Strategies for Generating Items based on the Difficulty Analysis of Problems

Rita Kuo
Dept. of Electronic Engineering, Chung-Yuan Christian University
No.22, Pu-Jen,Pu-chung Li, Chung-Li, 320, Taiwan

Pei-Ming Tien
Dept. of Visual Communication Design, Shu-Te University
No.59, Hengshan Rd., Yanchao Township, Kaohsiung County, 824, Taiwan

Maiga Chang
National Science & Technology Program for e-Learning in Taiwan
No.300, Jungda Rd, Chung-Li, 320, Taiwan

and

Jia-Sheng Heh
Dept. of Electronic Engineering, Chung-Yuan Christian University
No.22, Pu-Jen,Pu-chung Li, Chung-Li, 320, Taiwan

Abstract

This paper proposes a simple CAI system for iteratively generating item according to the previous *problem solving operations* from learners. The way to reach this goal needs *difficulty table* to determine the next item in the candidates which according to the value of *difficulty dimensions* in current item and the candidates. To select harder or easier item depends on correctness of each *diagnosis feature*. The way to analyze *diagnosis features* based on *difficulty features* from previous research. Using these analyses, in the end of this research constructs an *item generating system* with *diagnosis features* for learners practicing their *problem solving* skills by their own.

Keywords: Problem solving, Knowledge Map, Diagnosis, Difficulty of problems, Computer-assisted instruction systems

1. Introduction

Computer-Assisted Instruction Systems (CAI Systems) help teachers and learners with interactive programs while they are in the process of teaching and learning. [5] Teachers make use of CAI Systems as intermediary media to guide learners how to gain knowledge in various instructional models via multi-media, network *etc.* On the other hand, learners can pay attention to various types of teaching material and get knowledge through the interactive learning environment in CAI Systems. In CAI Systems, some researches focused on the *problem solving* techniques. These researches include how to transform a regular problem into computer, [11] how to build a problem solving environment *etc.* [1][2] Some CAI Systems also concentrate on the applications of Intelligent Agent. Several distance learning systems propose agents for managing system files [13], notifying grades in on-line quiz system [12] and so on.

The way to make CAI Systems more intelligent and powerful is knowledge structure. *Knowledge Map* is one of knowledge structures proposed in recent research. [6] It consists of two major parts, *Concept Hierarchy* and *Concept Schema*. *Concept Hierarchy* presents the hierarchical relationship among concepts. *Concept Schema* stores the remaining information, which integrates the definition, example, and other relations with associated concepts. In Figure 1, *Concept Hierarchy* is demonstrated in gray block with link among concepts and *Concept Schema* of each concept is demonstrated in the white block.

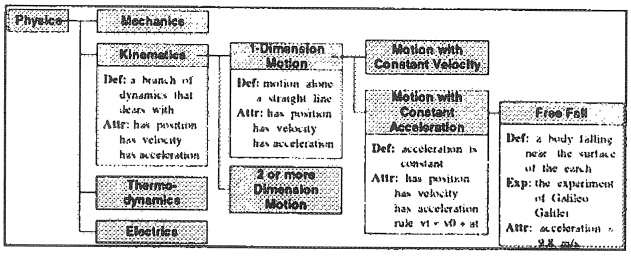


Figure 1. Example of Knowledge Map in Physics

This paper tries to build the architecture of CAI system which use *Knowledge Map* as its knowledge-base and have ability to train learners in problem solving operations. Section 2 describes some previous researches in *item generating steps* and *difficulty* definition. The method of how to select the next item with proper difficulty values would be analyzed in section

3. Section 4 designs *diagnosis features* to examine the correctness of *problem solving operations* from learners. An experiment system with example is demonstrated in section 5 and a section 6 gives a brief conclusion.

2. Problem Difficulties

In the previous research, the process of constructing problem use *Knowledge Map* as the knowledge provider. [7] Three major steps exist in the basic-problem constructing procedure, which are *concept selection*, *unknown designation*, and *proposition construction*, as shown in Figure 2. *Concept selection* sets the main concept of the basic problem. The next step, *unknown designation*, determines *given attributes* and *unknown attributes* in the problem. *Proposition construction* is the last step of the whole process, which builds sentences in natural language by using the result of setting-attributes from *unknown designation*. The entire procedure needs its knowledge base, *Knowledge Map*, and one specific problem metaphor, *Problem Matrix*.

