BEHAVIOUR & INFORMATION TECHNOLOGY, 1999, VOL. 18, NO. 6, 413–430

Development of a methodology for optimizing elicited knowledge

CHIN-JUNG CHAO†, GAVRIEL SALVENDY† and NANCY J. LIGHTNER‡

†School of Industrial Engineering, Purdue University, West Lafayette, Indiana 47907, USA

‡College of Business Administration, University of South Carolina, Columbia, South Carolina 29208, USA

Abstract. In this paper a conceptual framework and an operational methodology is presented for describing the most appropriate knowledge elicitation method (protocol, interview, induction and repertory grid) for three classes of tasks (diagnosis, debugging and interpretation) and for experts with strengths in various factors of cognitive abilities. Using the dependent variables of, (1) total knowledge captured; (2) time to acquire knowledge; (3) knowledge quality; (4) efficiency of the knowledge elicitation method; and (5) importance of resulting data, experimental results indicate the various strengths of the four knowledge elicitation methods. The knowledge acquired is also significantly affected by the combined factors of expert's strengths in different cognitive factors and the method of knowledge elicitation used. Based on these findings, a Matching Index for combining tasks, knowledge elicitation methods and cognitive abilities of the expert is described. The outcome of this research provides theoretical and practical implications for Human Computer Interaction (HCI) and training of knowledge engineers.

1. Objectives and Significance

Knowledge elicitation is the first step in building expert systems and it is a major bottleneck in the construction of expert systems (Hoffman 1987). Although many different methods of knowledge elicitation exist, the following issues are not known about these methods: (1) which knowledge elicitation methods are more suitable for different tasks; (2) which knowledge elicitation methods best extract different kinds of knowledge; and (3) how important of a factor is the strengths of factors in cognitive abilities of experts in the knowledge elicitation procedure. The primary objective of this research is to combine the above three questions into a new conceptual model, which will provide a framework and a methodology for selecting the most appropriate knowledge elicitation method for each task.

This derived conceptual model and methodological framework has an important impact on the design of human computer interaction and training of knowledge engineers.

2. Background literature and derivation of hypotheses

An expert system is a computer program that contains both declarative knowledge (facts about objects) and procedural knowledge (information about courses of action). The purpose of expert systems is to emulate the reasoning processes of human expert in a particular domain (Hunt 1986). Little research exists about the comparative effectiveness of different knowledge elicitation methods. Thus there are few guidelines to aid the knowledge engineer in the selection of the knowledge elicitation techniques, the anticipation of problems, or the estimations of progress (Fox *et al.* 1987). The study presented in this paper attempts to alleviate or reduce the problems associated with the selection of knowledge elicitation methods by establishing a conceptual framework for choosing an appropriate method. This framework consists of the following four parts:

1. knowledge elicitation methods
2. factors of cognitive abilities of humans
3. knowledge structures
4. task types

A description of each of the dimensions are presented in the following four sections.

Behaviour & Information Technology ISSN 0144-929X print/ISSN 1362-3001 online © 1999 Taylor & Francis Ltd
http://www.tandf.co.uk/JNLS/bit.htm
http://www.taylorandfrancis.com/JNLS/bit.htm

2.1. *Knowledge elicitation methods*

Knowledge elicitation is the process which extracts correct problem-solving expertise from knowledge sources (e.g. domain experts). Currently, there are numerous knowledge elicitation methods available, ranging from manual methods to automatic methods (Waterman and Hayes-Roth 1983). Based on the information presented by Lehto *et al.* (1992), a taxonomy of the most commonly used knowledge elicitation methods is derived (Table 1).

Many of the expert systems developed use either manual (interview and protocol) or automatic (induction and repertory) knowledge elicitation methods (Boose 1985, 1986, Marcus *et al.* 1985, Gaines and Shaw 1986). The following sections present a review and discussion of these elicitation methods.

2.1.1. *Manual knowledge elicitation methods*:

- Interview. This is the most widely utilized method of obtaining knowledge from a human expert (Tuthill 1990). During the interview, the knowledge engineer proposes some hypothetical problems pertaining to the tasks in question and asks the expert to solve them. During the problem-solving process, the expert should reveal the steps taken in making decisions and designing a solution (Olson and Rueter 1987). After the knowledge engineer obtains the knowledge from the human expert, he/she encodes them into the computer to form the basis for the knowledge structures in the expert systems.
- Protocol Analysis. This method asks the experts to 'think aloud' while performing a task or solving a

Table 1. Knowledge elicitation methods

Method	Repertory	Induction	Interview	Protocol	Reference
BLIP		*			Morik 1989
ID3		*			Quinlan 1987
INDUCE		*			Michalski *et al.* 1980
INDUCT		*			Gaines 1989
ATOM		*			Gaines 1977
AQ		*			Michalski 1983
CART		*			Grawford 1989
PRISM		*			Cendrowska 1987
CODE				*	Skuce 1989
LAPS				*	di piazza 1988
MACAO				*	Aussenac *et al.* 1988
MEDKAT				*	Jagannathan and Elmaghraly 1985
ARK			*		Tonn *et al.* 1989
KNACK			*		Hsieh *et al.* 1988
ELI			*		Silverman *et al.* 1989
MORE			*		Kahn *et al.* 1985
SALT			*		Marcus 1989
PROTOKI			*		Murray 1989
TEIRESIAS			*		Davis and Lenat 1982
MOLE			*		Eshelmen 1988
PLANET	*		*		Shaw 1984
IRA-GRID	*		*		Linster 1989
SMEE	*		*		Garg-Janardan and Salvendy 1987
COGNOSYS			*	*	Woodward 1988
ETS	*	*	*		Boose 1986
AQUINAS	*	*	*		Boose 1989
KSSO	*	*	*		Shaw 1989
KRITON	*		*	*	Linster 1989
KITTEN	*				Shaw and Gaines 1987
PATHFINDER	*				Vooke and McDonald 1987
DART	*				Boose 1989

Source: Reorganized from Lehto *et al.* (1992)

problem (Bainbridge 1979, Hoffman 1987, Neale 1988, Tuthill 1990). Johnson *et al.* (1987) define protocol analysis as the process used by cognitive psychologists to understand human problem-solving and decision making. The experts should report as much of what they are thinking about as possible, especially regarding the alternatives they have considered and their solutions and reasoning processes. For this method, a video or audio recorder can be utilized during the protocol analysis so that the analyst can further review the session. Ericcson and Simon (1984) have shown that verbal reports are a valuable and reasonably reliable source of information about human cognitive processes.

2.1.2. *Automatic knowledge elicitation methods*:

- Induction. Induction is the process of extracting knowledge from examples. In this method, the expert provides a set of examples, called a training set, consisting of different types of decisions in a specific task and the relevant attributes of each task. The attributes are characteristics of the examples that the expert uses to make decisions about problem-solving. The training set is then used to infer the decision processes of experts by using an inductive algorithm, which can induce a set of knowledge in a form of decision tree or decision rule. This method can predict the decisions for examples not included in the training set. However, experts are still required to validate the decision tree or rules.
- Repertory Grid. This method is derived from Kelly's (1955) personal construct theory. A repertory grid uses identified elements and constructs to describe objects. An element is what an expert considers relevant to the problem under consideration and a construct is a bipolar characteristic which each element has to some degree. The mapping of the elements onto the constructs produces the two-dimensional grid of relationships (Shaw and Gaines 1987). During the procedure, an expert is presented with three elements and is asked to differentiate any two of these elements from the third by pointing out the construct pole and the contrast pole. Then, all of the elements in the set are rated along this construct on a scale of 1 to 5 (scale 1 applies to the contrast pole, and scale 5 applies to the construct pole). This procedure is repeated until all elements of the grid have been identified. Following this a factor analysis or cluster analysis is used to identify the experts' use of these relationships. Knowledge elicitation methods, such as ETS (Boose 1985) and SMEE (Garg and Salvendy 1988), adopt this method.

Table 2 illustrates some of the published advantages and disadvantages of the above knowledge elicitation methods. These references seem to focus on the elicitation *process* with little mention of the suitability of the method to a specific task. The aim of this study is to provide direction in obtaining more effective knowledge elicitation results. This direction consists of a match between task type and the method of knowledge elicitation.

2.2. *Factors of cognitive abilities of humans*

During the knowledge elicitation process, individual differences in the expert's conceptual model are usually ignored (Sein and Bostrom 1989). Lehner and Kralj (1988) demonstrate that the 'cognitive model' is a dominant factor in the quality of user/expert system interaction. Gentner and Stevens (1983) suggest that experts use a 'mental model' to reason, and the expert's strengths in various factors of cognitive abilities allow them to manipulate their knowledge. Consequently, generating an accurate mental model may be a necessary step in the design and use of human—computer interface systems (Lehner and Zirk 1987). Of the twenty-three cognitive factors of human abilities identified by Ekstrom *et al.* (1976), ten were previously found to be significant in the ability to acquire knowledge in this task environment (Chao and Salvendy 1995).

