Int. J. General Systems, Vol. 28(2-3), pp. 227-241
Reprints available directly from the publisher
Photocopying permitted by license only

© 1999 OPA (Overseas Publishers Association) N.V.
Published by license under
the Gordon and Breach Science
Publishers imprint.
Printed in Malaysia.

A FUZZY REASONING APPROACH TO SUPPORT KNOWLEDGE ACQUISITION

CHAOCHANG CHIU[a,*], PEI-LUN HSU[b] and A.F. NORCIO[c]

[a]*Department of Information Management*, [b]*Department of Computer Engineering, Yuan Ze University, Chungli, Taiwan 320, R.O.C.;*
[c]*Department of Information Systems, University of Maryland Baltimore County, Baltimore, MD 21228, USA*

(Received in final form 20 December 1996)

Acquisition of a user's computer based problem-solving skill is an important research area in human–computer interaction. This type of information usually can be derived by careful inspection on the actual dialog behavior. To implement such a knowledge reasoning agent, several crucial issues are pointed out and carefully inspected. These include: (a) appropirtate knowledge representation schema that are able to demonstrate the causal relationship between pairs of dialog events; (b) the formulation of a valid formula in calculating the overall knowledge index from the background information; (c) determination of the minimum sufficient number of dialog events required to form a discernible pattern; and (d) generalization of categories of performance patterns that can be applied to all types of application domains. A prototype reasoning agent based on the proposed methodology is constructed and its effectiveness is verified with the dialog events during UNIX operations.

Keywords: Fuzzy cognitive map; human–computer interaction; knowledge acquisition; causal modeling

1. INTRODUCTION

Inferring a user's problem-solving skill for a given application domain is an important research area of knowledge discovery to support more effective human–computer interaction. This is because only when each user's performance can be more accurately captured, can the system better adapt an adequate subsequent dialog.

* Corresponding author.

In most cases, reasoning about the diversity of dialog interaction, while including the user's characteristics, dialog history, and the context of user's navigation, is a very complicated process. Developing a generic reasoning scheme to detect a user's knowledge level seems to be rather difficult due to the complexity and uncertainty of different tasks features. To capture a user's problem-solving skill, a methodology should help discern some dialog behaviors such as (a) casual errors due to careless behavior (*e.g.* typographical mistakes); (b) trials and errors; (c) command testing; (d) inadequate dialog event sequence; and (e) misplacement of variables, since these types of behavior are rather difficult to be discerned if the implicit knowledge or skills has been actually mastered by users.

In dealing with these situations, a reasoning agent should not only be able to differentiate normal dialog events from the abnormal ones but also be able to tolerate certain types of mistakes that are not exactly exhibiting lack of competence (Chiu *et al.*, 1994). In other words, to derive a more accurate understanding about a user's problem-solving knowledge, an agent should be able to filter the noise information and obtain the most critical information leading to valid judgment. There are many research works aiming to evaluate a user's problem-solving performance based upon a single dialog event. Most of them employ the evidential reasoning techniques that mostly are based on the use of Bayesian probability theory. Generally these approaches focus on the analysis of a single dialog event and often overlook the global evaluation of the entire set of consecutive dialog events. The primary objective of this research is to provide an innovative methodology for analyzing the observed user's dialog behavior and to evaluate the user's overall performance with consecutive dialog events.

2. THE KNOWLEDGE REASONING APPROACH

According to Miller's classic work on the magical number 7 ± 2, it is reasonable to assume that a user organizes the problem-solving procedures with five to nine steps (Miller, 1965). Likewise, Hamon and King suggest that a person can maintain from four to seven chunks of knowledge in short-term memory during problem-solving (Harmon and King, 1985). Therefore, an approach that can infer more information based upon existing known events seems valuable in gaining more insight to a user's problem-solving processes.

