate performance analysis.

Event-driven Accounting Approach and REA Modeling

The development of alternative accounting model to accommodate a broader spectrum of management information needs has been a topic of continued research interest for many years. However, most of the research concentrates on integrating Sorter's (1969) "event" accounting theories which advocates that an enterprise's approach to corporate information management should focus on managing relevant business events as opposed to managing values shown on the financial reports. Instead of collecting predetermined input values and deriving prespecified financial reports, the event-driven accounting approach collects all relevant information about an organization's business events and allows users to generate their input values for their own decision models. The overall goal of the event-driven approach is to correct the weaknesses of the traditional value-driven approach.

For corporate database design, McCarthy (1979) later applied the event-driven accounting approach, Chen's (1976) Entity-Relationship data modeling methodology, and Date's (1977) database theory to develop an entity-relationship view of accounting model, called the REA accounting model. The development of this model is not limited by the principles of double-entry and monetary measurement, but instead allows more of the multidimensional and disaggregated aspects of business events to be captured. In this model, an accounting system is viewed in a database environment as a collection of (1) real world *entities* and (2) *relationships* among

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

those entities, and the underlying structure of the database consists of entities (or objects) representing REA, an acronym for economic resources, economic events, and economic agents, plus relationships among those objects (see Figure 5.2).

The development of the REA accounting model is a process of database abstraction as shown in Figure 5.3. The process involves the following procedures:

(1) Identify (a) the entity sets in the conceptual world and (b) the relationship sets among those entities;

(2) Construct an Entity-Relationship (E-R) diagram to capture the semantic nature of the identified relationships;

(3) Define and map the characteristics of entity and relationship sets that is of interest to a particular system user;

(4) Organize the results of steps 1, 2 and 3 into entity-relationship tables and identify a key for each entity-relationship set.

McCarthy's REA accounting model provides a reliable means for identifying business events, and its E-R diagraming serves as a semantic modeling technique to capture meaningful information concerning these events. This can be illustrated by a partial E-R diagram shown in Figure 5.4. In this diagram, there are four entities ("purchase," "vendor," "inventory," and "accounts payable") and three relationship sets ("supplier of," "payment for," "and "line item"). The diagram can be interpreted as (1) "vendor" as an economic agent is "supplier of" "Purchase" as an economic event, (2) "accounts payable" as an economic event is "payment for" "vendor" as an economic agent, and (3) "inventory" as an economic resource is "line

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

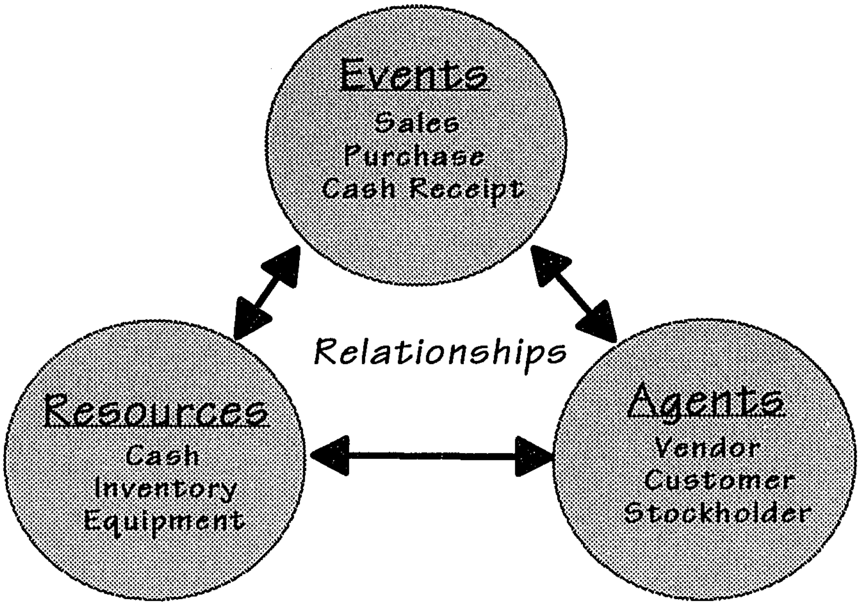


Figure 5.2. REA Objects and Relationships

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

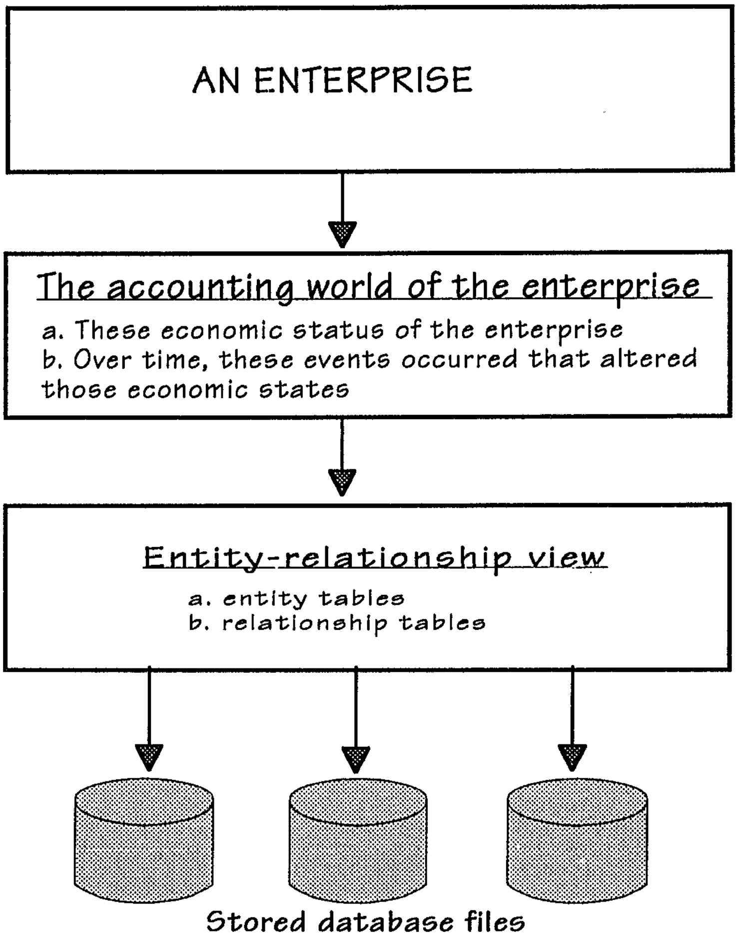


Figure 5.3. REA Model Abstraction Process

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

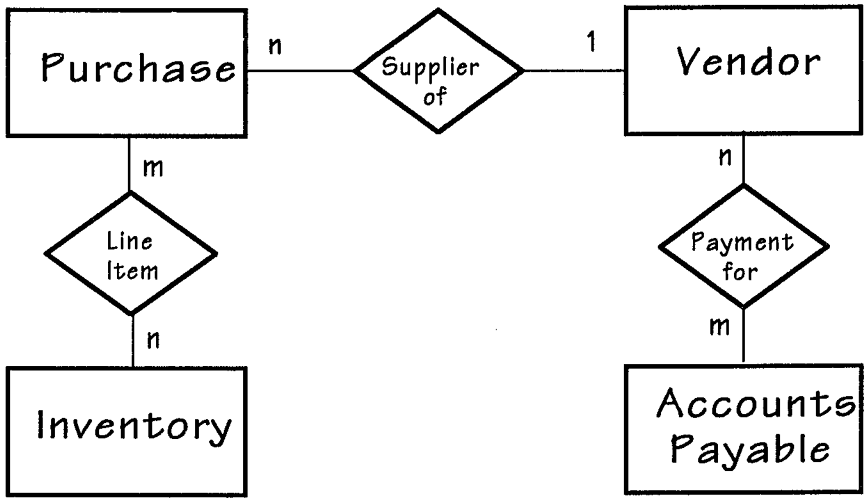


Figure 5.4. A Partial REA Model

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

item of "purchase" as an economic event. This diagram can eventually be converted into database structure to allow information about each entity and relationship to be capture and stored.

However, the semantic expressiveness of the REA model is still limited in its purpose for database construction which is essentially a process of aggregating diversified data. "Sales," for example, is an economic event that can be created as a single database, but much information about "sales," such as advertising, marketing, distributing, and so on, cannot be all captured by such a database or by creating related databases representing an unlimited number of relationship sets. To represent knowledge of corporate economics, a more sophisticated knowledge representation scheme is needed.

The concept of the REA modeling technique can serve as the foundation for the development of a semantic network to model knowledge of corporate economics. There are two reasons. First, because its representation of entity-relationships are exclusively in binary form, the concept of REA model can be used and expanded to develop a semantic network which also uses binary form to represent knowledge. Secondly, the idea of the REA model's representing corporate economics in three classes of economic objects (i.e., resources, events, and agents) and their relationships can also be used as a reliable and consistent knowledge representation scheme in the development of such a semantic network.

As discussed in Chapter 2, intelligent IR systems frequently use semantic networks to provide knowledge support for textual information retrieval. In the DSS

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

design proposed in this research, a semantic network capturing these three classes of economic objects and their relationships can serve as the knowledge base to assist the system in searching valid keys for retrieving integrated textual information to support corporate performance analysis.

Domain Knowledge Construction

This section further explores the concept of economic resources, economic events, and economic agents, and formalize the process of building a semantic network to represent the domain knowledge. The construction of a domain knowledge base requires an examination of three fundamental areas of knowledge work: knowledge representation, knowledge acquisition, and knowledge processing.

Knowledge Representation

Generally, there are two types of knowledge: surface knowledge and deep knowledge (Chandrasekaran and Mittal, 1983). Surface knowledge is the semantic relationship between objects without explicitly stating why the relationship exists. Deep knowledge, in contrast, explicitly describes the relationship. For example, surface knowledge is reflected in acknowledging that "sales" is the result of "marketing" activities, while acknowledging that "sales" is the result of "marketing" activities guided by "marketing strategies" and carried out by "market forces" reflects deep knowledge. Knowledge can also be declarative (factual) or procedural. Declarative knowledge captures what an expert know about the domain, while

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

procedural knowledge expresses the skills of how an expert actually solves problems. Intelligent IR systems typically use declarative and deep knowledge to retrieve relevant text from a large text base.

There are three commonly-used knowledge representation schemes: the rule-based, the frame-based, and the semantic network. Researchers generally agree that the intrinsic strength of the semantic network is its natural ability to represent declarative and deep knowledge (Bingi et al., 1993). The choice of using a semantic network to represent knowledge of corporate economics to support text retrieval is therefore considered as a natural one. A semantic network itself is a graph in which knowledge objects are the nodes connected by their declared relationships as the links. Thus, the development of a semantic network needs to define and identify knowledge objects and their relationships in the targeted domain.

In the semantic network representing knowledge of corporate economics, the three classes of economic objects are the knowledge objects. Among them, the economic events play a vital role in that

> "knowledge about relationships between events is a critical aspect of human knowledge. Knowing whether events are related, and how strongly they are related, enables individuals to explain the past, control the present, and predict the future" (Crocker, 1981, p. 272).

There exist many definitions for economic events. For Sorter (1969), an event is a business activity that has accounting significance. Johnson (1970, p. 643) later formally defines an event as "an action, occurrence, or happening that could be

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

described by one or more of an infinite number of properties, attributes, or characteristics." Yu (1976, p. 256), another accounting theorist, defines an event as "a class of phenomena which reflect changes in scarce means resulting from production, exchange, consumption and distribution." Finally, Denna et al. (1993, p.57), in their recent attempt to promote the use of the REA accounting model for business reengineering, defined a business event as "the essential organization activities that management wants to plan, control, and evaluate in order to achieve its business objects." In this dissertation, a new definition is considered unnecessary, rather all of the above definitions are considered plausible and used as the guidance for the understanding of business events.

The understanding of economic events provides a starting point for the definition of economic resources and agents. From the above definitions, it is obvious that the creation of economic events involves participation of humans (economic agents) and assumption of supplies (economic resources). Thus, in the making of economic events, economic resources are used by economic agents to pursue certain economic goals; and with those economic goals, economic agents are parties who have discretionary power to use or dispose of economic resources. A formal definition for economic resources, for example, is provided by Ijiri (1975) as the following:

"Resources can likewise be defined based on the goal of the entity and the concept of control. Namely, resources are objects that the entity intends to place under its control. This means that resources must have utility. However,

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

utility alone is not a sufficient reason for an entity to place an object under its control. The object must be scarce, thus ruling out free goods" (p.52). Moreover, economic resources can be tangible (e.g., cash, equipments, machines, etc.) or intangible (trade names, business goodwill, copyright, etc.).

The economic agents are equivalent to what Ijiri (1975) calls "entities." In Mattessich's (1977) terms, economic agents can be formally defined as "natural persons engaged in the economic activities (events) of producing, owning, managing, storing, transferring, lending, borrowing, and consuming economic objects (resources)" (p. 37, parentheses added). Economic agents can be individuals (accounts payable clerk, shipping manager, executive, etc.), groups (marketing department, board of directors, production cell, etc.), or organizations (vendor, business association, government agency, etc.).

Finally, the nominal constructs on financial statements can be treated as either economic resources or economic events. Elements of general ledger are generally classified as either balance sheet accounts, representing stocks of goods, services, and claims at a particular time, or income statement accounts, representing flows of the same items over a period of time. This dichotomy implies that stocks are economic resources under the control of the business enterprises, and the flows are economic events that make use of those stocks to meet the enterprises' goals.

Identified knowledge objects need to be related. In REA model building, two types of relationships are used: association and generalization. Association relationship connects objects that are related through causality. For example, the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

relationship "supplier of" is formed by connecting "purchase" and "vendor"; the relationship "payment for" is formed by connecting "vendor" and "accounts payable" (see Figure 5.3). Generalization is used to relate different subtypes or subsets of objects to a generalized type or superset. For example, the objects "cash," "inventory," and "equipment" are generalized to the object "asset."

In the semantic network, the relationship types, also called "semantic primitive," are theoretically infinite. In practice, however, one should define only a certain number of relationship types for the purpose of efficient knowledge processing. For example, Schank (1973) uses a reduced set of eleven semantic primitives (e.g., PTRAN, POSS_BY, MOVE, etc.) to represent the structure of natural language knowledge; Zarri's (1983) intelligent IR system RESEDA uses five semantic primitives (e.g., BE_AFFECTED_BY, BEHAVE, PRODUCE, etc.) to represent knowledge in several historical and socio-political fields. The reduction does not affect the semantic expressivity of the semantic network (Zarri, 1990), but enforces a consistent abstraction of the knowledge domain (Pau and Gianotti, 1990).

In the REA model, the association type of relationship can be expressed as a IS_CAUSED_BY semantic primitive, and the generalization type can be expressed as a IS_PART_OF semantic primitive. What the model is lacking of is a relationship type that can capture synonyms or things of the same nature. For example, a "salesman" can be called a "sales representative"; "revenue" is equivalent to "sales." The semantic primitive for this third type of relationship can be expressed as IS_A primitive. Together, these three fundamental semantic primitives are commonly used

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

in the development of many semantic networks, although they may be called differently by different people. For example, IS_CAUSE_BY is equivalent to BE_AFFECTED_BY, and IS_PART_OF is just another expression of IS_KIND_OF. Furthermore, these semantic primitives are all directional and have inverses. Thus, the inverse of IS_PART_OF is HAS_A, IS_A is HAS_INSTANCE, and IS_CAUSED_BY is AFFECT.

In the semantic network of corporate economics, the above three types of primitives will be used to represent relationships among identified knowledge objects (see Figure 5.5). Paired knowledge objects with the semantic primitives are then expressed in terms of semantic predicates. In terms of logic, the representation scheme of the semantic network can be viewed as a method to store binary propositions utilizing a set of predicates. These predicates can be written in Prolog statements as:

IS_A (knowledge object A, knowledge B),

IS_PART_OF (knowledge object A, knowledge B), and

IS_CAUSED_BY (knowledge object A, knowledge B)

Note that the purpose of using predicates is to represent knowledge in a simple form of natural language where knowledge objects are nouns (subjects or objects) and semantic primitives are verbs (actions of subjects or objects). This simple form is believe to be fundamentally sound in representing event-driven corporate economics, because REA objects as nouns are connected with each other through their relationships as verbs in the making of business events.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

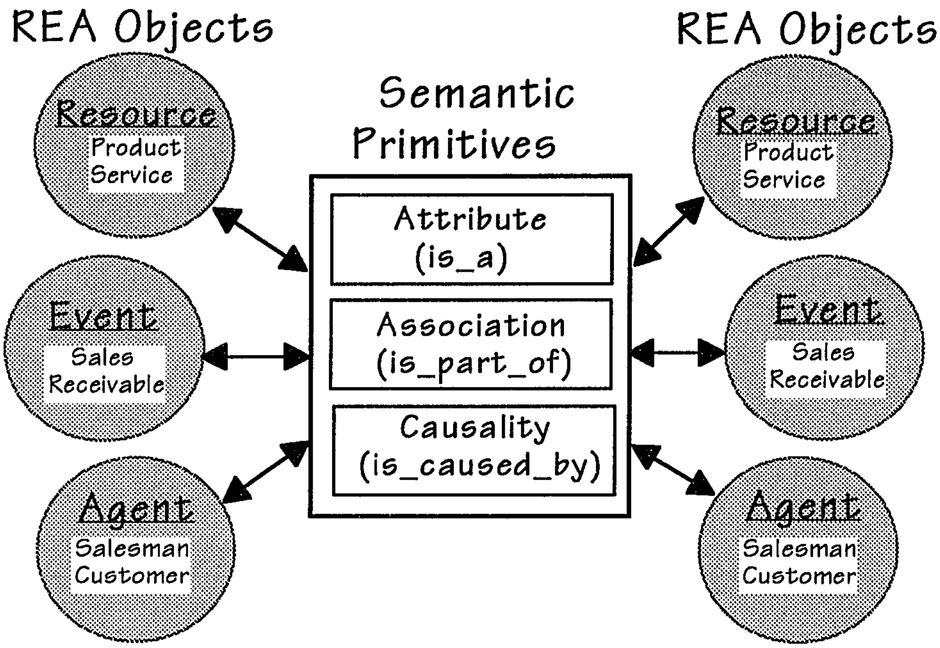


Figure 5.5. REA Objects Related by Semantic Primitive

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Knowledge Acquisition

There has been a steady stream of studies aimed at the acquisition aspects of domain-specific knowledge. Knowledge acquisition is the process of transferring knowledge from its source into the particular knowledge representation format. As McGraw and Harbison-Briggs (1989) summarize, there are generally three major approaches of knowledge acquisition: interviewing experts, learning by being told, and learning by observation. These approaches are equally important in that:

> "In reality, none is used to the total exclusion of the others. In some cases, a hybrid approach is adopted. In others, a single approach serves as primary for one phase of development, which additional approaches are adopted for use in later stages" (McGraw and Harbison-Briggs, 1989, p.9).

Although experts play an essential role in the knowledge acquisition process, the use of experts is not always practical. In most cases, as McGraw and Harbison-Briggs (1989) indicated, development of knowledge base takes place using rapid prototyping, in which knowledge is acquired in a very short term to allow developers to test their conceptions of design, knowledge representation, and processing for future improvements. For this reason, extensive interviews with experts is not appropriate for the purpose of prototyping. Furthermore, while knowledge may be easy to acquire in areas such as mathematics, engineering, and medicine, knowledge in business is generally considered more difficult to capture (Basden, 1983). Paradice and Courtney (1989) explains:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

"Mathematics and engineering have infallible laws that govern relationships between components of the field. In medicine, certain symptoms indicate with high probability the existence of specific medical problems. Business, however, has few such "laws." Although a price reduction may lead to increased sales, the impact of a price reduction depends on many other aspects of the business environment such as brand name recognition, relative market share, market maturity, and perhaps social attitudes. Because business relationships are so dynamic, identifying an expert that 'knows' all the relationships is practically impossible" (p.2).

Beside domain experts, Pau and Gianotti (1990) indicate that the sources of knowledge in economic and financial applications include:

(1) database containing historical or current data, including past decisions and performances;

(2) manuals, regulations and guidelines;

(3) other natural language sources, such as reports, newsletters, press-agency news;

(4) handbooks and textbooks; and

(5) interviews, rumors.

In developing the semantic network for the prototype DSS in this research, knowledge will be extracted from these resources through learning and observing by the developer, but in a consistent and systematic manner. In Denna et al.'s (1993) REA-based business re-engineering process, a systematic two-step method to identify

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

economic events is provided. The method first divides business activities into distinct processes or functions; these processes or functions are then decomposed into business events. The underlying premise of this method is that a business process or function is basically a sequence of business events. For example, in a mail-order company the sales process consists of the following business events: taking orders, packaging merchandises, and shipping merchandises. Once the economic events are identified, the last step is to include relationships among resources, events, and agents to complete the business model. The development of the semantic network using the REA concept follows the similar procedures as shown in Figure 5.6. The only difference is the types of semantic relationships used to represent these three classes of economic objects.