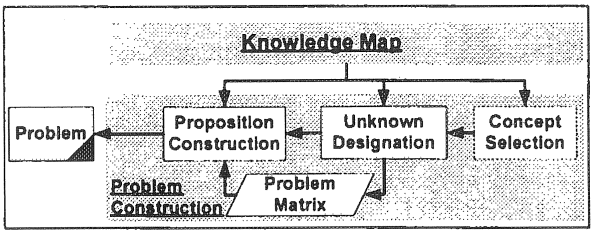


Figure 2. Process of constructing a basic problem

Problem Matrix displays the real manipulating concepts and relationships which are using for solving the problem. Figure 3 shows a *Problem Matrix* example for the Free Fall problem in Physics. Two Physics laws ("Displacement = 0.5 * Acceleration * Time ^ 2" and "Velocity = Acceleration * Time") present the relationships among four major concepts of Physics Quantity ("Displacement", "Time", "Velocity" and "Acceleration"). In the process of constructing a problem, *unknown designation* use *Problem Matrix* to know how to set *unknown attributes* and *given attributes*. For example of the *Problem Matrix* in Figure 3, *unknown designation* could choose "Displacement" as *unknown attributes*; "Time" and "Acceleration" as *given attributes* because learners could use the first manipulating relationship, "Displacement = 0.5 * Acceleration * Time ^ 2", for solving the problem.

	Displace-ment	Time	Velocity	Accele-ration
Displacement = 0.5 * Acceleration * Time ^ 2	1	1	0	1
Velocity = Acceleration * Time	0	1	1	1

Figure 3. Problem Matrix example

For deciding the difficulty of a problem, the previous study defined eight *difficulty features*, which are shown in Table 1 and based the problem constructing steps and the structure of *Knowledge Map*. [8] To simplify the problem, the study only considers six *difficulty features* for affecting the previous two steps of problem construction. The step of *concept selection* determines the features of *number of needed attributes*, *learning sequence*, and *concept depth*. *Number of unknown*, *number of given attribute*, and *number of elaborating attributes* would be influenced by the next step, *unknown setting*. *Difficulty features* may not be easily understood what learners get wrong in solving problem process. For this purpose, [8] also translates *difficulty features* as *difficulty dimensions*, which are according to the four problem solving knowledge schema proposed by Marshall. [3] [6] [10] [14]

Table 1. Difficulty Features and its denotation

Element	Difficulty Feature	Denotation
Selected Concepts	Number of Sub-Problems	γ_{sub_prob}
	Number of Needed Attributes	γ_{need_attr}
	Learning Sequence	γ_{learn_seq}
	Concept Depth	γ_{cpt_depth}
Attributes Setting	Number of Unknown	$\gamma_{unknown}$
	Number of Given Attributes	γ_{given_attr}
	Number of Elaborating Attributes	γ_{elb_attr}
	Mathematical Complexity	γ_{math_cpx}

Identification difficulty (γ_{idf}) focuses on the concept discussed in the problem. Two difficulty features are related to this dimension, which are *learning Sequence* and *concept Depth*. In the next three difficulties, the calculation takes more notice of attributes in the problem. *Elaboration difficulty* (γ_{elb}) emphasizes the attributes needed to solve the problem. *Number of needed attributes*, *number of given attributes*, and *number of elaborating attributes* influence this difficulty dimension. *Planning difficulty* (γ_{pln}) calculation needs to know the relation of unknown attributes in the problem. *Number of unknown* and *number of elaborating attributes* impact this difficulty dimension. *Number of unknown* also decides the difficulty dimension of *execution difficulty* (γ_{exc}). To transform difficulty features to difficulty dimensions, the transform formulas are listed below: (ω_{ij} indicates customized weight)

$$\gamma_{idf} = \omega_{21} * \gamma_{learn_seq} + \omega_{31} * \gamma_{cpt_depth}$$
$$\gamma_{elb} = \omega_{12} * \gamma_{need_attr} + \omega_{42} * \gamma_{given_attr} + \omega_{62} * \gamma_{elb_attr}$$
$$\gamma_{pln} = \omega_{53} * \gamma_{unknown} + \omega_{63} * \gamma_{elb_attr}$$
$$\gamma_{exc} = \omega_{54} * \gamma_{unknown}$$