Causal modeling techniques have been used to infer more information in various applications. Among these approaches the Bayesian conditional probability approach is the most common way to propagate the unknown information (Desmarais and Pavel, 1987; Martin and Vanlehn, 1993; Sime, 1993). For example, Desmarais and Pavel employ belief net (Pearl, 1986a; 1986b) to infer how users may know about other UNIX commands and a summed up value of knowledge index is assigned based on a command entered (Desmarais and Pavel, 1987). However, there is no further discussion of how to determine which index should be adopted when the inferred indexes are of high variation. According to Minasi, this approach falters because it assumes all attributes are independent (Minasi, 1990). Furthermore, the belief net propagation techniques based on the Bayesian technique are of exponential complexity and are not practical for complex software environments (Ng and Abramson, 1990). Computational efficiency is one of the most crucial concerns in implementing interactive human–computer systems, especially in the dynamic background reasoning settings.

On the other hand, though inferring the unknown information via causal modeling may lead to implicational information (*e.g.*, background knowledge), the potential conflicts resulting from different sources of evidence seem to be another issue that needs to be resolved. This is because every dialog event emerged may not ascertain the mastery of knowledge required for this event occurrence. Uncertainty about a user's dialog event may also be caused by the ambiguity of irrelevant sequences or by a misuse of commands. For example, it is not explicit for a reasoning agent to determine whether a wrong command is resulted from typographical mistakes or by guesses without the aid of other supporting information. These phenomena can make the agent's reasoning about the interactive performance more difficult.

Clearly, the dialog sequence is another important piece of evidence that can be considered along with other derived information. Instead of using the Bayesian probability approach, this study employs the fuzzy cognitive map (FCM) technique (Kosko, 1986) to continually infer both the dialog sequence correlationship and the implicit background information based upon every single dialog event. A FCM is very similar to a belief network (Pearl, 1986a); both of these techniques are classified as evidential reasoning schema that are used to extend knowledge by known evidence. Further discussion of the fuzzy mathematics operations can be found in (Kosko, 1986; 1992).

3. METHODOLOGY

A FCM can be represented as an $n \times n$ matrix (say M, shown in Table I) and be further processed by fuzzy mathematics operations to produce another $n \times n$ matrix (say M', shown in Table II). Due to the FCM's computational propagation effect, M is transformed (attributed to iterative activation by each identity vector) into M' that is a new form of another association matrix. This new matrix implies a synergy effects via all existing direct and indirect influences from all other matrix entities (i.e., node i, i from 1 to n). The third column in

TABLE I The vector representation of original association matrix M

Vector Index	*Original association matrix vector*	*Attributes of association vector*
1	A(1)	$(a_{11}, a_{12}, a_{13}, \ldots, a_{1n})$
2	A(2)	$(a_{21}, a_{22}, a_{23}, \ldots, a_{2n})$
3	A(3)	$(a_{31}, a_{32}, a_{33}, \ldots, a_{3n})$
$\vdots$	$\vdots$	$\vdots$
n	A(n)	$(a_{n1}, a_{n2}, a_{n3}, \ldots, a_{nn})$

$\forall a_{ij} \in (-1, 1)$

TABLE II The vector representation of propagated implicit background knowledge matrix M'

Activation Node (i.e., the index of identity vector)	*Propagated implicit background knowledge vector*	*Attributes of knowledge vector*
1	B(1)	$(b_{11}, b_{12}, b_{13}, \ldots, b_{1n})$
2	B(2)	$(b_{21}, b_{22}, b_{23}, \ldots, b_{2n})$
3	B(3)	$(b_{31}, b_{32}, b_{33}, \ldots, b_{3n})$
$\vdots$	$\vdots$	$\vdots$
n	B(n)	$(b_{n1}, b_{n2}, b_{n3}, \ldots, b_{nn})$

$\forall b_{ij} \in (-1, 0, \text{or } 1)$

Table II stores the output propagated knowledge vectors derived through iterative propagation with the product of the identity vector with the matrix M. The a_{ij}'s in Table I are values of real number in between −1 and 1. The value of each a_{ij}'s indicates the associationship from node *i* to node j. All the attributes b_{ij}'s within the propagated implicit background knowledge vector B(i) are either −1, 0, or, 1, which implies a collectively resulted negative association, an unrelated association, or a positive association, respectively.