2.3. *Task types*

Bylander and Chandrasekaran (1987) suggest that different knowledge elicitation methodologies should be required for different kinds of tasks. Application tasks in expert systems have been classified by many authors (Hayes-Roth *et al.* 1983, Clancey 1986 and Boose 1988). These classifications include interpretation, prediction, diagnosis, design, planning, monitoring, debugging, repair, instruction and control. Boose (1988) classifies the above tasks into two categories: analysis tasks and synthesis tasks.

Synthesis tasks obtain solutions from a set of component or sub-component solutions. Often there are too many components and too many possible solutions to the task. Typically, a specific knowledge elicitation method is developed to solve a particular synthesis task, but it is difficult to determine the optimal method of knowledge elicitation because there are no reliable and valid criteria for evaluating the knowledge derived from synthesis tasks (Boose 1988).

Table 2 Comparison of knowledge elicitation methods.

Method	Advantages	Disadvantages	Reference
Interview	*most prevalent *simple *fills in the gaps resulting from the knowledge engineer's descent into the deeper aspects of the knowledge domain *quickly generates a lot of knowledge *elicits unforeseen information *little demand on expert other than time *requires little equipment *highly flexible, portable	*time-consuming *expensive *little methodology to guide the interaction between expert and knowledge engineer *subjective *poorly defined process *lack of direction *inefficiencies, frustrations limit value *highly dependent on the knowledge engineer *not all experts actually does reflects what he thinks	Hart 1985 Olson and Rueter 1987 Gammack 1987 Parsaye and Chignell 1988 Neale 1988 Gammack and Young 1984 Tuthill 1990 Hoffman 1987
Protocol analysis	*less time consuming for an initial prototype system *elicit procedure that experts use in problem-solving which they may not be able to articulate *no delay between the act of thinking of something and supporting it *creates a detailed picture of the representation	*difficult and time consuming *takes longer to perform and analyze *gaps and jumps in verbalization *retrieves smaller amount of the necessary information *no necessary correlation between verbal report and mental behaviour *expertise-intensive *vulnerable to biases dereived from the idiosyncrasies of the individual	Olsen and Rueter 1987 Tuthill 1990 Parsaye and Chignell 1988 Rurton et al. 1987 Kuipers et al. 1987 Shaw and Gain 1987 Neale 1988 Gammack and Young 1984
Repertory grid	*more effective in very complex applications *provides method of automatic induction that capture without computer assistance *creates a foundation for a conceptual framework for knowledge	*restricted to analysis problems *distinction may not be publicly agreed upon *larger concept sets require more expert time *becomes unmanageable with more than about 100 subjects	Olsen et al. 1987 Tuthill 1990 Parsaye and Chignell 1988 Parsaye et al. 1988 Shaw and Gain 1987 Neale 1988 Gammack and Young 1984
Induction	*describes the decision-making process itself *unbiased and objective *can detect things of which the expert is unaware *indicates gaps and problems *consistent, repeatable	*results can be inaccurate for insufficient examples *domains where rules are not appropriate are unsuitable for induction *incomplete or inadequate set is likely to result without explanation	Hart 1985 Hart 1987 Neale 1988

On the other hand, analysis tasks involve the interpretation of information through components. Hence, all the possible solutions in certain problems can be identified reliably, validly and objectively.

Basically, an analysis task is composed of the following three kinds of sub-tasks:

- Debugging. This task prescribes remedies for malfunctions and offers suggestions for correcting the problems which have been diagnosed.
- Diagnosis. This is the process of fault-finding in a system based on the observed data of a specific task. As such, diagnosis is the intermediate procedure to the debugging task.
- Interpretation. This task infers a situation description from observation of an expert performing a task. An interpretation system explains observed data by assigning symbolic meanings to the task. These meanings describe the situation or system state which accounts for the observed data (Hayes-Roth *et al.* 1983).

2.4. *Knowledge structures*

Human knowledge structures can be classified in a variety of ways. But independent of their classifications, they require different knowledge elicitation methods to capture the knowledge most effectively (Gammack and Young 1984).

The expert's knowledge is composed of two types of knowledge: procedural knowledge and declarative knowledge. Procedural knowledge is defined as the strategies and sequences of operations used in problem-solving (know-how). Because it is concerned with a process of completing a task, it is called procedural knowledge (Nagao 1990). Declarative knowledge is composed of facts and the meanings stored in memory. Declarative knowledge is composed of the expert's background knowledge of the problem. Their relations can be explained by the following example:

IF A B C => D

In this example, the whole production rule is procedural knowledge, and it includes three parts of declarative knowledge (A, B and C). A rule is said to be triggered if the premise (A, B and C) is satisfied and the condition (D) is performed.

2.5. *Development and statement of hypotheses*

A major problem in knowledge elicitation is not only which knowledge elicitation method is best for which task, but also whether there is a model that can account for their relationship. Based on the background literature presented in Section 2.1. and 2.2., the four most commonly used knowledge elicitation methods (protocol, interview, induction and repertory grid), and three analysis sub-tasks (diagnosis, debugging and interpretation) are used to test the proposed conceptual framework.

The rationale for the three dimensional conceptual model presented in Figure 1 is based on the following assertions:

- Different tasks require different human abilities, which has been illustrated by Chao and Salvendy (1995).
- Different knowledge elicitation methods may be viewed by the knowledge engineer as different tasks. Consequently, the point made above regarding tasks and human abilities is also applicable here.
- Knowledge elicitation methods and task characteristics are hypothesized to be interrelated. Thus, if the same abilities are needed for both the task and the knowledge elicitation method, there is a match between the two and the best knowledge elicitation can be achieved. However, when there is no match, the knowledge elicitation method may not give good results.

In this model, ten dominant factors of cognitive abilities needed for each knowledge elicitation method are used from the study by Chao and Salvendy (1995). That study showed that the factors of cognitive abilities of experts in these ten areas determined the success of each of the knowledge elicitation methods. A knowledge elicitation method, which is most appropriate for one expert for a specific task may not be effective for the same expert on a different task or by another expert on the same class of tasks. Therefore, developing a model and methodology for matching the attributes of tasks, the cognitive characteristics of individuals and the knowledge elicitation methods is suggested. The better this match or the higher this index is, the better the captured knowledge will be. Indeed, this conceptual model lays the foundation for developing an operational model for selecting the most efficient knowledge elicitation method for each task.

Three hypotheses are proposed to test the validation of the model.

1. Different knowledge elicitation methods extract different amounts or types of procedural and declarative knowledge from the domain experts.
2. Each knowledge elicitation method is best suited for a specific task or class of tasks.
3. If the above two hypotheses are supported, then the following function is hypothesized:

   Matching Index = f (knowledge elicitation method, cognitive ability, task attribute)

3. Method

3.1. *Rationale for testing hypotheses*

Based on the hypotheses proposed above, a statistical experiment is used to test whether one knowledge elicitation method is superior to other methods for various analysis task types. The experimental design uses three task types and four knowledge elicitation methods as the treatments in a nested factorial design. This design is used to test Hypotheses 1 and 2. By using the all-possible-regression selection procedure (Mendenhall and Sincich 1988) to identify the important variables in the multiple regression equations (which include tasks, knowledge elicitation methods, factors of cognitive abilities of human and the interactions of these three factors), then Hypothesis 3 can be tested. The detailed description of this experiment is presented below.

3.2. *Task description*

The tasks selected involve the understanding of two computer programs written in the FORTRAN language. One is used in the diagnosis task and the other is used in the debugging task. In the interpretation task, subjects have to explain the possible reasons of the observed error message in a FORTRAN program. The reason for selecting these three task types in the experiment is that the knowledge acquired by using each knowledge elicitation method can be objectively compared with a 'standard solution' for each task. The following is the detailed description of each task.