This paper enhances the architecture of problem constructing steps and makes the unknown designation more intelligent according to the answer from the learners'. Two important issues exist in this research. How to decide next item from number of candidates according to multiple dimensions is one critical point of the architecture. Another

one is how to diagnose learners operation with various dimensions in the process of problem solving.

3. Item Determination

Item response theory is a method for considering the latent traits of the examinee. [9] This theory can be used to construct computerized adaptive testing system. Unidimensionality, local independence, and nonspeedness are three kinds of assumption that IRT must satisfies. [4] However, the definition of problem difficulty in this paper has four dimensions, inconsistent with the first assumption of IRT, and makes it not easy to decide the next item/problem. To solve this problem, this research proposes another way and use *Difficulty Table* to decide next item/problem.

Difficulty Table is a 2-dimension matrix shows the corresponding of each possible generated problem and *difficulty dimensions*. Figure 4 shows the concept of *Difficulty Table*, which denotes P_j as predicate j, γ_i as ith difficulty dimension, and d_{ji} as difficulty value of ith difficulty dimension in predicate j.

	γ_1	γ_2	γ_3	γ_4
P_1	d_{11}	d_{12}	d_{13}	d_{14}
P_2	d_{21}	d_{22}	d_{23}	d_{24}
P_3	d_{31}	d_{32}	d_{33}	d_{34}
P_4	d_{41}	d_{42}	d_{43}	d_{44}
P_5	d_{51}	d_{52}	d_{53}	d_{54}

Figure 4. The concept of *Difficulty Table*

Before using *Difficulty Table* for determining the next item, the value of current *difficulty dimensions* (D_i^k) and the correctness of learner's operation (a_i) in each *difficulty dimension* are two important data which are listed below.

- $a_i = \begin{cases} +1 & \text{if learner's operation in } \gamma_i \text{ is correct} \\ -1 & \text{if learner's operation in } \gamma_i \text{ is incorrect} \end{cases}$
- D_i^k the value of γ_i in kth item

The idea of selecting next item can use Figure 5 to explain. If the current difficulty value of γ_i is D_i^k and five possible difficulty values (d_{1i}, d_{2i}, d_{3i}, d_{4i}, d_{5i}) could be chosen for the next item, the correctness of learner's operation (a_i) makes the step of *unknown designation* pick harder (d_{4i}, d_{5i}) or easier (d_{2i}, d_{1i}) one for the setting. The sorting list of selecting difficulty value should be $d_{4i} > d_{5i} > d_{3i} > d_{2i} > d_{1i}$ if $a_i = +1$; or $d_{2i} > d_{1i} > d_{3i} > d_{4i} > d_{5i}$ if $a_i = -1$. To make the sorting list, a matrix S stores the priority of selecting value and the calculation of each item in S (which denoted as s_{ji}) can be defined as below:

- $s_{ji} = strength(a_i(d_{ji} - D_i^k)) + 1 - |d_{ji} - D_i^k| \quad s_{ji} \in S$

where $strength(a_i(d_{ji} - D_i^k)) = \begin{cases} 1 & \text{if } a_i(d_{ji} - D_i^k) > 0 \\ 0 & \text{if } a_i(d_{ji} - D_i^k) \le 0 \end{cases}$

After the calculation of the priority matrix, the *unknown selection* step could find the predicate which have most high priority value in matrix S.