Example Consider the association matrix as shown in Table III. If a user enters command 4 (*pack*) then

$$\text{Let } N_1 = (0,0,0,1,0,0,0,0)$$
$$\text{then } N_1 \otimes M = (0,0.9,0.9,0,-1,-1,0.9,0) = T$$

Let the threshold operation (H) apply to the transition vector (T) with threshold values, 0.5 and −0.5. Then the first iteration is obtained:

$$N_2 = HT = (0,1,1,1,-1,-1,1,0)$$

Subsequent iterations yield:

$$N_2 \otimes M = (1.8,1.4,1.8,-0.1,-01.9,-1,0,1.8) = T;$$
$$N_3 = HT = (1,1,1,1,-1,-1,0,1);$$
$$N_3 \otimes M = (2.7,1.9,1.8,-0.1,1.9,\ -1.9,0.9) = T$$
$$N_4 = HT = (1,1,1,1,-1,-1,1,1);$$
$$N_4 \otimes M = (2.7,1.9,1.8,-0.1,1.9,-1,.9,2.7);$$
$$N_5 = HT = (1,1,1,1,-1,-1,1,1);$$

TABLE III Association matrix for UNIX file security commands

No.	1	2	3	4	5	6	7	8
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.9
2	0.9	0.0	0.9	−1	0.0	0.9	0.0	0.9
3	0.9	0.5	0.0	0.9	0.0	0.0	0.0	0.0
4	0.0	0.9	0.9	0.0	−1	−1	0.9	0.0
5	0.0	0.0	0.0	0.0	0.0	0.9	0.0	0.0
6	0.0	0.0	0.0	0.0	0.9	0.0	0.9	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.9
8	0.9	0.5	0.0	0.0	0.0	0.0	0.9	0.0

1—cat, 2—crypt, 3—unpack, 4—pack, 5—ed, 6—vi, 7—s, 8—m.

Finally, the iteration process stops due to the result $N_4 = N_5$. Thus, $N_5 = (1, 1, 1, 1, -1, -1, 1, 1)$ is the solution.

Therefore, the resulted background knowledge vector is $(1, 1, 1, 1, -1, -1, 1, 1)$ which indicates that normally a user knows what to adopt subsequent node except nodes 5 and 6. Thus, every time a user inputs a new command, a corresponding entry in the resulting B(*i*) is extracted.

The Knowledge Refinement Process

Two types of information—an association vector itself B(*i*) and the individual attribute b_{ij} in Table II are crucial to the identification of irregular dialog events as well as the computation of a user's knowledge index.

In Fig. 1, for example, by referencing the propagated background knowledge vector, the association index b_{ij} of the second command is equal to 1 when the j'th command (*i.e.*, dialog event) is followed by the k'th command. In the same way, the association index b_{jk} of the third command is equal to -1 when the j'th command is followed by the k'th command.

To compute the knowledge index CK_j, a general way of encoding the formula can be expressed with the Eq. (1) as follows:

$$CK_j = \frac{\sum_{i=1}^{n} f_i \times w_i}{\sum_{i=1}^{n} f_i}, \tag{1}$$

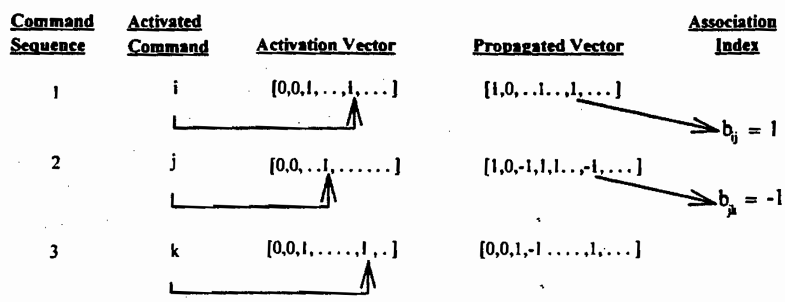

FIGURE 1 The Derivation of association index—b_{ij} and b_{jk}.

where $w_j \in W = (w_1, w_2, w_3, \ldots, w_n)$; w_j indicates the degree of the level of expertise of the mastery of the j'th concept node B(j); (a dialog event is treated as concept node here); $f_j = 0$ if $b_{ij} \leq 0$ or $f_j = 1$ if $b_{ij} > 0$; and if $\sum_{i=n-s+1}^{p} f_i$ then $CK_j = 0$.