Knowledge Processing

The last facet of domain knowledge construction is the design of search algorithm to reason or to make assertions about knowledge in the knowledge base. Depending on their representation structure, search algorithms vary from knowledge base to knowledge base.

One of the key advantages of semantic network is that, beside providing a formalism for representing knowledge, it provides at the same time a structure for retrieving it. Thus, objects related to a given object in the domain can be found by search algorithm that follow the links from the node associated to the given object in the graph.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

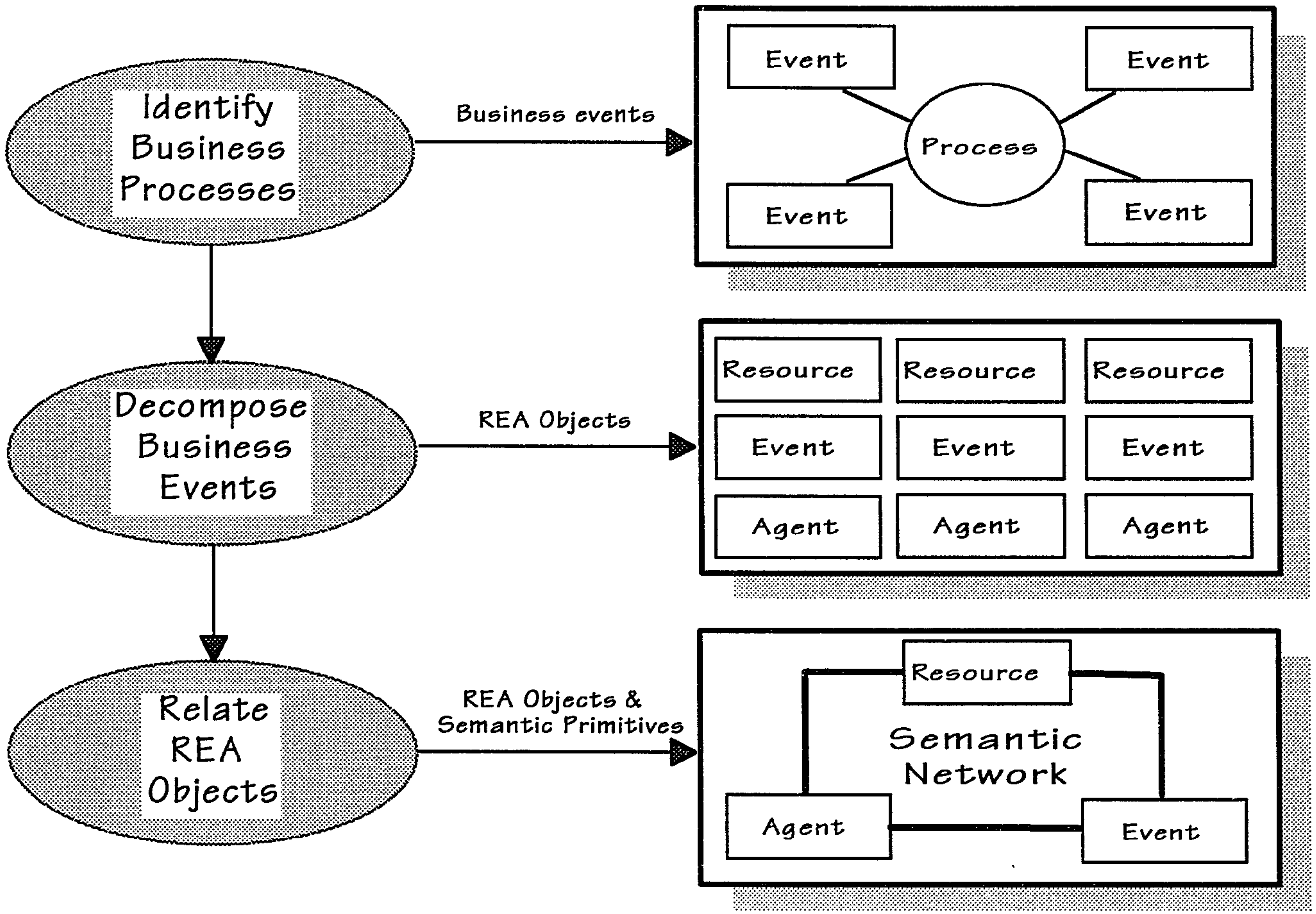


Figure 5.6. The Process of Developing the Semantic Network of Corporate Econominc

Most of semantic networks use the "spreading activation" search algorithm, which navigates the knowledge base by first activating the key objects, followed by objects that are directly related to the objects, followed by objects that are directly related to the objects previously activated, and so on. As Cohen and Kjeldsen (1987, p. 257) describe, "the activation spreads across relations in the network like ripples in a pond." Psychologists have long recognized that human memory is a system of associated concepts or "chucks" of symbolic information" (Simon and Newell, 1970). The structure of semantic network is analogous to that of the human memory (Quillian, 1968), and the mechanism of spreading activation mimics how human retrieve memory (Cohen Kjeldsen, 1987).

There are two types of spreading activation search algorithms: ordinary spreading activation and constrained spreading activation. Ordinary spreading activation searches objects in the network without any restrictions. The problem is that without restrictions the search will eventually touch every object in the network. In contrast, constrained spreading activation works under a set of rules to favor particular pathways through the network, and at the same time, avoid irrelevant ones.

The constrained spreading activation is ideal for searching knowledge of a specific topic or purpose. However, the general question is: where should the search stop? Cohen and Kjeldsen (1987) propose three kinds of rules that can be used to guide the search. First, the search can cease at a predetermined *distance* (e.g., 3 links or 4 nodes) from the original activation. Secondly, a *fan-out* algorithm can be used to stop the search at objects that have very high connectivity, or fan-out. The

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

third type of constraint uses the idea of likelihood of the relationship between objects. In Cohen and Kjeldsen's (1987) GRANT, an intelligent IR system, a minimally constrained spreading activation of a given *distance* (4 links) is used with satisfactory rates of precision and recall. The application and implementation of the constrained spreading activation algorithm will be further discussed in Chapter 8.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

CHAPTER VI

THE CONCEPTUAL DESIGN

Based on the conceptual framework formulated in Chapter 4 and the domain analysis conducted in Chapter 5, the conceptual design for the integrated DSS proposed in this dissertation is presented as follows:

Decision Process

As discussed in Chapter 4, a generic model of decision process that requires an integration of quantitative and qualitative information would consist of four phases: (1) model selection, (2) quantitative data retrieval for the selected model, (3) model execution, and (4) qualitative (textual) information retrieval. For decision making involved in corporate performance analysis, the quantitative aspect of the decision problems can be quantitatively formulated in terms of financial ratios, and the decision maker analyzes the corporate performance by repeating the following cycle of decision process.

(1) constructing (or choosing) the financial ratio(s);

(2) retrieving financial data (quantitative) from the financial statements for the selected ratio(s);

(3) calculating the ratio(s); and

(4) retrieving textual (qualitative) information that contains other salient, semantic attributes that is relevant to the ratio analysis but cannot be represented quantitatively.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Structure of the DSS Design

Based on the system requirements derived in the conceptual design framework, the structure of the proposed DSS should operate on an integrated information base that contains ratios, financial data needed for the ratio calculation, corporate performance literatures stored for retrieval, and semantic-sensitive knowledge required to retrieve relevant text for integrated qualitative information support. An intelligent system interface should also be developed to integrate information supports from multiple data resources and to coordinate the interactions between the system and the users. The top-level conceptual design of this system is illustrated in Figure 6.1.

This conceptual DSS design contains two groups of components: (1) four data resources (i.e., a data base, a model base, a knowledge base, and a text base) as the integrated information base and (2) a knowledge-based intelligent agent (IA) as the interface for integrated information processing and user interfacing. As discussed in Chapter 4, a unique advantage for systems developed from the scratch is that the data integration problem with heterogeneous databases can be avoided by adapting a uniform data model with its "database management system." Thus, data, model, knowledge, and text can all be stored in one particular data model and manipulated by the model's "database management system." Since this management system handles the information base, it is best to call it "information base management system" to distinguish it from other "management systems." The intelligent agent can then be built as an application in such an information management system.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

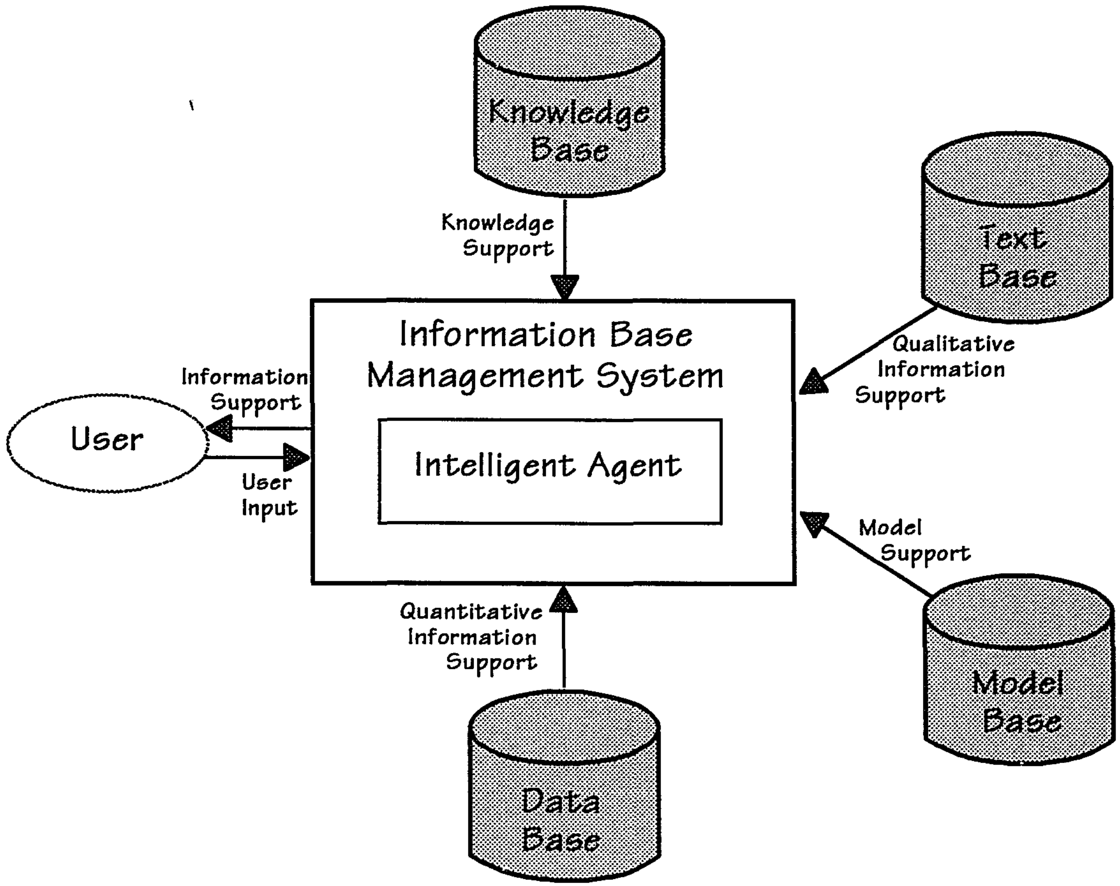


Figure 6.1. A Conceptual Design for the Proposed DSS

Overall, the DSS components are structured and interconnected in a way that allows the system to provide financial data support from the data base, ratio construction support from the model base, textual financial information support from the text base, and semantic knowledge support from the knowledge base. Through the functioning of the intelligent agent, this system will allow the tasks of ratio construction, financial data retrieval, ratio calculation, and the retrieval of textual financial information to be integrated and intelligently supported.

Structure of the Information Base

The integrated information base in this DSS design is composed of a data base, a model base, a knowledge bass, and a text base. The data base will contain financial data from income statements and balance sheets. In order to provide qualitative information support, documents related to current corporate performance will be collected, organized, and stored in the text base. These documents can then be organized into discrete elements such as paragraphs and sentences, in a text base and indexed by their entry labels (a text organization method used in a well-known text management product BRS/SEARCH produced by BRS Software). For example, a news article can be stored as a record consisting of fields such as date, source, subject, and text. The text base can also be automatically indexed by an inverted indexing method and made available for full-text retrieval.

To support ratio analysis of financial statements, various model components (financial statement constructs) will be stored in the model base. Because the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

numerators and denomerators (model constructs) in those financial ratios (models) are themselves constructs in financial statements and semantically represent objects or concepts in corporate performance analysis (problem domain), these objects with their semantic relationships can be captured and stored in a semantic network which can be used by the intelligent agent to generate a list of search keys for text retrieval.

Functions of the Intelligent Agent

The working of the proposed DSS is made possible by the functioning of the intelligent agent. Based on the system design principles suggested in Chapter 4, the intelligent agent of the current DSS design should have the following functions (also see Figure 6.2):

Flexible Model Construction or Selection

The menu-driven interface mode is chosen for this intelligent agent design for three reasons. First, menu-driven interface design has been widely adapted by most of the existing information systems. Today, many end users have been cognitively trained to interact with information systems through such an interface mode. Secondly, recent advances in software engineering have substantially improved the user-friendliness of menu-driven interface design by introducing the graphical user interface (GUI) technology. GUI technology provides remarkable support for the concept of object-oriented work flow that is closer to how human actually work. Thirdly, the number of financial statement constructs in limited, and so, a menu-

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

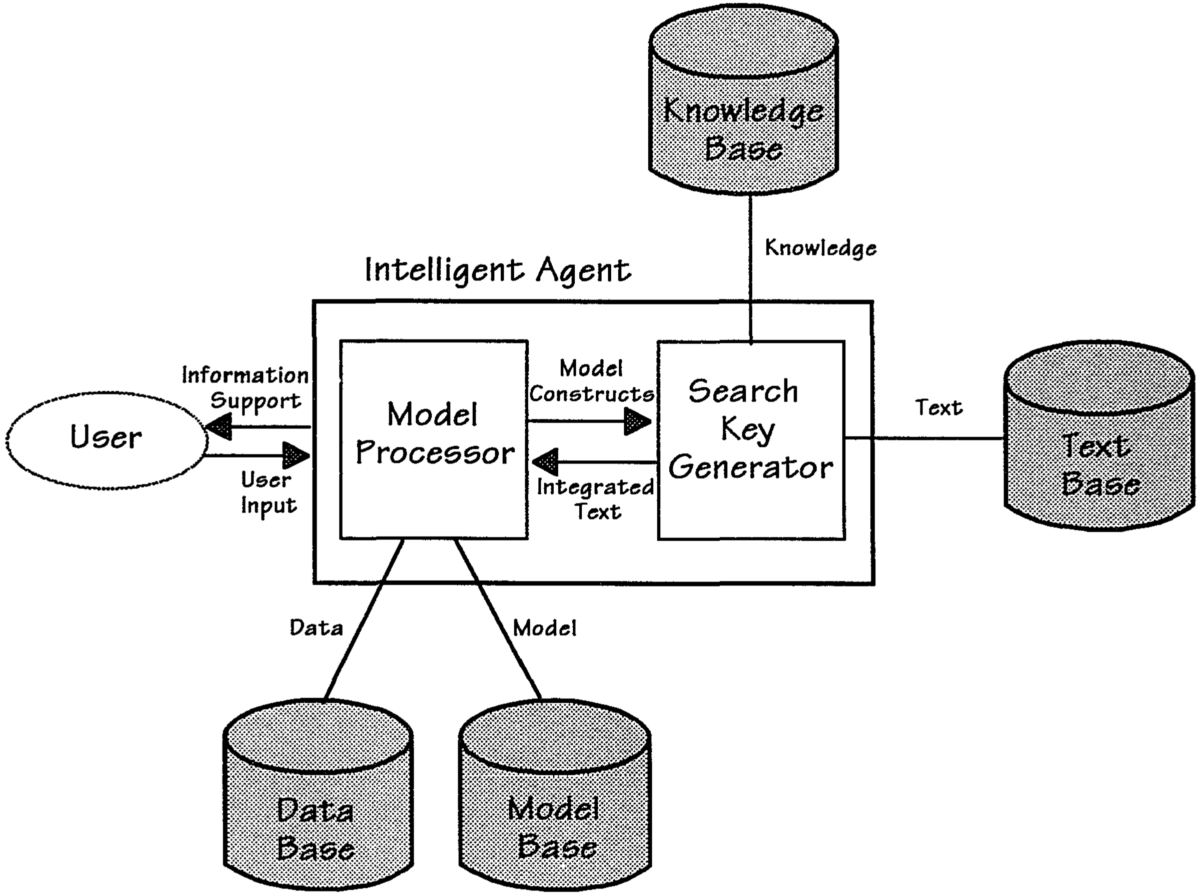


Figure 6.2. A Conceptual Design for the Intelligent Agent

driven interface, rather than a natural language interface, is sufficient to handle user requests.

Automatic data retrieval

The goal of this function is to automate the task of retrieving financial data from financial statements for calculating ratios being selected. Because of the use of a uniform data modeling scheme in this DSS design, the imcompability problem of heterogeneous data structures is completely eliminated. Thus, the retrieval of financial data from the data base for ratio calculation can be easily handled by referring to the data dictionary in a straightforward manner.

Intelligent search key derivation

Following the conceptual design framework, an intelligent search key generator and a semantic network can be developed to handle the task of deriving a list of search keys for text retrieval. In the domain of corporate performance analysis, the numerators and denomerators (model components) in financial ratios (models) are themselves constructs in financial statements and semantically represent objects or concepts in corporate performance analysis (problem domain). To support financial ratio analysis, the semantics embedded in these variables and their relationships provide a legitimate search space or contextual base that can be used for the exploration of search keys to retrieve relevant text from a financial text base. For example, suppose that the *receivables turnover ratio* is selected, the intelligent

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

agent will construct a search key list by tracing objects in the network that are semantically related to the two model constructs, "net sales" and "accounts receivable." As a result, search keys such as "sales," "revenue," "marketing strategy," "accounts receivables," "credit policy," "bad debt," and so on might be identified through recognizing the following relationships in the network: "*sales* IS_CAUSE _BY *marketing*," "*revenue* IS_CAUSE_BY *sales*," "*bad debt* IS_CAUSE_BY *credit policy*," and so forth.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

CHAPTER VII

SYSTEM PROTOTYPING

System Development Environment

This prototype system was implemented by using FoxPro for Windows v.2.6, a microcomputer DBMS product by Microsoft, for its provision of database management and programming facilities. FoxPro is a member of the so-called xBASE DBMS family but runs the fastest because of its copyrighted "Rushmore" query technology. For prototyping purposes, FoxPro for Windows is an appropriate choice because its high-level and user-friendly programming environment offers opportunity for efficient system prototyping. Speed is also important because text processing generally takes considerable CPU time.

FoxPro for Windows is the Windows-interfaced version of FoxPro that provides programming facilities to develop GUI-based applications. This feature makes the development of a GUI-based, Window-running DSS possible. Like other programmable xBASE DBMSs, FoxPro for Windows has built-in mathematical functions that are sufficient to support ratio analysis.

Finally, FoxPro for Windows is a relational DBMS (not by Cobb's standards). Thus the data base, model base, knowledge base, and text base can all be created in a relational format, making the handling of data query less complex (an argument made in the last section).

To take advantage of its CPU performance, a Pentium-90 based microcomputer was used as the hardware platform. Accompanied with its graphic

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

acceleration card, the system has made the tasks of testing and debugging much more efficient, because the interface design of the system is heavily graphic-oriented.

Data Preparation

For prototyping purposes, the dissertation does not intend to create a gigantic information base that contains data of a large number of companies. Rather, six companies (15%) out of the computer and peripherals industry listed in the *Value Line* (edition 7, January 27, 1995) were selected. In the selection process, companies that have limited public exposure were avoided, because published articles addressing these companies are usually hard to find. Also, note that the number of companies selected does not represent a statistically significant sample of the whole industry out of all the business entities in the U.S. and abroad. Corporate performance analysis is considered as a relatively generic domain, in which demographic characteristics of the companies are not of major concern.

The financial statements (balance sheets and income statements) of those six companies were downloaded from the Compact Disclosure Database developed by Disclosure Incorporated. The current version of this database contains financial and management information extracted from over 10,000 companies' annual and periodic reports (from 1991 to 1994) filed with the U.S. Securities and Exchange Commission (SEC).

Since the financial statements collected in the Compact Disclosure Database were all in standard 10k report format, these standard financial constructs will be

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

used as the base ratio components to build the model base. A list of 61 ratio components is shown in Appendix A.

The sources of text include newspapers, trade magazine, business journals, companies' annual reports, and so on. The president's letter and management discussion in corporate annual reports were downloaded from the Compact Disclouse Database. Other published articles were obtained from the General Periodicals Index for Academic Library Edition (GPIA) of InfoTrac developed by the Information Access Company. GPIA contains citations, and in some cases, brief abstracts of approximately 1,100 general-interest and scholarly publications, and covers a wide range of subject areas including social sciences, general sciences, humanities, business and management, economics, and current affairs. This indexed text base was used to collect citations of published articles concerning the selected companies. Since textual information is valuable for its timeliness, only articles published between January 1994 and May 1995 were collected. In the collection process, articles that are highly technical or out of business interest are avoided.