- $count_j$ number of jth predicate that is the max difficulty value of each *difficulty dimension*

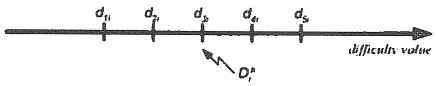


Figure 5. Difficulty value selection example

An example of previous definitions is shown in Figure 6. There are five candidate predicates of the problem and the difficulty of all predicates are shown in Figure 6(a). Assume that the current difficulty setting and the correctness of learner's problem solving operations are listed in Figure 6(b). Using the formula described above, table with double line frame in Figure 6(c) displays the priority calculation matrix, S, and find out predicate P_2 have high priority value at most. According to the calculating result in Figure 6(c), Figure 6(d) indicates the value of the next difficulty dimensions (D_i^{k+1}).

	$\gamma_1 = \gamma_{diff}$	$\gamma_2 = \gamma_{clth}$	$\gamma_3 = \gamma_{pth}$	$\gamma_4 = \gamma_{exc}$
P_1	$d_{11} = 0.3$	$d_{12} = 0.7$	$d_{13} = 0.3$	$d_{14} = 0.5$
P_2	$d_{21} = 0.6$	$d_{22} = 0.3$	$d_{23} = 0.6$	$d_{24} = 0.7$
P_3	$d_{31} = 0.4$	$d_{32} = 0.5$	$d_{33} = 0.4$	$d_{34} = 0.4$
P_4	$d_{41} = 0.7$	$d_{42} = 0.4$	$d_{43} = 0.7$	$d_{44} = 0.6$
P_5	$d_{51} = 0.5$	$d_{52} = 0.6$	$d_{53} = 0.5$	$d_{54} = 0.6$

(a)

$a_1 = +1$	$a_2 = -1$	$a_3 = -1$	$a_4 = +1$
$D_1^k = 0.5$	$D_2^k = 0.2$	$D_3^k = 0.5$	$D_4^k = 0.8$

(b)

	γ_1	γ_2	γ_3	γ_4	*count*
P_1	0.8	0.5	1.8	0.7	0
P_2	1.9	0.9	0.9	0.9	3
P_3	0.9	0.7	1.9	0.6	1
P_4	1.8	0.8	0.8	0.8	0
P_5	1.0	0.6	1.0	0.5	0

(c)

$D_1^{k+1} = 0.6$	$D_2^{k+1} = 0.3$	$D_3^{k+1} = 0.6$	$D_4^{k+1} = 0.7$

(d)

Figure 6. An example of difficulty decision making. **(a)** Difficulty Table; **(b)** The value of each difficulty dimension (D_i^k) and the correctness of learner's operation (a_i); **(c)** The priority calculation matrix S (table in double line frame) and the number of difficulty value which fits the consideration ($count_j$); **(d)** The value of difficulty dimensions (D_i^{k+1}) of the next predicate, P_2.

4. Problem Solving Operation Diagnosis

Using the analysis in the previous section, this paper enhances the procedure of item generation as shown in Figure 7. Beside adding *problem solving system* for getting correct answer of the problem, the details in the former two steps, *concept selection* and *unknown designation*, have made some variation and store relative information for comparing *problem solving operations* from learners'. The enhancement of step *unknown designation* has been mentioned in section 3, and this section will describe other two essential mechanism of system design, *conception-selecting strategy* and *problem-solving operation analysis*.

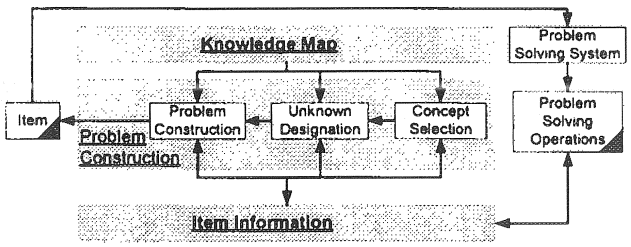


Figure 7. Item Generation Procedure

To make the whole system more automatically, the enhanced part in *Concept Selection* should have the ability of traversing all concepts in a concept-set voluntarily. In the architecture of *Knowledge Map*, all concepts are organized as tree structure, and any sub-tree in the *Knowledge Map* can be treated as a concept-set. To automatically journey over each item in the sub-tree (concept-set), many tree-structure traversing algorithms described in textbook have been development. Depth-first search is chosen for our algorithm.