However, only evaluating the current knowledge index (*i.e.*, CK_j) is prone to mislead the judgement of a user's actual knowledge level especially when there are various possibilities of dialog scenarios. For instance, inappropriate dialog behavior could be due to trial and error, typing errors, commands that are violating system process regulations, or other irregular events.

Generally, these situations are difficult to be identified and to be analyzed for determining a user's actual knowledge level with only a single dialog event (Chiu *et al.*, 1994). Consequently, a further refinement procedure is needed to resolve this problem when the irregular events occur. The following formula Eq. (2) is developed for determining the average knowledge index, which can eliminate certain intrusive effects resulting from careless or inappropriate dialog behavior using the referential information of $b_{ij'}$'s (*i.e.*, unrelated or negative related events). As shown in Eq. (2), the f_i is a control flag used especially for filtering irrelevant or negatively related events. That is, through this screening process the noises are excluded from the overall knowledge evaluation.

$$AK_p = \frac{\sum_{i=p-s+1}^{p} f_i \times CK_i}{\sum_{i=p-s+1}^{p} f_i} \tag{2}$$

where s can be a pre-defined value that indicates the number of consecutive dialog events; and p is integer and $s-2 < p < n+1$; $f_i = 0$ if $b_{ij} < 0$ and $j <> n$; or $f_i = 1$ if $b_{ij} \geq 0$ or $j = n$; if $\sum_{i=p-s+1}^{p} f_i = 0$ then AK_p isassigned with a default minimum value; this is because all the dialog event sequences indicate overall irrelevance.

4. IMPLEMENTATION RESULTS

This research conducts an experiment to simulate how a user's knowledge indek is computed and the noise information is eliminated with the aid of the association indexes. To evaluate the effectiveness of this methodology, the degrees of fluctuation for both CK_i and AK_i are continually collected and evaluated. To focus on analyzing the system trace, some factors such as current time, the length of time the simulation is run, and the time of arrival of the next commands issued are not considered in this experiment. The UNIX file security commands are used as subjects that are treated as dialog events. The contents of the selected subjects are listed in Appendix. This simulation encompasses all possible combination of dialog events that can be considered as equivalent to the real-world dialog setting.

Three different categories of sample sizes (8, 16, and 24 commands) are selected from the entire population of 26 UNIX commands. Each sample size construct has a session by group (15×3) design. Every construct has the same groups: CK and AK. Each group is divided into *Standard Deviation* (STD) and *Coefficient of Variation* (CF) dichotomy with fifteen sessions; so that a total of 120, 240. and 360 command subjects are randomly generated for each sample size.

As a prototype implementation, the number of consecutive dialog events 5 is tentatively defined as five. That is the five commands are continuously analyzed in sequence. As discussed above, five-steps is a reasonable number for general problem-solving activities. The command sequence can be viewed as a reflection of the user's cognitive organization for accomplishing the task. The outcomes of 8 items with 15 sessions, 16 items with 15 sessions, and 24 items with 15 sessions are summarized in Table IV. At the bottom of each category, total amount and average for attributes STD and CF are listed. In the first block for 8 items, it shows that the average of STD, 0.266 for AK is less than 1.475 for CK and the average of CF, 0.047 for AK is less than 0.278 for CK. In the second block for 16 items, the average of STD, 0.592 for AK is less than 1.640 for CK and the average of CF, 0.117 for AK is less than 0.324 for CK. In the third block for 24 items, it shows that the average of STD, 0.660 for AK is less than 1.545 for CK and the average of CF, 0.123 for AK is less than 0.283 for CK. As shown in Fig. 2, AK still maintains the minimum STD of 0.506 than 1.553 for CK; the minimum of CF of 0.091 than 0.295 for CK.

