한국산업정보학회논문지 제5권 제3호 (2000. 9)

MS/OR EDUCATIONAL SOFTWARE PACKAGES: ARE THEY EFFECTIVE TUTORING PROGRAMS?

Eyong B. Kim* Sangjin Yoo**

요 약 오늘날 대단히 많은 MS/OR 교육용 소프트웨어 패키지들이 널리 이용되고 있다. 이들 패키지들은 모두 학생들이 MS/OR 기법들을 보다 효과적으로 이해하는데 도움이 될 것이라 기대되고 있으나 때로는 이들 패키지들이 MS/OR 기법들을 가르치는데 사용되기보다는 구축된 MS/OR 모형의 해(Solution)를 보다 효율적으로 구하기 위한 도구로 사용되는 경우가 많다. 본 연구는 이러한 MS/OR 소프트웨어 패키지들이 효과적인 교육용 도구로 사용되지 못하는 이유들을 찾아내 보려 시도되었으며, 연구결과는 다음과 같다. 첫째 기술적인 한계, 둘째, MS/OR 전문가들의 소프트웨어 패키지에 대한 인식부족이다.

Abstract Management science/operations research (MS/OR) educational software packages are widely used at the present time. Those software packages are expected to help students understand MS/OR techniques better. However, MS/OR educational software packages are often used as computational tools to obtain model solutions efficiently rather than as the tutoring software packages. Several possible reasons for the lack of effective tutoring capacity in MS/OR educational software packages are identified in this paper. The authors believe that the deficiency of tutoring capacity in those softwares is mainly due to technological limitations (computers and artificial intelligence) and the MS/OR professionals' perception about those software packages. Given technological limitations, feasible design and development approaches are provided to improve the tutoring effectiveness of MS/OR educational software packages.

1. INTRODUCTION

Many years have passed since microcomputer based management science/operations research educational software packages (MS/OR ESPs) were introduced. Because of these software packages, many changes have taken place in MS/OR education. For example, the emphasis of MS/OR education has shifted from obtaining the model solution to the model formulation and the implementation issues of the model solution [6]. A recent survey demonstrated that many MS/OR professionals agree that it is desirable to have more of model formulation and computer solutions in an MS/OR course; and less of simplex method mechanics [10].

When MS/OR ESPs were introduced, many MS/OR professionals believed that these software packages could become an effective tutoring aid because the students could understand the MS/OR techniques more easily by relieving the mathematical burdens of MS/OR solution algorithms. In current MS/OR education practice, students are allowed to use the MS/OR ESPs to minimize the mathematical burden of solution algorithms. However, if relief from the mathematical burden is the main benefit the students can gain from the MS/OR ESPs, those software packages cannot be considered an effective tutoring aid. The question we as MS/OR professionals should answer is that we did not or can not develop an effective MS/OR tutoring software.

The concept of "learning by doing" has been emphasized for more effective learning in the education domain [22]. One study reports that people retain about 45 percent of what they see and hear, but about 70 percent of what they see, hear, and do [7]. It

* Marketing/Management Dept. Barney School of Business University of Hartford
** Department of MIS, Keimyung University

implies that the interaction between a system and a student is one important element of an effective tutoring software. However, most currently available MS/OR ESPs are designed to obtain the model solutions as quickly as possible rather than providing enough interactions with students. With minimal interaction with students, these software packages cannot effectively help students understand the different dimensions of MS/OR techniques such as problem recognition and model selection, model formulation, solution procedure, and interpretation of model solution.

Thus, in this paper, the authors will discuss possible explanations for the lack of effective tutoring capacity in currently available MS/OR ESPs. The practical design and development approaches will be provided to enable those software packages to become more effective tutoring aids.