3.2.1. *Diagnosis task*: The diagnosis task consists of error detection in a FORTRAN program consisting of 278 lines of code. This task requires subjects to detect

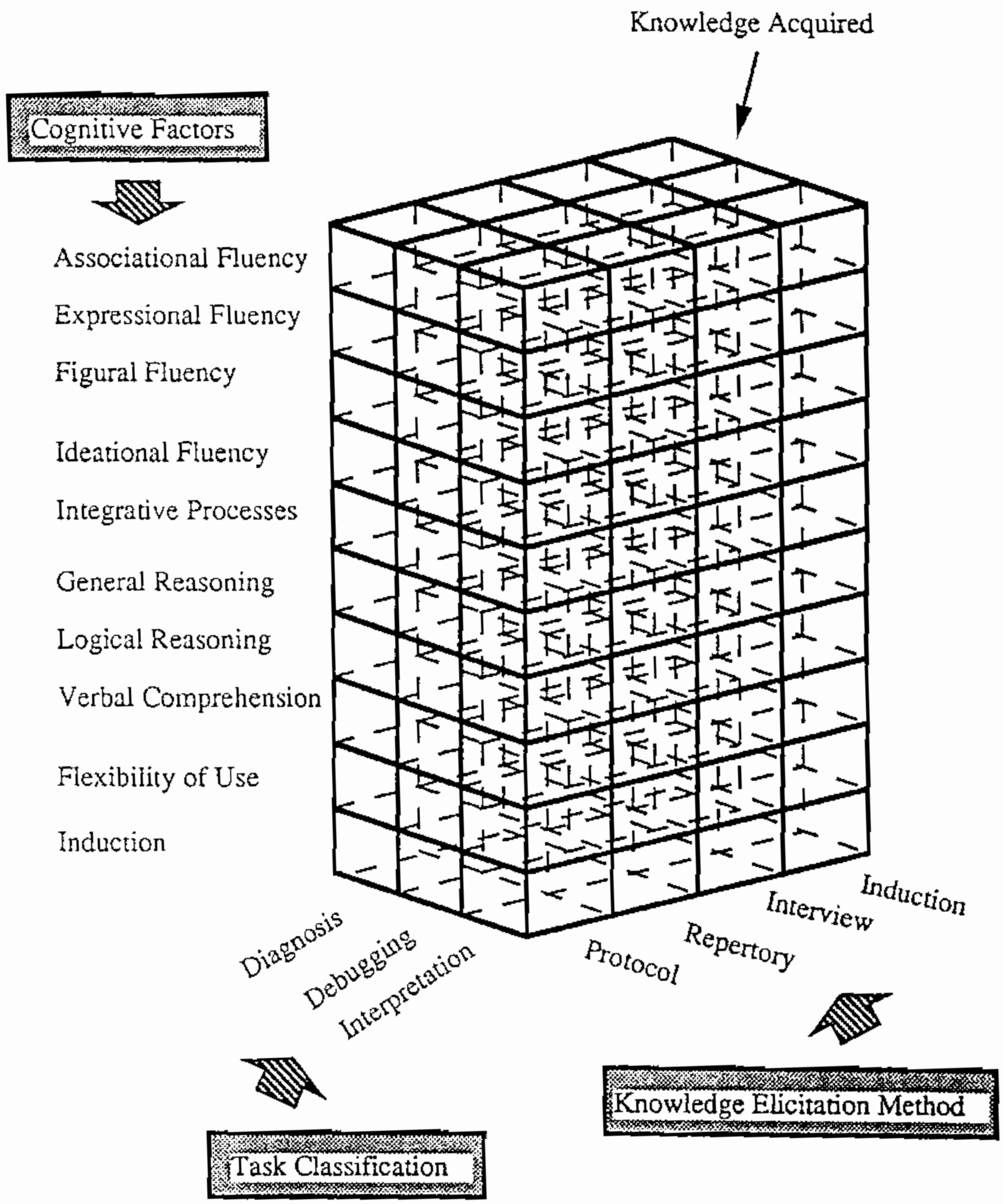

Figure 1. A conceptual model for optimizing knowledge elicitation.

errors in the program. A randomly assigned knowledge elicitation method is used by the subject to find all the errors in the program. The subject does not need to correct them.

3.2.2. *Debugging task*: For the debugging task, the FORTRAN program has been altered to accommodate company policy changes. Some statements have been changed into invalid conditions, and these invalid conditions are indicated by marking the yellow symbols on these specific statements. The subjects have to use the assigned knowledge elicitation method to correct these errors.

3.2.3. *Interpretation task*: In the interpretation task program, one type of program error message—'error in format'—is shown on a computer screen, and the subject has to use the specified knowledge elicitation method to explain all of the possible reasons why this format error occurred. There are total of 65 reasons why an error message appeared in this task.

3.3 *Knowledge elicitation method*

3.3.1 *Overview*: Based on the review of the background literature, the four knowledge elicitation methods most often used are selected as the treatments in the experiment. Two of them are manual knowledge elicitation methods and the other two are automated methods. A detailed description of each method is presented below.

3.3.2. *Protocol*: This method asks the subject to think aloud when solving any one of the three tasks, and the knowledge engineer does not prompt the subject except when the subject forgets to talk during the problem-solving procedure, as suggested by Ericcson and Simon (1984). During the elicitation process, both video and audio recorders are used to monitor and record the whole procedure for the purpose of transcribing the context later.

3.3.3. *Interview*: During the interview procedure, the knowledge engineer prepares a list of possible open and closed questions about the specific tasks in advance and the remaining interview conversation depends on the specific conditions at that time. Video and audio recorders are used during the interviews.

3.3.4. *Induction*: In this experiment, the C4.5 induction system, developed by Quinlan (1992), is used as the induction software. This software can induce rules in the form of a decision tree and decision rules.

3.3.5. *Repertory grid analysis*: A repertory grid analysis method, DART (Design Alternatives Rationale Tool), is used in this experiment. DART, developed by Boeing (Boose 1988) employs a repertory grid as the form of knowledge representation and is generated into four hierarchies of data types. The first interview with the expert for building the grid can be done by the software. This software also has the functions of similarity analysis, which can check the similarity between constructs and elements, respectively.

3.4. *Subjects*

Twenty-four subjects are randomly selected from the top 9% of a group of eight hundred students. These students, taught by the Computer Science Department at Purdue University, are expert computer programmers and have especially good skills and knowledge in debugging, diagnosis and interpretation tasks utilized in the FORTRAN language. Six randomly selected students are trained in one of the four knowledge elicitation methods.

3.5. *Dependent variables*

The dependent variables are used to assess the model's usefulness and are outlined below:

1. Completeness—percentage of total knowledge captured. This measurement unit is represented by the percentage of total knowledge captured from the experts divided by total standard solution of the knowledge in each task.
2. Time—total time is represented in minutes. The total time includes the elapsed time from the start of the knowledge elicitation session to the finish of the rule reviewing by the expert. This time is measured with a stopwatch by the experimenter.
3. Inconsistency (Merlevede and Vanthienen 1991)—conflicting rule. Conflicting rules refer to rules with the same premises, but leading to contradictory conclusions. The unit is expressed by the total number of conflicting rules in the final analysis result.
4. Importance of data—This measurement unit is obtained from three super experts using the following scales to estimate each acquired rule's importance. The sum of each acquired rule's weight equals the total importance of data acquired.

1. – not important
2. - less important
3. – moderate importance
4. - important
5. – extremely important

5. Efficiency— number of rule/time. The unit equals the total correct rules divided by the total time elapsed.

3.6. *Independent variables*

The independent variables include three analysis sub-tasks (diagnosis, debugging and interpretation) and four knowledge elicitation methods (protocol, interview, induction and repertory grid).

3.7. *Experimental design*

A nested factorial design (Hicks 1973) is used in the experiment. Every treatment combination includes the same number of subjects to obtain a balanced design. Six subjects are randomly assigned to each knowledge elicitation method. Since the same subjects can not appear in different groups, these subjects are nested within groups. A model that recognizes the restriction error and assumes all errors are not correlated is as follows:

$Y_{ijk} = \mu + M_i + S_{(i)j} + \delta_{(ij)} + T + MT_{ik} + ST_{(i)jk} + \varepsilon_{(ijk)}$

where i = 1,2,3,4 j = 1,2,3,4,5,6 k = 1,2,3

Y_{ijk} = value of the dependent variable from the k^{th} task by the j^{th} subjects, and in the ith knowledge elicitation method.

μ = overall mean

M_i = effect of the i^{th} knowledge elicitation method

$S_{(i)j}$ = effect of the j^{th} subject in the i^{th} knowledge elicitation method

$\delta_{(ij)}$ = restriction error caused by the three tasks being done by the j^{th} subject in the i^{th} knowledge elicitation method, $NID(0,\sigma_\delta^2)$

T_k = effect of the k^{th} task

MT_{ik} = effect of the interaction of the ith knowledge elicitation method with the k^{th} task

$ST_{(i)jk}$ = effect of the interaction of the j^{th} subject in the i^{th} knowledge elicitation method with the k^{th} task

$\varepsilon_{(ijk)}$ = within error, $NID(0,\sigma^2)$

3.8. *Procedure*

At the beginning of the experiment, each subject is randomly assigned one of the following four knowledge elicitation methods: interview, protocol, induction and repertory grid analysis. A training manual is provided to each subject, so that the method of knowledge elicitation can effectively be learned and practised on three practical tasks which are similar to the test tasks. All the subjects are exposed to these tasks for their training. All subjects are trained in the elicitation method until they completely understand how to use the method. After the training period, each subject receives a set of three tasks in a random order. All experts perform the same tasks under the same experimental conditions. All subjects completed the same ten cognitive tests, which were randomly ordered. For the interview and the protocol methods, video and audio recorders were used to record all of the experimental procedure for subsequent review and analysis. After the analysis, rules are derived by the knowledge engineer/computer and the subjects are asked to update all of the rules without missing any useful information in solving each task.

