Mutual Fund Selector – An Intelligent Object-Oriented Decision Support Environment

Rashmi Malhotra¹⁾ & D. K. Malhotra²⁾

¹⁾ Saint Joseph's University, Management & Information System's Department (<u>rmalhotr@sju.edu</u>)

²⁾ Philadelphia University, School of Business Administration (MalhotraD@philau.edu)

Abstract:

The investment in mutual funds has steadily increased in this decade. This growth has created a need for comprehensive and expandable financial decision support systems (DSS) that embody the knowledge of an expert in making investment decisions. This study illustrates the design and development of an intelligent, object-oriented decision support system: MFS —Mutual Fund Selector. MFS takes into account an investor's unique requirements and personal characteristics, and recommends an investment strategy.

1. Introduction

Mutual funds have been one of the fastest growing institutions in the United States. Funds assets have soared from less than \$150 billion in 1980 to over \$5.5 trillion in 1999. An estimated 48.4 million or 47.4 percent of all U.S. households owned mutual funds in 1999, up 9 percent from 44.4 million in 1998. Technology has been one of the key drivers behind the growth of the mutual fund industry. The use of personal computers (PCs) in the home has exploded in the last five years, with estimates of U.S. household PC ownership climbing from 16 percent in 1990 to 33 percent in 1995. PC ownership among mutual fund shareholders has consistently outpaced the national average, growing 28 percent in 1990 to 56 percent in 1995. This growth has created a need for comprehensive and expandable financial decision support systems (DSS) that embody the knowledge of an expert in making investment decisions. Over the last two decades, with technological advances in computer hardware and software, the original view of decision support systems has continuously improved. In the 1990s, hybrid systems that combine aspects of different artificial intelligence techniques such as knowledge-based expert systems, neural network methods, and genetic algorithms are growing in their development and use. Moreover, within the decision support systems area of research, there is a growing body of work focusing on the topic of knowledge management. This study illustrates the design and development of an intelligent decision support system: MFS – Mutual Fund Selector that interfaces between knowledge-based system and database management system to form an intelligent decision support environment. Further, to increase the productivity in organizational modeling and decision-making activities, this study investigates the use of object-oriented paradigm in coupling with database management systems and knowledge-based systems within a decision support framework. The rest of the paper is organized as follows. Section II describes the pertinent literature review; section III discusses the design and development of the system; section IV illustrates the study with a demonstration example; and finally, section V summarizes and concludes the study.

2. Literature Review

The impact of computer technology on organizations and society is increasing as new technologies evolve and existing technologies expand. Nearly all business executives state that information technology is vital to the business and they are using technologies extensively [Caldwell, 1995, Morrison, 1994]. There is a trend to provide managers with decision support systems that can assist them directly in making decisions at all levels of management. The overall results of using a DSS have been very impressive, as indicated in Udo's and Guimaraes' study [1994]. In the early 1970s, Scott Morton first articulated the major DSS concepts [Gorry and Scott Morton, 1971]. The main direction of this improvement has linked DSS to the computational approach in electronic data processing (EDP) applications [Alter, 1980], the knowledge-based symbolic approach (Bonezek, et. al., 1980), and the system architecture and development process (Keen, 1980; O'Keefe, 1986). In the 1990s we see group support systems and neural computing emerging, as well as many hybrid integrated computer systems [Turban, et. al., 1996, 1998]. The objective of a computer-based information system regardless of its name or nature is to assist management in solving managerial or organizational problems faster and better than what can be done without computers. To attain this objective, they may use one or more information technologies. The benefits of integrating the technologies were investigated by Forgionne and Kohli [1995], who found significant improvements when integrated systems are used. A literature survey on the development of decision support systems by Eom, Lee, Kim, & Somrajan, 1998 showed a growing tendency towards group-based decision support systems, such as development of group decision support systems, executive support systems, and knowledge-based system applications. Furthermore, Mamghani (1997) describe the potential of knowledge-based systems to provide effective computer support for IS strategic planning. Therefore, we need holistic management systems that are open-ended, flexible, and can easily integrate model management, and data management, with artificial intelligence and software engineering principles.

