Mansoura Engineering Journal(MEJ),Vol,20,No.4,Dec. 1995 E.48

Quantitative inference of decision making strategies under inherent knowledge uncertainties

الإستنتاج الكمى لإستراتيجيات إتخاذ القرار فى حالات المعرفة الغير مؤكدة

Guirguis S.K.*, El-Racy M.E.* and Aly E.M.**

*** Institute of Graduate Studies & Research, Alexandria University, Egypt.**
**** Statistics Dept., Alexandria Customs Computer Center, Alexandria, Egypt.**

ملخص

يقدم هذا البحث نموذجا رياضيا و البرمجيات الخاصة به لتحديد الإستراتيجيات المثلى و التى يتم تطبيقها بواسطة متخذى القرار فى الظروف المختلفة و بطريقة آمنة.

إن متخذى القرار لمتوقع أحيانا أن يتخذوا قراراتهم طبقا لمشاهداتهم و ليس طبقا للحدث الحقيقى المسبب للمشاهدة حيث أن العلاقة بين الأحداث و المشاهدات متعددة بالإضافة إلى حالات عدم التأكد فى عملية تقدير الخسائر نفسها و هى التى أحيانا تكون بسبب بعض القصور فى معلومات متخذ القرار أو كنتيجة لطبيعة المشكلة نفسها.

هذا و يتم فى هذا البحث تطبيق النموذج على منظومة حقيقية لدراسة نتائج مراحل التنفيذ المختلفة و تأثيرها على إستكمال المعلومات القاصرة.

ABSTRACT

In this paper a model and its associated software system, that is being used for designation of optimal strategies are presented. Obtained optimal strategies are employed by decision makers to safely react to various conditions. The fact that decision makers are expected to take actions according to given observations is discussed along with emphasizing the relationships between these observations and the underlying states of nature that caused them. This is a many-to-many relation and care must be taken to avoid unreliable identification of the real cause or causes of an observed situation.

The inherent uncertainties in the assessment process have been identified and accounted for during the losses estimation phases. These uncertainties are partially attributed to limitations in the decision maker's knowledge and partially due to the obscured problem nature.

The software application to a real system is illustrated. The application has been chosen to model the operations of an existing information center as an example of a wide class of similar systems. The application aimed at experimenting the phases of the software system and identifying the merits of using such quantitative models in furnishing knowledge gaps.

1. INTRODUCTION

Decision support systems (DSSs) have been in use since the early seventies. Applications of DSS systems are well known in a wide range of activities, including: office automation, manufacturing, marketing, accounting, budgeting, purchasing, banking, finance, sales, insurance, facility planning ... etc., [1]. Their role in management is rather distinct when regularly equipped with up-to-date information and full decision maker's knowledge and experiences.

Several aspects have been studied by a number of researchers in order to pin point the capabilities and limitations of such DSS systems. Abou-Kahf [2] reviewed the elements of decision making process given as: selection, alternatives, objectives & motives, time, available human and other resources and internal and external environments. He also summarized the steps of decision making for problem solving to be: a) definition of the objectives and problem, b) determination of the alternatives to achieve the objectives, c) assessment of these alternatives under the constraints of cost, time, revenue as well as social and cultural aspects, d) choice of the optimal alternative, and finally e) decision making.

Decision making can be made by individuals, majority or consensus and can be accomplished by using either traditional or quantitative techniques. Traditional decision making techniques include: experience, intuition, trial & error, follow an example and brain storming. Unfortunately, these are vulnerable to errors due to: lack of experience, limited memory, emotional effects,...etc. On the other hand, quantitative decision making techniques including: Expected value used in risk analysis, Checklist models, Decision tree, Weighting factor or Operations research techniques, comprise more safer decision support tools.

The development and implementation approaches for DSS systems have been, and seemingly will be, under investigation for a considerable time for a variety of reasons. For instance, the most important question, which is: *Who does what in DSS developing?*, [3] , is still to be answered. While Sprague [1] noticed that traditional analysis and design techniques have prove inadequate because there is no single comprehensive theory of decision making, he also quoted that DSS systems need to be built with short and rapid feedback from users to ensure that development is proceeding correctly, i.e. iterative design is employed. This means that the system must be adaptive in nature.