4. Results

4.1. *Data synthesis*

At the completion of each task, the results are analyzed immediately. For the interview and protocol methods, both the video and audio tapes are reviewed by the knowledge engineer to obtain the expert's actual procedural and declarative knowledge. The prototypes of the results are expressed in rules and these results are reviewed by the knowledge engineer with the expert to make sure that the knowledge engineer understands the expert's meaning in deriving the rules. For the induction method, the training set is the input into the C4.5 induction software, and the IF-THEN rules are derived and printed by the computer. As with the other procedure, the results are reviewed by both the knowledge engineer and the expert. For the repertory grid method, the input data are the construct names, element names, grid scales and weights, all of which are provided by the experts. By using DART software, the possible rules are generated and the knowledge engineer reviews these results with the expert.

4.2. *Data analysis*

The multivariate analysis of variance (MANOVA) in a statistical software SAS (1989) program is used to analyze the results of this nested factorial design. In case where there is an overall significant effect due to method or method X task, a Student-Newman-Keuls (SNK) multiple range test is performed to investigate all possible pairs of means in a sequential manner. The SNK results are considered significant when the p-value <0.05. Table 3 shows the descriptive statistics and practical differences for each dependent variable. Practical differences are based on the issues of concern when actually using each elicitation method and reflect the dependent variables. They are: 1) The total amount of knowledge captured by the method; 2) The amount of time required to use the method; 3) The inconsistency between the resultant rules; 4) The rule generation efficiency, calculated as the number of rules per hour; and 5) The importance of the data that results from the method. Detailed discussions of these results are below.

4.2.1 *Total knowledge captured*: The results indicate that there is no significant difference among the methods in acquiring total knowledge ($f(3,20) = 1.12$,

Table 3 The descriptive statistics and practical differences in the use of different knowledge elicitation methods for three tasks.

Knowledge elicitation method		Protocol		Interview		Induction		Repertory grid		% change
Tasks	Dependent variables	$\bar{x}$	S.D.	$\bar{x}$	S.D.	$\bar{x}$	S.D.	$\bar{x}$	S.D.	
Diagnosis	Total knowledge acquired (%)	40.3	4.6	39.3	9.4	41.5	10.3	40.5	15.5	–
	Total time (min.)	265.5*	140.2	258.8	77.7	118.3*	22.3	120.8	34.6	124
	Inconsistency	8.3	2.5	9.8*	3.1	6.0	1.8	2.3*	2.4	322
	Rule/time (#/hr.)	4.9	1.9	4.1*	3.4	9.4*	2.3	9.0	2.1	128
	Importance of data	55.8	5.6	53.6	14.0	56.0	14.7	55.5	22.6	–
Debugging	Total acknowledge acquired (%)	37.2	12.8	52.7*	6.3	50.0	8.7	37.7	12.9	41
	Total time (min)	266.8	97.6	309.7*	52.4	136.7	25.7	128.2*	42.1	141
	Inconsistency	9.8*	4.0	5.3	2.7	6.2	2.6	4.5*	2.6	118
	Rule/time (#/hr)	4.1*	1.3	5.0	0.6	10.8*	2.8	8.4	1.3	162
	Importance of data	60.0*	22.2	85.6	7.9	80.7	13.2	63.2	10.9	44
Interpretation	Total knowledge acquired (%)	25.3	6.1	27.8	4.4	21.2	6.2	21.2	6.2	
	Total time (min)	77.8	48.3	125.3	32.0	48.2	18.4	67.8	31.8	–
	Inconsistency	2.2	1.6	2.5	1.8	0.7	0.8	1.0	1.3	
	Rule/time (#/hr)	15.8	5.3	9.2*	1.7	18.3*	4.8	14.1	3.4	99
	Importance of data	57.3	12.1	61.7	9.0	44.5	8.4	49.2	11.0	–

⌴ significant difference at $\alpha = 0.5$ level

*selected in calculating the % change

$p = 0.3648 > 0.05$). However, the interaction between method and task is significantly different ($f(2,40) = 58.45$, $p = 0.0001 < 0.05$). From the results of SNK test, the conclusions are as follows.

1. For the diagnosis task, the methods are not significantly different
2. For the debugging task, the interview (mean = 52.7, SD = 6.3) and induction (mean = 50.0, SD = 8.7) methods are not significantly different from each other. The protocol (mean = 37.2, SD = 12.8) and repertory grid (mean = 37.7, SD = 12.9) methods also do not produce significantly more knowledge from each other. However, the interview method produced significantly more knowledge than either the protocol or repertory grid methods.
3. For the interpretation task, there is no significant difference between methods.

4.2.2 *Required total time to acquire knowledge*: The results indicate that there is a significant difference for different methods ($f(3,20) = 12.03$, $p = 0.0001 < 0.05$), tasks ($f(2,40) = 47.37$, $p = 0.0001 < 0.05$) and the interaction between method and task ($f(6,40) = 3.43$, $p = 0.0001$) in total time. The SNK test concludes the following

1. For the diagnosis task, the protocol (mean = 265.5, SD = 140.2) and interview (258.8, SD = 77.7) methods took significantly more time than the repertory grid (mean = 120.8, SD = 34.6) and induction (mean = 118.3, SD = 22.3) methods.
2. For the debugging task, the interview (mean = 309.7, SD = 52.4) method took significantly more time than the induction (mean = 136.7, SD = 25.7) and repertory grid (mean = 128.2, SD = 42.1) methods. The protocol (mean = 266.8, SD = 97.6) method took significantly more time than the repertory grid (mean = 128.2, SD = 42.1) method.
3. For the interpretation task, there is no significant difference between the methods.

The results show that the interview and protocol methods take a great deal of total time to elicit the knowledge from the expert. However, there is no significant difference in the elicitation time as shown in Table 3. This suggests that time in forming rules and/or reviewing rules is the possible cause of the time difference between the different knowledge elicitation methods.

4.2.3. *Inconsistency of knowledge*: The results indicate that there is a significant difference among different methods ($f(3,20) = 9.81$, $p = 0.0003 < 0.05$), task ($f(2,40) = 15.74$, $p = 0.0001 < 0.05$) and the interaction between method and task ($f(6,40) = 6.85$, $p = 0.0001$) in acquiring conflict rules. The SNK test concludes the following:

1. For the diagnosis task, the protocol (mean = 8.3, SD = 2.5) and interview (mean = 9.8, SD = 3.1) methods resulted in significantly more resultant rules in conflict than the other two methods (means = 6.0 & 2.3, SDs = 1.8 & 2.4).
2. For the debugging task, the protocol (mean = 9.8, SD = 4.0) method results in significantly more rules in conflict than the interview (mean = 5.3, SD = 2.7) and induction (mean = 6.2, SD = 2.6) methods.
3. For the interpretation task, there is no significant difference between the methods.

4.2.4. *Efficiency of knowledge elicitation*: The results indicate that there is a significant difference between different methods ($f(3,20) = 9.80$, $p = 0.003 < 0.05$), tasks ($f(2,40) = 48.98$, $p = 0.0001 < 0.05$) and the interaction of method and task ($f(6,40) = 4.64$, $p = 0.0011$) in efficiency of acquiring rules/time. The SNK test concludes the following:

1. For the diagnosis task, the induction (mean = 9.4, SD = 2.3) and repertory grid (mean = 9.0, SD = 2.1) methods are significantly more efficient than the interview (mean = 4.1, SD = 3.4) method.
2. For the debugging task, the induction (mean = 10.8, SD = 2.8) method is significantly more efficient than the interview (mean = 5.0, SD = 0.6) and protocol (mean = 4.1, SD = 1.3) methods.
3. For the interpretation task, the induction (mean = 18.3, SD = 4.8) and protocol (mean = 15.8, SD = 5.3) methods are significantly more efficient than the interview (mean = 9.2, SD = 1.7) method.