Because InfoTrac stores only the citation or a brief abstract of the articles, a hard copy of each selected article can only be obtained by tracing back to the original sources. A list of the publications containing articles collected in this dissertation is shown in Appendix B.

Functions of the Prototype

This section illustrates the functions and structure of the prototype. Because

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

the system interface is graphic-oriented, the illustration can be best done by the use of screen displays.

To demonstrate its use in the domain of corporate performance analysis, this prototype system was named "Corporate Performance Analyzer" by the developer. As shown in Figure 7.1, the name of the system is displayed on the title screen. Note that the picture on the screen was designed with the purpose of illustrating the theme of the system, which is to integrate information from the data base, model base, knowledge base, and text base for better decision support.

After a key is pressed, the system displays the main menu screen as shown in Figure 7.2. Because the system was implemented as an application in Microsoft FoxPro for Windows, the name of the system is displayed in a large font in the background to differentiate this application from the others. The screen also displays a "status bar" on the bottom as one of the basic features of FoxPro for Windows.

According to the conceptual design presented in the previous Chapter, the proposed system should have an information base incorporating four data resources, and an intelligent agent performing information integration and services. This functional requirement was implemented by designing and structuring individual functions in different modules as displayed in the menu bar. The main menu shows that there are three groups of modules in the system: the information base, the intelligent agent, and the utilities. *File*, *Statement*, *Ratio*, *Knowledge*, and *Text* are modules in the information base group. *Tool*, *Window*, and *Help* are in the utilities Group. *Analysis* itself represents the intelligent agent group. Functions of each

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

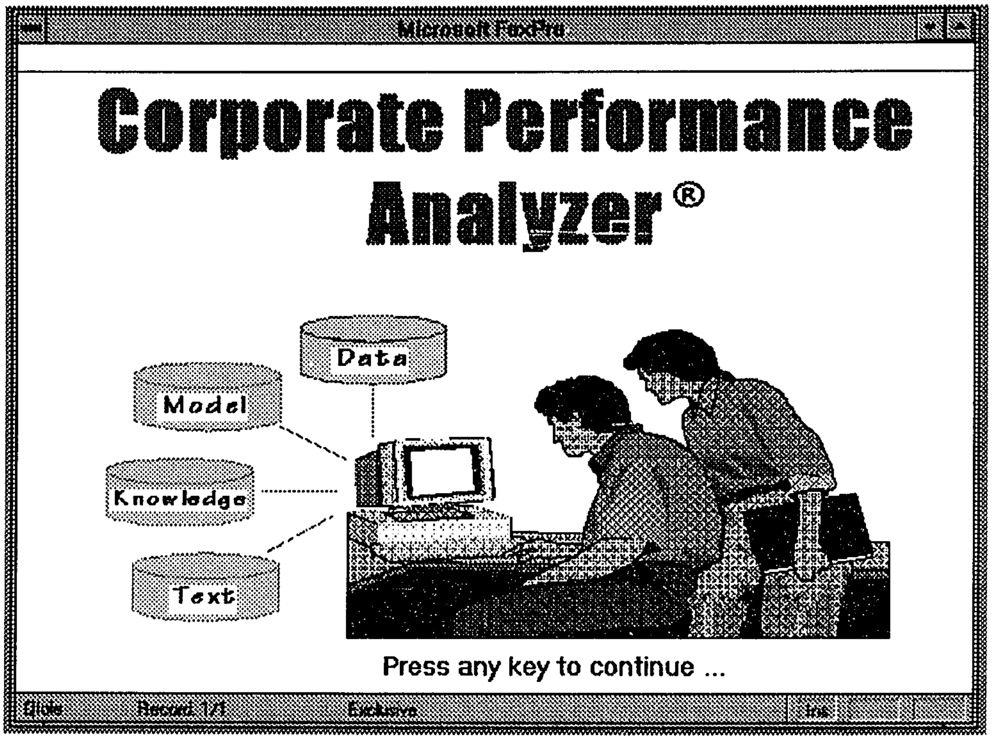


Figure 7.1. The Title Screen

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

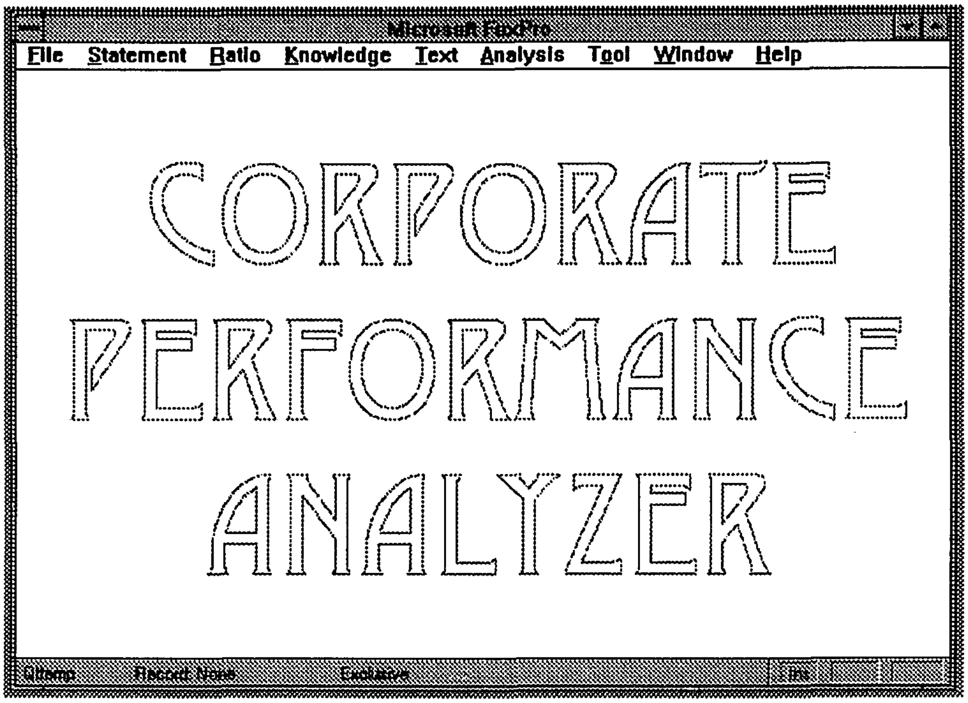


Figure 7.2 The Main Menu Screen

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

module are shown in Figure 7.3. The following briefly discusses the functions of these modules:

File

Similar to most Windows applications, this module was designed for file controls and printing purposes. The *Open* function allows the user to select the company he is interested in analyzing. The *Add* function allows a new company to be added to the database. The *Print* is reserved for function that lets the users print a hard copy of financial statements, ratio analysis history, and retrieved text for the company being analyzed. Finally, the *Exit* allows the user to quit the system.

Statement

This module allows users to view income statement or balance sheet of the company being analyzed. The modules controls the data resource of the information base.

Ratio

There are four functions in this module. To assist users in the analysis, the *Ratio Family Chart* can be viewed at any time when needed. This chart is also shown in Figure 5.1.

For users to conduct ratio analysis, the *Ratio Construct* function allows ratio components to be added, and using these components, the *User-Defined Ratio*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

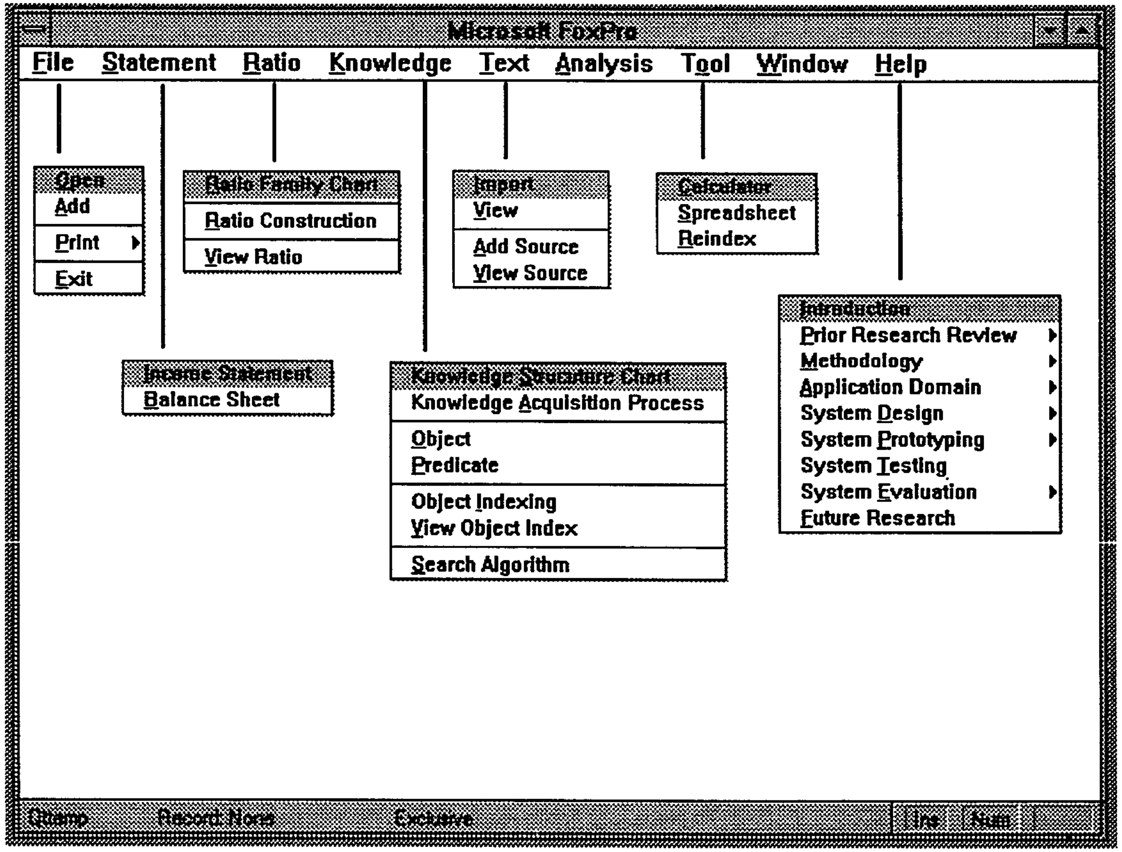


Figure 7.3. Functions of the Prototype System

function allows ratios to be defined and stored. The *View Ratio* function displays existing ratios with their types and definitions.

Knowledge

This module allows the semantic network to be built and maintained, and the search algorithm to be configured. The *Knowledge Structure Chart* and the *Knowledge Acquisition Process* functions can be activated to display Figure 5.4 and 5.5, respectively, to facilitate the tasks of knowledge engineering. The *Object* function allows REA knowledge objects (i.e., economic resources, economic events, and economic agents) to be added, edited, viewed, and deleted. Once knowledge objects are defined and stored in the system, the *Predicate* function allows semantic predicates to be added, edited, viewed, and deleted. The *Object Indexing* function enables the system to index text by knowledge objects captured in the *Object* function, and the *View Object Index* function allows the index file to be viewed as needed. Finally, the *Search Algorithm* function allows users to assign parameters to the constrained spreading activation employed in this system.

Text

This module builds and maintains the text base. The *Import* function accepts text files in ASCII form, reads them, and indexes them by objects captured in the *Object* function in the knowledge module. The *View* module allows the user to read an article collected in the text base. Texts stored in the text base are also indexed by

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

their sources. To do this, the *Add Source* function allows information about the source to be added and maintained, while the *View Source* allows existing sources to be viewed.

Analysis

This module itself is the intelligent agent that allows the user to select ratios, and enables the system to calculate the selected ratios, keep ratio analysis history, derive search key lists for text retrieval, and retrieve texts.

Tool

In order to facilitate the ratio analysis, two on-line tools were built in the system. First, the *Calculator* function allows a FoxPro for Windows Calculator to be activated. Secondly, using the Dynamic Data Exchange (DDE) technique, the *Spreadsheet* function is reserved to provide the system an opportunity to expand its capabilities by allowing the user to access a Microsoft-developed spreadsheet software package like Excel. The *Reindex* function is a routine which reindexes all databases in the system when needed.

Window

Like the *Calculator* in the tool module, the *Window* function is a FoxPro for Windows built-in utility program that can be activated to manage the window-operated environment when needed.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Help

This utility module is reserved to provide a copy of this dissertation and the system documentation. Text of this dissertation is displayed to help users understand the underlying concepts of the system design and development, while system documentation is displayed to assist the user in the use of the system.

Information Base Development

One of the component of this prototype DSS is the information base that contains all four data resources (e.g., data, models, knowledge, and texts). Since FoxPro for Window was chosen as the software development platform, these data resources need to be modeled and stored in relational databases. The processes of building the information base are illustrated in Figure 7.4. The following discusses the development of the information base in detail:

Data

Because the downloaded financial statements were all contained in ASCII files, these files need to be converted into .dbf format used by FoxPro for Windows. The conversion process includes the following procedures:

(1) trim unnecessary margins in the text;

(2) import the text into Lotus 1-2-3;

(3) convert text strings into data in numeric format; and

(4) export the file into .dbf.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

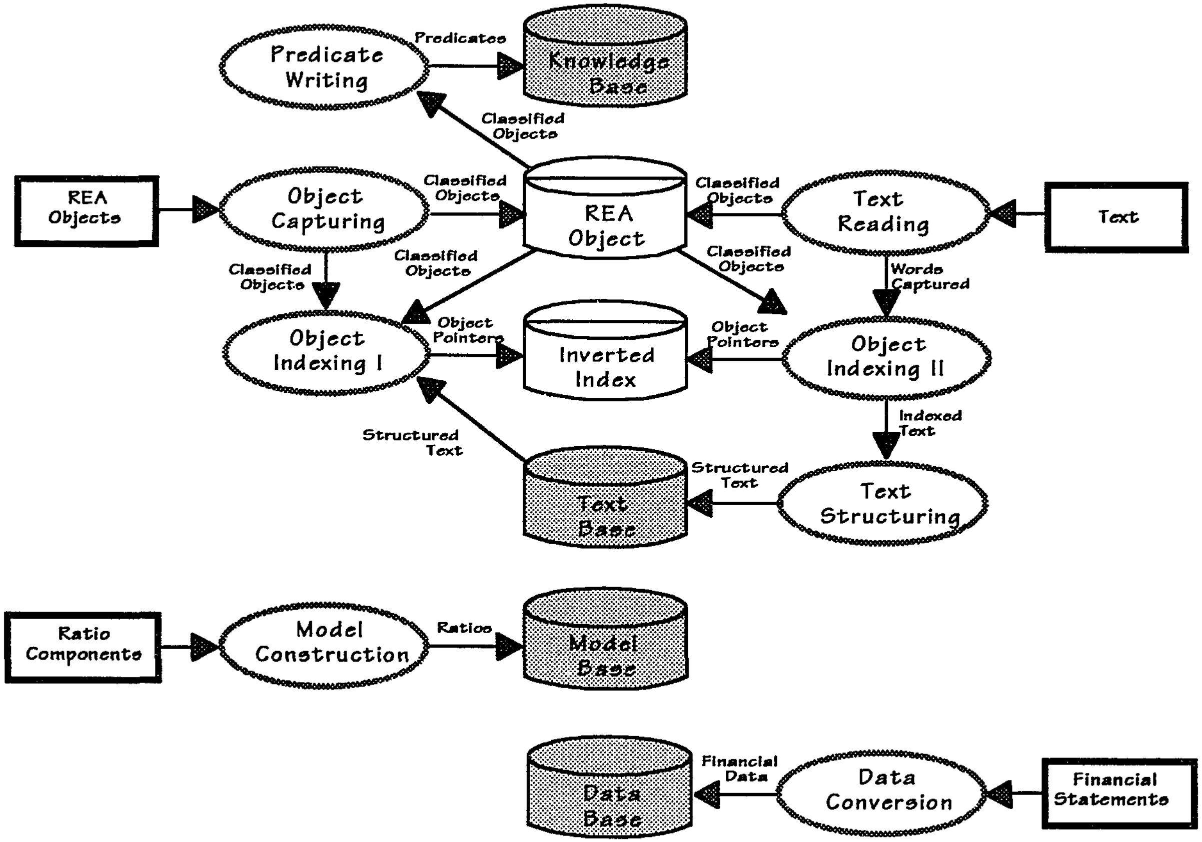


Figure 7.4. The Development of the Information Base

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

As a result, all four years (1991-1994) of financial statements (balance sheets and income statements) were stored in the database file called *qfs.dbf* and indexed by the company identification number.

Models

The 61 standard financial constructs of the 10k report were stored in a database named *qratioc.dbf*. Using these financial constructs as ratio components, ratios can be defined and stored in another database, called *qratio.dbf*, as the model base.

For experimentation purpose, five ratios (one in each type of the ratio family suggested by Morrow) were defined and stored in the model base by the developer. They are: quick ratio (financing structure ratio), gross working capital (asset allocation ratio), receivable turnover (asset utilization ratio), gross profit margin (sales profitability ratio), and earning per share (return on investment ratio). Ratios can be defined through the *Ratio Construction* function under the *Ratio* module in the main menu. There are five steps in this ratio definition process:

(1) choose ratio components from a list of existing ratio components;

(2) assign the chosen ratio components to either the numerator group or the denomerator group with positive or negative sign;

(3) classify the ratio into a specific ratio type;

(4) give the ratio a distinct name; and

(5) save the ratio.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

As shown in Figure 7.5, for example, the quick ratio can be defined by choosing one positive component ("+total current assets) and one negative component (-inventory) into the numerator, and one positive component (+total current liab) into the denomerator, and classifying it as one of the financing structure ratios Furthermore, during the above process, the *View Ratio* function can be called up to view existing ratios in a window at any time.

Knowledge

As discussed in Chapter 5, the semantic network of corporate economic can be built in two steps: capturing knowledge objects (i.e., economic resources, economic events, and economic agents) and writing semantic predicates (i.e., IS_A, IS_PART_OF, and IS_ CAUSED_BY). Knowledge objects can be captured through the *Object* function which allows objects to be:

(1) entered in three forms: the original form, the plural form, and the abbreviation.;

(2) classified into different REA object types; and

(3) edited, viewed, or deleted.

For example, Figure 7.6 shows that the object "general manager" can be entered and classified as an economic agent in three forms: "general manager," "general managers," and "gm." Note that there are two rules in the object capturing process:

(1) objects are all entered in lower case for consistency (the system will also convert objects in upper case into lower case if the user fails to do so); and

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

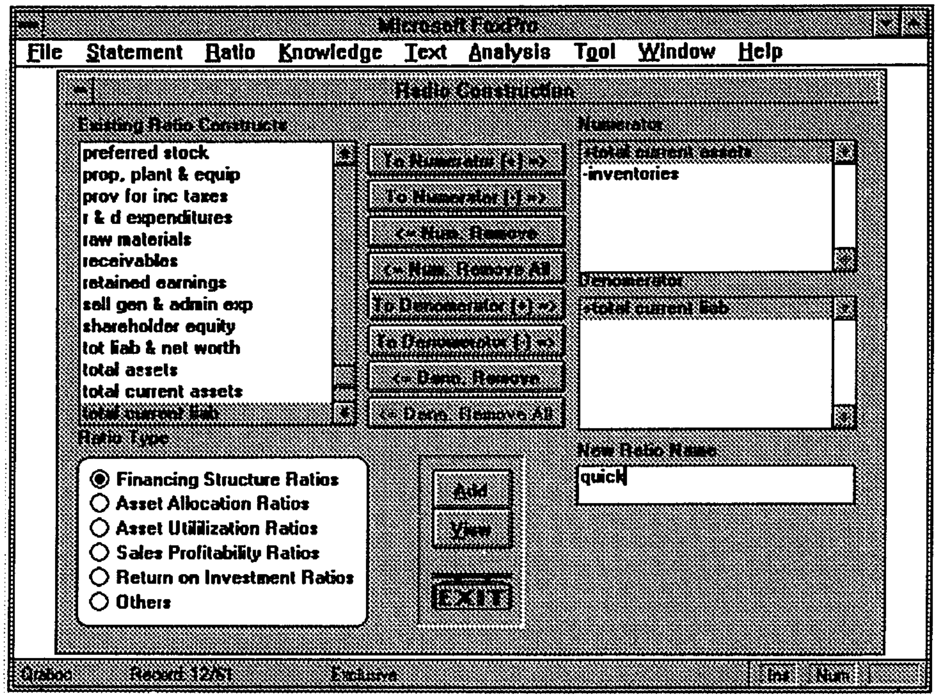


Figure 7.5. The Ratio Construction Screen

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

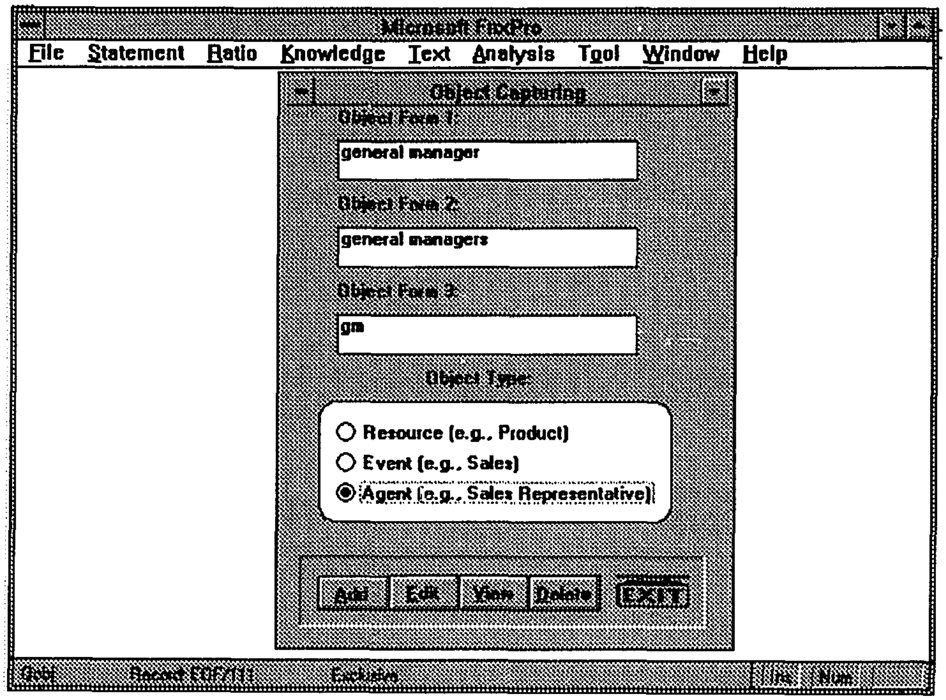


Figure 7.6. The Object Capturing Screen

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(2) the number of words that make up the object is limited to four (a number that is believed to be large enough to cover most of the knowledge objects).