Aspects of DSS implementation success have been investigated by several researchers. Alavi and Joachimsthaler [4] determined the implementation success variables of a DSS system to be: system use, decision making performance (cost or profit), decision making time, user satisfaction with the system, user confidence in the decision and the user attitude towards DSS. They have classified the user factors affecting the implementation success to be: 1) cognitive style, 2) personality, 3) demographic variables: age, sex and education, and 4) user-situational variables: involvement, training and experience. During their study, they noticed that there is no single approach to the definition of DSS implementation success existed in the literature and they argued that the user- situational variables are the most important factors to DSS implementation success and can improve it by as much as 30%.

Kottemann and Remus [5], conducted a study which utilized the bootstrapping paradigm of psychological research to help answer the question of how reliably are decision model naturalness and performance related?. Their study results, however, did not have clear-cut policy implications, but raised some interesting issues about the value matching the structure of a decision model with the cognitive style of its users. Silver [6], studied the directed and non directed change induced by DSSs. Also, empirical investigations have been conducted on the relationship between DSS usage and the system performance, e.g. Le Blanc and Kozar [7].

The issue of the user interface is of recognized importance as it is directly contributing to the DSS implementation success. For instance, Subramanian *et al.* [8] performed a comparison of the decision table and tree concerning their effectiveness in the user interface. In their experiment they concluded that the tree performs better than the table. Also, Todd and Benbasat [9] performed an experimental investigation and addressed the impact of computer based decision aids on decision making strategies.

The role of DSS can be extended for strategic planning, where one of the important aspects to be addressed in this process is the knowledge level during the occurrence of certain situations. This issue means that if a given condition occurs then a pre specified action is to be taken with the aim of reducing the losses (or cost) or increasing the revenue. A condition, in this context, means a situation where its environment and all associated and governing parameters are uniquely identifiable. Under the constraint of full knowledge, estimating the effects of these parameters, and therefore assessing the losses (or revenue), would result in a straight forward strategic planning.

This situation of full knowledge is not always valid, simply because not all conditions can usually be uniquely identified but rather only associated observations are monitored. The problem arises from the unawareness of the real underlying phenomena that caused the observation. Several observations may result from a given state of nature and inversely, several states of nature would have a common observation. This situation exhibits an inherent, and of chief importance, uncertainty as the question: "which state of nature resulted in the monitored observation?", needs to be answered in order to take a proper action.

In this paper a model has been employed and a software system has been developed and used for strategic planning under these uncertainty conditions. The presented software system is meant to model real systems characterized by a many-to-many relation between possible states of nature and observations. It requires the knowledge of the decision makers to be provided in the form of past experiences. Hypothetical situations can also be synthesized and then experienced decision makers are asked to react in view of given observations. The acquired knowledge can be amended to the knowledge base for possible investigation. The software examines an automatically generated exhaustive set of possible strategies and locates the ones which would result in minimum losses. The losses function is chosen to have a static component and a time related component. The static component represents the fixed cost associated with the action while the time related component represents the estimated value of the elapsed time required to overcome a given problem by taking a proposed action. This lost time estimation process constitutes another source of uncertainty. Other elements of uncertainty may be attributed to the decision maker style of reaction as exposed to different situations.

2. DESIGN & IMPLEMENTATION

Figure (1) illustrates the main units of the software system where it consists of three major functional components: the preprocessor, the strategy synthesizer and the strategy evaluation subsystems.