2. PROBLEMS IN CURRENT MS/OR EDUCATIONAL PACKAGES

There could be several reasons to explain why most currently available MS/OR ESPs failed to help students understand the different dimensions of MS/OR techniques effectively. Developing a tutoring software which can help to define a problem and to formulate an MS/OR model is not an easy task because of the "natural language understanding" problem of artificial intelligence (AI) which is known as "Achilles' heel" of potential tutoring systems [1]. Thus, in general, human intelligence is heavily emphasized in the problem recognition and the MS/OR model formulation. It means that instructors, as well as the managers, believed that the skills of recognizing the problem and formulating an MS/OR model can be obtained through experience and effort [2]. The difficulty of natural language understanding in the rich business environment may be one reason (but not a sole reason) to explain why most currently available MS/OR ESPs lack the ability to support problem recognition and model formulation.

Many MS/OR professionals often consider MS/OR ESPs as tools to obtain MS/OR model solutions rather than as software packages which can enhance students' understanding of MS/OR techniques. Due to this perception, up to now, many MS/OR researchers evaluate (or select) MS/OR ESPs based mainly on the mathematical aspects of ESPs such as the accuracy of model solution, the computational speed, the model size limitation, the coverage of different MS/OR models, and so on [15, 25, 32]. The perception of those software packages as calculating tools may deter MS/OR professionals from attempting to improve the tutoring capacity of MS/OR ESPs fast enough.

Currently available MS/OR ESPs include too many different MS/OR techniques in a relatively small memory space (diskettes). This strategy may satisfy most instructors' needs of covering many different MS/OR techniques in their MS/OR classes. However, to be an effective MS/OR ESP, a software should emphasize popular models rather than provide a limited sample of all the MS/OR models which suffer from the limitations of size and other options [33]. A desirable MS/OR ESP is the one which has several popular MS/OR modules with enhanced explanation capacity.

It is desirable to analyze the students' difficulties more carefully when developing a MS/OR ESP. Developers of MS/OR ESP should pay more attention to actual users' (students') information needs and system needs. The information needs may include explanations about every step of MS/OR model use (the processes of model selection, formulation, solution, and solution interjection), the availability of graphical information, the availability of help whenever needed, and some others. The system needs may include easy-to-use system (i.e., adequate help system and others), easy-to-understand input and output screen (in terms of format and readability), editing capacity of input data, and so on. This is the same as the user requirements analysis phase in a system development life cycle approach. Often many of these requirements are determined by the MS/OR professionals who can assume what the students may need, not what students actually need. Thus, currently available MS/OR ESPs are often easy to use by MS/OR professionals, but not by students.

A standardized measuring tool is important for assessing the impact of MS/OR ESP on the learning process of students. A standardized measuring tool is a set of questions which can be used to assess the effects of treatment on the subjects objectively. To be a standardized measurement tool, a test set should satisfy all three standards of validity - prediction validity, content validity, and construct validity.

Considering the characteristics of MS/OR education, content validity is the most important validity to standardize the achievement test (or the test of mastery learning) [21]. Without using the standardized measuring tool, it is difficult to specify what benefit(s) or damage(s) the students gain from using MS/OR ESPs.

Table 1 Processes of Learning and Types of Instructional Events That May Support Them

Learning Process		Instructional Event	
1	Altering the learner toreceive stimulation	1	Gaining attention
2	Obtaining an expectancy of the results of learning	2	Show to learner the lesson objectives
3	Retrieval of long term memory to work	3	Stimulating recall of prior memory
4	Selective perception of the patterns enter into learning	4	Presenting stimuli with distinctive features
5	Semantic encoding of presented material for long term storage	5	Provide learning guidance
6	Responding with a performance that verifies learning	6	Eliciting performance
7	Reinforcement to establish the learning	7	Provide feedback
8	Cues to recall	8	Assisting performance
9	Generalizing knowledge acquired to new situations	9	Enhancing retention and learning transfer

Note: Excerpted from Gagne and Briggs [(8) 1979]

Due to the reasons mentioned above, we develop MS/OR ESPs which do not satisfy most instructional events on Table 1. Each internal learning process on Table 1 transforms the information affecting a learner and brings about a change in state that advances learning [16]. Thus, depending on the type of learning outcome (what is to be learned), it is desirable to determine how to arrange the sequence of instructional events in an educational software [9]. The desirable tutoring software is expected to support all instructional events on Table 1 [8].