TABLE IV The individual simulation results

Subjects		*CK*		*AK*	
		STD	*CF*	*STD*	*CF*
8 Items	Total	22.130	4.177	3.983	0.702
	Average	1.475	0.278	0.266	0.047
16 Items	Total	24.597	4.250	8.885	1.749
	Average	1.640	0.324	0.592	0.117
24 Items	Total	23.179	4.250	9.902	1.845
	Average	1.545	0.283	0.660	0.123

(STD: Standard Deviation, CF: Coefficient of Variation).

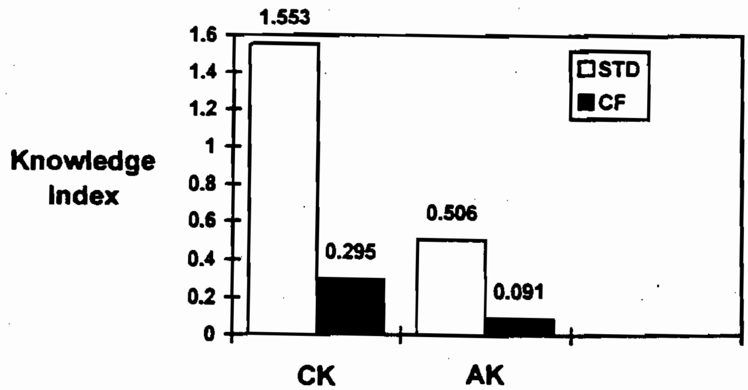

FIGURE 2 Graphical chart for the overall statistics comparison.

5. DISCUSSIONS

Based on the simulation experiment of knowledge evaluation for the 45 sessions of total 720 commands, the statistical analysis provides evidences that support the use of fuzzy based approach for identifying a user's knowledge level during dynamic human–computer interaction.

The results indicate that AKs outperform CKs and are effective in maintaining the minimum variation of data fluctuation. These imply the proposed reasoning mechanism is able to tolerate and handle unusual dialog events (*i.e.* outlying data); therefore, the inferred outcomes of a user's knowledge level may closer to the judgment of human observers who make decisions with global and approximate information rather than with a discrete and single dialog event.

Yet, evaluating a user's knowledge level only based on knowledge index may seems not sufficient. There are several possibilities that the use of a set of consecutive commands may be coincidentally assigned with high knowledge indices while the overall behavior cannot be justified in terms of task accomplishment. This could also be resulted from a rambling style of interaction. To minimize these potential anomalies occurred, further study may need to explore other ways to fine-tune a user's knowledge level by considering more information such as the dialog performance pattern or timing factor as well as the knowledge index (*i.e.*, AK).

System overhead, validation, and maintenance are some of the major problems occurred to a large rule-based system especially in a real-time process control environment. A truly adaptive interface system is not only required to detect a user's needs of help, but also to determine how and when to assist a user and activate a help solution. Monitoring a user's dialog behavior requires intensive system computation due to the large amount of input data needed to be processed. This research employs a fuzzy based and a simplified arithmetic operation to accomplish the task in determining a user's performance pattern based on the event logs. This study doesn't claim that the proposed method is the optimum solution to effectively accomplish this type of reasoning. For instance, among these crucial factors affecting the system overhead, the minimum steps (*i.e.*, the consecutive dialog events) that are required to determine the AK value needs to be carefully determined in order to balance the system overhead as well as system effectiveness.

Another issue such as the technique in determining the accurate FCM is also critical to this research. As noted by Kosko that the links (or edges) between concepts in a FCM can be assigned intuitively by domain experts or allow causal adaptation law to infer causal links from real data (Kosko, 1986). This adaptation law is so called differential Hebbian learning law expressed as (Kosko, 1986; Kosko, 1992)

$$\dot{m}_{ij} = -m_{ij} + S_i S_j + S_i S_j$$

or with its simpler formula

$$\dot{m}_{ij} = -m_{ij} + S_i S_j$$

where the overdot denotes time differentiation and m_{ij} denotes the synaptic efficacy of the synapse along the directed axonal links from the *i*th neuron in the input neuron filed to the jth neuron. According to Kosko, the time derivatives measure changes and the product of derivatives correlate changes. Thus, when the signal node S_i is interpreted as command node C_i, the so called concomitant-variation law that corresponds to the simpler form of Hebbian learning method leads to (Kosko, 1992):

$$\dot{e}_{ij} = -e_{ij} + \dot{C}_i\dot{C}_j,$$

where $-e_{ij}$ becomes zero causality between unchanged commands nodes. The concomitant-variation term $\dot{C}_i\dot{C}_j$ indicates causal increase or decrease according to joint command movement. If C_i and C_j both increase or decrease, the product of derivative is positive; if both of them hold opposite sign, the product of derivative is negative.