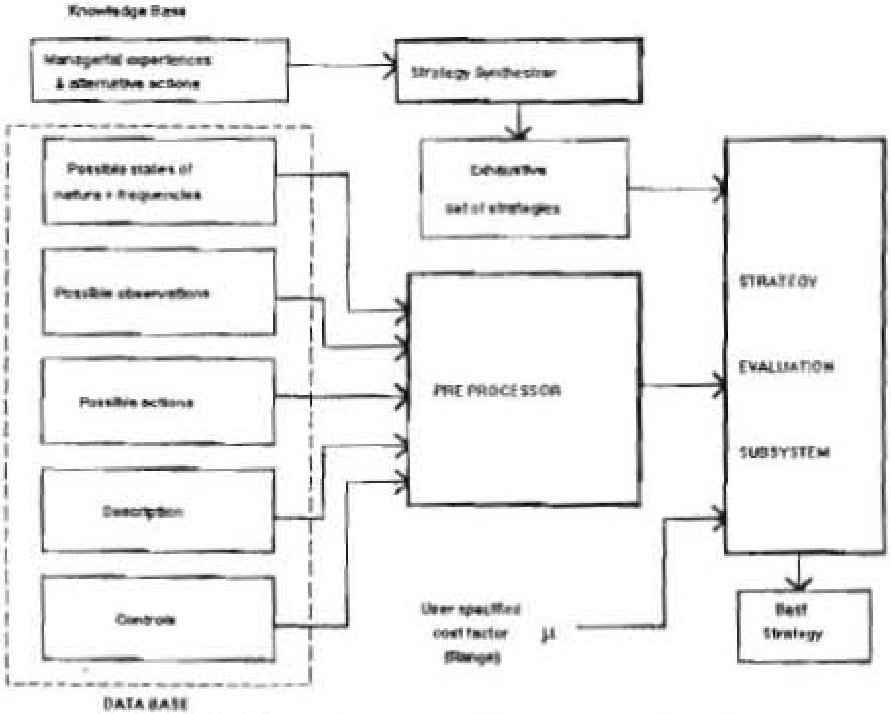

Figure 1. Block diagram of the proposed system

The preprocessor's main function is to compile the user specified data and measures of the real system performance history and produce a set of data files suitable for processing by the strategy evaluation subsystem component along with the appropriate documentation. Its input data base consists of a number of data sets of :

. possible states of nature $\{\theta_1, \theta_2, ... \theta_n\}$ along with their frequencies of occurrence as statistically computed from historical data
. possible observations $\{z_1, z_2, ... z_m\}$ associated with the states of nature
. possible actions $\{a_1, a_2, ... a_n\}$ to be taken in view of given observations

Also, the input data base contains a description of the real system historical performance data as well as control information describing the real system parameters.

The preprocessor then computes the following:

1. The loss table $L_{\theta,a}$ where it gives the loss expected from taking a specific action (a) given the occurrence of the state of nature (θ). It has been chosen in this study to be composed of two parts as shown in equation (1).

$$L_{\theta,a} = Co_{\theta,a} + \mu\, T_a \qquad (1)$$

where $Co_{\theta,a}$ is a fixed cost component, T_a is the computed average elapsed time while taking the associated action (a) and μ is a cost factor to be specified interactively by the user while running the strategy evaluation subsystem component. This is supplied as a range of values to allow for sensitivity studies. This system parameter is employed to account for the inherent uncertainties in evaluating the overall losses. The entries of the loss table are computed from the compiled past-performance information.

2. The probability of occurrence of a given observation (z) given that a state of nature (θ) has occurred: $f(z|\theta)$. As mentioned earlier, it is possible of course that a given state of nature may exhibit more than one observation and an observation may be shared by a number of states of nature.

A strategy synthesizer is then used to build an exhaustive set of all possible strategies $\{s_1, s_2, ..., s_r\}$ from past experiences encountered by the decision makers. The decision makers knowledge and experiences are acquired from their description of the actions they have, or would have, taken in view of given observations. This compiled knowledge base is fed to the strategy synthesizer which constructs an exhaustive set of possible strategies to be evaluated by the strategy evaluation subsystem. Alternative and even hypothetical actions can also be amended at this stage for possible consideration in strategic planning.

A strategy, in this context, is essentially a list of actions associated with observations, therefore a table $S_{s,z}$ is constructed such that its elements are *indices to actions.*

The strategy evaluation subsystem's main function is to locate the strategy with minimal cost. It starts with computing the average loss table $\Gamma_{\theta,s}$ which relates the states of nature to the strategies according to equation (2):

$$\Gamma_{\theta,s} = \sum_z f(z|\theta)\, L_{\theta,S_{s,z}} \qquad (2)$$

Given the probability of occurrence of each state of nature $f(\theta)$, as computed from historical data, the strategy evaluation subsystem then locates the strategy for which the computed total average cost C_s for all possible states of nature is minimal, where C_s is given by equation (3) below:

$$C_s = \sum_{z} f(\theta)\, \Gamma_{\theta,s} \tag{3}$$

In cases where the probabilities of occurrence of states of nature are not known, the minimax rule is usually applied by choosing Cs to have the maximum $\Gamma_{\theta,s}$ as suggested by equation (4) and then choose the strategy associated with the smallest C_s value.