3. APPROACHES TO IMPROVE MS/OR EDUCATIONAL PACKAGES

As mentioned earlier, to improve the educational effectiveness of MS/OR ESP, the most important step is changing the MS/OR professionals' perception about those software packages. If MS/OR ESPs are regarded as big calculators, there is no need to improve the educational effectiveness of those software packages any further. If MS/OR professionals agree to improve the effectiveness of MS/OR ESPs, we may consider the use of expert system techniques for enhanced tutoring.

Developing an expert system as a part of a tutoring system is conceptually feasible. However, while there have been significant expert system applications in related areas, such as statistics, there is little evidence for the use of expert systems in MS/OR classrooms [30]. The major advantage of an expert system as a tutor is that it can make learning an active process rather than a passive one. To review previous researches on how an expert system approach is used as a tutor in MS/OR or modeling support tools, refer to Turban and Trippi [30].

3.1. Expert System for MS/OR Model Selection

Selecting an appropriate MS/OR model is one of the most important steps in applying the MS/OR technique to real world problems. However, it is also true that it is one of the most difficult tasks due to the limitation of AI technologies. Many MS/OR educators often begin with an assumption that the students know which MS/OR model can provide the best solutions for a particular problem. Students often mechanically choose an MS/OR model without understanding why.

They often choose a model by matching key words or recalling the chapter of textbook in which a similar problem was discussed, and so on. This selection approach works well in current MS/OR classrooms because most MS/OR problems in textbooks are well structured and from a narrow context [5].

Generally, a real-world problem is often so ambiguous that students have difficulty in defining the problem itself [5]. In addition, given the same problem, there could be several feasible solution approaches. Because a decision maker chooses a solution model based on the context of a particular problem and his/her preference, it is possible to use a rules-based-model-attribute matrix to select the appropriate model [14].

If we develop a model selection expert system, it should not only choose the best MS/OR model for a specific problem, but provide the proper explanations about the questions students have in the process of choosing a particular MS/OR model(s). Without this explanation capacity, it is not an effective model selection helper which can be used with MS/OR ESPs.

A system should illustrate all possible MS/OR techniques to solve a specific problem and explain to students which one is better and why. For example, if a system suggests a goal programming (GP) technique instead of multiple objective linear programming (LP) technique, it should explain the differences between these two techniques and what are the advantages of using that technique for that specific problem.

3.2. Knowledge Based MS/OR Mode Formulation Support System

The MS/OR model formulation support system may be one of the most important functions that an MS/OR ESP should include to become an effective tutoring system. This system is also difficult to develop due to the limitation of AI technology. For example, human experts use a *different kind of knowledge in building* LP formulations, namely, syntactic LP knowledge, semantic LP formulation knowledge, semantic LP domain knowledge, and semantic world knowledge (common sense) [28].

There have been many studies in computer-based support for model formulation [3, 17, 19, 23, 26, 27, 29]. All these systems support the formulation of LP models. For example, Murphy and Stohr [19] described an overview of a system which uses AI and database techniques to help expert users formulate a large LP model by automating many of the tedious processes associated with large-scale modeling. This research by Murphy and Stohr complemented the works of Greenberg [11, 12] in computer-aided analysis, in which Greenberg developed tools and mathematical techniques to help understand LPs after they have been formulated.

Recently, there was a study which reported that if a human modeler uses a computer assisted model formulator (MODFORM), s/he outperforms the experienced human modelers [23]. The developers of MODFORM found that the human experts use "chunks" which serve as cues to recognize a particular problem as being similar to a standard problem type and to model the problem in a routine manner. The mechanism, by which the "chunks" are mapped to the problem types, helps students understand the problem recognition and model selection process. As the developers of MODFORM [23] noted, the mapping mechanisms human experts use to recognize the problem are largely unknown. However, those developers claimed that once the problem type and variables are recognized, the formulation remains largely a synthetic process in which domain knowledge and modeling principles together help a modeler formulate an appropriate model.