Captured REA objects are stored in an object database called *qobj.dbf.*

Once knowledge objects are captured and stored, they can be used to write semantic predicates. This is done through the *Predicate* function by:

(1) entering two REA objects as component A and B of the predicate;

(2) giving the predicate a short annotation (e.g., generation, facilitation, transformation, measurement, etc.);

(3) specifying a predicate type; and

(4) editing, viewing , or deleting.

As shown in Figure 7.7, for example, a IS_CAUSED_BY and "transformation" type of predicate can be formed by entering "product" as object A and "raw material" as object B. Written semantic predicates are then stored in three predicate databases: *qkisa.dbf* for IS_A predicates, *qkisp.dbf* for IS_PART_OF predicates, and *qkisc.dbf* for IS_CAUSED_BY predicates.

For experimentation purposes, 357 knowledge objects and 377 semantic predicates were entered by the developer (see Appendix C and D), following the acquisition processes suggested in Chapter 5. In particular, articles of three companies (Compaq Computer, Dell Computer, and Stratus Computer) were selected and reviewed to supplement the knowledge acquisition work.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

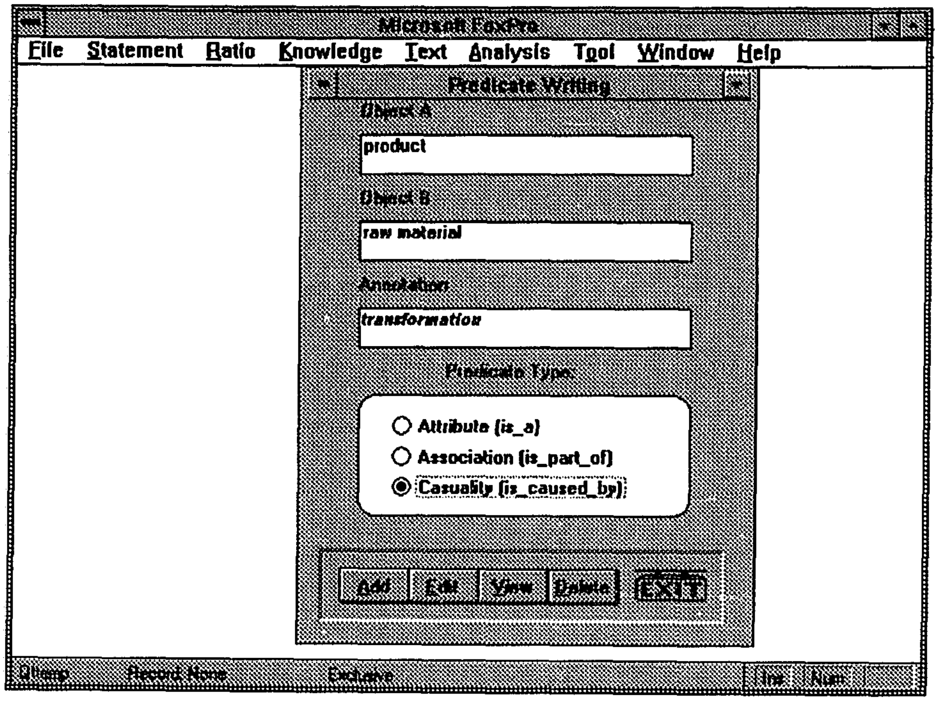


Figure 7.7. The Predicate Writing Screen

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Texts

To create documents in electronic form, the collected articles were scanned by Envision Scanner NV6100 developed by Envision Inc. and transferred into ASCII files by using TextBridge, an Optical Character Recognition (OCR) software created by Xeror. Because no OCR program can produce usable text files with 100% accuracy, and very frequently, the accuracy depends largely on the physical condition of the hard copy, the resultant text files need to be further checked for spelling and manually trimmed for unwanted margin discrepancies. Scanned text files were then saved for further processing.

Scanned text files were then processed by the *Import* function to create a text base and an inverted index file. To start the processing, the text files should be first identified. For example, Figure 7.8 shows that an article concerning the Apple Computer, published in the issue of October 10, 1994, in Business Week is about to be processed. The process takes the following three major steps (see also Figure 7.4):

(1) *Text Reading*:

a. read the text files four words a time;

b. form captured words into permutations of one word to four words (there are totally ten different permutations).

(2) *Object Indexing*:

a. match the permutated words to objects stored in the object database, but skip words that match the "dummy" words (e.g., of, at, and,

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

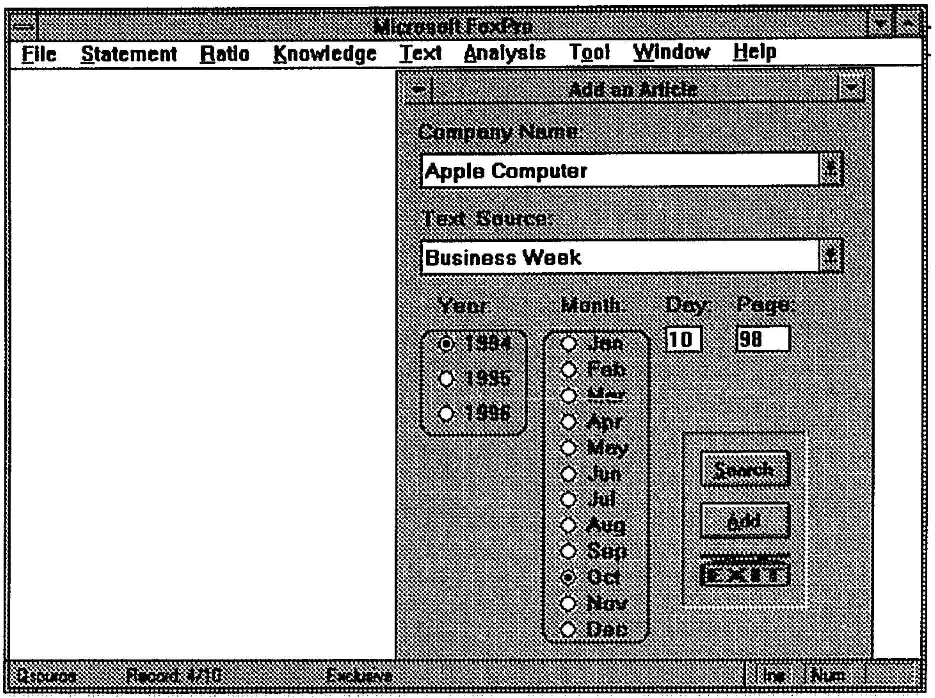


Figure 7.8. The Text Importing Screen

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

never, hence, etc.) contained in a database file, *qtcomm.dbf.*

b. if an object is matched, save the location of the permutated words as the pointer in an inverted index file named *qtinvert.dbf* .

(3) *Text Structuring*:

a. save the read text in the text base, *qt.dbf* , by treating a line of the text as a record (a length of 80 characters);

b. save heading information (e.g., time and resource) of this article in a text master file called *qtmaster.dbf.*

Note that the pointers of text is independent of the knowledge structure used in the semantic network. Thus, changes in the semantic predicates will not affect how the text base is indexed, because only knowledge objects are used for indexing. If objects are added into the object database after certain text files are converted and stored in the text base, the *Object Indexing* function in the knowledge model can be used to add pointers to the inverted index file. In Figure 7.4, this process is referred as *Object Indexing I* as opposed to *Object Indexing II* that is done through the *Import* function in the *Text* module.

The Intelligent Agent

As suggested in Chapter 6, the functions of the intelligent agent include (1) flexibly interacts with users during model construction or selection (the user-model interaction), (2) automatically determines the location of the data needed for model execution (the model-data interaction), and (3) intelligently derives search keys for

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

textual information relevant to the model (the model-text interaction). These functions were carried out in the design of the *Analysis* module.

As shown in Figure 7.9, the *Analysis* module allows users to select ratios previously constructed, calculate the selected ratios, view ratio analysis history and the generated search key list, and retrieve relevant text. For instance, when the quick ratio is selected and the *Calc* bottom is pressed, the intelligent agent will calculate the ratio of four years and show the result by displaying the calculations in the *Ratio* window. If the *Text* bottom is pressed, the intelligent agent will further develop a list of search keys, retrieve texts from the text base, and show the results in the *View Supporting Text* window. During the analysis, the search key list can be viewed in the *Search Key List* window, and ratio previously calculated can also be viewed in the *Ratio* windows. As shown in Figure 7.10, the intelligent agent's graphic-oriented, window-operated interface enables the analysis of corporate performance in an integrated environment.

In order for the intelligent agent to develop a list of search keys, users have to specify parameters to the spreading activation algorithm in the *Knowledge* module. As discussed in Chapter 5, effective search in the semantic network can only be achieved by the use of constrained spreading activation algorithms.

The task of constructing the search key list is carried out by the *Search Key Generator*, an inference engine used by the Intelligent Agent. The *Search Key Generator* works by the following steps:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

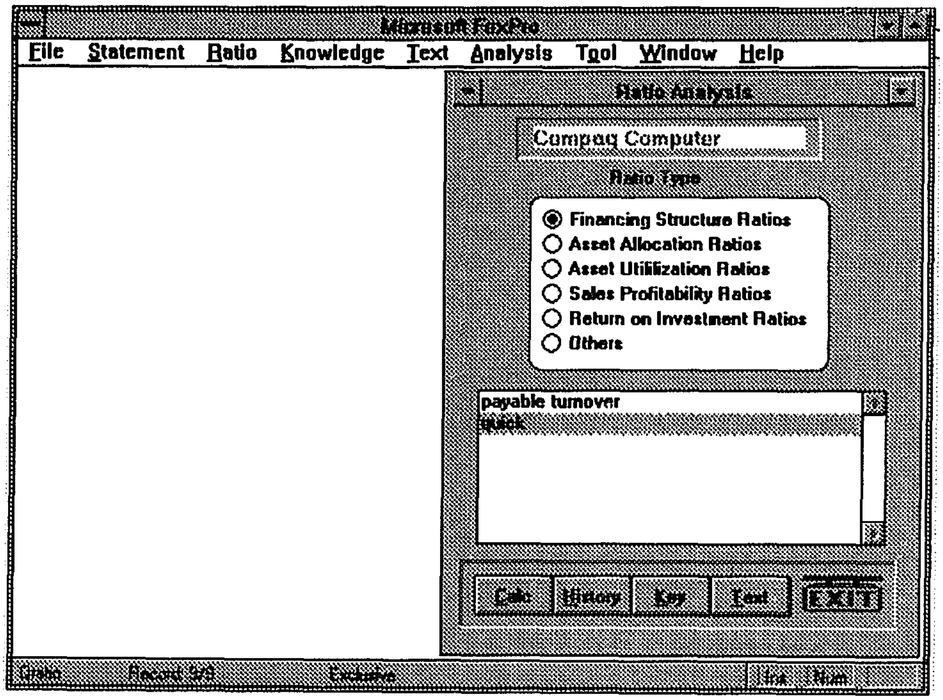


Figure 7.9. The Ratio Analysis Screen

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

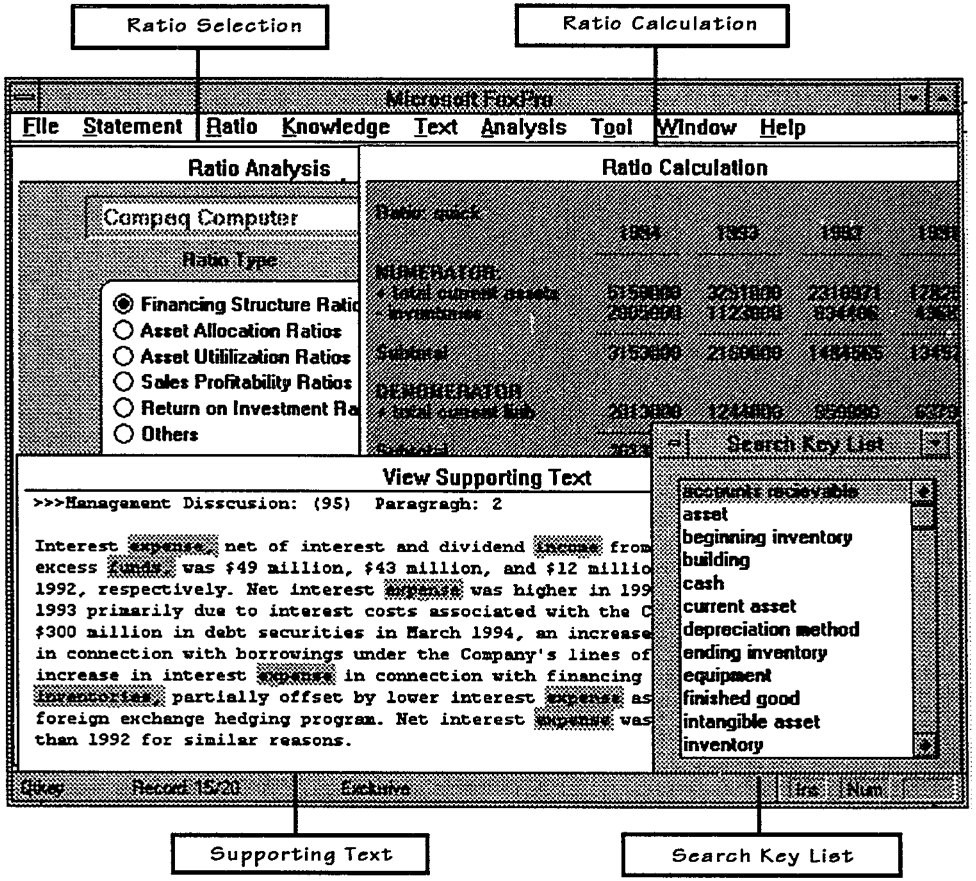


Figure 7.10. Functions of the Intelligent Agent in an Integrated Working Environment

(1) identify ratio components of the selected ratio;

(2) read the component strings four words at a time;

(3) form captured words into a permutations of one word to four words;(4) match the permutated words with objects stored in the object database, but skip words that match the "dummy" words contained in the database *qtcomp.dbf*;

(5) store all matched objects (primary search keys) in a temporary database (the original search key list);

(6) search predicate databases using the search parameters previously configured;

(7) add objects of the predicates spreaded (or touched) by the spreading activation search algorithm to the search key list;

(6) obtain pointers from the inverted index file for the search key list;

(7) retrieve text.

Experiments on the search performance will be conducted in system testing stage, and the results will be discussed in Chapter 8.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

CHAPTER VIII

SYSTEM TESTING

In retrospect, the goal of the current study is to formulate a conceptual DSS design that incorporates a text base, and based on the design, to develop a workable and testable prototype DSS to be used in corporate performance analysis. This prototype should allow a decision process consisting such tasks as ratio selection, quantitative data retrieval for the ratio, ratio execution, and textual information retrieval to support the ratio analysis to be integrated and intelligently supported. Using an intelligent agent as an interface, the system should also be capable of flexibly interacting with users in model construction, automatically determining the location of the financial data needed for ratio execution, and intelligently deriving search keys to retrieve relevant textual information.

This chapter describes the processes and results of a formal testing of the developed prototype. The testing was done in two parts: functionality validation and performance benchmarking. Functionality validation was a walkthrough of the functioning of the prototype. The goal was to validate whether principles of the conceptual system design are followed in the implementation of the prototype system. Performance benchmarking was used to appraise and document the capabilities of the system, with the goal to provide valuable advice for further improvement.

Functionality Validation

This part of testing was done using two illustrative examples of ratio analysis

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

on two companies: Compaq Computers and Dell Computers.

Example I

Suppose that a loan officer is evaluating a loan application filed by Compaq Computer, he/she wants to use the system to examine the company's ability to meet its debts. Since the loan officer is interested in the liquidity aspect of the company's performance, he/she chooses to initiate his/her analysis by using the current ratio. The following describes the interactions between the loan officer and the system in such a ratio analysis:

Step 1. *Open the company*. The *Open* option in the *File* menu is activated to choose "Compaq Computer" as the company to be analyzed.

Step 2. *View financial statements* (optional). If needed, the *Statement* menu can be activated to view the balance sheet or income statement of Compaq Computer.

Step 3. *Construct the ratio*. The *Ratio Construction* option in the *Ratio* menu is first activated. The ratio is then constructed by:

a. choosing the "current assets" as the numerator and the "current liabilities" as the denomerator;

b. giving the ratio a distinct name ("Current Ratio");

c. assigning the ratio to a ratio group ("Financing Structuring Ratios"); and,

d. saving the constructed ratio in the model base.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Step 4. *Verifying the ratio.* The *View Ratio* option in the *Ratio* menu can be activated to verify whether the current ratio has been accurately constructed.

Step 5. *Conduct the ratio analysis.* After the ratio is constructed and saved in the model base, the ratio analysis is conducted in the following five substeps:

a. select the "Current Ratio" from the model base.

b. activate the *Calc* function for the system to calculate the ratio and to display the result,

c. activate the *Search Algorithm* option in the *Knowledge* menu to select an algorithm with a particular configuration;

d. activate the *Text* function for the system to generate search keys, to retrieve relevant text, and to display the results; and

e. activate the *Key* option to view the search key list when needed.