Over the last two decades, the most commercially applied area of knowledge-based systems and artificial intelligence is expert systems. Expert systems as a decision aid have been applied in almost every functional area of business and industry [Wong and Monaco, 1995] such as engineering and manufacturing [Badire, 1992], finance and accounting [Foltin and Smith, 1994, O'Keefe and Rebne, 1993], production management [Byrd, 1993], and strategic decision-making [Zuylen, 1993]. Traditional expert systems suffer from inherent problems such as combinatorial explosion, memory management, and unwieldy knowledge bases. With the increased complexity of problems for which knowledge-based systems are constructed there is a trend to increase the use of frames and object-oriented implementations [Bogarin and Ebecken, 1996]. Object-oriented paradigm is a fast growing technique that has many advantages over conventional knowledge-based technique such as extendibility, reusability, reduced code size, and the process worldview. These systems help the modeler perceive things with a "real world" viewpoint. Yoon and Guimaraes [1992] proposed to create an object-oriented organizational knowledge base. Furthermore, OOP tools are especially appropriate for applications related to multimedia and for delivering knowledgebased systems capabilities to Internet Web Servers. Besides, objects remain more or less stable, while functions tend to adapt to changing needs or circumstances. Moreover, new objects can be added to the existent system because objects and their management have a uniform and consistent definition. Moreover, an object-oriented framework provides a unifying context for model management research and decision support system development environment that integrates model management and data management utilizing modern software engineering principles and intelligent knowledge-based techniques. Besides, OOP allows model reuse, sharing, and integration. Therefore, the use of OOP facilitates the development of extremely flexible decision support systems that support intelligent interface, domain-independent model management, reusable and integrated model sharing, and an information-based methodology.

One of the most commonly used models in a DSS environment is a simulation modeling technique. The knowledge management module has been successfully applied to simulation systems also. Allen, Mabrouk, & Weigel (1996) describe the design of an expert system interface to simulation that encounters the problem of modeling complex systems and improving their efficiency using IF-THEN statements. Pflughoeft, Hutchinson, and Nazareth (1996) describe the architecture of an intelligent knowledge-based simulator KBSim that provides systematic research capability. As knowledge is now recognized as a major organization resource [Santosus, 1996], some studies have also reported the use of more sophisticated development methodologies such as OOP. Klein & Grubbstrom (1998) propose a model to effectively design DSS applications in the field of production through the use of OPTRANS object. Karacal, Cem, & Mize (1998) describe the development of a prototype object-oriented software system for discrete event simulation and the embedded decision processes. Booth (1998) describe how the object modeling techniques can be sued to develop integrated factory models that embrace factory process modeling as well as policy modeling.

With the growing popularity of object-oriented paradigm and a need for integrated systems, researchers have proposed different ways of managing knowledge, and developing knowledge management system of a decision support system. Muhanna (1993) illustrates that an object-oriented framework provides a unifying context for model management research and decision support system development. Muhanna (1994) describes the aspects of a software prototype MMS, called SYMMS, that is designed to provide for model sharing, reusability, data management, and integration. Karcal, Cem, and Mize (1998) address the modular and structured representations of the modular and structured representations of physical and logical entities of a manufacturing system for simulation modeling in the form of reusable software objects. Joines & Roberts (1996) create a set of object frames that encapsulate simulation requirements. Balci, Bertelrud, Esterbrook, & Nance (1997) describe the creation of a library of reusable model components that enable discrete-event, domain-independent, object-oriented, picture-based, component-based, visual simulation model development and execution for solving complex problems. On the other hand, Chang, Holsapple, & Whinston (1993) propose the use of hyper knowledge paradigm for knowledge management in a decision support system. Basu & Blanning (1998) propose the use of metagraphs, an analytical approach to represent assumptions about the validity of decision model bases. Silverman (1995) suggests three ways for knowledge-based expert systems to be integrated with mathematical modeling: knowledge-based decision aids that support the steps of the decision process unaddressed by mathematics; intelligent decision modeling systems that help users build, apply, and manage libraries of models; and decision analytic expert systems that integrate theoretically rigorous methods of uncertainty into the expert knowledgebase. The object-oriented approach toward systems analysis, design, and programming is directly affecting hybrid approaches to knowledge representation. Devedzic et al. (1996) describe how knowledge can be represented as objects in expert systems, alongwith the reasoning process. Their model, Reasoning Objects for Building Inference Engines (ROBBIE), is based on a hierarchy, where each level is more detailed. Kuechler et al. (1995) use a variety of knowledge representation and reasoning strategies in one object-oriented system.

Therefore, we need a well-integrated, flexible, intelligent decision modeling knowledgebase management system that can aid non technical users build, apply, and manage expert support systems. This study illustrates the design and development of an intelligent decision support system: MFS – Mutual Fund Selector. MFS can be used by a non-technical user to make decisions regarding an investment in mutual funds given an investor's characteristics. MFS was developed using Ellipse – an object-oriented knowledge systems development tool [citation from NEDSI proceedings]. Furthermore, MFS is also coupled with the morning star's mutual funds database. Once MFS identifies the optimal choice for an investor, the system retrieves the fund characteristics for the recommended fund. The next section describes the design and development of the mutual fund selector.