As mentioned earlier, a fully automated formulating system of MS/OR model is not yet achieved. However, from previous studies we learned the method of structuring the formulation process of LP model which can be applied to other MS/OR techniques. Thus, in a MS/OR ESP, we may implement a smaller size of a previously developed computer supported model formulating system with more explanation capacity.

3.3 Drill and Practice Program with the "Difference Analyzer"

The current trend of MS/OR education is to have less emphasis on the solution algorithm of MS/OR techniques such as simplex method [10]. However, up to a certain limit, it is necessary to know the solution algorithm to understand an MS/OR model in depth. This approach is intended to help students learn the solution process of an MS/OR technique effectively.

A drill and practice program asks questions and provides feedback to the students to enhance the retention and the transfer of knowledge learned previously [31]. The drill and practice program satisfies instructional events 6, 7, 8 and 9 on Table 1.

There are several issues in the cognitive literature which we need to consider more in designing a drill and practice program [24]. Those issues are automacy of subskills, inference, spaced practice, spaced review, capacity of short term memory, and representation of information in memory. A "difference analyzer" means finding the differences (if any) through comparing the user's solution with that generated by an expert module in a system. In the difference analyzer, the user enters his/her solution as an input.

Using these two approaches together, it is possible to develop an MS/OR ESP which can help students understand the solution process more effectively. In other words, a student provides the answer to a question a program asks and a program will provide a correct answer (if the student makes a mistake) with the proper explanation.

To provide the proper explanation on the student's mistake, the "predetermined explanation string" may be utilized. It means that the system has the sentences (text strings) which were determined in advance by a system developer. To provide an explanation(s), a system retrieves a corresponding text string(s) and add some variable names(s) or values. This approach is somewhat limited in flexibility, but more practical in implementing on a MS/OR ESP than the natural language understanding approach of AI. For example, if a student makes a mistake in determining the entering variable of a LP model (choosing X1), the system suggests the correct answer (X2) and provides the reason for choosing that answer, e.g., "The correct entering variable is X2 because X2 has bigger Cj-Zj value of 40 than X1 which has 30." In this sentence, every word is predetermined to explain the reason for choosing the entering variable except the variable names and values (X1, X2, 30, and 40). Only the variable names (X1 and X2) and values (30 and 40) have to be added to the predetermined explanation string. It is recommended that whether the student's answer is correct or not, a system should provide proper reasoning for choosing the correct answer because most students solve MS/OR models mechanically without knowing why [18].

Due to limited memory space, it is difficult to include the explanations for all possible students' mistakes. Thus, to develop those predetermined explanation strings, a system developer should examine the students' possible mistakes (or difficulties) in understanding MS/OR techniques carefully and determine how to provide an explanation for each mistake in advance. An example study of analyzing the students' behavior was conducted by Lee and Kim [18] which recognized the difficulties students have in understanding a goal programming model in the actual classroom environment. The careful analysis of student behavior is important to provide a set of predetermined explanation strings which can explain the main difficulties many students have. A suggested framework for this approach is provided on Figure 1.