As shown in Figure 8.1, the "Ratio Calculation" window displays the current ratios for four years. Figure 8.2 shows that the "View Supporting Text" window displays a retrieved paragraph. By activating the "arrow" functions on the bottom of the screen, different paragraphs can also be viewed. Note that messages shown on the top of the window indicate that the paragraph being displayed is the fourth paragraph out of the 38 paragraphs retrieved, and it is the 17th paragraph in the "Management Discussion" section of Compaq's annual report published in 1995. The highlighted terms, such as "cash," "assets," and so on, are knowledge objects stored in the knowledge base. The use of these terms in the paragraph is the reason that the paragraph is retrieved. Table 8.1 shows a list of the search keys.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

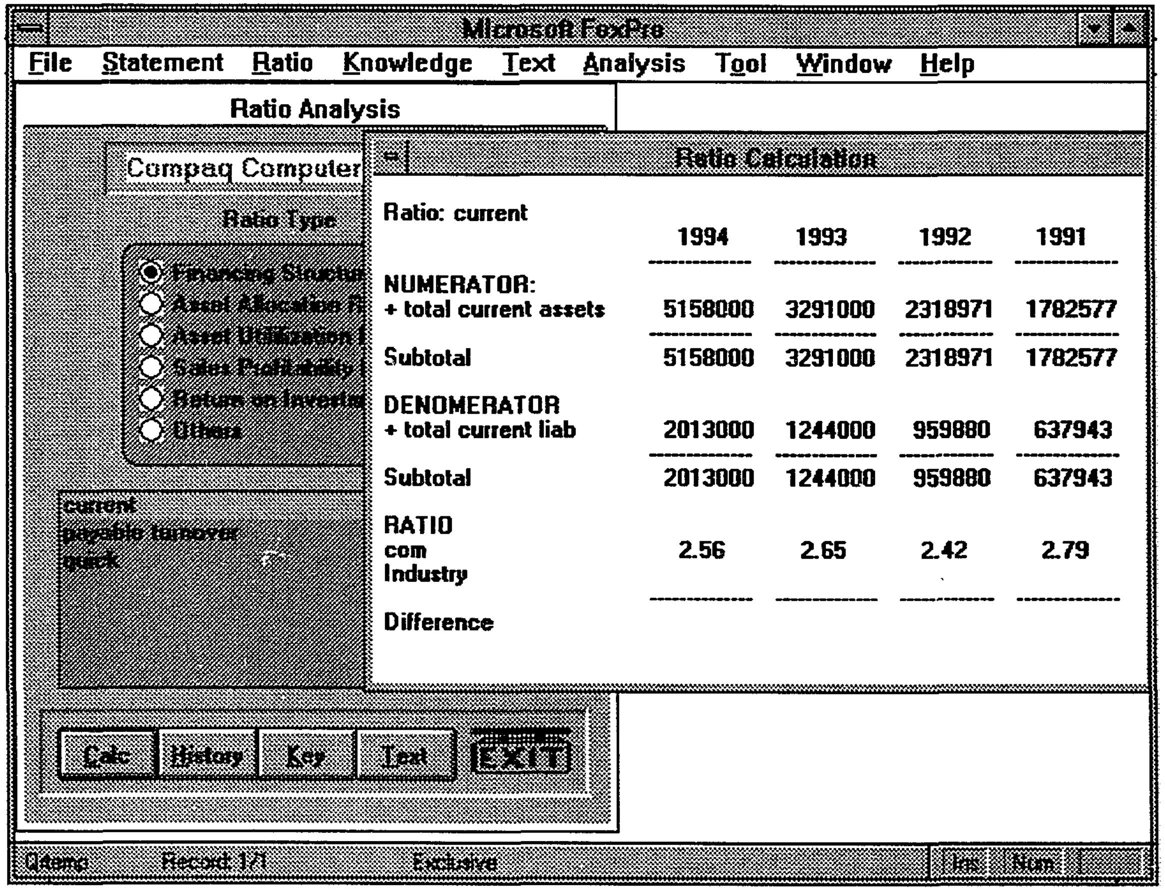
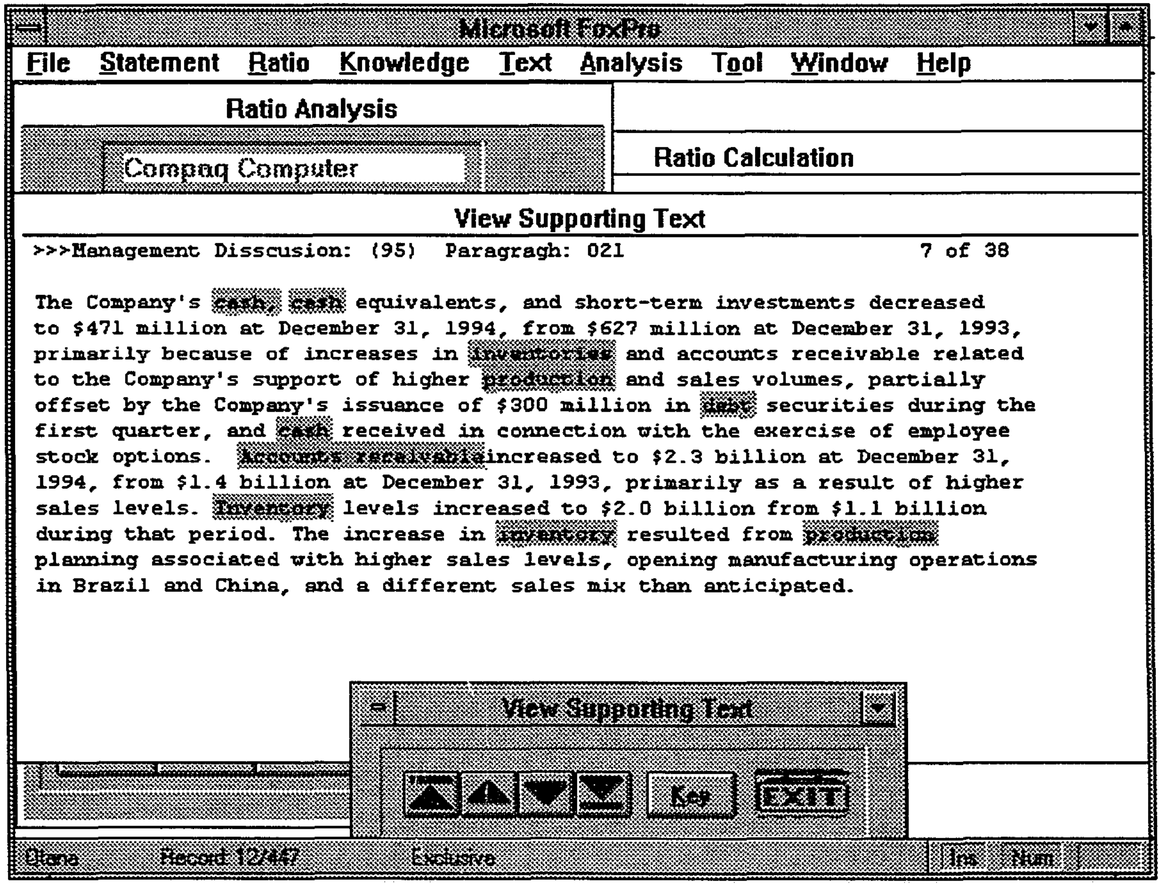


Figure 8.1. The "Ratio Calculation" Window in Example I

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.



Fiugure 8.2. The "View Supporting Text" Window in Example I

Search Key List
Company: Compaq Computer
Ratio: Current
Algorithm: Attribute-activated
Configuration: 1:1:1

---

accounts receivable
accounts receivables
accumulated depreciation
amortization
amortizing
asset
assets
bank deposit
bank deposits
borrowing
borrowings
building
buildings
capital expenditure
capital expenditures
cash
cash flow
cash flows
chief financial officer
chief financial officers
credit rating
credit ratings
current asset
current assets
current liabilities
current liability
current ratio
currentl liab
debt
debts
deferred charge
deferred charges
deposit
deposits
depreciation
depreciation
depreciation method
depreciation methods
equipment
equipments
financial management
financial planning
financial risk
financial risks
gross working capital
gross working capitals
intangible asset
intangible assets
interest payable
interest rate
interest rates
inventories
inventory
investment
investmetns
liab
liabilities
liability
liquidity
long-term liabilities
long-term liability
machine
marketable securities
marketable security
mrktable securities
nonoperating asset
nonoperating assets
operating asset
operating assets
payable
payables
plant
plants
prepaid expense
prepaid expenses
production
quick ratio
return on asset
return on assets
saving account
saving accounts
short-term liabilities
short-term liability
tangible asset
tangible assets
working capital

Table 8.1. The Search Key List in Example I

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Example II

In this example, suppose that an individual investor is thinking about purchasing Dell Computer's stock, and he/she wants to use the system to support his decision making. Since this individual in interested in the company's profitability, he/she chooses to initiate his/her analysis by using the ratio of return on stockholders' equity. The following describes the interactions between the investor and the system in such a ratio analysis:

Step 1. *Open the company.* The *Open* option in the *File* menu is activated to choose "Dell Computer" as the company to be analyzed.

Step 2. *View the financial statements* (optional). If needed, the *Statement* menu can be used to view the balance sheet or income statement of Dell Computer.

Step 3. *Construct the ratio.* The *Ratio Construction* option in the *Ratio* menu is first activated. The ratio is then constructed by:

a. choosing the "net income" as the numerator and the "shareholder equity" as the denomerator;

b. giving the ratio a distinct name ("Return on Stockholders' Equity Ratio");

c. assigning the ratio to a ratio group ("Return on Investment Ratios"); and

d. saving the constructed ratio in the model base.

Step 4. *Verifying the ratio.* The *View Ratio* option in the *Ratio* menu can be activated to verify whether the current ratio has been accurately constructed.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Step 5. *Conduct the ratio analysis*. After the ratio is constructed and saved in the model base. The ratio analysis is conducted in the following five substeps:

a. select the "Return on Stockholders' Equity" ratio from the model base.

b. activate the *Calc* function for the system to calculate the ratio and to display the result,

c. activate the *Search Algorithm* option in the *Knowledge* menu to select an algorithm with a particular configuration;

d. activate the *Text* function for the system to generate search keys, to retrieve relevant text, and to display the results; and

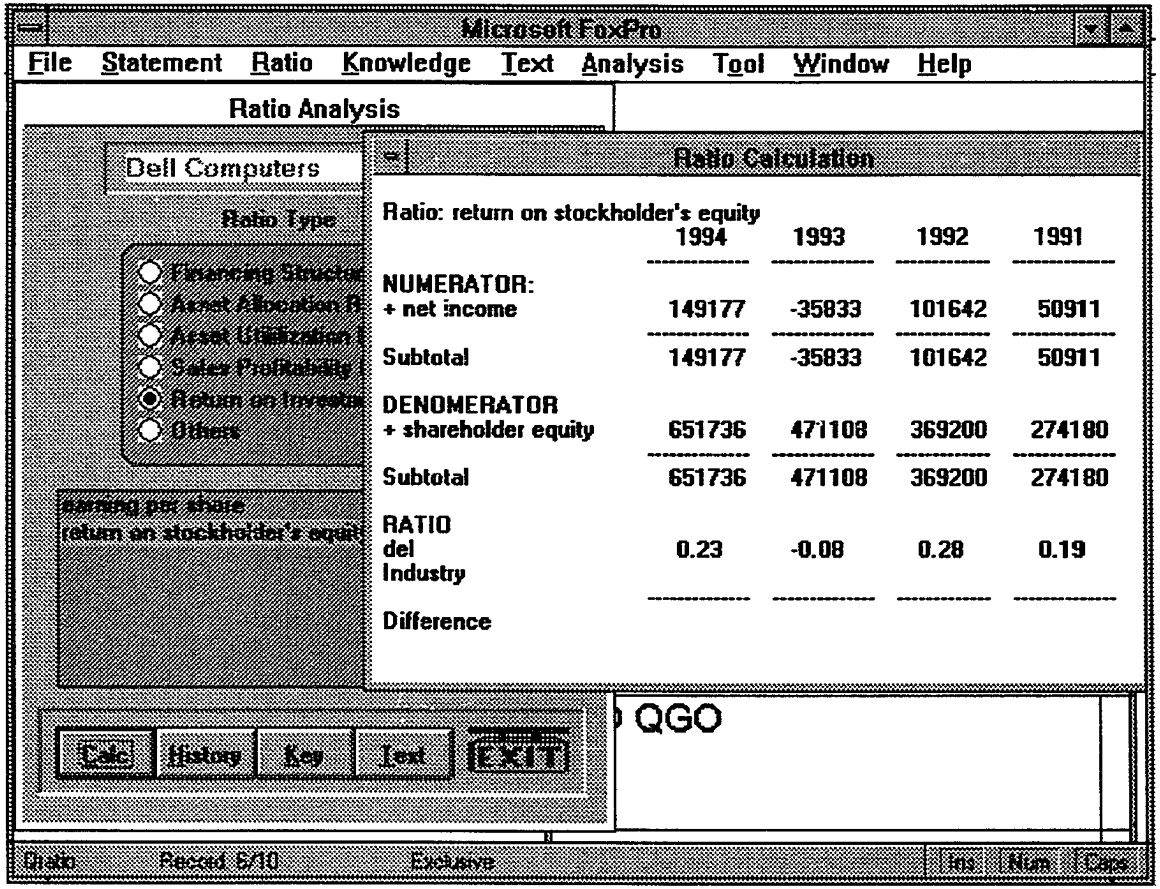
e. activate the *Key* function to view the search key list when needed.

Figure 8.3 shows that the "Ratio Calculation" window displays the ratios for four years. Figure 8.4 shows that the "View Supporting Text" window displays a retrieved paragraph. The paragraph being displayed is the first paragraph out of the 68 paragraphs retrieved, and it is the second paragraph in an article of *Barron's* published in October 11, 1995. The highlighted word "profit" is one of the knowledge object identified by the system and used as the key to retrieve the displayed paragraph. Table 8.2 shows a list of the search keys.

Through the above walkthroughs, it appears that the prototype DSS has the capability to provide integrated financial information (quantitative) and relevant textual information (qualitative) to support corporate performance analysis. Based on the conceptual design, the system provides necessary and logically interconnected

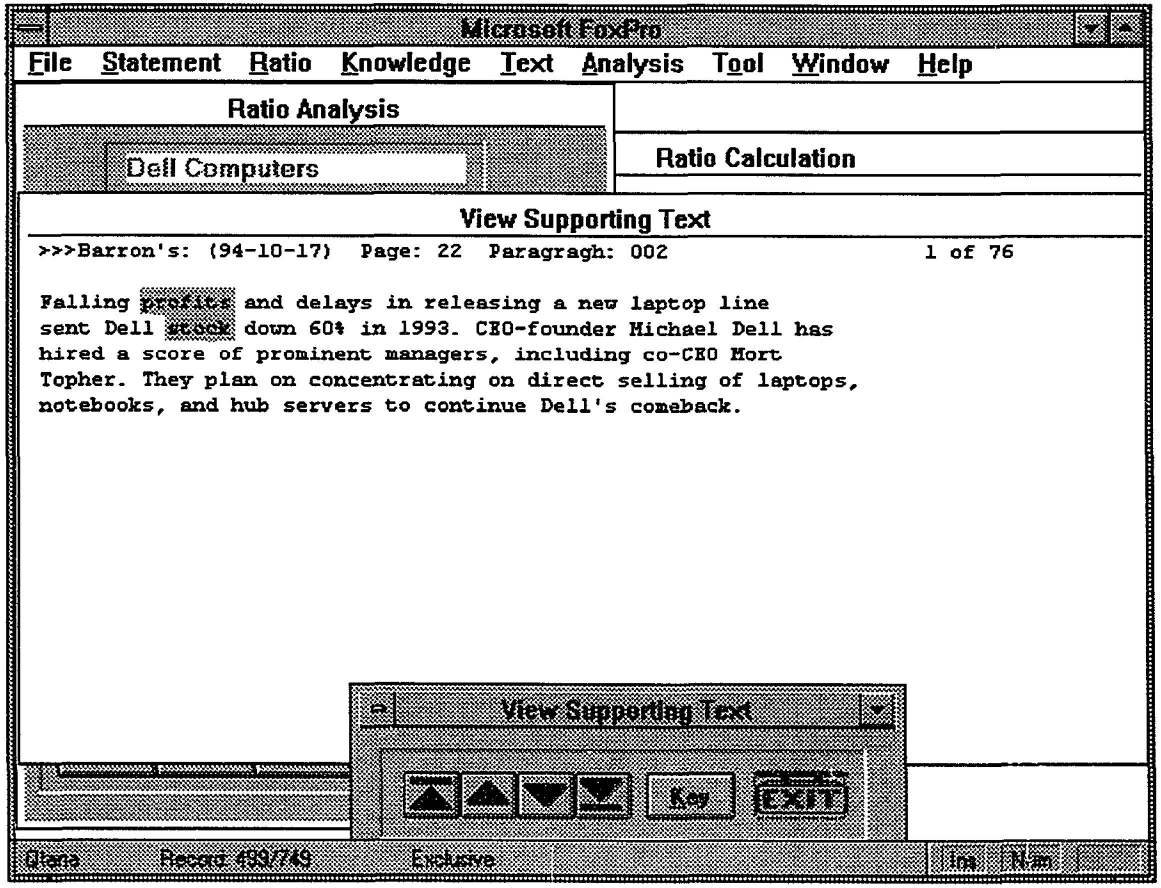
Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.



Fiugure 8.3. The "Ratio Calculation" Window in Example II

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.



Fiugure 8.4. The "View Supporting Text" Window in Example II

Search Key List
Company: Dell computer
Ratio: Return on Stockholder's Equity
Algorithm: Causality-activated
Configuration: 1:1:1
----

administrative expense
administrative expenses
earning
earning per share
earnings
equity
equity
expense
expenses
income
income before tax
income tax
income taxes
incomes
indirect labor
insurance expense
insurance expenses
interest expense
interest expenses
loss
losses
marketing expenditure
marketing expenditures
marketing expense
marketing expenses
net income
operating cost
operating expense
operating expenses
operation costs
profit
profits
r & d expenditure
r & d expenditures
r & d expense
r & d expenses
revenue
revenues
selling expense
selling expenses
share-net
shareholder
shareholders
stockholder equity
stockholders' equity

Table 8.2. The Search Key List in Example II

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

functions to allow the decision task of ratio construction or selection, financial data retrieval, ratio calculation, and textual information retrieval to be supported in an integrated, windows-running environment. Such a decision process is also supported in an intelligent manner in that the users do not have to know where the data is stored and look for search keys for text retrieval.

In regard to whether the functionality of the prototype has been reliably implemented, the above two walkthroughs may not be sufficient. The reliability validation, therefore, was conducted in the performance benchmarking which required 180 tests that in turn required 180 walkthroughs of the system.

Performance Benchmarking

Tests were also conducted to measure the performance of the system. Three major performance benchmarks used in the testing include speed, precision, and recall. Processing speed is a general performance indicator to all computer information systems, because one of the major objective of computer information systems is to provide efficient data processing in terms of satisfactory speed. Precision and recall are two performance measures commonly used in text retrieval. Precision is the percentage of relevant text retrieved to all text retrieved. Recall is the percentage of relevant text retrieved to all of the relevant text stored in the text base. These two measures are used jointly to provide an accurate appraisal of text retrieval, which goal is to retrieve the most relevant text in the least amount of total text retrieved.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

For the current prototype DSS, the precision and recall were tested to measure how effectively the system integrates quantitative and qualitative (textual) information in the analysis of corporate performance. Since the integration task is mainly carried out by the search key generator of the intelligent agent, the test is also used to examine how effectively the search key generator works using different search algorithms with different configurations. Chapter 5, 6, and 7 have discussed how the search key generator functions to construct a search key list that can be used to retrieve integrated textual information. In brief, the construction of the search key list is accomplished through the following major steps:

(1) identifying the qualitative determinant(s) associated with the quantitative constructs of the ratios as the primary search keys;

(2) configuring and choosing a search algorithm; and

(3) searching the semantic network to identify more keys that are semantically related to the primary search keys.

As also discussed in Chapter 5, the effectiveness of text retrieval relies primarily on the search algorithm configured and selected to construct the search keys list. Thus, the performance of text retrieval may vary as different search algorithms are configured and selected.

The testing for speed, precision, and recall covered all three different search algorithms (i.e., causality-activated, association-activated, and attribute-activated) provided by the system. Each algorithm was also tested by six different configurations in which different number of links ("distance" parameter) were

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

assigned to the three different types of semantic relationships (i.e., IS_A, IS_PART_OF, IS_CAUSED _BY). These six different configurations were predetermined for testing based on the experience learned from the system prototype stage. The parameters for these configurations were stated in a standard form as:

number of IS_A link: number of IS_PART_OF link : number of IS_CAUSED_BY.

For example, the configuration 1:2:2 stands for one IS_A link, two IS_PART_OF links, and three IS_CAUSED_BY links. Thus, the configuration activates the search from the primary key for one link of the IS_CAUSED_BY relationship, followed by two links of the IS_PART_OF, and by two links of IS_A relationships.

The testing was done on five ratios, including quick ratio, gross working capital ratio, receivable turnover ratio, gross profit margin ratio, and earning per share ratio. These ratios represented the five individual types of the ratios family suggested by Morrow (1991), and were predefined and stored in the model base previously in the system prototyping phase.

Two companies used for testing were Digital Equipment Corporation and Sun Microsystems. A statistical summary of the collected text for these two companies are shown in Table 8.3. For Digital Equipment Corporation, 16 documents with a total of 174 paragraphs and 11,027 words were used as the base for testing. There were 780 knowledge objects captured in these documents, representing an average

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Source	Number of Paragraphs	Number of Knowledge Objects	Number of Words
Digital Equipment Corp.			
(1) AB	2	5	79
(2) AB	2	10	86
(3) AB	2	2	84
(4) BA	2	4	74
(5) BM	2	3	63
(6) BM	2	2	96
(7) BW	4	9	153
(8) BW	14	41	1121
(9) BW	7	32	655
(10) BW	12	49	770
(11) CI	2	9	107
(12) IF	2	7	106
(13) MD	90	492	5913
(14) PL	25	63	1147
(15) VL	4	22	407
(16) WS	2	10	166
Total	174	760	11027
Sun Microsystems			
(1) AB	2	3	90
(2) BW	10	30	648
(3) CC	2	5	97
(4) MD	85	488	6125
(5) PL	19	71	1120
(6) VL	6	23	395
(7) WS	2	12	207
Total	126	632	8682

Table 8.3. Characteristics of the Tested Text Base

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

of one knowledge object in every 14 words (11,027/780) and 4.5 knowledge objects (780/174) in every document. For Sun Microsystems, 6 documents with a total of 126 paragraphs and 8,682 words served as the testing base. In this testing base, there are 632 identified knowledge objects, representing an average of one knowledge object in every 13.8 words (8,682/632) and 5 knowledge objects in every document. On the surface, these two testing bases had a very similar density of knowledge objects.