3. Design and Development of MFS

To develop MFS we used Ellipse, an object-oriented and database-coupled expert system development tool. Ellipse enables non-technical users (human experts) to develop expert systems interactively with very little help from the knowledge engineers, depending on the complexity of the

system. Ellipse was developed using the object-oriented paradigm. Ellipse creates and maintains the knowledgebase as a hierarchical tree. Typically, the nodes of tree are objects that correspond to a complete menu, and have appropriate pointers to pertinent subtrees (nodes/objects). Further, knowledge-based systems running in the Ellipse environment have access to generalized objects that run the inference engine and perform standard expert system activities such as going back to the previous level, stopping consultation, and referring back to the shell environment. Furthermore, Ellipse separates intensional knowledge (factual knowledge) from the extensional knowledge (knowledge beyond the factual knowledge), and represents the knowledgebase as a tight KB/DB coupled system. The domain database handles large volume of domain-specific knowledge (menu options), and has pointers (object identifiers) to other subtrees that should be instantiated if a particular option is selected. Besides the standard expert system capabilities, Ellipse also offers an explanation facility and modification capability. A knowledgebased system can explain the sequence of steps taken to reach a particular solution/recommendation. Also, if a user feels that a knowledge-based system has inappropriate/insufficient information, Ellipse allows the user to modify the knowledgebase interactively at any level. Finally, being modular and developed using an object-oriented platform, Ellipse-based systems can be easily interfaced with other databases (internal and external). Thus, exploiting this capability, we interfaced Ellipse-based knowledge system with the Morning Star database of mutual funds to develop an intelligent decision support environment that can aid a non-technical user in making investing decisions. Figure 1 illustrates the design process of MFS. As illustrated before, we used Ellipse as the knowledge systems development tool.

The development of MFS proceeded in three phases: the knowledge-understanding phase, the prototyping phase, and the systems development phase. In the knowledge-understanding phase, the user expert was presented with an introductory session of Ellipse. To convert the human expertise in the form of a knowledge tree, we exchanged many ideas in our cross-disciplinary fields. The knowledge engineer had sessions with the expert to explain how the knowledge should be organized. Once the expert was well acquainted with the tree format used by Ellipse to store knowledge, the expert was ready to develop an experimental prototype system MFA - Mutual Fund Advisor. With minimal assistance from the knowledge engineer, the user expert developed the MFA system. Although, the prototype did not include detailed heuristics procedures and other informational procedures, the expert was able to interact with the actual system, and get a general idea of the overall capabilities and the limitations of Ellipse. Therefore, after getting acquainted with the shell environment, the user expert developed MFS interactively with Ellipse by providing all the knowledge through the knowledge-acquisition module of Ellipse. The user expert was only responsible for entering all the information in the Ellipse environment. Ellipse created the database of factual information (intensional knowledge) and the knowledgebase of objects to store the extensional knowledge. Furthermore, since Ellipse is not a procedure writer, the expert also worked with the knowledge engineer to develop a template to store heuristical knowledge and interface with the Morning Star database of mutual funds. Screen forms 1-4 illustrate the user interface with the knowledge acquisition module of Ellipse.

Screen form 1 displays the main menu of Ellipse. Once the user selects the "New expert system" option, the expert is prompted to supply housekeeping information such as expert system's name and the files Ellipse should use. After performing the domestic chores, Ellipse prompts the user to enter the knowledgebase in the form of menus of user options, and arranges the information as a hierarchical tree. Screen form 3 illustrates the development of the hierarchical tree. Ellipse recursively requests the user to enter an entity and its characteristics till the user reaches the lowest level of the tree. Further, as illustrated in screen form 4, once the user reaches the grassroots level (the solution to the path traversed), Ellipse asks the user if the solution will be a simple recommendation or an information retrieval procedure that interactively works with the user to provide a recommendation for the given path¹. For the MFS system, at the end of each solution path, MFS conducts a dialog with the user to recommend mutual funds that ideally satisfy the user's requirements. Thus, the user enters option 2 so that Ellipse creates a partially written procedures file. The next section illustrates the functioning of the MFS system.