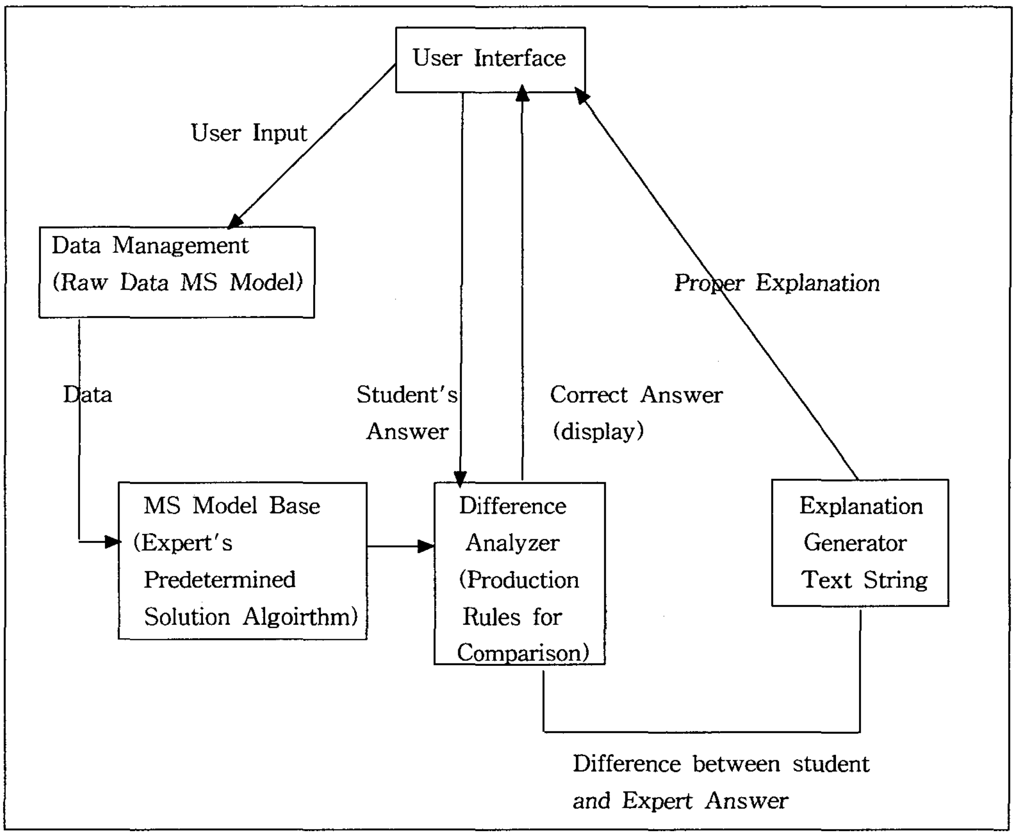
This predetermined explanation string approach can be utilized not only for explaining the solution process but also for explaining the other dimensions of MS/OR education. If we successfully develop the predetermined explanation strings, it is much easier in the future to develop a "bug-part-library" which can be used in intelligent tutoring system (ITS) such as BUGGY did [4]. Because it is difficult for a system developer to know all of the students' possible mistakes (bugs) in advance (which is the bug library), a bug may be constructed from a library of bug parts (primary bugs) rather than extracted from the provided bug library. For example, a mistake of economic order quantity may be explained through many different combinations of the mistakes of calculating annual demand, carrying cost, ordering cost, and so on.

Rather than developing a huge combinations of possible mistakes, the bug-part-library approach tries to keep only the main causes (primary bugs) of mistakes and assemble possible reason(s) using those primary bugs.

Instead of simply displaying model solutions, the other possible improvement of the currently available MS/OR ESPs is adding an explanation module of the model solution. Previously, Greenberg [13] utilized the limited natural language understanding capabilities and graphics to assist model building and the interpretation of LP model solution. However, it is a prototype with limited syntax and semantic capacity which has limited usability.

Practically, it is possible to interpret most MS/OR

Figure 1 Suggested Framework for "Difference Analyzer



model solutions by following a certain sequence of rules. Thus, we may add an explanation module (which has if-then rules) to the currently available MS/OR ESPs through utilization of the "predetermined explanation string" as it was suggested for solution algorithm explanation. It is also an option to utilize the hypertext which is well suited for open learning applications where the student is allowed freedom of action and encouraged to take the initiative [20, p.65].

The hypertext allows each student to move from a piece of model solution (or information) down to lower and lower levels of detail at the time of interpreting MS/OR model solution because it is nonsequential (sometimes it is called the "generalized footnote").

The emphasis is on making students understand the meaning of model solutions rather than making them interpret solutions mechanically.

4. CONCLUSION

Even though the MS/OR professionals have been leaders in using advanced computer technologies, MS/OR educational software packages are out of date in terms of educational effectiveness. The era of first generation MS/OR educational software packages which provides the MS/OR model solutions quickly has to be ended. It is the time to start the development of MS/OR educational software packages which can actually teach MS/OR techniques to students.