In summary, the testing was done on five ratios of two companies, using all three search algorithms each having six different configurations. The experiment setting, therefore, required 180 individual tests.

In order to calculate the precision and recall, documents collected for testing were first reviewed paragraph by paragraph to determine their relevancy to those five ratios. A database was created to record the identification number of the reviewed paragraphs with true/false marks to indicate if they were relevant to a particular ratio. This database was then used to compare with the results of each individual test. The speed, precision, and recall of the 180 tests are collected in Appendix E. Findings resulted from the experiments are discussed in the following sections:

Precision

Overall, the three search algorithms with the six configurations achieved an average of 81.23% precision (See Figure 8.5(a)). The causality-activated search algorithm with configuration 1:1:1 had the highest precision at 91.52%; the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(a) Precision

Parameter \ Algorithm	Causality-activated	Association-activated	Attribute-activated	Average
1 : 1 : 1	91.52	86.82	88.88	89.07
1 : 2 : 2	90.98	86.46	88.36	88.60
1 : 3 : 3	87.28	86.12	87.57	86.99
1 : 4 : 4	88.27	86.12	87.57	87.32
2 : 1 : 1	71.46	65.09	67.77	68.11
2 : 2 : 2	71.47	63.87	66.48	67.27
Average	83.50	79.08	81.10	81.23

(b) Recall

Parameter \ Algorithm	Causality-activated	Association-activated	Attribute-activated	Average
1 : 1 : 1	92.01	88.45	85.51	88.66
1 : 2 : 2	93.51	89.61	86.67	89.93
1 : 3 : 3	94.01	89.80	88.16	90.66
1 : 4 : 4	93.81	89.80	88.16	90.59
2 : 1 : 1	95.05	91.56	89.58	92.06
2 : 2 : 2	96.22	92.83	91.70	93.48
Average	94.10	90.29	88.30	90.90

(c) Speed

Parameter \ Algorithm	Causality-activated	Association-activated	Attribute-activated	Average
1 : 1 : 1	4.29	4.11	3.13	3.84
1 : 2 : 2	7.61	6.42	3.95	5.99
1 : 3 : 3	11.33	9.08	5.02	8.48
1 : 4 : 4	15.05	11.57	6.33	10.98
2 : 1 : 1	6.65	6.34	4.78	5.93
2 : 2 : 2	12.04	9.107	5.99	9.04
Average	9.50	7.77	4.87	7.38

Figure 8.5. Averages of the Three Performance Benchmarks

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

association-activated algorithm with 2:2:2 configuration had the lowest precision at 63.87.

In additions, the results also show that the proportion of relevant text in the retrieved set significantly decreased when the number of IS_CAUSED_BY links increased from one to two. The percentage differences among these configurations with either one of two IS_CAUSED_BY link, however, were within a very narrow range from 1 to 4. This reflects the fact that among the three different semantic primitives in the semantic network, the IS_CAUSED_BY link has the highest connectivity which is an important reason for retrieving irrelevant text.

In some cases, the association-activated algorithm and the attribute-activated algorithm were able to achieve better rates of precision. For example, for Sun Microsystems' ratio of gross profit margin, the association-activated algorithm with 1:1:1 configuration was able to provide the highest precision at 98.3%, followed by 98% of the attribute-activated algorithm and 95.5% of the causality-activated algorithm (see Appendix E). However, although the association-activated algorithm and the attribute-activated algorithm were capable of providing the highest precision, the number of relevant paragraphs they retrieved are less than the one by the causality-activated algorithm by four (64-60) and twelve (64-52) paragraphs respectively. Thus, the increased precision of the association-activated algorithm and the attribute-activated algorithm were achieved only at the cost of the decreased recall. In order to find an explanation for the above phenomenon, a careful examination was conducted to trace how the search key lists were generated in the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

searching. The result showed that the association-activated algorithm and the attribute-activated algorithm were able to generate less irrelevant knowledge objects for these two ratios than the causality-activated algorithm, because a lot of knowledge objects pertaining to the profitability aspect of the corporate performance are represented by IS_A and IS_PART_OF links.

Recall

As shown in Figure 8.5(b), the testing results of the experiments indicate that the three algorithms with the six configurations achieved an average recall at 90.90%. The causality-activated algorithm with 2:2:2 configuration provided the best recall at 96.22%; the attribute-activated algorithm with 1:1:1 provided the lowest recall at 85.51%.

The results also indicate that the causality-activated algorithm was able to provide better recall than the attribute-activated algorithm, and the configurations with two IS_CAUSED_BY links performed better in recall than the ones with one IS_CAUSED_BY link and one or two IS_PART_OF and IS_A links. An explanation for this finding is that the algorithm which activates the search from the IS_CAUSED_BY links and the configuration with more IS_CAUSED_BY links are able to generate more search keys which in turn retrieve more relevant text out of all the relevant text stored in the text base.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Speed

The results of the testing reveal that the strength of the three algorithms was reflected on their required speed of text retrieval. First, since the causality-activated algorithm starts the search with the IS_CAUSED_BY links that have the highest connectivity, it generally generate the largest search key list and requires the longest search time. In contrast, the attribute-activated algorithm generally requires the least search time, because it tends to generate the smallest search key list. Secondly, the speed of text retrieval was also a function of two factors: the number of the IS_CAUSED_BY links and the number of total links of IS_A and IS_PART_OF. More links generally means more search keys and more time to retrieve text.

Trade-off and Ceiling Effect

When examined more closely, the testing results of both precision and recall shows that there were two searching conditions commonly found in these six configurations: the *ceiling effect* and the *trade-off*. The *trade-off* was the condition in which increased recall resulted in decreased precision, or vice versa. For example, in Sun's quick ratio analysis, the change from configuration 1:1:1 to configuration 1:2:2 (the number of links of the IS_A and IS_PART_OF increased) led to an increase in the recall from 93.1% to 96.5%, but an decrease in the precision from 90.0% to 87.5% (see Appendix E). Another example was found in Digital Equipment's ratio analysis of receivable turnover, where the change from configuration 1:1:1 to configuration 2:1:1 (the number of IS_CAUSED_BY links

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

increased) resulted in an increase in the recall from 85.2% to 92.6%, but an decrease in the precision from 95.0% to 58.9% (see also Appendix E). The explanation for such an effect is that by increasing the number of links in the configuration, the search key generator will generate more valid keys to retrieve more relevant text, but only at the expense of decreasing precision which happens when the number of irrelevant text is also increased.

The *ceiling effect* was the condition that, with one IS_CAUSED_BY link in the search, the precision decreased and the recall increased when the number of links of the IS_A and IS_PART_OF increased, but only to a certain level. For example, for Digital Equipment's ratio of gross profit margin, the precision and recall reached a ceiling level of 97.8% and 90.1% respectively when the number of links of the IS_A and IS_PART_OF increased from 1 to 2; and after that, the precision and recall stayed the same regardless of any increase in the number of links of the IS_A and IS_PART_OF (see Appendix E). This *ceiling effect* was mostly found when the number of links of the IS_A and IS_PART_OF increased from 2 to 3. The explanation for this effect is that when all of the increased IS_A and IS_PART_OF links reach nodes in the semantic network that do not have any extension nodes, the size of the search key list stays fixed.

In summary, the testing results of precision, recall, and speed indicate that:

(1) Among the three types of search algorithms, the causality-activated algorithm worked the most effectively. In some cases, although the association-activated algorithm and the attribute-activated algorithm were able to provide higher

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

precision, the increased precision were achieved only at the cost of decreasing the recall.

(2) The strength of the three algorithms with the six configurations was principally reflected on their required speed of text retrieval. In that, the causality-activated algorithm generally required the longest search time; and the more IS_CAUSED_BY links required to search or the more the number of total links of IS_A and IS_PART_OF presented in the configuration, the longer it took to complete the key searching and text retrieval.

(3) The *trade-off* condition was found common in most configurations, in which increased recall resulted in decreased precision, or vice versa.

(4) *Ceiling effect* was also found common, in which it suggests that the precision decreased and the recall increased when the number of links of the IS_A and IS_PART_OF increased. Furthermore, this effect was mostly found when the number of links of the IS_A and IS_PART_OF increased from 2 to 3.

Sources of the Imperfect Precision and Recall

Efforts were also made to identify the sources for imperfect precision and recall. The result shows that there are three types of such sources: the semantic complexity, the search link inadequacy, and the knowledge engineering bottleneck. In general, the semantic complexity was the main cause for imperfect precision rate; the inadequate search links in a algorithm was the main cause for imperfect recall rate; and the knowledge engineering bottleneck was a cause for both.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The semantic complexity refers to problems in which the semantics of the raw text cannot be accurately captured and represented through the structure of a semantic network. There are three types of semantic complexity:

(1) *Semantic ambiguity.* There are some terms that have multiple meanings. Without further semantic or programmatic analysis, the terms can be easily misinterpreted. Some of the examples include:

a. *Shipment* can mean finished products shipped by the company to the customers or raw materials shipped from the vendor to the company.

b. *Forecast* can mean prediction from the perspective of the supplier market or from that of the buyer market. It can also be interpreted as the prediction of product demand, money market demand, political weather, and so on.

c. *Production* can be or cannot be the same as *manufacturing* for different writers. A general understanding of the business literatures indicates that *production* refers to the management and control of producing products, while *manufacturing* means the actual production process.

d. *Equipment* which means machinery used in the manufacturing process happens to be one of the words in the company name, Digital Equipment Corp.

e. *Building* can be referred to a physical property or used as a verb.

(2) *Message emptiness.* The message being conveyed in a paragraphs may be too general. For example, a paragraph in an article that appeared in one issue of *Forbe* magazine (Churbuck, 1994) reads:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

" Legend has it that John D. Rockefeller was asked, How much money is enough? His answer: "Just a little more than what you have" (p.162). Obviously, these is no knowledge object that can accurately represents the message that is being conveyed by the writer. Experience learned from the system prototyping shows that this type of paragraph appear mostly in the introduction of an article or in the President's Letters.

(3) *Syntactic variation*. There are some knowledge objects that can be expressed in many different forms. For example, *product life cycle* can be written as *the cycle of the product life*, *return on investment* can be written as *return of our investment*, or *domestic market* can be written as *domestic home market*. Since the system only takes up to four words for each knowledge object, not all of the possible expressions of these knowledge objects can be represented and stored by the developed semantic network.

The knowledge engineering bottleneck has been recognized as a common problem with the development of knowledge-based information system. In the semantic network developed for this system, knowledge objects and their relationships can be wrongly represented or structured. For example, in the testing, the knowledge object "market" was found not to relate to "sales" through an IS_CAUSED_BY links. This problem has caused a drop in the recall when only one IS_CAUSED_BY link was used.

Finally, a natural cause for the imperfect precision and recall was the algorithms themselves. Throughout the tests, it was found that, under certain

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

circumstances, using inferior search algorithms or configurations would result in decrease precision or recall. Detailed explanations for this cause has been documented previously in this Chapter.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

CHAPTER IX

SYSTEM EVALUATION

In this Chapter, an overall appraisal of the prototype DSS is conducted. Insights and recommendations are made for further improvements.

An Overall Appraisal

The overall appraisal of the developed prototype DSS is based on Shannon's (1975) six sufficiency criteria for total system evaluation. These six criteria are:

(1) The prototype should be goal or purpose directed.

(2) The prototype should be robust, in that it does not give absurd answers.

(3) The internal representation of the concepts should have "face validity" based on a priori knowledge, past research, and existing theory;

(4) The user interface should be appropriate. It should be easy for the user to control and manipulate;

(5) The prototype should be adaptive and flexible, having an easy procedure to modify and update representations; and

(6) The prototype must be complete on important issues.

The developed prototype DSS satisfies all the above criteria. First, the prototype DSS is goal-directed, because it has been built for the purpose of integrating quantitative and qualitative (textual) information to support unstructured decision making. More specifically, the system has been built to fulfil the two research objectives: (1) to build a conceptual DSS design framework to integrate a

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

text base and (2) to develop a workable and testable DSS prototype based on the conceptual design framework.

Secondly, the robustness of the prototype is evidenced by the testing results discussed in the previous Chapter. The results show that the prototype has the capability to provide integrated financial information (quantitative) and relevant textual information (qualitative) to support corporate performance analysis. The reliable and logically interconnected system functions allow the decision task of ratio construction or selection, financial data retrieval, ratio calculation, and textual information retrieval to be supported in an integrated working environment. By using an intelligent agent as the interface, the system is capable of flexibly interacting with the users in ratio construction, automatically determining the location of the financial data needed for ratio calculation, and intelligently deriving search keys for text retrieval. By using the causality-activated algorithm with one to two links of IS_A and IS_PART_OF search, the system has achieved a satisfactory precision and recall averaging over 80%.

Thirdly, the internal representation of the prototype has "face validity" based on a priori knowledge, past research, and existing theory. The premise that unstructured decision making needs integrated quantitative and qualitative information support is supported by empirical evidence provided by past research. The DSS design that integrates a text base is based on a thorough review of research and development in the related fields such as decision support systems, database management, and intelligent text processing. In the implementation of integrating

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

quantitative and qualitative information, a knowledge base was built based on a detailed domain analysis and some existing theories such as the event-driving accounting approach and the REA database modeling.

Fourthly, the user interface of the prototype is appropriate because it allows the user to control and manipulate the system with ease. The graphics-oriented menu interface does not require the user to memorize menu options and the flow of the functions. Figures built in the system help the user in capturing important principles and concepts in ratio analysis. The window-running screen design enables the user to solve the decision problems in an integrated working platform.

Fifthly, the prototype is adaptive and flexible, with an easy procedure to modify and update representations. On the information base side, the system allows companies, financial statements, ratios, knowledge, sources of document, and text to be added, modified, viewed, or deleted. Knowledge base can be updated without affecting the structure of the text base, because the text base is indexed by knowledge objects only. Text base can be indexed in two ways: when the text is imported or after new knowledge objects are added. On the system design side, FoxPro for Windows has the capability of expanding the prototype to a full-scale system. For example, the maximum number of records allowed for a relation is as large as 1 billion and the number of relations that can be operated is unlimited. Furthermore, FoxPro allows Dynamic Data Exchange (DDE) and Object Linking and Embedding (OLE) applications to be incorporated into the prototype.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Finally, the prototype is complete because all the important aspects of the proposed DSS design are fully implemented and tested. The issues, for example, include the possibility of integrating a text base with the traditional DSS design, the feasibility of adopting TBIS technologies for intelligent text processing and retrieval, the applicability of the proposed conceptual design framework, the compatibility of heterogeneous data models, the implementation practicality of the design from the conceptual level to the physical level, the cohesiveness of the developed system functions, the effectiveness of text retrieval, and so on.

Insights and Recommendations

This section discusses some of the insights learned from the system testing and recommendations for possible system improvements:

(1) *Data model compatibility.* The current system was developed by employing a uniform data model. The complexity of interfacing and communicating the four data sources (i.e., data, model, knowledge, and text), therefore, has been substantially reduced. For existing DSS not having such an advantage, cost-effectiveness will be the factor.

(2) *Insufficient knowledge engineering.* Based on the testing results discussed in the previous Chapter, the semantic network has demonstrated itself as a reliable and effective knowledge representation scheme of capturing business knowledge. However, it was also found that imperfect precision and recall was a result of the limitation of the semantic network. One of the major problems with the semantic

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

network is its use of binary representation which structures knowledge solely on a one-to-one relationship. To improve the precision and recall, rule-based knowledge engineering can be applied. For example, rule-base reasoning can tell whether a shipment contains finished products or raw material by manipulating a set of knowledge rules such as:

if (vendor) then (raw material) or if (customer) then (finished good).

(3) *Intestable text base.* Practically, the precision and recall can only be tested during the system development stage, in which the size of the text base is still manageable. To test whether a paragraph is relevant, the collected documents need to be manually reviewed. This review process may become impossible when the text base grows to a very large size. Hence, before the system is expanded to a full-scale system, alternatives that can improve the precision and recall need to be fully explored and tested.

(4) *Text base administration.* Text base is a large data source needed to be effectively managed. In analogy with the database administration, the text base needs to be administered through formal, organizational efforts. Before a document can be scanned and imported into the text base, it needs to be reviewed for relevancy and redundancy. Once the document is imported, the document in electronic form needs to be verified for accuracy. These can be done by a text base administrator (TBA).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

CHAPTER X

SUMMARY AND CONCLUSIONS

Summary of the Research

This dissertation attempts to explore the possibility of developing a DSS to automatically provide integrated quantitative and qualitative (textual) information support for unstructured decision making. A life cycle of system development is used as the research methodology to provide a systematic research process, in which the research problems are formulated, important concepts are integrated, the system design is substantiated, and the resultant system is validated.

The dissertation first reviews research and technologies related to the design and development of a DSS incorporating a text base. Based on such a review, a conceptual design framework is developed as a high-level actualization of the envisioned system. Followed by an analysis of the selected domain and knowledge construction, a logical design of the system is developed as a blue-print for prototyping. A prototype DSS is then developed, tested, and evaluated.

The two research questions have been answered by building the conceptual design framework presented in Chapter IV, and by developing such a workable and expandable DSS prototype described in Chapter VII, tested in Chapter VIII, and evaluated in Chapter IX. The prototype has demonstrated that it has the desired capabilities to provide integrated financial information (quantitative) and relevant textual information (qualitative) to support corporate performance analysis. The reliable and logically interconnected system functions allow the decision task of ratio

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

construction or selection, financial data retrieval, ratio calculation, and textual information retrieval to be supported in an integrated working environment. By using an intelligent agent as the interface, the system is capable of flexibly interacting with the users in ratio construction, automatically determining the location of the financial data needed for ratio calculation, and intelligently deriving search keys for text retrieval. By using the causality-activated algorithm with one to two links of IS_A and IS_PART_OF search, the system has achieved satisfactory precision and recall.

According to Shannon's (1974) criteria for system evaluation, this developed DSS prototype is goal-directed, robust, user-friendly, and internally validated, and designed with flexibility ad adaptivity. Experience and insight learned from prototyping the system provide valuable advice for further system improvements. The prototype can serve as an artifact for expanded and continuing research.

Limitations of the Research

Providing computer-integrated quantitative and qualitative information support requires substantial work in the consolidation of systems design and the integration of information technologies. This research serves only as an initial inquiry into this new research area. There are some limitations that are worthy of noting.

The major limitation of this research stems from the dynamic nature of information technologies. When the technologies used in this research become obsolete, the system design framework will have to be modified and updated.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Evolutionary technology may makes the system design outdated in a short term.

The second limitation of the research is inherited from the prototyping methodology itself. Since the prototype is only a try-out version of the final system, problems may be encountered when the system is later expanded or needed to interface with other systems.

Finally, the prototype is limited in its domain portability, which is also common to all knowledge-based systems. This prototype is designed to be used in the domain of corporate performance analysis. There is no assurance that the system can be easily ported to other domains without any difficulty.

Suggestions for Future Research

Further research in the following four directions should be promising:

First, the possibility of incorporating other types of TBIS technologies into the current DSS design can be explored. Hypertext technology can possibly be applied to structuring the retrieved text for more efficient review. This is important when the number of retrieved paragraphs gets very large. Advanced text analysis technologies can also be applied to support the text indexing process, where knowledge objects can be identified more effectively.

Secondly, the methodology of business knowledge engineering used in this research can be modified, improved, and expanded. Although the developed semantic network with three types of links provides satisfactory precision and recall, the possibility of using different types of links may be worthy of examining.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Thirdly, the developed prototype can be used as an instrument for behavioral study that focuses on the effectiveness of decision making that requires computer-integrated quantitative and qualitative information support. Empirical research in this direction is important, because DSS that incorporates textual information has rarely been found.

Finally, the possibility of using this prototype for educational purposes is also a fertile research area. Business courses, such as Financial Management and Corporate Policy, can utilize this system to teach students how to analyze corporate performance based on integrated quantitative and qualitative information in a simulated environment.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

LIST OF REFERENCES

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

LIST OF REFERENCES

Adams, D. A., Courtney, J. F., and Kasper, G. M. "A Process-oriented Method for the Evaluation of Decision Support System Generators," *Information and Management* (19), 1990, pp. 213-225.

Agarwal, T., Tanniru, M. R., and Dacruz, M. "Knowledge-based Support for Combining Qualitative and Qualitative Judgements in Resource Allocation Decision," *Journal of Management Information Systems* (9:1), Summer 1992, pp.165-184.

Akscyn, R. M., McCracken, D. L., and Yoder, E. A. "KMS: A Distributed Hypermedia System for Managing Knowledge in Organizations," *Communications of the ACM* (31:7), July 1988, pp. 820-835.

Aragon, G. A. *Financial Management*, Allyn and Bacon, Bonston, MA, 1989.

Ariav, G. and Ginzberg, M. J. "DSS Design: A Systematic View of Decision Support,." *Communications of the ACM* (28:10), October 1985, pp. 1045-1052.