4. A Demonstration Application

Once all the stages in the development of MFS were completed, we tested the system to ensure that the system was working as intended. This section illustrates a sample run of the MFS system. Screen form 5, 6, and 7 illustrates the sample user session. To begin with, the user is presented the main menu of Ellipse. The user selects MFS, and Ellipse presents the uppermost level of MFS' knowledge tree. The user selects the mutual funds with low risk. Within the low risk option, MFS further inquires if the user wants regular income, or regular income with capital appreciation. The user selected the second option, indicating that regular income with capital appreciation is the priority. Once the MFS reached the grass root level, the heuristical/informational procedure further inquires if the user wants load funds and funds with 12b-1 fees. Based on user's response MFS recommends three funds and displays their pertinent statistics to the user. Finally, MFS continues the session by presenting more options to the user.

5. Summary and Conclusion

The investment in mutual funds has steadily increased in the past decade. The increase in mutual fund investment has partially been contributed by the advances in technology and use of personal computers for online investing. Mutual fund shareholders with home personal computers have 40 percent of household financial assets invested in mutual funds, and own a median of three funds from two companies. This growth has created a need for comprehensive and expandable financial decision support systems (DSS) that embody the knowledge of an expert in making investment decisions. This study illustrates the design and development of an intelligent decision support environment: MFS – Mutual Fund Selector. Although the study concentrates on the knowledgebase management system of the DSS environment, the system offers an open architecture that can be easily interfaced with many other applications, and can offer an elaborate modelbase management system through the unifying object-oriented platform. We use Ellipse – an objectoriented and database-coupled knowledge system development environment. To understand the functionality of Ellipse, the expert first develops a prototype of the expert system. After becoming familiar with the hierarchical representation of knowledge, the user expert interactively entered the knowledge for the MFS system. The Ellipse software further organizes the knowledge into two data structures: a knowledgebase of objects (extensional knowledge) and the factual database (intensional knowledge), and created a hierarchical tree of knowledge objects with appropriate pointers to the domain database. Furthermore, to recommend suitable funds and their pertinent features, we develop an interface between MFS' knowledgebase and an external database (Morning Star's database of mutual funds). Moreover, MFS also uses user-defined procedures. The Ellipse software prompts the user to enter procedure names, establishes appropriate links with the procedural objects, and creates a partially written procedure file. To create the procedures for MFS, the user expert used a template provided by the knowledge engineer. The partially filled template included the headers, empty routines, and an interface with the database of mutual funds. The user filled in the body of routines. Thus, we illustrate a flexible, knowledge-based system that can be interfaced with many other systems.

The development of MFS using object-oriented workbench illustrates a methodology that has many advantages such as reusability, code sharing, implementation language independence, modular decomposition, effective memory management, and an easy interface with other applications. Thus, the design of MFS represents a new generation of tools that are more iterative, support the concept of objectorientation, and live in an open architecture. This architecture offers future for the systems that are integrated with multimedia to provide an integrated environment that has visual, audio, and graphics support combined with the advice-rendering and decision-making support of the knowledge-based systems. Furthermore, MFS offers a set of objects that are flexible and reusable, and have the potential to be applied to complex systems. Thus, the extendible architecture of MFS is more conducive to be integrated with other systems such as office automation systems, transaction-processing systems, and other enterprise information system. Further, enterprise information systems can benefit from knowledge-based systems (KBS) with CASE tools that permit the development of AI-based systems.

Another advantage of MFS is that the decision support environment is not limited due to memory requirements/memory management problems. The use of KB/DB coupling provides efficient dynamic memory management. Although the study illustrates a tight coupling, the database is an independent data structure, and therefore can also exist as a loose coupling. Thus, MFS can be easily interfaced with other databases (internal and external). This capability can be extended to interface with other decision support systems, group support systems, and executive support systems that can be used to automate the execution of individual and collaborative decision-making processes performed by organizational members. In addition, MFS can also be integrated with other internet-based systems that use web objects. Besides interfacing MFS with other organization information systems and internet-based applications, another inevitable extension of MFS is the addition of a modelbase. Although the study does not illustrate the design of a modelbase management system (MBMS), the architecture of MFS can easily be extended to include a modelbase that stores traditional management science models and the state-of-the-art neural network models and genetic algorithm application. Thus, MFS can also be extended to analyze data warehouses and offer data mining tools. Finally, the architecture of MFS can also be extended to implement distributed management that is most suitable for larger databases and client/server information technology architecture.

References

Alter, S. L. Current Practices and Continuing Challenges, Addison-Wesley, Reading, MA, 1980.

Allen, G.J., Mabrouk, K. M., & Weigel, M. L. Utilizing an Expert System for Conflict Resolution in a "Train siding selection" Simulation Model. *Proceedings of the 1996 Winter Simulation Conference*, 1347-52, 1996.