4.2.4. *Importance of data*: The results indicate that there is no significant difference in acquiring importance of data among methods ($f(3,20) = 1.11$, $p = 0.3667 > 0.05$), but there is a significant difference among tasks ($f(2,40) = 22.45$, $p = 0.0001 < 0.05$) and the interaction of method and task ($f(6,40) = 3.88$, $p = 0.0038$), separately. The SNK test concludes the following:

1. For the diagnosis task, all methods are not significantly different.
2. For the debugging task, the interview (mean = 85.6, SD = 7.9) method is significantly different from the protocol (mean = 60.0, SD = 22.2) and repertory grid (mean = 63.2, SD = 10.9) methods.
3. For the interpretation task, the methods are not significantly different

4.3. *Match index among task, factors of cognitive ability and method of knowledge elicitation*

The above results indicate that task type, factors of cognitive abilities and methods of knowledge elicitation have effects on the acquired knowledge. Therefore, a multiple regression equation including these three variables is used to predict their matching index. A three-stage process is used to determine which knowledge elicitation method is the most effective for certain tasks and certain individuals (Figure 2). Using the available data from the independent variables, stage one provides the regression equations for the following five dependent variables: index for completeness, index for time, index for inconsistency, index for importance of data and index for efficiency.

For example, the regression equation for determining the index of completeness can be expressed as follows:

$$R = a + b\ \text{Method}(i) + c\ \text{Task}(j) + d\ \text{Cognitive}(k) + e\ (\text{Method} * \text{Task}) \quad (1)$$

where a, b, c, d and e are the relative coefficients obtained by using the multiple regression skills with the available data. R is the dependent variable used as the criterion to be predicted in the regression. In this equation, the other factors included are: (1) Cognitive test scores derived from the 24 subjects; (2) Three tasks; and (3) The interaction between method and task. Because both knowledge elicitation methods and tasks are qualitative variables, the following dummy variables are introduced.

X1 = 1 if protocol method is used, 0 otherwise
X2 = 1 if interview method is used, 0 otherwise
X3 = 1 if induction method is used, 0 otherwise
X4 = 1 if diagnosis task is used, 0 otherwise
X5 = 1 if debugging task is used, 0 otherwise

In the case where an appropriate knowledge elicitation method must be selected for a new expert, then all the possible combinations of tasks, methods and cognitive scores for the new expert are used to determine the index. The higher the index, the more effective the combination of different knowledge elicitation methods and tasks.

The procedures to find the 'best' regression equation for each matching index are stated as following:

1. Identify the possible set of independent variables. The criteria for selecting the best independent variables are by C_p, MSE_p and R^2.
   a. C_p is based on the Total Mean Square Error (TMSE) for each of the regression equation, where

   $$TMSE = E\left\{\sum_{i=1}^{n}[y_i - E(y)_i)]^2\right\} \quad (2)$$

   The standard for selecting C_p for the best regression model is either a smaller C and /or a value of C_p near p+ 1.
   b. R_p^2 is the multiple coefficient of determination, and it is used to find the suitable regression model when more independent variables are added to the model, but the R^2 only increases a small value.
   c. MSE_p (or R_a^2) is used to account for the number of β parameters in the model. A model with the minimum MSE is preferred.
2. Form the hypothetical regression model
3. Determine the model coefficients
4. Check the distribution of the random error term
5. Check the utility of the model

In order to establish a compatible value for each criterion, all the dependent variables are transferred into 100% percentile values. This 100% percentile value will be used to specify the referenced relative value for each criterion value acquired in the process of knowledge elicitation.

- Index for completeness: The total percentile knowledge acquired is the criterion utilized in the regression to estimate the completeness of the elicited knowledge. From the primitive analysis of the regression, the variance of the error term can not satisfy the assumption of homoscedasticity. A LOG transformation of the dependent variable data is used to stabilize the variance. The higher the index is, the more complete.
- Index for total time: The total percentile time spent is the criterion utilized in the regression. The higher the index, the worse the result.
- Index for inconsistency: The sum of the percentile of conflict rules is the dependent variable utilized

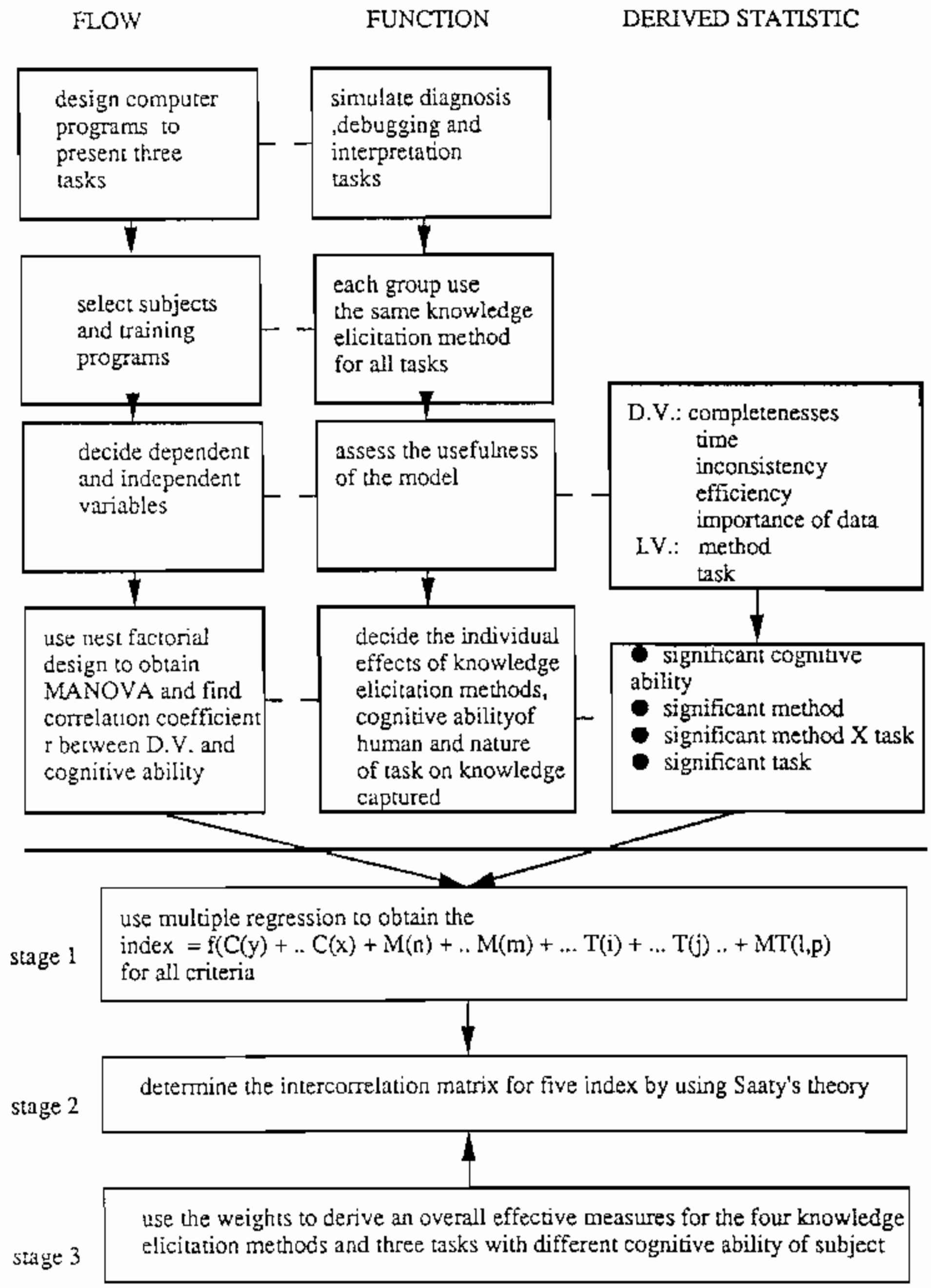

Figure 2 Flow chart to determine the matching index.

in the regression, which is calculated to estimate the inconsistency. The higher the index is, the less the consistency. From the primitive analysis of regression, the variance of the error term can not satisfy the assumption of homoscedasticity. A LOG transformation of the dependent variable data is used to stabilize the variance.

- Index for efficiency: The percentile rules acquired per unit time is the criterion utilized in the regression, calculated to estimate the efficiency of the model. From the primitive analysis of regression, the variance of the error term can not satisfy the assumption of homoscedasticity. A SQRT transformation of the dependent variable data is used to stabilize the variance. The higher the index is, the more the efficiency.

- Index for importance of data: The total percentile importance of data acquired is the criterion utilized in the regression to estimate the index. From the primitive analysis of regression, the variance of the error term can not satisfy the assumption of homoscedasticity. A SQRT transformation of the dependent variable data is used to stabilize the variance. The higher the index is, the greater the amount of important data acquired.