The analysis of problem solving operations from learners derives from the *Difficulty Features* mentioned in section 2. Extracting six *difficulty features* in use through [8], each of them corresponds to one *Difficulty Dimensions* as shown in the left side of Table 2, which of the origin comes from the transformation between *difficulty features* and *difficulty dimensions*. To make it easier to use in problem solving operation diagnosis, this paper defines five *diagnosis features* (right part of Table 2) which are used to identify problem solving abilities of learners'.

Major concept, the first diagnosis features in Table 2, composes *difficulty features* of *learning sequence* and *concept depth* and influence *difficulty dimensions* of *identification*. In *elaboration difficulty dimension*, *number of given attributes*, *elaborating attributes*, and *needed attributes* are useful diagnosis features, but *diagnosis features* only use *given attributes* and *elaborating attributes* for analysis because of the hard detecting of *needed attributes'* operation. *Unknown attributes* influence *difficulty dimensions* of *planning* and *execution*. It is referred to the *diagnosis feature* of *answer*, which are determined for diagnosing *difficulty dimension* of *execution*. An external *diagnosis feature*, *problem solving path*, is joined in for verifying the *difficulty dimension* of *planning* in the bottom of the table. These five *diagnosis features* will be transformed into user interface of the assistant system for deciding difficulty value of next item.

Table 2. Analysis of problem solving operations

Difficulty Features	Difficulty Dimensions	Diagnosis Features	Difficulty Dimensions
Learning Sequence	Identification	Major Concept	Identification
Concept Depth	Identification		
Number of Given Attributes	Elaboration	Given Attributes	Elaboration
Number of Elaborating Attributes	Elaboration / Planning	Elaborating Attributes	Elaboration
Number of Needed Attributes	Elaboration		
Number of Unknown	Planning / Execution	Problem Answer	Execution
		Problem Solving Path	Planning

For applying diagnosis of *problem solving operations*, this paper also designs dialogic interface for learners answering relative question, which can get their idea of how to solve this problem. To make the answer convergent, *selection type* of the question is one way proceed to create conversation. The way to generate proper sentence of question depends on the respective information of the problem. For this purpose, three essential structures, *concept information*, *difficulty table*, and *problem matrix*, gives diverse information in different steps.

Concept information records the concept of the entire test and the major concept of the current item. This information can be used to ask learners the diagnosis feature, *major concept*, of the problem. *Difficulty table* lists values of *difficulty dimension* including the current item and the possible predicates of the next item. The last data structure, *problem matrix*, stores many information of the problem, including *given attributes* and *elaborating attributes* which also parts of *diagnosis features*. Other information of the *problem matrix*, manipulating relation, can be considered as the *diagnosis features* of *problem solving path*. The last feature, *problem answer*, can not be gained from the problem directly but can be acquired from using *problem solving system*.

5. Enhanced Item Generating System

The new Item generating system (Figure 8), which based on the previous research in [8], integrates diagnosing problem solving operations in to the original one. The learner could select a *concept set*, which comes from *declarative knowledge*

in *knowledge map*, and the system would generate one item at first. The generated item is delivered to *dialogue interface*, which is an interface for learners operating problem solving strategies. The item also would be transferred to *problem solving system* in order to get the correct answer for comparing learners' response. The operation of learners' would compare data in the *item information* then decide the *difficulty features* of the next item.

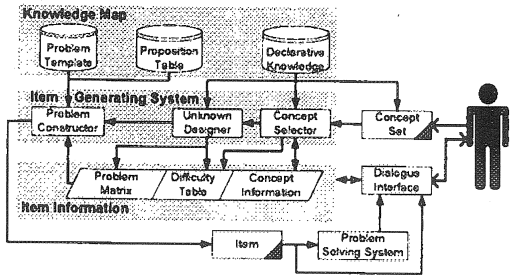


Figure 8. System architecture of enhanced *Item Generating System*

This research also uses Physics for example. Figure 9 displays the snapshot of selecting *concept set* from the learner. After selecting the concept set, which contains *1-dimension motion* as the root node, the system generate a new item and create dialogue of asking question for diagnosing different *difficulty features*. To simplify the procedure of predicting *difficulty value* in the next item, each *difficulty dimension* use only one *diagnosis feature* to estimate the next possible value.