$$C_s = \underset{\theta}{\mathrm{MAX}} \{ \Gamma_{\theta,s} \} \tag{4}$$

Following is an application of the system described above to model the operations of Alexandria Customs Computer Center in order to experiment this proposed framework and the developed software system, Aly [10]. The acquired data have been assembled and categorized strictly according to the major categories as suggested by the decision makers' past experiences. Although other researchers would be interested in measuring and enhancing the productivity of such computer systems, see for example Kriebel and Raviv [11], the current work has been done without any attempt from the authors to interpret the significance or correctness of the data elements regarding whether a state of nature or an observation have been identified and/or classified properly or whether an action is suitable for a given situation. This approach is adopted in order to model a real system without any externally intervening effects or biasing agents.

3. APPLICATION AND RESULTS

The following tables have been compiled from a monthly report produced by Alexandria Site of the Egyptian Customs Computer Network. Data have been collected for 14 months (December 90-February 92, except for July 91 where no data was available). This monthly report contains information on the resource failure times (in minutes) as recorded by the system manager. Failure causes and the taken remedial actions are also included. These data have been analyzed and then categorized into the following tables. Table I contains the possible states of nature (θ) and their computed frequencies of occurrence $f(\theta)$.

Table I. Possible states of nature and their frequencies

State of nature	Frequency
1 CPU down	0.008175
2 Power failure	0.004762
3 Terminal error	0.053056
4 Communications error	0.024666
5 File system damage	0.014683
6 Printer error	0.012952
7 MT fault	0.004762

Table II summarizes all possible observations as have been recorded during the study period.

Table II. Possible observations

1	Cache memory defective
2	Faulty battery
3	UPS battery defective
4	UPS power problem
5	Processor board faulty
6	Circuit breaker problem
7	Air condition problem
8	Dead lock
9	Transaction problem
10	No polling
11	Terminal give error message
12	Video problem
13	Internal processor board error
14	Cable problem
15	Video amplifier problem
16	Interference
17	Operator fault
18	Slow response time
19	Keyboard locked
20	Modem-terminal cable cut
21	No communication
22	Files corrupted
23	No output
24	Bad printout
25	General printer failure
26	MT not running
27	MT head damaged
28	Read error

Actions to be taken in view of a given observation have been categorized into four possibilities as given in table III.

Table III. Possible actions

1	Wait
2	Fix
3	Repair
4	Replace

Table IV illustrates a compiled sample of the past-performance information file contents. It is worth noting that different managers may exhibit different reactions towards observations and therefore different actions may be taken.

Table IV emphasizes the fact that several observations may share the same state of nature as well as several states of nature may have common observations. Also, it emphasizes that varying actions may be taken in view of a given observation and this is mainly due to the past experience of a certain site manager.

Table IV. A compiled sample of past-performance information

State of nature	Observation	f Action	C_a	Fault duration (minutes)
CPU down	Cache memory defective	1 Replace	1000.0	105
	Faulty battery	2 Replace	500.0	15,15
	UPS battery defective	1 Repair	100.0	60
	UPS power problem	1 Repair	300.0	60
	Processor board faulty	2 Replace	2000.0	400,80
	Circuit breaker problem	1 Repair	300.0	145
	Air condition problem	2 Wait	0.0	75,75
Power failure	Dead lock	1 Wait	0.0	600
Terminal error	Transaction problem	2 Fix	0.0	15,15
	No polling	3 Wait	0.0	225,225,510
	Terminal give error msg.	1 Wait	0.0	15
	Video problem	1 Fix	0.0	175
	Internal processor board	2 Replace	500.0	15,90
	Cable problem	2 Replace	30.0	985,985
	Video amplifier problem	1 Replace	100.0	220
	Interference	2 Repair	100.0	105,105
	Cable problem	2 Repair	30.0	1200,555
	Operator fault	2 Fix	0.0	555,40
	Transaction problem	1 Replace	3000.0	175
	Cable problem	2 Replace	20.0	60,70
	Slow response time	2 Repair	50.0	170,120
	Keyboard locked	1 Fix	0.0	55
Communications error	Modem-termnl cable cut	1 Replace	50.0	15
	No communication	1 Wait	0.0	2848
	Modem-termnl cable cut	2 Fix	0.0	90,90
	No communication	2 Replace	500.0	25,40
File system damage	Files corrupted	3 Fix	0.0	630,620,600
Printer error	No output	1 Replace	200.0	70
	No output	1 Replace	30.0	25
	Bad printout	1 Replace	50.0	25
	Bad printout	1 Fix	0.0	30
	General printer failure	6 Replace	3000.0	300,80,45,25,12,15
	Transaction problem	2 Replace	200.0	65,40
	Cable problem	1 Replace	30.0	900
MT fault	MT not running	1 Replace	50.0	120
	MT head damaged	1 Replace	200.0	240
	Read error	1 Fix	0.0	240