Ayati, M. B. "A Unified Perspective on Decision Making and Decision Support Systems," *Information Processing and Management* (23:6), 1987, pp. 616-628.

Baldwin, D. "Principles of Design for a Multiple Viewpoint Problem Formulation Support Systems," Unpublished Doctoral Dissertation, Texas Tech University, 1989.

Barlow, J. F. "Structuring Informal Information," *Journal of Systems Management* (39:1), 1988, pp. 28-32.

Barnes, P. and Brown, C. "What the Chairmen Say: A Help or Hindrance?" *Accountancy*, November 1981, pp. 161-162.

Basden, A. "On the Application of Expert Systems," *International Journal of Man-Machine Studies* (19:5), 1983, pp. 461-477.

Benbasat, I. "An Analysis of Research Methodologies," In *The Information Systems Research Challenge*, W. F. McFarlan (ed.), Harvard Business School Press: Cambridge, MA, 1984, pp. 47-85.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Bieber, M. P. and Kimbrough, S. O. "On Generalizing the Concept of Hypertext," *MIS Quarterly* (16:1), March 1992, pp. 77-93.

Bingi, R., Khazanchi, D. and Yadav, S. B. "Comparative Analysis of Knowledge Representation Schemes," *Information Processing & Management* (31:2), 1995, pp.233-245.

Blair, D. C. and Gordon, M. D. "The Management and Control of Written Information," *Information & Management* (20), 1991, pp. 239-246.

Blake, S. P. *Managing for Responsive Research and Development*, W. H. Freeman and Company, San francisco, 1978.

Blanning, R. W. "A Relational Framework for Join Implementation in Model Management Systems," *Decision Support Systems* (1:1), January 1985, pp. 69-82.

Blanning, R. W. "A Survey of Issues in Expert Systems for Management," in *Expert Systems for Business*, Barry G. Silverman (ed.), Reading, MA: Addison-Wesley, 1987.

Blanning, R. W. "Conversing with Management Information Systems in Natural Language," *Communications of the ACM* (27:3), March 1984, pp.201-207.

Blanning, R. W. "Model Management Systems," *Decision Support Systems* (9:1), 1993, pp. 9-18.

Bonczek, R. H., Holsapple, C. W., and Whinston, A. B. "Computer-based Support of Organizational Decision Making," *Decision Science* (10:2), April 1979, 269-291.

Bonczek, R. H., Holsapple, C. W., and Whinston, A. B. "The Evolution from MIS to DSS: Extension of Data Management to Model Management." In *Decision Support Systems*, M.J. Ginzberg, W.R. Reitman, and E.A. Stohr (eds.), North-Holland, Amsterdam, 1982, pp. 61-78.

Bouwman, M. J. "Human Diagnostic Reasoning by Computer: An Illustration from Financial Analysis," *Management Science* (29:6), June 1983, PP. 653-672.

Burgin, R., and Dillion, M. "Improving Diambiguation in FASIT," *Journal of the American Society for Information Science* (43:2), March 1992, pp. 101-114.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Campagnoni, F. R. and Ehrlich, K. "Information Retrieval Using a Hypertext-based Help System," *ACM Transactions on Information Systems* (7:3), July 1989, pp. 271-291.

Carmel, E., McHenry, W. K., and Cohen, Y. "Building Large, Dynamic Hypertexts: How Do We Link Intelligently?" *Journal of Management Information Systems* (6:2), Fall 1989, pp. 33-50.

Chandrasekaran, B. and Mittal, S. "Deep Versus Compiled Knowledge Approaches to Diagnostic Problem Solving," *International Journal of Man-Machine Studies* (19:5), 1983, pp. 425-436.

Chang, A., Holsapple, C. W., and Whinston, A. B. "Model Management Issues and Directions," *Decision Support Systems* (9), 1993, pp. 19-37.

Chang, H. "Design of an Inquiry Support System for Managerial Problem Diagnosis," Unpublished Doctoral Dissertation, Texas Tech University, 1993.

Chen, P. P. "The Entity-relationship Model: Toward a Unified View of Data," *ACM Transactions on Database Systems*, March 1976, pp. 9-36.

Chugh, L. C. and Meador, J. W. "The Stock Valuation Process: The Analysts' View," *Financial Analysts Journal*, November/December 1984, pp. 41-48.

Coats, P. K. "Why Expert Systems Fail," *Financial Management*, Autumn 1988, pp. 77-86.

Cohen, P. R. and Kjeldsen, R. "Information Retrieval by Constrained Spreading Activation in Semantic Networks," *Information Processing & Management* (32:4), 1987, pp. 255-268.

Conklin, J. and Begeman, M. L. "gIBIS: A Tool for All Reasons," *Journal of the American Society for Information Science* (40:3), May 1989, pp. 200-211.

Conlon, S., Reithel, B., Aiken, M., and Shirani, A. "A Natural Language Processing Based Group Decision Support System," *Decision Support Systems* (12:3), July 1994, pp. 181-188.

Covaleski, M. A., Dirsmith, M. W., and Ketz, J. E. "The Impact of Qualitative Factors in Corporate Performance Appraisal: Implications for Managers," *The Journal of Applied Business Research* (7:3), Summer 1991, pp. 8-14.

:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Courtney, J. F., Paradice, D. B., and Mohammed, A. "A knowledge-based DSS for Managerial Problem Diagnosis," *Decision Sciences* (18:3), 1987, pp. 373-399.

Crocker, J., "Judgment of Covariation by Social Perceivers," *Psychological Bulletin* (90:2), 1981, pp. 272-292.

Croft, W. B. and Thomson, R. H. "I^3R: A New Approach to the Design of Document Retrieval Systems," *Journal of the American Society for Information Science* (38:6), November 1987, pp. 389-404.

Daft, R. L. and Lengel, R. H. "Organizational Information Requirements, Media Richness, and Structural Design," *Management Science* (32:5), 1986, pp. 554-571.

Dalal, N. P. and Yadav, S. B. "The Design of a Knowledge-based Decision Support System to Support the Information Analyst in Determining Requirements," *Decision Science* (23:6), November/December 1992, pp. 1373-1388.

Date, C. J. *An Introduction for Database Systems*, Addison-Wesley, MA, 1977.

Davis, G. B. and Olson, M. H. *Management Information Systems: Conceptual Foundations, Structure, and Development*, McGraw-Hill, New York, 1985.

Denna, E. L., Cherrington, J. O., Andros, D. P., and Hollander, A. S. *Event-driven Business Solutions: Today's Revolution in Business and Information Technology*, Business One Irwin, Homewood, IL, 1993.

Denning, P.J. et al. "Computing as a Discipline." *Communications of the ACM* (32:3), January 1989, pp. 9-23.

Dewire, D. T. *Text Management*, McGraw-Hill, NY, 1994.

Dickson, G. W., Senn, J. A., and Chervany, N. L. "Research in Management Information Systems: The Minnesota Experiments," *Management Science* May 1977, pp. 913-923.

Duda, R. O. "State of Technology in Artificial Intelligence." in *Research Directions in Software Technology*, P. Wenger (ed.), MIT Press, Cambridge, MA, 1979, pp. 729-749.

EDP Analyzer. "Tools for Building on EIS," *EDP Analyzer* (17:12), 1979.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Emory, C. W. and Cooper, D. R. *Business Research Methods*, Irwin, Homewood, IL, 1991.

Flores, F., Graves, M., Hartfield, B., Winograd, T. "Computer Systems and the Design of Organization Interaction," *ACM Transaction on Office Information Systems* (6:2), April 1988, pp. 153-172.

Fox, E. A. "Development of the CODER System: A Testbed for Artificial Intelligence Methods in Information Retrieval," *Information Processing and Management* (23:4), 1987, pp. 341-366.

Frisse, M. E., and Cousins, S. B. "Guides for Hypertext: An Overview," *Artificial Intelligence* (2), 1990, pp. 303-314.

Galbraith, J. R. "Organization Design: an Information Processing View," *Interfaces,* (4:3), 1974, pp. 23-36.

Galliers, R. D. and Land, E. F. "Choosing Appropriate Information Systems Research Methodologies," *Communications of the ACM* (32:5), May 1989, pp. 594-605.

Ginsberg, A. "A Unified Approach to Automatic Indexing and Information Retrieval," *IEEE Expert* (8:5), October 1993, pp. 46-56.

Gorry, G. A. and Scott-Morton, M. S. "A Framework for Management Information Systems," *Sloan Management Review* (13:1), Winter 1971, pp. 55-70.

Goul, M., Shane, B., and Tonge, F. "Using a Knowledge-based Decision Support System in Strategic Planning Decisions: An Empirical Study," *Management Information Systems* (2:4), 1986, pp. 70-84.

Grishman R. "PROTEUS report," in *Proceedings DARPA Speech and Natural Language Workshop,* Cape Cod, MA, 1989.

Halasz, F. G. 'Reflections on Notecards: Seven issues for the Next Generation of Hypermedia Systems," *Communications of the ACM* (31:7), July 1988, pp. 836-855.

Hamilton, S. and Ives, B. "MIS Research Strategies," *Information & Management* (13), December 1982, pp. 339-347.

Hawkins, D. T. "Growth Trends in the Electronic Information Services Market: Part 2," *ONLINE*, September 1993, pp. 105-108.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Hobbs, J. R., Applelt, D. E., Bear, J., Tyson, M., and Magerman, D. "Robust Processing of Real-world Natural Language Texts. In *Text-Based Intelligent Systems: Current Research and Practice in Information Retrieval.* P. S. Jacobs (eds.). Lawrence Erlbaum, Hillsdale, NJ, 1992, pp. 13-34.

Hsu, C., Bouzine, M., and Rattner, L. "Information Resources Management in Heterogeneous, Distributed Environments: A Metadatabase Approach," *IEEE Transactions on Software Engineering* (17:6), June 1991, pp. 604-625.

Houghton, K. A. and Woodliff, D. R. "Financial Ratios: The Prediction of Corporate 'Success' and Failure," *Journal of Business Finance & Accounting* (14:4), Winter 1987, pp. 537-554.

Huber, G. P. "The Nature of Design of Post-industrial Organizations," *Management Science* (30:8), 1984, pp. 928-951.

Huber, G. P. "A Theory of the Effects of Advanced Information Technologies on Organizational Design, Intelligence, and Decision Making," *Academy of Management Review* (15:1), 1989, pp. 47-71.

Huber, G. P. and McDaniel, R. R. "The Decision-making Pparadigm of Organizational Design," *Management Science* (32:5), 1986, pp. 572-589.

Ijiri, Y. *Theory of Accounting Measurement*, American Accounting Association, 1975.

Ives, B., Hamilton, S. and Davis, G. B. "A Framework for Research in Computer-based Management Information Systems," *Management Sciences* (26:9), September 1980, pp. 910-934.

Jacobs, P. S. "Introduction: Text Power and Intelligent Systems," In *Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval.* P. S. Jacobs (eds), Lawrence Erlbaum, Hillsdale, NJ , 1992, pp. 1-7.

Jacobs, P. S. and Rau, L. F. "SCISOR: Extracting Information from On-line News," *Communication of the ACM* (33:11), November 1990, pp. 88-97.

Jackson, I. F. *Corporate Information Management*, Prentice-Hall, Englewood Cliffs, NJ, 1986.

Johnson, O. "Toward an 'Event' Theory of Accounting," *The Accounting Review* (45:4), October 1970, pp. 641-653.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Johnson, H. T. and Kaplan, R. S. "Management by Accounting is not Management Accounting," *The Accounting Systems Journal*, Fall 1989, pp. 14-17.

Jones, J. W. and McLeod, R. "The structure of executive information systems: An exploratory analysis." *Decision Sciences*, 17, 2, '986, 220-249.

Keen, P. G. W. and Scott Morton, M. S. *Decision Support Systems*, Addison-Wesley: Redding, MA, 1978.

Kieso, D. E. and Weygandt, J. J. *Intermediate Accounting*, John Wiely & Sons, NY, 1983.

Kirshnamurthy, V., Su, Y. W., and Lam, H. "IMDAS - An Integrated Manufacturing Data Administration System," *Data & Knowledge Engineering* (3:4), 1988, pp. 109-131.

Laitinen, E. K. "The Use of Information Contained in Annual Reports and Prediction of Small Business Failures," *International Review of Financial Analysis* (2:3), 1993, pp. 153-166.

Lieberman, A. Z. and Whinston, A. B. "A Structuring of an Events-accounting Information System," *The Accounting Review* (50:2), April 1975, pp. 246-258.

Liu, J. I. C.; Yun, D. Y. Y.; and Klein. "An Agent for Intelligent Model Management," *Journal of Management Information Systems* (7:1), Summer 1990, pp. 101-122.

Lucas, H. "Empirical Evidence for a Descriptive Model of Implementation," *MIS Quarterly* (2:1), June 1978, pp. 27-42.

Lyytinen, K. J. "Implications of Theories of Language for Information Systems," *MIS Quarterly* (9:1), 1985, pp. 61-74.

Mann, J. "Software to Manage the Paper Mountain," *DATAMATION* (37:14), July 15, 1991, pp. 79-80.

Mattessich, R. *Accounting and Analytical Methods: Measurements and Projection of Income and Wealth in the Micro- and Macro- Economy*, Scholar Book, Houston, TX, 1977.

McCarthy, W. E. "An Entity-relationship View of Accounting Models," *The Accounting Review* (54:4), October 1979, pp. 667-687.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

McGraw, K. L. and Harbison-Briggs, K. *Knowledge Acquisition: Principles and Guidelines*, Prentice Hal, Englewood Cliffs, NJ, 1989.

Medina-Mora, R. Winograd, T., Flores, R., and Flores, F. "The Action Workflow Approach to Workflow Management Technology," *The Information Society* (9), 1992, pp. 391-404

Minch, R. P. "Application and Research Areas for Hypertext in Decision Support Systems," *Journal of Management Information Systems* (6:2), Winter 1989-1990, pp. 119-138.

Mintzberg, H. *Impediments to the Use of Management Information*, National Association of Accountants, NY, 1975.

Mintzberg, H. Raisinghani, D., and Theoret, A. "The Structure of 'Unstructured' Decision Processes," *Administrative Science Quarterly* (21), June 1976, pp. 246-275.

Morrow, V. *Handbook of Financial Analysis for Corporate Managers*, Prentice Hall, Englewood Cliffs, NJ, 1991.

Naumann, J. D. and Jenkins, A. M. "Prototyping: The New Paradigm for System Development," *MIS Quarterly* (6:3), September 1982, pp. 29-44.

Nagy, R. A. and Obenberger, R. W. "Factors Influencing Individual Investor Behavior," *Financial Analyst Journal* (5:4), 1994, pp. 63-69.

Newell, A. and Simon, H. *Human Problem Solving*, Prentice-Hall, Englewood Cliffs, NJ, 1972.

Nunamaker, J. F., Chen, M., and Purdin, T. D. M. "Systems Development in Information Systems Research," *Journal of Management Information Systems* (7:3), Winter 1990-1991, pp. 89-106.

Orman, L. "Functional Semantics of Accounting Data," *The Accounting Systems Journal*, Fall 1990, pp. 6-29.

Paice, C. D. "Construction of Literature Abstracts by Computer: Techniques and Prospects," *Information Processing & Management* (26:1), 1990, pp. 171-186.

Palmer, M. S., Passonneau, R. J., Weir, C., and Finin, T. "The KERNEL: Text Understanding System," *Artificial Intelligence* (63:1-2), October 1993, pp. 17-68.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Paradice, D.B., "Causal Modeling as a- Basis for the Design of an Intelligent Business Problem Formulation System," Unpublished Doctoral Dissertation, Texas Tech University, 1986.

Paradice, D. B. and Courtney, J. F. "Organizational Knowledge Management," *Information Resources Management Journal* (2:3), 1989, pp. 1-13.

Parker, B. J. and Al-Utaibi, G. A. "Decision Support Systems: The Reality that Seems Hard to Accept?" *OMEGA International Journal of Management Science* (14:2), 1986, pp. 135-143.

Pau, L. F. and Gianotti, C. *Economic and Financial Knowledge-Based Processing*, Springer-Verlag, NY, 1990.

Peel, M. J., Peel, D. A., and Pope. P. F. "Predicting Corporate Failure: Some results for the UK Corporate Sector." *Omega* (14:1), 1986, pp. 5-12.

Pollitt, A. S. "CANSEARCH: An Expert Systems Approach to Document Retrieval," *Information Processing and Management* (23:2), 1987, pp. 119-136.

Pritchard-Schoch, T. "Natural Language Comes of Age," *ONLINE*, May 1993, pp. 33-43.

Rappaport, A. "Stock Market Signals to Managers," *Harvard Business Review* (87:6), November/December 1987, pp. 57-62.

Raymond, D. R. and Tompa, F. W. "Hypertext and the Oxford English Dictionary," *Communications of the ACM* (31:7), July 1988, pp. 871-877.

Remus, W. and Kottemann, J. E. "Semi-structured Recurring Decisions: An Experimental Study of Decision Making Models and Some Suggestions for DSS," *MIS Quarterly* (11:2), June 1987, pp. 233-243.

Riecken, D. "Intelligent Agents," *Communications of the ACM* (37:7), July 1994, pp. 18-21.

Ross, D.T. and Schoman, K.E., "Structured Analysis for Requirements Definition," *IEEE Transactions on Software Engineering* (3:1), 1977, pp. 6-15.

Salton, G. *The Smart Retrieval System - Experiments in Automated Document Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1971.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Salton, G., Allan, J., and Buckley, C. Automatic Structuring and Retrieval of Large Text Files," *Communications of the ACM* (37:2), February 1994, pp. 97-108.

Samstag-Schnock, U. and Meadow, C. T. "PBS: An Economical Natural Language Query Interpreter," *Journal of the American Society for Information Science* (44:5), 1993, pp. 265-272.

Senn, J. A. *Analysis and Design of Information Systems*, McGraw-Hill, NY, 1984.

Shank, R. C. "Identification of Conceptualizations Underlying Natural Language," In *Computer Models of Thought and Language*, R. C. Schank and K. M. Colby (eds.),W. H. Freeman and Co., San Francisco, 1973.

Shoval, P. "Principles, Procedures and Rules in an Expert System for Information Retrieval," *Information Processing and Management* (21:6), 1985, pp. 475-487.

Simon, H. A. *The New Science of Management Decision*, Harper and Brothers, New York, 1960.

Simon, H. A. "Applying Information Technology to Organization Design," *Public Administration* Review (33:3), 1973, pp. 268-278.

Simon, H. A. and Newell, A. "Human Problem Solving: The State of the Theory in 1970," *American Psychologist*, February 1971, pp. 145-159.

Smith, P. J., Shute, S. J., Galdes, D., and Chignell, M. H. "Knowledge-based Search Tactics for an Intelligent Intermediary System," *ACM Transactions on Information Systems* (7), 1989, pp. 246-270.

Sorter, G. H. "An 'Event' Approach to Basic Accounting Theory," *The Accounting Review* (44:1), January 1969, pp. 12-19.

Sprague, C. *The Philosophy of Accounts*, Scholars Book Co., Lawrence, Kansas, 1972.

Sprague, R. H. and Carlson, E. D. *Building Effective Decision Support Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

Steele, A. "The Accuracy of Chairmen's Nonqualified Forecasts: An Exploratory Study," *Accounting and Business Research* (12), Summer 1982, pp. 215-230.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Suh, E. H. and Hinomoto, H. "Use of a Dialog Base for Integrated 'Relational' Decision Support Systems," *Decision Support Systems* (5), 1989, pp. 277-286.

Templeton, M., Brill, D., Dao, S., Lund, E., Ward, P., Chen, A., and MacGregor, R. "MERMAID - A Front-end to Distributed Heterogeneous Database," in *Proceedings of IEEE* , 1987, pp. 695-708.

Tennyson, B. M., Ingram, R. W., and Dugan, M. T. "Assessing the Information Content of Narrative Disclosures in Explaining Bankruptcy," *Journal of Business Finance and Accounting* (17), Summer 1990, pp. 391-410.

Tong, R. M. and Shaprio, D. G. "Experimental Investigations of Uncertainty in a Rule-based System for Information Retrieval," *International Journal of Man-Machine Studies* (22), 1985, pp. 265-282.

Turban, E. and Watkins, P. R. "Integrating Expert Systems and Decision Support Systems," *MIS Quarterly* (10:3), June 1986, pp. 121-136.

Turtle, H., and Croft, W. B. "Inference Networks for Document Retrieval," in *Proceedings of he13th Annual International Conference Research and Development in Information Retrieval*, ACM, New York, 1990, 1-24.

Van Horn, R. "Empirical Studies in Management Information Systems," *Data Base*, Winter 1973, pp. 172-180.

Vickery, A. and Brooks, H. M. "PLEXUS - The Expert System for Referral." *Information Processing & Management* (23:2), 1987, pp. 99-117.

Weber, E. S. "Systems to Think with: A Response to 'A Vision for Decision Support Systems'." *Journal of Management Information Systems* (2:4), 1986, pp. 85-97.