It may be difficult to develop a fully automated MS/OR tutoring system with currently available computer technologies and artificial intelligence techniques. Thus, it is more reasonable to develop MS/OR educational software packages which can tutor students up to a certain limit. Toward to this end, as an initial step, the authors provided several suggestions in this paper: the change of MS/OR professionals' perception about MS/OR educational software packages, the recognition of difficulties students actually have in understanding MS/OR techniques, and several development approaches to improve the educational effectiveness of MS/OR educational software packages.

The authors suggest that it is desirable to search for a method of structuring the unstructured portion of MS/OR education practice as much as possible. If MS/OR professionals do not step forward to improve MS/OR software packages at this time, it will be difficult to develop a truly effective MS/OR tutoring software in the future when technologies are ready.

A final recommendation is that it is desirable to develop an MS/OR educational software package as a joint project among cognitive psychology specialists, AI specialist, information science specialist, and MS/OR specialist to enhance the tutoring capabilities. Thus, the desirable development approach is that MS/OR specialists provide overall expertise of MS/OR education, cognitive scientists analyze student behavior, AI specialists represent those experts' knowledge in the proper format, and the information scientists design and develop an MS/OR educational software. Without this corporation and mutual support, it will be difficult to improve the effectiveness of MS/OR educational software package fast enough.

Reference

[1] J. R. Anderson, The Expert Module. AFHRL Research Planning Forum for Intelligent Tutoring Systems. San Antonio, Texas, September, 1986.

[2] M. Angelides and G. Doukidis, Is There a Place in OR for Intelligent Tutoring System? Journal of the Operational Research Society. 41(6) 491-503 (1990).

[3] M. Binbasioglu and M. Jarke, Domain Specific DSS Tools for Knowledge-Based Model Building. Decision Support Systems. 2, 213-223 (1986).

[4] J. S. Brown and R. B. Burton, Diagnostic Models for Procedural Bugs in Basic Mathematical Skills. Cognitive Science. 2, 155-192 (1978).

[5] J. A. Chisman, Preparing Students and New Practioners for the Real-World of OR/MS. Computers and Operations Research. 18(7), 585-589 (1991).

[6] R. A. Deininger, Teaching Linear Programming on a Microcomputer. Interfaces. 13(4), 30-33 (1983).

[7] M. Edwards, The Mercedes Benz of Interactive Video. Hardcopy. 14(5), 74-80 (1985).

[8] R. M. Gagne and L. J. Briggs, Principles of Instructional Design. 2d ed. New York, Holt, Rinehart and Winston, (1979).

[9] R. M. Gagne, Learning outcomes and their effects: Useful Categories of Human Performance. American Psychologist. 39, 377-385 (1984).

[10] G. Gamini, Trends in Teaching Management Science in Undergraduate Business Programs. Interfaces. 21(5), 16-21 (1991).

[11] H. J. Greenberg, A Tutorial On Computer-Assisted Analysis eds.) In Advance Techniques. In: The Practice of Operational Research. Greenberg, H.J.; Murphy, F.H., Shaw, H. (Eds.), North-Holland, Amsterdam, 212-250 (1982).

[12] H. J. Greenberg, A Functional Description Of ANALYZE: A Computer-assisted Analysis System For Linear Programming Models. ACM Transactions On Mathematical Software. 9(1), 18-56 (1983).

[13] H. J. Greenberg, A Natural Language Discourse Model To Explain Linear Programming Models And Solutions. Decision Support Systems. 3(4), 333-342 (1987).

[14] I. B. Hong and V. R. Doug, Data and Model Management in a Generalized MCDM-DSS. Decision Sciences. 22(1) 1-25 (1991).

[15] W. K. Janczyk and J. E. Beasley, Multiple-Model OR Packages. Journal of Operations Research Society. 39(5), 487-509 (1988)..

[16] R. L. Klatzky, Human Memory: Structures and Processes. 2d ed., San Francisco, Freeman, (1980).