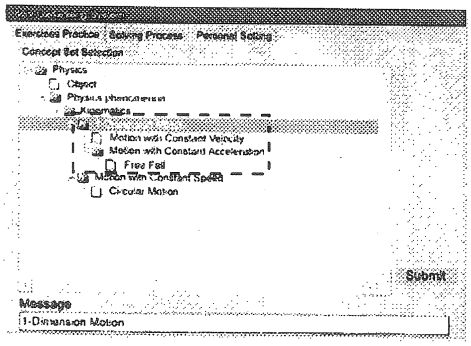


Figure 9. Selecting concept set

Figure 10 ask leaner the major concept of the item and influence the value of *identification difficulty* in next item. Furthermore, Figure 11 shows the question of *given attributes*, which determines the value of *elaboration difficulty*. The dialogue of asking *problem solving path* is demonstrated in Figure 12 and shows that the accurately use of physics law in the problem solving process is to find to proper way to get the answer. Figure 13 is the last question of all diagnosis and query the final answer. This step affects the *execution difficulty* value of the next item. After the whole diagnosing process, the system will displays all the answer the learner replied and asks him to generate another item or reselect the concept set (Figure 14). This result will influence the next item.

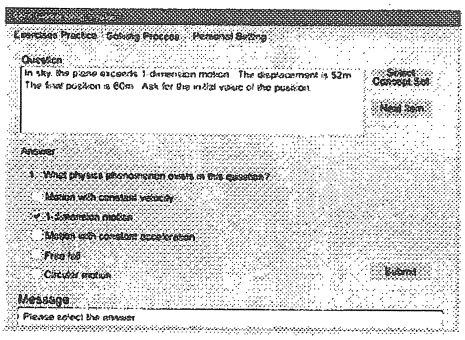


Figure 10. *Identification* difficulty diagnosis

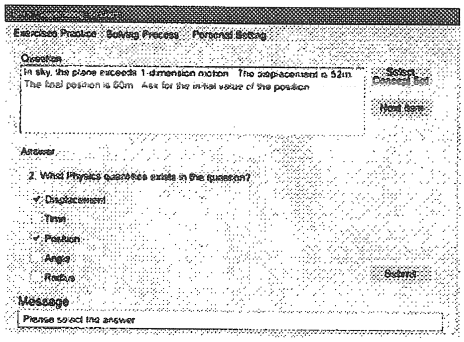


Figure 11. *Elaboration* difficulty diagnosis

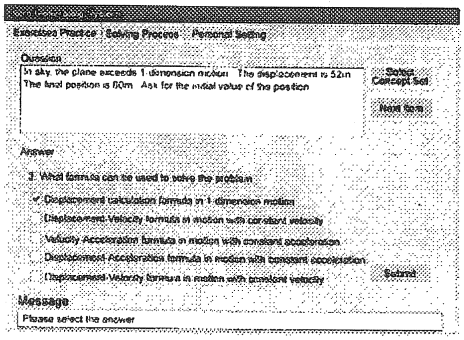


Figure 12. *Planning* difficulty diagnosis

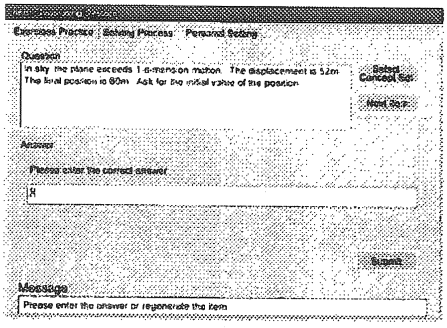


Figure 13. *Execution* difficulty diagnosis

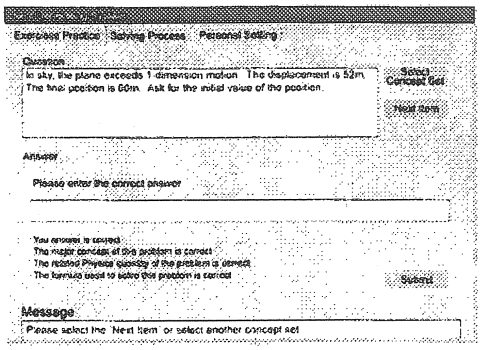


Figure 14. Diagnosis displacement

6. Conclusion

This paper proposes a strategy of generating items iteratively. Based on the previous research in [7], the process of *problem construction* retain three stages but the previous two are enhanced including the way to select next concept and the algorithm for choosing next item based on *difficulties* setting. The whole procedure needs to define several data structures as *item information* to execute each step.

Concept Information keeps data from *knowledge map* and use depth first search to decide the next major concept of the next item. Another important data in the entire procedure is *Problem Matrix*, which can be used to define *diagnosis features* to estimate the ability of problem solving operation from learners and use those data to evaluate the difficulty value of next item with *Difficulty Table*. In the end, an enhanced *item generating system* is constructed and has a small example to demonstrate the process.

The future work of this paper is integrating the idea of *item response theory* in to the architecture. Furthermore, how to make the interface of *problem solving operation diagnosis* more diversity and accurately is another task of the research.