The software system produces a document containing all computed elements along with the best strategy to be followed. It produces the loss table $L_{\theta,a}$ as shown in Table V from which useful management information can be extracted.

Table V. Estimated loss $L_{\theta,a}$

States of nature	Actions			
	Wait	Fix	Repair	Replace
CPU down	150μ	∞	$233+88\mu$	$2000+205\mu$
Power failure	600μ	∞	∞	∞
Terminal error	488μ	214μ	$120+752\mu$	$840+520\mu$
Communication error	2848μ	180μ	∞	$525+40\mu$
File system damage	∞	1850μ	∞	∞
Printer error	∞	30μ	∞	$3118+267\mu$
MT fault	∞	240μ	∞	$125+180\mu$

The symbol (∞) in Table V represents an infinite cost when applying a certain action for a given state of nature to emphasize that it is either inapplicable, or no such action has been taken before and worth to be investigated. For instance an inapplicable action would arise as follows: "when the CPU is down there is no fixation action that can be taken". Also, candidate actions to be investigated would be for example: "repair the printer instead of replacing it". It is rather noted that, for instance and due to the manager's past experience "the CPU was down because of an air condition problem!" (see Table IV) and therefore nothing could be done but "waiting!" until the problem has been eliminated. It is therefore suggested that information gathered from Table V can be adopted to overview and rectify the decision makers experiences. Consequently, this rectification process would reclassify the states of nature and the observations and would reform their mutual relations.

Also the system computes $f(z|\theta)$ which can be utilized by the decision makers to identify the possible states of nature that could have occurred associated with a given observation. Table VI represents such frequencies:

Table VI. Computed $f(z|\theta)$*

States of nature	OBSERVATIONS																											
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
CPU down	10	20	10	10	20	10	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Power failure	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Terminal error	0	0	0	0	0	0	0	0	13	13	4	4	8	26	4	8	8	8	4	0	0	0	0	0	0	0	0	0
Comm. error	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	50	0	0	0	0	0	0	0
File system damage	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
Printer error	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	15	15	44	0	0	0
MT fault	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33	33	33

* all numbers are multiplied by 100.

Inspection of Table VI gives clues for the decision makers towards the possible cause or causes of a given observation. For instance "If a deadlock occurs (observation 8 in Table II) then it is definitely a power failure problem and vice versa" as previously experienced. Also "if cache memory is defective (observation 1 in Table II) then the CPU is down", this rule is inferred from the fact that all other entries of $f(z|\theta)$ for the rest of states of nature are zeros for this particular observation. Also, simultaneous observations may uniquely identify the occurrence of a certain state of nature, e.g. observation 9 would give rise to two possibilities: "Terminal error" or "Printer error", while if observation 10 is monitored simultaneously, the problem is then uniquely identified as "Terminal error".

Synthesized strategies shown in Table VII are drawn from all possible actions for each observation as derived from the compiled knowledge base.

Table VII. Synthesized strategies

Strategy	Observations 1111111111222222222 1234567890123456789012345678
1	4433431121124443232212424442
2	4433431141124443232212424442
3	4433431121124443232412424442
4	4433431141124443232412424442
5	4433431121124443232242424442
6	4433431141124443232242424442
7	4433431121124443232442424442
8	4433431141124443232442424442
9	4433431121124443232212444442
10	4433431141124443232212444442
11	4433431121124443232412444442
12	4433431141124443232412444442
13	4433431121124443232242444442
14	4433431141124443232242444442
15	4433431121124443232442444442
16	4433431141124443232442444442

In a typical session, the strategy assessment process would request the user to interactively supply a range of expected values for the cost factor (μ) and then the strategy evaluation subsystem would locate the best strategy associated with each (μ) value. Figure (2) represents the lower bound of C_s values and, hence, the suggested strategy to be followed for a given (μ) range.