In this study, the FCM proposed is obtained through referencing technical user menus. Yet this can be alternatively done by self-organizing approach that collects the real data from user's actual dialog history. The newly obtained FCM can be compared with the one already derived through technical menus.

6. CONCLUSION AND FUTURE DEVELOPMENT

This paper proposes a method in acquiring a user's problem-solving skill during human–computer interaction. Based on this experiment, the statistics analysis supports the effectiveness of the proposed method in two aspects: (a) a user's background knowledge can be explicitly expressed; and (b) the unusual dialog events (*i.e.*, the noise) misleading the agent's judgment can be detected and eliminated; by which the agent is able to maintain the minimum variation of skill evaluation.

Human's perception of performance is generally based upon various perspectives of information rather than only the reliance on a single event or partial facts. Though, this method has been demonstrated its usefulness in determining the knowledge index or background knowledge information, more supportive information such as performance pattern seems to help better understanding about the user's overall interactive performance within complex problem-solving environments. Performance patterns can be independent of application domain and is well suited as a guidance to aid evaluating a user's actual performance. Constructing a user's performance model resembles human decision making with its ability to synthesize from approximate information sources and find precise solution. Future research would consider to explore other techniques that can make use of the descriptive strength to determine a user's domain knowledge index as well as the overall performance. According to Bezdek, the major types of uncertainty such as inaccuracy, randomness, and vagueness, are inherent to man–machine systems (Bezdek, 1981). Watanabe suggests that fuzzy logic provides a mathematical appearance of the idea that in human cognition the concepts have only vaguely defined extensions (Watanabe, 1985). It has also been suggested that fuzzy logic, with its natural language features of reasoning, is well suited for modeling the highly sophisticated performance of human cognitive processes (Zimmermann, 1991). Hence, using a fuzzy linguistic description for judging a user's problem-solving performance, based on the interaction behavior, seems to be more congruent to the human reasoning style in nature.

On the other hand, to dynamically process the collected dialog information, the embedded background reasoning agents need to be computationally efficient to alleviate the entire system overhead. Traditional pattern-matching approaches such as production expert systems tend to incur high system overhead when the reasoning components have complicated inferencing processes. Furthermore, maintainability is another major problem with knowledge-based systems that contain a large number of inferential rules (Caudill, 1990). Therefore, selecting appropriate techniques becomes one of the most important tasks for the success of implementing a feasible reasoning agent in this type of research domains.

Acknowledgements

This research was supported by National Science Council, Taiwan, Republic of China, under the contract number NSC84-2511-S-155-001CL.

References

Bezdek, J. (1981) *Pattern Recognition with Fuzzy Objective Function Algorithms* (Pergamon, NY).

Caudill, M. (1990) Expert Networks. In *Neural Network PC Tools: A Practical Guide.* R.C. Eberhart and R.C. Dobbins (Academic Press) pp. 189–213.

Chiu, C., Norcio, A.F. and Hsu, C. (1994) "Reasoning on domain knowledge level in human–computer interaction." *Information Sciences (Applications): An International Journal*, **1**, 31–46.

Desmarais, M.C. and Pavel, M. (1987) User Knowledge Evaluation: An Experiment with UNIX. *Human–Computer Interaction–INTERACT'87*, H.J. Bullinger and B. Shackel (eds) (Elsevier Science Pub., North-Holland), pp. 151–156.

Harmon, P. and King, D. (1985) *Expert Systems: Artificial Intelligence in Business.* (John Wiley, NY.)

Kosko, B. (1986) "Fuzzy cognitive maps," *International Journal of Man-Machine Studies*, January, 65–75.