[17] R. Krishnan, Automated Model Construction: A Logic Based Approach. Annals of Operations Research. 21, 195-226 (1989).

[18] S. M. Lee and E. B. Kim, An Analysis of

Pedagogical Effects of Alternative Teaching Approaches of Goal Programming. Decision Sciences. 23(4), 991-1002 (1992).

[19] F. H. Murphy and E. A. Stohr, An Intelligent System For Formulating Linear Programs. Decision Support Systems. 2(1), 39-47 (1986).

[20] J. Nielsen, HyperText & HyperMedia. Academic Press, Inc., Boston, (1990).

[21] J. C. Nunnally, Psychometric Theory. 2d. Ed., McGraw-Hill, New York, (1978).

[22] J. Piaget, Science of Education and the Psychology of the Child. New York: Viking, (1970).

[23] S. Raghunathan, R. Krishnan, and J. May, MODFORM: A Knowledge-based Tool to Support the Modeling Process. Information Systems Research. 4(4), 331-358 (1993).

[24] D. Salisbury, Cognitive Psychology and Its Implications for Designing Drill and Practice Programs for Computers. Journal of Computer Based Instruction. 17(1), 23-30 (1990).

[25] R. Sharda, The State of Art of Linear Programming on Personal Computers. Interfaces. 18(4), 49-58 (1988).

[26] M. Sklar and R. Pick, A Knowledge-Based Linear Programming Formulation Assistant. Proceedings of HICSS-23. IEEE Computer Society Press, Los Alamitos, C.A. 3, 269-278 (1990).

[27] M. Sklar, R. Pick, and J. Evans, A Knowledge-Based Approach to LP Model Formulation. Proceedings of DSI Meeting. 1, 226-228 (1984).

[28] M. Sklar, R. Pick, and J. Evans, On the Automatic Formulation of LP problems. Presented at ORSA/TIMS. St. Louis, October 25-28, (1987).

[29] M. Sklar, R. Pick, and J. Evans, Eliciting Knowledge Representation Schemes for Linear Programming. in D. E. Brown and C. C. White (Eds.), Operations Research and Artificial Intelligence: The Integration of Problem Solving Strategies. Kluwer, 279-316 (1990).

[30] E. Turban and R. Trippi, The Utilization of Expert Systems in OR/MS: An Assessment. OMEGA. 17(4), 311-322 (1989).

[31] W. Wager and R. M. Gagne, Designing Computer-Aided Instruction. in Instructional Designs for Microcomputer Courseware. by D.H. Jonassen, (Ed.), Lawrence Erlbaum Associates Publishers, Hillsdale, NJ, 35-60 (1988).

[32] J. Yurkiewicz, Educational Operations Research Software: A Review. Interfaces. 18(4) 59-71 (1988).

[33] S. H. Zanakis, A. Kara, S. Sahay, and N. Sivasubramaniam, An Analytic Evaluation of Microcomputer Educational OR/MS Software. OMEGA. 19(6), 639-649 (1991).

유 상 진

서강대학교 물리학과 (이학사)및 경영학과 (경영학사) 졸업
미국 University of Nebraska-Lincoln에서 MS 전공으로 Ph.D 취득
미국 Ohio주 Bowling Green State University MIS담당 조교수 근무
현재 계명대학교 경영학부 경영정보학 교수 , 한국정보시스템 학회장, 대구경북 전자상거래 지원센타 전문위원, 대한상사 중재원중재인, 대구경북 정보화 추진단장

주요 연구관심분야 : 정보기술을 활용한 경영혁신, 전자상거래, 정보기술의 전략적 활용 등

Eyong B. Kim is an Assistant Professor of MIS at the Barney School of Business at University of Hartford. He received his BS degree from Minnesota State University at Mankato, and MBA and Ph.D from University of Nebraska-Lincoln. His primary research areas include E-Commerce, Decision Support Systems, Pedagogy Systems, and Network Security. He has published several refereed journal articles including Decision Sciences, OMEGA, and Journal of Operational Research Society