When these regression equations are obtained, the match index can be estimated. At stage two, an intercorrelation matrix is developed by using Saaty's analytical hierarchy process (AHP, Saaty 1980), and this method is used to calculate a weight for each of the five dependent variables. In this research, the relative weights are given by three super experts who have excellent experience in the FORTRAN language for at least 11 years. The basic algorithm for AHP is to solve the homogeneous linear equations $AW = \lambda W$, where vector A is a 5-by-5 matrix which represents the pairwise comparison of the five criteria, and W is a length 5 column vector representing the weight for the five criteria.

$$A = \begin{bmatrix} w1/w1 & w1/w2 & w1/w3 & w1/w4 & w1/w5 \\ w2/w1 & w2/w2 & w2/w3 & w2/w4 & w2/w5 \\ w3/w1 & w3/w2 & w3/w3 & w3/w4 & w3/w5 \\ w4/w1 & w4/w2 & w3/w3 & w4/w4 & w4/w5 \\ w5/w1 & w5/w2 & w5/w3 & w5/w5 & w5/w5 \end{bmatrix}$$

$$W = \begin{bmatrix} W1 \\ W2 \\ W3 \\ W4 \\ W5 \end{bmatrix}$$

Then by using the EIG function in MATLAB, the eigenvalues and eigenvectors can be generated. Based on this intercorrelation, the relative weight of each dependent variable in relation to the overall weights of the five variables is derived.

In stage three, the weights derived from the intercorrelation matrix are used to derive an overall effective measure for the four different knowledge elicitation methods and the three different tasks. This is done by multiplying the relative index Y_i derived from stage one with the weights obtained in the intercorrelation matrix. The index for time and inconsistency are negative due to their opposite effects in knowledge elicitation. The total index can be expressed below:

$$\text{Total Index} = [\text{Weight}] \times [\text{Task}] \times [\text{Each Index}]$$

$$\text{Where weight} \begin{bmatrix} 0.2474 & 0.3112 & 0.4167 \\ 0.2207 & 0.3218 & 0.2718 \\ 0.1449 & 0.0667 & 0.0544 \\ 0.2100 & 0.2366 & 0.1746 \\ 0.2402 & 0.0636 & 0.0877 \end{bmatrix}$$

$$\text{Task} = \begin{bmatrix} X4 \\ X5 \\ 1 - X4 - X5 \end{bmatrix}$$

Each index = $[Y_{com} - Y_{time} - Y_{incon}\ Y_{eff}\ Y_{imp}]$ (see Appendix A)

The largest summed value represents the best combination of task, method and cognitive ability found in this vector operation.

4.4. *Testing of hypotheses*

The three hypotheses have been tested by using a statistical experiment in the analysis task mentioned and the results are illustrated below.

- *Hypothesis one. Different knowledge elicitation methods do extract different amounts of procedural and declarative knowledge from the domain experts.*

This hypothesis is supported for the debugging task only. The interview and induction methods are better than the protocol and repertory grid analysis methods.

- *Hypothesis two. Each knowledge elicitation method is suited for a specific task or tasks.*

This hypothesis is supported. It indicates that each knowledge elicitation method is suitable for a different task under the requirements of different dependent variables. For diagnosis and debugging tasks, certain knowledge elicitation methods are better than others given elicitation methods. The criterion utilized effects the selected methods. For the interpretation tasks, only for the criterion of efficiency did the method of Interview have inferior results compared to the other methods

- *Hypothesis three. If the above hypotheses are supported, then the following function is hypothesized:*

  Match Index = f

  (*knowledge elicitation methods, factors of cognitive abilities, tasks*)

This hypothesis is supported since the interaction among tasks, knowledge elicitation methods and factors of cognitive abilities exists. This hypothesis has also been demonstrated successfully to form a total match index, which combines the importance of data, completeness, time, inconsistency and efficiency.

5. Discussion

The power of an expert system depends on the quality of the knowledge base (Harandi and Lange 1990). This paper provides a descriptive model as to how knowledge elicitation methods, the nature of the task and human factors of cognitive abilities are interrelated. This section discusses in three sequential parts the effect of knowledge elicitation methods on the diagnosis, debugging and interpretation tasks and the effects of individual factors of cognitive abilities on the effectiveness of the elicited knowledge.

5.1. *Effects of the knowledge elicitation methods on a diagnosis task*

The results of this research show that any method of knowledge elicitation can elicit only 40% of the whole knowledge in this domain (Table 3). Therefore, developing a new knowledge elicitation method, combining different knowledge elicitation methods used in the same task, or extracting knowledge from multiple experts is recommended (Chao and Salvendy 1994) if a more complete knowledge-base for building an expert system is expected. While the repertory grid and induction methods take longer to learn than do the interview or protocol methods, the latter two methods take longer to elicit the knowledge and generate less rules per unit time than the former two. Hart (1987) indicates that if a training set in the induction method is available, then induction is rapid. The above theory is consistent with the experimental outcomes.

The protocol, interview and induction methods extract more conflicting rules than does the repertory grid method It is possible that the repertory grid analysis method uses identified elements and constructs to describe the objects, and as such it limits the domain to a known content. This suggests that the protocol and interview methods generate more conflicting rules because experts propose facts and rules of knowledge and can not easily check them simultaneously when they are doing it.

Since the four methods of knowledge elicitation obtain the same amount of knowledge, the acquisition efficiency for each knowledge elicitation method depends on the reciprocal of the total time. The induction and repertory grid methods are more efficient in acquiring facts than the interview and protocol methods.

5.2. *Effects of the knowledge elicitation methods on a debugging task*

The interview or induction method acquires more procedural or declarative knowledge than the repertory grid or protocol method. They can derive only 50% of the whole knowledge; however, this is better than the 40% derived for the diagnosis task. It is due to the fact that the subject already knows where the malfunctions are in the debugging task, so they concentrate on these areas for correcting the problems. In order to obtain a more complete knowledge data base, developing a new and more effective knowledge elicitation method, combining different knowledge elicitation methods or extracting knowledge from multiple experts is recommended (Chao and Salvendy 1994). It suggests that the protocol method generate more conflicting rules then the interview and induction methods. Hart (1987) also points out that induction is consistent in the derived rules, which supports the results here. The induction method is more efficient in acquiring rules per unit time than the interview method. Both the interview and induction methods acquire more important data than the other two methods of knowledge elicitation.

5.3. *Effects of the knowledge elicitation methods on an interpretation task*

The maximum percentage of knowledge obtained using any of the knowledge elicitation methods tested is 27%, and this result is obtained by using the interview method. The reason for such low knowledge acquired may be due to the fact that this task has many reasons to account for the error message, hence the subjects locate only a few of the reasons. In the other two tasks, the subjects can cross-reference other material, and this increases the percentage of knowledge acquired. All methods have no significant difference in the total time. The induction method is more efficient in acquiring rules per unit time than the interview or repertory grid method. The interview method obtains more important data than the repertory grid method.

Because factors of cognitive abilities impact on the effectiveness of the elicited knowledge (Chao and Salvendy 1995), in order to maximize the elicited knowledge the expert whose knowledge is elicited should have high performance on the cognitive tests where the correlation coefficients are positive.

The discussions here indicate that a selection of a suitable knowledge elicitation method should be based on task type and human factors of cognitive abilities. The content of the knowledge base is more complete and more reliable when eliciting expertise from several