Kosko, B. (1992) Neural Networks and Fuzzy Systems: *A Dynamic System Approach to Machine Intelligence* (Prentice Hall Pub., New Jersey).

Martin, J.D. and Vanlehn, K. (1993) "OLAE: Progress toward a multi-activity, Bayesian student modeler," *The Proceedings of AI-ED*, **93**, 410–417.

Minasi, M. (1990) "Exhibiting uncertainty," *AI EXPERT*, June, 13–15.

Miller, G.A. (1956) "The magical number seven, plus or minus two: some limits on our capacity for processing information," *Psychological Review*, **63**(2), 81–96.

Ng, K.C. and Abramson, B. (1990) "Uncertainty management in expert systems," *IEEE Expert*, April, 29–46.

Pearl, J. (1986a) "Fusion, propagation, and structuring in belief networks," *AI*, **29**, 241–288.

Pearl, J. (1986b) "On evidential reasoning in a hierarchy of hypothesis," *AI*, **28**, 9–15.

Sime, J. (1993) "Modeling a learner's multiple models with Bayesian belief networks," *The Proceedings of AiED'93*, 426–433.

Watanabe, S. (1985) *Pattern Recognition: Human and Mechanical*, (John Wiley & Sons, Inc., New York).

ZimmerMann, H.J. (1991) "Cognitive sciences, decision technology, and fuzzy sets," *Information Sciences*, **57**, 287–295.

APPENDIX

The subsets of UNIX file manipulation commands

Index	*Command*	*Level*	*Description*
A	cat	1	concatenate and print files
B	cd	1	change working directory
C	chgrp	3	change group of a file or directory
D	chmod	2	change mode of a file or directory
E	chown	3	change owner of a file or directory
F	cp	1	copy files
G	crypt	3	encrypt/decrypt files
H	cut	3	cut out selected columns from each line of a file
I	echo	2	print argument
J	ed	2	text editor
K	fsck	3	check the consistency of file systems and directories
L	grep	3	search a file for a pattern
M	id	2	print user and group id's and names
N	ls	1	list contents of directories
O	makekey	3	generate encryption key
P	mkdir	1	make directory
Q	mv	1	move files
R	pack	3	compress files
S	pwd	2	working directory name
T	rm	1	removes files
U	sort	2	sort and/or merge files
V	su	3	become super-user or another user
W	unpack	3	uncompressed files
X	vi	2	screen editor
Y	TypeError	*	spelling error of UNIX-based commands
Z	UnRecognized	**	unrecognized non UNIX-based commands

Note: In this table, the first column indicates the command index used for identification purpose; the second is the actual command; the third is the level of expertise that a user types this command; and the fourth is the brief description of command usage. There are several levels of command expertise; "1" here indicates the easiest level; "2" middle level; and "3" the most proficient level. Each level is assigned with weights (*i.e.*, W_{ij} of 4, 7, and 10 respectively. The "*" is assigned with weight of 0 assuming the user knows the right command while issuing a wrong one due to typographical error. The "**" is assigned with weight of −4 assuming that this command can't be recognized by the UNIX operating system and is not caused by a typographical error. The "**" is used especially for differentiating the unusual and unrecognized commands from mis-spelled commands.

Chaochang Chiu is an associate professor in Department of Information Management at Yuan Ze University. He received his Ph.D. in Information Systems from University of Maryland, Baltimore County in 1993. His research interests are adaptive human–computer interfaces, genetic programming, and expert decision support systems.

Pei-Lun Hsu is currently a Ph.D. student of computer engineering at Yuan Ze University. He holds a MS in computer science from Tamkang University, Taipei, Taiwan in 1993. His research interests include fuzzy constraint-based seasoning and AI applications.

A.F. Norcio is a professor in Information Systems Department at University of Maryland, Baltimore County. He received his Ph.D. from Catholic University of America, 1978. His research interests are in the theoretical and applied areas of software/systems design, intelligent human–computer interfaces (including voice systems), and health informatics. He currently serves as a Computer Scientist at the Artificial Intelligence Center of the Naval Research Laboratory. He has also served as the Scientific Advisor to the Mathematical, Computer, and Information Sciences Division of the Office of Naval Research.