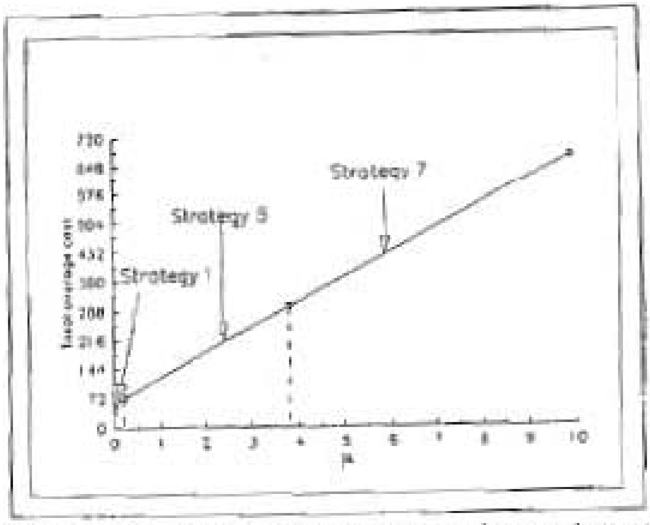

Figure 2. Total average cost lower bound

4. CONCLUSION

The framework presented in this paper is suggested to model a range of real systems possessing similar features characterizing their behavior. These real systems may be subjected to some unbalanced situations or problems which require some decision maker to react and take proper actions. The uncertainties which would affect the performance of the decision maker are grouped into three most prevailing sources: a) the inherent uncertainties in the real systems as regard to the real causes which lead to certain observations, where the former are probably obscured, b) Losses estimation criteria and c) the decision maker style of reaction.

The proposed system's goal is to set up optimal strategies in the form of rules to be followed when problems occur. These rules are inferred from the overall experiences and knowledge of the decision makers, and therefore constitute a safe way for action taking. In addition, hypothetical situations can be synthesized and then proposed actions are suggested to be experimented to decide upon their merits. The information obtained at various phases while applying such systems can also be utilized in spotting the knowledge gaps and rectifying and updating the gained experiences and knowledge of the decision makers on the performance of the real systems.

In this study the cost estimation function used in computing the elements of the loss table has been chosen to be a linear function of time, which to some extent allowed for allocating several strategies versus the cost factor. If required, other functions can be used according to the user specific needs.

REFERENCES

[1] Sprague R.H., 1989, A Framework for the Development of Decision Support Systems, Ed. R. Sprague Jr. and H. Watson, Prentice-Hall International Inc.

[2] Abou-Kahf A.S., 1990, Fundamentals of Management, Business Administration Dept., Faculty of Commerce, Alexandria University.

[3] Young L.F.,1989, A Corporate Strategy for Decision Support Systems, Ed. R. Sprague Jr. and H. Watson, Prentice-Hall International Inc.

[4] Alavi M. and Joachimsthaler E.A., 1992, Revisiting DSS implementation research: A Meta-Analysis of the Literature and Suggestions for Researchers, MIS Quarterly, pp. 95-116.

[5] Kottemann J.E. and Remus W.E., 1989, A Study of the Relationship Between Decision Model Naturalness and Performance, MIS Quarterly, pp. 171-181.

[6] Silver S.M., 1990, Decision Support Systems: Directed and Non directed Change, Information Systems Research 1:1.

[7] Le Blanc L.A. and Kozar K.A., 1990, An Empirical Investigation of the Relationship between DSS Usage and System Performance: A Case Study of Navigational Support System, MIS Quarterly.

[8] Subramanian G.H., Nosek J., Raghunathan S.P. and Kanitkar S.S., 1992, A Comparison of the Decision Table and Tree, Communications of the ACM, Vol. 35, No. 1.

[9] Todd P. and Benbasat I., 1991, An Experimental Investigation of the Impact of Computer Based Decision Aids on Decision Making Strategies, Information Systems Research 2:2.

[10] Aly E.M., 1994, Decision Support Systems and the Sociotechnical Aspects of Information Technology in the Customs Sector, M.Sc. thesis, Dept. of Information Technology, IGSR, Alex. Univ.

[11] Kriebel C.H. and Raviv A., 1980, An Economic Approach to Modeling the Productivity of Computer Systems, Management Science, Vol. 26, No. 3.