Balci, O., Bertelrud, A. I., Esterbrook, C. M., & Nance, r. E. Developing a Library of Reusable Model Components by Using the Visual Simulation Environment. *Proceedings of the 1997 Summer Computer Simulation Conference, Simulation and Modeling Technology for the Twenty-First Century*, 253-8, July, 1997.

Basu, A. & Blanning, R. W. The Analysis of Assumptions in Modelbases using Metagraphs. *Management Science*, 44:7, 982-995, July, 1998.

Bogarin, J. A. G. & Ebecken, N. F. F. Integration of Knowledge Sources for Flexible Pipe Evaluation and Design. *Expert Systems with Applications, Vol. 10:*1, 1996.

Bonczek, R. H., Holsapple, C. W., & Whinston, A. B. The Evolving Roles of Models in Decision Support Systems. *Decision Sciences, Vol. 11*:2, 1980.

Byrd, T. A. Expert Systems in Production and Operations Management: Results of a Survey. *Interfaces, Vol. 23:2*, March/April, 1993.

Caldwell, B. CEOs click on IT. Information Week, May 8, 1995.

Chang, A., Holsapple, C. W., & Whinston, A. B. Model Management: Issues and Directions. *Decision Support Systems*, 9:1, 19-37, January. 1993.

Eom, S. B., Lee, s. B., Kim, E. B., & Somrajan, C. A Survey of Decision support System Applications (1988-1994).

Journal of the Operational Research Society, 49:2, 109-120, February, 1998.

Foltin, L. C. & Smith, L. M. Accounting Expert Systems. The CPA Journal, Vol. 64:11, November, 1994.

Forgionne, G. A. & Kohl, R. Integrated MSS Effects: An Empirical Health Care Investigation. *Information Processing and Management, Vol. 31:6*, November, 1995.

Gorry, G. A. & Scott-Morton, M. S. A Framework for Management Information Systems. *Sloan Management Review*, 11:2, 1971.

Joines, J. A. & Roberts, S. D. Design of Object-Oriented Simulations in C++. *Proceedings of the 1996 Winter Simulation Conference*, 65-72, 1996.

Karcal, S. C. & Mize, J. H. Object-Oriented Software Implementation for Manufacturing Systems. *IIE Transactions*, 30:3, 217-26, March, 1998.

Keen, P. G. W. Adaptive Design for Decision Support Systems. Database, Vol. 12:1, 1980.

Klein, M. & Grubbstorm, R. W. Using OPTRANS objetc as KB-DSS Development environment for Designing DSS for Production Management (Knowledge-based Decision Support System). *European Journal of Operations Research*, *109*:2, 264-85, September 1, 1998.

Mamaghani, F. Improved Strategic Planning for Information System: A Knowledge-Based System Approach. *International Journal of Management*, *14*:4, 679-89, December, 1997.

Morrison, D. CEOs and PCs: Do they Compute? Beyond Computing. Jan/Feb, 1994.

Muhanna, W. A. SYMMS: A Model Management System that supports Model Reuse, Sharing, and Integration. *European Journal of Operations research*, 72:2, 214-242, January 27, 1994.

Muhanna, W. A. An Object-Oriented Framework for Model Management and Decision Support Development. *Decision Support Systems*, *9*:2, 217-229, February, 1993.

O'Keefe, R. Simulation and Expert Systems: A Taxanomy and some Examples. *Simulation, Vol.* 46:1, 10-16, January, 1986.

O'Keefe, R. M. & Rebne, D. Understanding the Applicability of Expert Systems. *International Journal of Expert Systems, Vol. 1:*1, September, 1993.

Pflughoeft, K. A., Hutchinson, G. K., & Nazareth, D. L. Intelligent Decision Support for Flexible Manufacturing: Design and Implementation of a Knowledge-based Simulator. *Omega*, 24:3, 347-360, June, 1996.

Santosus, M. Knowledge Management. CIO, June 1, 1996.

Turban, E. & Aronson, J. E. Decision Support Systems and Intelligent Systems. *Prentice Hall*, Upper Saddle River, New Jersey, Fifth Edition (1998).

Udo, G. J. & Guimaraes, T. Empirically Assessing Factors Related to DSS Benefits. *European Journal of Information Systems*, July, 1994.

Wong, B. K. & Monaco, J. A. Expert Systems in Business: A Review and Analysis of Literature. *Information and Management, Vol.* 29:3, September, 1995.

Yoon, Y. & Guimaraes, T. Developing Knowledge-based Systems: An Object-Oriented Organizational Approach. *Information Resource Management Journal*, Summer, 1992.

Zulyen, H. J. From Scientific Computation to Decision Support. Knowledge Systems, Vol. 6:1, 1993.

Note: The figures and screen forms are available upon request.