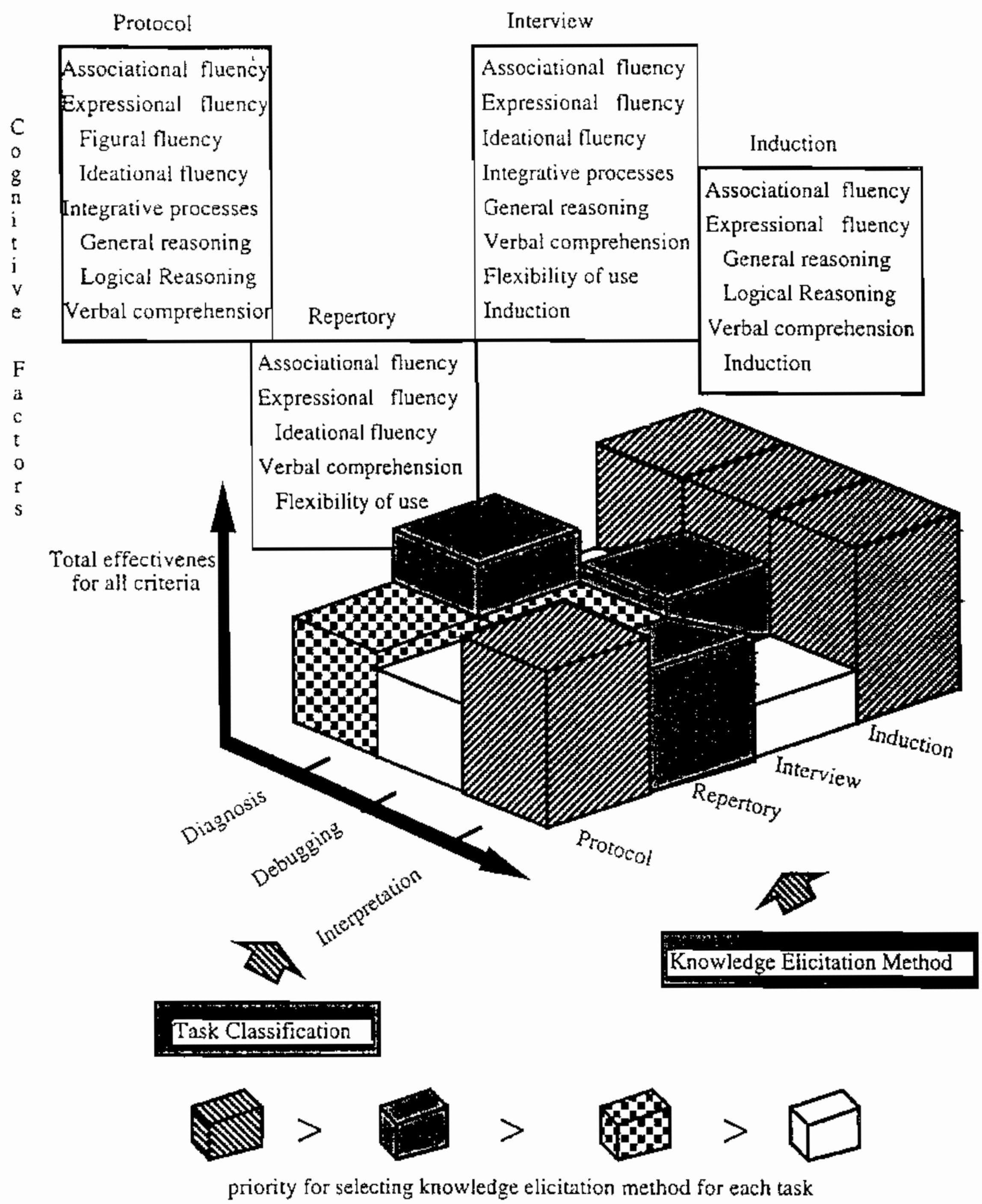

Figure 3. Conceptual model for selecting knowledge elicitation methods and the cognitive factors associated with the use of each method

experts. This study also describes how each method of knowledge elicitation captures the expert's knowledge completely, consistently and efficiently. Figure 3 clearly indicates which combination of task and knowledge elicitation method is best with certain factors of cognitive abilities of experts. For the three analysis type tasks used in this experiment (diagnosis, debugging and interpretation), the induction method showed overall superiority and required the use of different cognitive factors than those associated with the other three methods. In this study the impact of the elicitation methods and cognitive factors were tested only on the tasks considered as analysis type, which represents three out of the ten classes of task categories (interpretation, prediction, diagnosis, design, planning, monitoring, debugging, repair, instruction and control). Hence, based on the current study it is not known which elicitation methods would do the best job with the remaining synthesis tasks.

6. Conclusions and Implications

6.1. *Conclusions*

From the study presented here, the following conclusions may be drawn:

1. The repertory grid and induction methods of knowledge elicitation require significantly less time (by one- to six-fold) to elicit knowledge than the interview and protocol methods in the diagnosis or debugging task.
2. For eliciting knowledge in the debugging tasks, the induction or interview method elicits an average of 40% more knowledge than using either the protocol or the repertory grid method.
3. The induction method of knowledge elicitation acquires about 100% more declarative knowledge per hour than the protocol method in an interpretation task.

6.2. *Implications*

Knowledge elicitation is the most important procedure in building an expert system. The outcome of this research provides theoretical and practical implications for Human Computer Interaction (HCI) and training of knowledge engineers. Both of these issues are discussed below.

6.2.1. *HCI implications*: The proposed model provides a matching index among knowledge elicitation methods, tasks and expert's factors of cognitive abilities for five important criteria. This matching index results in a more precise and complete depiction of the mental model of the experts than otherwise is possible. This study shows that the interface designer can use the index to embed the mental model of an expert in the specific task into the HCI interface design.

Norman (1986) suggests that the design model conceptualization is based on the user's tasks, requirements and capabilities. Thus, the types of problem-solving processes and reasoning methods used by the experts determine the appropriate knowledge representation in the computer. The interface designer can design another knowledge elicitation method or combine different methods during the interactive knowledge acquisition processes for the different tasks and factors of cognitive abilities of the expert.

6.2.2. *Training of knowledge engineer*: The knowledge engineer has the important role of extracting the domain related knowledge from the expert, and then integrating that knowledge into a knowledge base for the expert system. One of the most difficult aspects of the knowledge engineer's task is to help the domain expert identify the domain and implication knowledge, and to structure and formalize the knowledge into some form of knowledge representations (Noelke 1988). An essential step of this task is to determine which knowledge elicitation technique should be used.

Once this conceptual model and methodological framework of knowledge elicitation is known, knowledge engineers can be trained efficiently and effectively in selecting the appropriate method of knowledge elicitation for each task.

Acknowledgements

The authors wish to thank CSIST in R.O.C. and the NEC professorship for support with this work.

References

BAINBRIDGE, L. 1979, Verbal reports as evidence of the process operator's knowledge, *International Journal of Man-Machine Studies*, **11**, 441 – 436.

BOOSE, J. H 1985, A knowledge acquisition program for expert systems based on personal construct psychology, *International Journal of Man-Machine-Studies*, **20**(1), 21 - 43.

BOOSE, J. H. 1986, Rapid acquisition and combination of knowledge from multiple experts in the same domain, *Future Computering Systems*, **1**, 191 - 216.

BOOSE, J. H. 1988, A research framework for knowledge acquisition techniques and tools, in J. Boose, B. Gaines and M. Linster (eds), *Proceedings of the European Knowledge Acquisition Workshop for Knowledge-Based Systems (June 19–23)* (Gesellschaft für Mathematik und Daterverarbeitung, Sankt Augustin), 10 – 1 – 10 – 3.

BYLANDER, T, and CHANDRASEKARAN , B. 1987, Generic tasks for knowledge based reasoning: the 'right 'level of abstraction for knowledge acquisition, *International Journal of Man-Machine Studies*, **26**, 231 – 243.

CHAO, C. J. and SALVENDY, G. 1994, Percentage of Procedural Knowledge Acquired as a Function of the Number of Experts from Whom Knowledge is Acquired for Diagnosis, Debugging and Interpretation Tasks, *International Journal of Human-Computer Interaction*, **6**(3), 221 – 233.

Chao, C. J. and Salvendy, G. 1995, Impact of Cognitive Abilities of Expert on the Effectiveness of Elicited Knowledge, *Behaviour & Information Technology*, **14**(7), 174 – 182.

Clancey, W. J. 1986, Heuristic classification, in J. S. Kowalik (ed.) *Knowledge Based Problem-solving*, (Englewood Cliffs, New Jersey: Prentice Hall), 1 67

Ekstrom, R. B., French, J. W., Harman, H. H., and Dermen, D. 1976, *Manual for Kit of Factor-Referenced Cognitive Tests* (Princeton : Educational Testing Services).

Ericsson, K A., and Simon, H. A. 1984, *Protocol Analysis: Verbal Reports as Data.*, (Cambridge, Massachussetts: MIT Press).

Fox, J., Myers, C. D., Greaves, M. F., and Pegram, S. 1987, A systematic study of knowledge base refinement in the diagnosis of leukemia, in A. Kidd (ed.), *Knowledge Acquisition for Expert Systems: A Practical Handbook*, (New York Plenum Press)

Gaines, B. R. and Shaw, M. L G. 1986, Knowledge engineering technique, *Proceedings of AUTOFACT'86 (November 12 – 14)*, Dearborn, Michigan: CASE/SME) 8 – 79 – 8 – 96.

Gammack, J. G. and Young, R. M. 1984, Psychological techniques for eliciting expert knowledge, in M. A. Bramer (ed.), *Research and development in expert systems*, (New York: Cambridge University Press), 105 – 112.

Gammack, J. G. 1987, Different techniques and different aspects on declarative knowledge, in A. Kidd (ed.) *Knowledge acquisition for Expert Systems: A Practical Handbook*, (New York: Plenum Press).