Winograd, T. "A Language/action Perspective on the Design ofCcooperative Work," *Human-Computer Interaction* (3:1), 1977/1978, pp. 3-30.

Wolfe, M. "Development of the City of Quality: A Hypertext-based Group Decision Support System for Quality Function Deployment," *Decision Support Systems*, (11), May 1994, pp. 299-318.

Wu, W. and Dilt, D. M. "Integrating Diverse CIM Data Bases: The Role of Natural Language Interface," *IEEE Transaction on Systems, Man and Cybernetics* (22:6), November/December, 1992, pp. 1331-1347.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Young, L. F. "Knowledge-based Systems for Idea Processing Support," *DATA BASE*, Fall 1990, pp. 27-33.

Yu, S. C. *The Structure of Accounting Theory*, The University Presses of Florida, 1976.

Zahedi, F. *Intelligent Systems for Business: Expert Systems with Neural Networks*, Wadsworth Publishing, Belmont, CA, 1993.

Zarri, G. P. "An Outline of the Representation and Use of Temporal Data in the RESEDA System," *Information Technology: Research and Development* (2), 1983, pp. 89-108.

Zarri, G. P. "A Knowledge Representation Language for Large Knowledge Bases and 'Intelligent' Information Retrieval Systems," *Information Processing & Management* (26:3), 1990, pp. 349-370.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

APPENDIX A

A List of Ratio Components

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Balance Sheet	Income Statem
CASH	NET SALES
MRKTABLE SECURITIES	COST OF GOODS
RECEIVABLES	GROSS PROFIT
INVENTORIES	R & D EXPENDITUR
RAW MATERIALS	SELL GEN & ADMI
WORK IN PROGRESS	INC BEF DEP & AM
FINISHED GOODS	DEPRECIATION & A
NOTES RECEIVABLE	NON-OPERATING I
OTHER CURRENT ASSETS	INTEREST EXPENS
TOTAL CURRENT ASSETS	INCOME BEFORE T
PROP, PLANT & EQUIP	PROV FOR INC TAX
ACCUMULATED DEP	MINORITY INT (INC
NET PROP & EQUIP	INVEST GAINS/LOS
INVEST & ADV TO SUBS	OTHER INCOME
OTHER NON-CUR ASSETS	NET INC BEF EX IT
DEFERRED CHARGES	EX ITEMS & DISC O
INTANGIBLES	NET INCOME
DEPOSITS & OTH ASSET	
TOTAL ASSETS	
NOTES PAYABLE	
ACCOUNTS PAYABLE	
CUR LONG TERM DEBT	
CUR PORT CAP LEASES	
ACCRUED EXPENSES	
INCOME TAXES	
OTHER CURRENT LIAB	
TOTAL CURRENT LIAB	
MORTGAGES	
DEFERRED CHARGES/INC	
CONVERTIBLE DEBT	
LONG TERM DEBT	
NON-CUR CAP LEASES	
OTHER LONG TERM LIAB	
TOTAL LIABILITIES	
MINORITY INT (LIAB)	
PREFERRED STOCK	
COMMON STOCK NET	
CAPITAL SURPLUS	
RETAINED EARNINGS	
TREASURY STOCK	
OTHER EQUITIES	
SHAREHOLDER EQUITY	
TOT LIAB & NET WORTH	
FISCAL YEAR ENDING	

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

APPENDIX B

A List of Documents Collected in the Text Base

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Source#	Source Name	Number Documents Coll
AA	Advertising Age	1
AB	American Bankers	4
BA	Barron's	3
BI	Business Insurance	1
BM	Business Marketing	4
BW	Business Week	20
CC	J. of Commerce and Comme	1
CI	Computer Industry Report	1
FB	Forbe	5
FT	Fortune	1
FW	Financial World	1
IF	Interface	1
MD	Management Disscusion	6
MO	Money	1
MT	Management Today	1
NU	National Underwriter	1
NY	New York Times	1
PL	President's Letter	3
TE	The Economist	2
VL	Value Line	6
WS	Wall Street Journal	4
	Total	68

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

APPENDIX C

A Partial List of Knowledge Objects

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

REA Type*	Object Form 1	Object Form 2	Object Form 3
1	asset	assets	
1	cash		
1	equipment	equipments	
1	inventory	inventories	
1	finished good	finished goods	
1	current asset	current assets	
1	current liability	current liabilities	currentl liab
1	currency	currencies	
1	coin	coins	
1	fund	funds	
1	bad debt	bad debts	
2	accounts receivable	accounts receivables	
2	depreciation method	depreciation methods	
2	accounts payable	accounts payables	
2	aging method	aging methods	
2	credit policy	credit policies	
2	allowance method	allowance methods	
2	credit sales		
2	cash sales		
2	aging schedule	aging schedules	
2	aged balance		
2	amortization	amortizing	
2	cash purchase	cash purchases	
3	creditor	creditors	
3	marketing force	marketing forces	
3	president	presidents	
3	board of directors		
3	customer	customers	
3	ceo	chief executive officer	chief executive
3	general manager	general managers	gm
3	consumer	consumers	
3	distributor	distributors	
3	chief operating officer		
3	chief financial officer	chief financial officers	
3	corporate client	corporate clients	

* 1 = Resource 2 = Event 3 = Agent

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

APPENDIX D

A Partial List of Knowledge Predicates

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Predicate Type*	Object A	Object B
1	advertising expenditure	advertising expense
1	bankruptcy	business failure
1	capital expense	capital expenditure
1	cash	money
1	ceo	executive
1	client	customer
1	common share	common stock
1	competitor	rival
1	consumer	customer
1	consumer	shopper
1	corporate goal	company goal
2	accounts receivable	receivable
2	administrative expense	operating expense
2	advertising	marketing strategy
2	building	asset
2	cash	current asset
2	cash cow	marketing strategy
2	cash sales	sales
2	coin	cash
2	credit sales	sales
2	current asset	working capital
2	current liability	working capital
2	customer satisfaction	customer
3	accounts recievable	aging method
3	accounts recievable	allowance method
3	accounts recievable	credit sales
3	asset	depreciation method
3	asset	capital expenditure
3	bad debt	credit policy
3	bad debt	uncollectible recievable
3	bank deposit	cash
3	capital expenditure	expense
3	cash	accounts receivable
3	credit loss	bad debt
3	currency	exchange rate

* 1 = IS_A 2 = IS_PART_OF 3 = IS_CAUSED_BY

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

APPENDIX E

Individual Testing Results

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Company: Digital Equipment Corp.
Ratio: Quick

Algorithm	Causality-activated					Association-activated					Attribute-activated				
Measure / Parameter	Precision		Recall		Speed	Precision		Recall		Speed	Precision		Recall		Speed
	Result	%	Result	%		Result	%	Result	%		Result	%	Result	%	
1 : 1 : 1	47/58	81.0	47/51	92.2	4.38	47/62	75.8	4/51	92.2	5.93	46/58	79.3	5/51	90.2	4.39
1 : 2 : 2	48/59	81.4	48/51	94.1	8.26	48/63	76.2	3/51	94.2	10.19	47/59	79.7	4/51	92.2	5.98
1 : 3 : 3	49/60	81.7	49/51	96.1	13.01	49/64	76.6	2/51	96.1	15.12	48/60	80.0	3/51	94.1	7.97
1 : 4 : 4	48/60	81.7	48/51	94.1	16.79	49/64	76.6	2/51	96.1	17.68	48/60	80.0	3/51	94.1	10.21
2 : 1 : 1	48/64	75.0	49/51	96.1	5.84	48/91	52.8	3/51	94.2	11.56	48/90	53.3	3/51	94.1	6.53
2 : 2 : 2	49/65	75.4	49/51	96.1	11.1	48/103	46.7	3/51	94.2	14.23	48/112	47.1	3/51	94.1	8.97

Company: Digital Equipment Corp.
Ratio: Gross Working Capital

Algorithm	Causality-activated					Association-activated					Attribute-activated				
Measure / Parameter	Precision		Recall		Speed	Precision		Recall		Speed	Precision		Recall		Speed
	Result	%	Result	%		Result	%	Result	%		Result	%	Result	%	
1 : 1 : 1	43/54	79.6	43/44	97.7	4.46	43/58	74.1	44/44	97.7	4.79	43/55	78.1	43/44	97.7	3.59
1 : 2 : 2	44/55	80.0	44/44	100	7.71	44/59	74.5	44/44	100	8.19	44/56	78.5	44/44	100	5.21
1 : 3 : 3	44/55	80.0	44/44	100	11.07	44/59	74.5	44/44	100	11.80	44/56	78.5	44/44	100	6.79
1 : 4 : 4	44/55	80.0	44/44	100	14.13	44/59	74.5	44/44	100	15.33	44/56	78.5	44/44	100	8.38
2 : 1 : 1	43/81	53.1	43/44	97.7	6.03	43/98	43.8	43/44	97.7	6.06	43/98	43.9	43/44	97.7	6.04
2 : 2 : 2	44/52	53.6	44/44	100	10.85	44/99	44.4	44/44	100	7.63	44/99	44.4	44/44	100	7.89

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Company: Digital Equipment Corp.
Ratio: Receiable Turnover

Algorithm	Causality-activated					Association-activated					Attribute-activated				
Measure / Parameter	Precision		Recall		Speed	Precision		Recall		Speed	Precision		Recall		Speed
	Result	%	Result	%		Result	%	Result	%		Result	%	Result	%	
1 : 1 : 1	58.61	95.0	10/68	85.2	4.73	56/74	75.6	56/68	82.3	3.77	56/73	76.7	56/68	82.3	2.73
1 : 2 : 2	59.62	95.1	59/68	86.7	7.68	56/74	75.6	56/68	82.3	5.30	56/73	76.7	56/68	82.3	3.37
1 : 3 : 3	61/79	77.2	61/68	89.7	12.51	56/74	75.6	56/68	82.3	8.16	56/74	75.6	56/68	82.3	3.91
1 : 4 : 4	61/70	87.1	61/68	89.7	17.64	56/74	75.6	56/68	82.3	11.25	56/74	75.6	56/68	82.3	4.81
2 : 1 : 1	63/107	58.9	63/68	92.6	7.66	70/101	59.4	60/68	88.2	5.83	57/90	63.3	57/68	83.8	4.22
2 : 2 : 2	63/107	58.9	63/68	92.6	14.01	70/101	59.4	60/68	88.2	8.85	57/90	63.3	57/68	83.8	4.78

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Company: Digital Equipment Corp.
Ratio: Gross Profit Margin

Algorithm	Causality-activated					Association-activated					Attribute-activated				
Measure / Parameter	Precision		Recall		Speed	Precision		Recall		Speed	Precision		Recall		Speed
	Result	%	Result	%		Result	%	Result	%		Result	%	Result	%	
1 : 1 : 1	91/92	98.9	91/101	90.1	5.17	88/89	98.8	88/101	87.1	3.97	77/78	98.7	77/101	76.2	3.08
1 : 2 : 2	91/93	97.8	91/101	90.1	8.93	88/89	98.8	88/101	87.1	5.61	77/78	98.7	77/101	76.2	3.29
1 : 3 : 3	91/93	97.8	91/101	90.1	12.65	88/89	98.8	88/101	87.1	6.86	77/78	98.7	77/101	76.2	3.73
1 : 4 : 4	91/93	97.8	91/101	90.1	16.38	88/89	98.8	88/101	87.1	8.26	77/78	98.7	77/101	76.2	4.05
2 : 1 : 1	97/106	9105	97/101	96.0	8.77	94/99	94.9	94/101	93.0	6.23	90/95	94.7	90/10	89.1	4.15
2 : 2 : 2	97/106	91.5	97/101	96.0	15.67	94/99	94.9	94/101	93.0	8.60	90/95	94.7	90/101	89.1	4.80

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Company: Digital Equipment Corp.
Ratio: Earning Per Share

Algorithm	Causality-activated					Association-activated					Attribute-activated				
Measure / Parameter	Precision		Recall		Speed	Precision		Recall		Speed	Precision		Recall		Speed
	Result	%	Result	%		Result	%	Result	%		Result	%	Result	%	
1 : 1 : 1	65/67	97.0	65/70	92.8	3.60	61/63	96.8	61/70	87.1	2.83	61/63	96.8	61/70	87.1	2.66
1 : 2 : 2	65/67	97.0	65/70	92.8	5.77	61/63	96.8	61/70	87.1	3.52	61/63	96.8	51/70	87.1	2.74
1 : 3 : 3	65/67	97.0	65/70	92.8	7.88	61/63	96.8	61/70	87.1	4.02	61/63	96.8	51/70	87.1	3.38
1 : 4 : 4	65/67	97.0	65/70	92.8	10.0	61/63	96.8	61/70	87.1	4.63	61/63	96.8	51/70	87.1	3.63
2 : 1 : 1	68/90	75.5	68/70	97.1	5.56	64/86	74.4	64/70	91.4	4.01	64/86	74.4	64/70	91.4	3.58
2 : 2 : 2	68/90	75.5	68/70	97.1	8.62	64/86	74.4	64/70	91.4	5.10	64/86	74.4	64/70	91.4	3.92

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Company: Sun Microsystems
Ratio: Quick

Algorithm	Causality-activated					Association-activated					Attribute-activated				
Measure / Parameter	Precision		Recall		Speed	Precision		Recall		Speed	Precision		Recall		Speed
	Result	%	Result	%		Result	%	Result	%		Result	%	Result	%	
1 : 1 : 1	27/30	90.0	27/29	93.1	4.20	28/33	84.85	28/29	96.55	5.12	27/30	90.0	27.29	93.1	3.64
1 : 2 : 2	28/32	87.5	28/29	96.6	8.34	29/35	82.86	29/29	100	9.61	28/32	87.5	28.52	96.5	5.33
1 : 3 : 3	28/33	84.85	28/29	96.6	12.36	29/36	80.56	29/29	100	14.49	29/33	84.85	28.29	96.5	7.19
1 : 4 : 4	28/33	84.85	28/29	96.6	16.72	29/36	80.56	29/29	100	18.96	29/33	84.85	28.29	96.5	9.36
2 : 1 : 1	27/36	75.0	27/29	93.10	5.65	28/57	49.12	28/29	96.5	7.60	28/56	50.0	25/29	96.5	5.98
2 : 2 : 2	28/38	73.68	28/29	96.55	10.89	29/69	42.03	29/29	100	13.36	29/69	42.0	29/29	100	8.31

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

170

Company: Sun Microsystems
Ratio: Gross Working Capital

Algorithm	Causality-activated					Association-activated					Attribute-activated				
Measure / Parameter	Precision		Recall		Speed	Precision		Recall		Speed	Precision		Recall		Speed
	Result	%	Result	%		Result	%	Result	%		Result	%	Result	%	
1 : 1 : 1	23/26	88.4	23/26	88.4	3.67	23/27	85.1	23/26	88.4	4.55	23/26	88.4	23/26	88.4	3.28
1 : 2 : 2	24/28	85.7	24/26	92.3	7.24	24/29	82.7	24/26	92.3	7.87	24/28	85.7	24/26	92.3	4.79
1 : 3 : 3	24/28	85.7	24/26	92.3	10.71	24/29	82.7	24/26	92.3	11.7	24/28	85.7	24/26	92.3	6.40
1 : 4 : 4	24/28	85.7	24/26	92.3	14.13	24/29	82.7	24/26	92.3	15.19	24/28	85.7	24/26	92.3	8.03
2 : 1 : 1	23/45	51.1	23/26	88.4	5.72	23/62	37.10	23/26	88.4	6.95	23/62	37.10	23/26	88.4	6.19
2 : 2 : 2	23/47	51.06	24/26	92.3	10.55	24/64	37.50	24/26	92.3	11.49	24/64	37.50	24/26	92.3	7.46

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Company: Sun Microsystems
Ratio: Receiable Turnover

Algorithm	Causality-activated					Association-activated					Attribute-activated				
Measure / Parameter	Precision		Recall		Speed	Precision		Recall		Speed	Precision		Recall		Speed
	Result	%	Result	%		Result	%	Result	%		Result	%	Result	%	
1 : 1 : 1	45/49	91.8	45/49	91.8	4.45	41/52	78.8	41/49	83.6	3.59	41/50	82.0	41/49	83.6	2.64
1 : 2 : 2	45/49	91.8	45/49	91.8	7.88	41/52	78.8	41/49	83.6	5.15	41/50	82.0	41/49	83.6	2.97
1 : 3 : 3	45/60	75.0	45/49	91.8	12.45	41/53	77.3	41/49	83.6	7.99	41/51	80.3	41/49	83.6	3.58
1 : 4 : 4	45/60	75.0	45/49	91.8	17.81	41/53	77.3	41/49	83.6	11.32	41/51	80.3	41/49	83.6	4.57
2 : 1 : 1	47/76	61.9	47/49	95.9	7.56	44/69	63.7	44/49	89.9	5.51	42/61	68.8	42/49	85.7	3.74
2 : 2 : 2	47/76	61.9	47/49	95.9	14.16	44/69	63.7	44/49	89.9	8.57	42/61	68.8	42/49	85.7	4.26

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Company: Sun Microsystems
Ratio: Gross Profit Margin

Algorithm	Causality-activated					Association-activated					Attribute-activated				
Measure / Parameter	Precision		Recall		Speed	Precision		Recall		Speed	Precision		Recall		Speed
	Result	%	Result	%		Result	%	Result	%		Result	%	Result	%	
1 : 1 : 1	64/67	95.5	64/69	92.7	4.67	60/61	98.3	60/69	86.9	3.76	51/52	98.0	18/69	73.9	2.66
1 : 2 : 2	64/67	95.5	64/69	92.7	8.71	60/61	98.3	60/69	86.9	5.29	51/52	98.0	18/69	73.9	2.92
1 : 3 : 3	64/67	95.5	64/69	92.7	12.69	60/61	98.3	60/69	86.9	6.63	60/63	95.2	60/69	86.9	4.05
1 : 4 : 4	64/67	95.5	64/69	92.7	16.61	60/61	98.3	60/69	86.9	8.34	60/63	95.2	60/69	86.9	4.60
2 : 1 : 1	66/73	90.4	66/69	95.6	8.35	62/66	93.9	62/69	89.8	5.86	57/61	93.4	57/69	82.6	4.16
2 : 2 : 2	66/73	90.4	66/69	95.6	15.63	62/66	93.9	62/69	89.8	8.44	57/61	93.4	57/69	82.6	4.41

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Company: Sun Microsystems
Ratio: Earning Per Share

Algorithm	Causality-activated					Association-activated					Attribute-activated				
Measure / Parameter	Precision		Recall		Speed	Precision		Recall		Speed	Precision		Recall		Speed
	Result	%	Result	%		Result	%	Result	%		Result	%	Result	%	
1 : 1 : 1	50/51	98.0	50/52	96.1	3.54	43/43	100	43/52	82.6	2.80	43/43	100	43/52	82.6	2.62
1 : 2 : 2	51/52	98.0	51/52	98.0	5.63	43/43	100	43/52	82.6	3.43	43/43	100	43/52	82.6	2.93
1 : 3 : 3	51/52	98.0	51/52	98.0	8.02	43/43	100	43/52	82.6	4.07	43/43	100	43/52	82.6	3.23
1 : 4 : 4	51/52	98.0	51/52	98.0	10.26	43/43	100	43/52	82.6	4.71	43/43	100	43/52	82.6	3.63
2 : 1 : 1	51/62	82.2	51/52	98.0	5.36	45/55	81.8	45/52	86.5	3.82	45/55	81.8	45/52	86.5	3.24
2 : 2 : 2	52/63	82.5	52/52	100	8.87	45/55	81.8	45/52	86.5	4.77	51/62	82.2	51/52	98.0	5.18

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Company: Sun Microsystems
Ratio: Earning Per Share

Algorithm	Causality-activated					Association-activated					Attribute-activated				
Measure / Parameter	Precision		Recall		Speed	Precision		Recall		Speed	Precision		Recall		Speed
	Result	%	Result	%		Result	%	Result	%		Result	%	Result	%	
1 : 1 : 1	50/51	98.0	50/52	96.1	3.54	43/43	100	43/52	82.6	2.80	43/43	100	43/52	82.6	2.62
1 : 2 : 2	51/52	98.0	51/52	98.0	5.63	43/43	100	43/52	82.6	3.43	43/43	100	43/52	82.6	2.93
1 : 3 : 3	51/52	98.0	51/52	98.0	8.02	43/43	100	43/52	82.6	4.07	43/43	100	43/52	82.6	3.23
1 : 4 : 4	51/52	98.0	51/52	98.0	10.26	43/43	100	43/52	82.6	4.71	43/43	100	43/52	82.6	3.63
2 : 1 : 1	51/62	82.2	51/52	98.0	5.36	45/55	81.8	45/52	86.5	3.82	45/55	81.8	45/52	86.5	3.24
2 : 2 : 2	52/63	82.5	52/52	100	8.87	45/55	81.8	45/52	86.5	4.77	51/62	82.2	51/52	98.0	5.18