Garg, C. K. and Salvendy, G. 1988, A structure knowledge elicitation methodology for building expert systems, *International Journal of Man-Machine Studies*, **29**, 377 – 406.

Gentner, D. and Stevens, A. 1983, *Mental Models*, (Hillsdale, New Jersey: Lawrence Erlbaum).

Harandi, M. T. and Lange, R. 1990, Model-based Knowledge Acquisition, in H. Adeli (ed.) *Knowledge Engineering*, (New York McGraw-Hill), 103 - 129.

Hart, A 1985, The role of induction in knowledge elicitation. *Expert System*, **2**(1), 24 28

Hart, A. 1987, Role of induction in knowledge elicitation, in A. Kidd (ed.) *Knowledge acquisition for Expert Systems: A Practical Handbook*, (New York: Plenum Press).

Hayes-Roth, F., Waterman, D. A. and Lenat, D. B. 1983, *Building Expert Systems*, (Reading, MA: Addison-Wesley).

Hicks, C. R. 1973, *Fundamental Concepts of Design of Experiment*, 2nd edn, (New York: Holt, Rinehart and Winston)

Hoffman, R. R. 1987, The problem of extracting the knowledge of experts from the perspective of experimental psychology, *AI Magazine*, **8**(2), 53 – 64.

Hunt, V. D. 1986, *Artificial Intelligence and Expert Systems Sourcebook*, (New York: Chapman and Hall).

Johnson, P. E., Zualkernan, I. and Garber, S. 1987, Specification of expertise, *International Journal of Man-Machine Studies*, **26**, 161 – 182.

Kelly, G. A. 1955, *The psychology of personal constructs*, (New York: W.W. Norton).

Kuipers, B. and Kassirer, J. 1987, Knowledge acquisition analysis of verbation protocols, in A. Kidd (ed.) *Knowledge acquisition for Expert Systems: A Practical Handbook*, (New York: Plenum Press).

Lehner, P. E. and Kralj, M. M. 1988, Expert systems: The user interface, in J A. Hendler (ed.) *Human/Computer Interaction*, 307 318.

Lehner, P. E. and Zirk, D A. 1987, Cognitive factors in user/expert system interaction, *Human Factors*, **29**(4), 97 – 109.

Lehto, M. R., Boose, J, Sharit, J. and Salvendy, G. 1992, Knowledge Acquisition, in G. Salvendy (ed). *Handbook of Industrial Engineering*, 2nd edn, (New York: John Wiley & Sons), 1495 - 1545.

Marcus, S., McDermott J. and Wang, T. 1985, Knowledge acquisition for constructive systems, in A. Joshi (ed.), *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (August 18 – 23)* (Los Angeles IJCAI), 637 – 639.

Mendenhall, W. and Sincich, T. 1988, A second course in business statistics: regression analysis, 3rd edn, (San Francisco: Dellen Publishing).

Merlevede P. and Vanthienen J. 1991, A Structured Approach to Formalization and Validation of Knowledge, in J. Feinstein, E. Awad, L. Medsker and E. Turban (eds), *Proceedings of the 1991 IEEE/ACM Conference on Developing and Managing Expert Systems Programs (September 30 – October 2)*, Los Alamitos, California: IEEE Computer Society Press,), 149 – 158.

Nagao, M 1990, *Knowledge and inference*, (London: Academic Press).

Neale, I. 1988, First generation expert systems: A review of knowledge acquisition methodologies, *The Knowledge Engineering Review*, **3**(1), 105 – 145.

Noelke, U. 1988, The essence of knowledge engineering, in S. Savory (ed.) *Artificial intelligence and expert systems*, (New York: John Wiley & Sons), 116 – 132.

Norman, D. A. 1986, Cognitive engineering, in D. A. Norman and S. W. Draper (eds) *User centered system design: new perspectives on human-computer interaction*, (Hillsdale, New Jersey: Laurence Erlbaum Associates), 31 – 61.

Olson, J. R. and Reuter, H. H. 1987, Extracting expertise from experts: Methods for knowledge acquisition, *Expert System*, **4**(3), 152 168.

Parsaye, K. and Chignell, M. 1988, *Expert systems for experts*, (New York: John Wiley & Sons).

Quinlan, R. 1993, C4. 5: Program for Machine Learning, (San Mateo, CA: Morgan Kaufmann).

Saaty, T. L. 1980, *The analytic hierarchy process: planning, priority setting, resource allocation*, (New York: McGraw-Hill).

SAS Institute Inc. 1989, *SAS Procedures Guide*, Release 6.03 Edition, (Cary, North Carolina, SAS Institute, Inc.).

Sein, M. K. and Bostrom, R. P. 1989, Individual differences and conceptual models in training novice users, *Human-Computer Interaction*, 4, 197 - 229.

Shaw, M. L. G. and Gaines, B. R. 1987, An interactive knowledge-elicitation technique using personal construct technology, in A. Kidd (ed.) *Knowledge acquisition for expert systems : a practical handbook*, (New York: Plenum Press), 109 – 135.

Tuthill, G. S. 1990, *Knowledge engineering: concepts and practice for knowledge-based systems*, (Blue Ridge Summit, Pennsylvania: Tab Books).

Waterman, D. A. and Hayes-Roth, F. 1983, An investigation of tools for building expert systems, in F. Hayes-Roth, D. A. Waterman and D. B. Lenat (eds) *Building Expert Systems*, (Reading, MA: Addison-Wesley), 169 – 215.

Appendix A. Parameter estimates for each index

Y*com = LOG (Ycom) = −1.01 + 4.43X1 + 5.39X2 + 4.31X3 + 0.52X4 + 14.47X5 + 0.03C3 − 0.02C4 − 0.03C6 + 0.05C7 − 10.96X1*X5 − 1.19X2*X4 − 14.55X2*X5 − 13.59X3*X5 + 0.05C2*X5*(1 − X2 − X3 − X1) − 0.48X1*X5*C4 + 0.05*X5*C9*(1 − X1 −X2 −X3) + 0.54C2*(1 − X4 − X5)*(1 − X1 −X2 −X3)

Ytime = 47.98 − 21.09X1 + 29.69 X2 − 21.16X3 − 91.98X4 + 224.27X5 + 4.52C1 + 2.06C3 − 1.92C4 + 0.63C8 − 3.95C9 + 114.98X1*X4 − 206.94X1*X5 + 86.98X2*X4 − 224.94X2*X5 + 93.65X3*X4 − 221.6X3*X5 − 221.6X3*X5 + 9.23X4*C2*(1 − X1 − X2 − X3) − 32.48C9*X5*(1 − X1 −X2 − X3)

Y*incon = LOG(Yincon) = 3.32 + 0.64X1 + 0.89X2 − 0.93X4 − 0.33X5 + 0.08C3 − 0.06C4 − 0.14C5 − 0.06C6 + 0.06C7 + 0.05C8 + 0.08C10 + 1.02X1*X4 + 1.12X2*X4 − 0.29X2*X5 + 1.34X3*X4 + 0.8X3*X5 + 0.04 X1*X5*C4

Y*eff = SQRT(Yeff) = 6.63 + 2.62X1 − 2.18X2 + 1.54X3 + 1.45X4 + 1.04X5 − 0.4C1 − 0.16C3 + 0.15C4 + 0.07C5 − 0.06C6 + 0.09C7 − 0.05C8 + 0.27C9 − 1.39X1*X5 − 1.1X2*X4 − 1.02X3*X4 − 0.24X1*X4*C4 − 0.18X1*X5*C4

Y*imp = SQRT(Yimp) = 0.22 + 7.49X2 + 3.78X3 − 2.18X4 + 26.85X5 + 0.31C2 − 0.06C4 + 0.11C7 − 0.35C9 + 0.07C10 + 8.39X1*X4 − 11.37X1*X5 − 27X2*X5 + 3.93X3*X4 − 24.35X3*X5 + 0.99X4*C2*(1 − X1 − X2 − X3) − 1.46X1*X5*C2 − 2.95X5*C9*(1 − X1 −X2 −X3) + 0.59*X1*C1*(1 − X4 − X5) + 0.65*C2*(1 − X1 −X2 −X3)*(1 − X4 − X4)

Total Matching Index = a Y*com + b Ytime + cY*incon + dY*eff + eY*imp

where X1–X5 as shown in the context

- C1: associational fluency
- C2: expressional fluency
- C3: figural fluency
- C4: ideational fluency
- C5: integrative processes
- C6: general reasoning
- C7: logical reasoning
- C8: verbal reasoning
- C9: flexibility of use
- C10: induction

a, b, c, d and e are the weights for each individual index