Reference

[1] Julie Baher, (1998) "How Articulate Virtual Labs can Help in Thermodynamics Education: A Multiple Case Study," *Frontiers in Education Conference, 1998. FIE '98. 28th Annual , Volume: 2*, pp. 663 –668 vol.2

[2] William H. Blackmon and Steven J. Fenves, (1997) "An Environment for Modeling Structural Analysis Problem Strategies," *Frontiers in Education Conference, 1997. 27th Annual Conference. Teaching and Learning in an Era of Change. Proceedings. , Volume: 3*, 1997, pp. 1338 –1342 vol.3

[3] Kun-Fa Cheng, Rita Kuo, Kun-Yuan Yang and Jia-Sheng Heh (2001), "Intelligent Agent Construction for Problem Solving with Knowledge Schemas," *the 17th Workshop on Science Education*, Kaohsuang, Taiwan, 2001 (published)

[4] Hambleton, R. K., & Swaminathan, H. (1985), *Item Response Theory: Principles and Applications*, Boston: Kluwer-Nijhoff

[5] Robert Heinich, Michael Molenda and James D. Russel, (1982) *Instructional Media and the New Technologies of Instruction*, Macmillan Publishing Company, 1982

[6] Rita Kuo, Maiga Chang, Da-Xian Dong, Kun-Yuan Yang, and Jia-Sheng Heh(2002), "Applying Knowledge Map to Intelligent Agents in Problem Solving Systems," *World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-Media 2002)*, Denver, Colorado, USA, Jun. 24-29, 2002, pp.1053-1054

[7] Rita Kuo, Wei-Peng Lien, Maiga Chang and Jia-Sheng Heh (2003), "Problem Model Analysis in Inter-Correlated Knowledge by Implementing Item Generating System in Knowledge Map," *AACE World Conference on Educational Multimedia, Hypermedia & Telecommunications, (ED-Media 2003)*, Hawaii, USA, June 23-28, 2003, pp.1150-1153.

[8] Rita Kuo, Wei-Peng Lien, Maiga Chang and Jia-Sheng Heh (2003), "Difficulty Analysis in Basic Problem based on Four Steps of Problem Solving Strategies by Knowledge Map," *IEEE International Conference on Advanced Learning Technologies, (ICALT 2003)*, Athens, Greece, July 09-11, 2003 , pp. 386-387.

[9] Kuo, T. H. (1997), "Distance adaptive testing system - Design, implementation and evaluation," Unpublished master's thesis. National Taiwan Normal University. Taipei, Taiwan.

[10] Marshall, Sandra P., (1995) *Schemas in Problem Solving*, Cambridge University Press, 1995

[11] Richard E. Mayer, (1992) "Teaching for Transfer of Problem –Solving Skills to Computer Programming," *Computer-Based Learning Environments and Problem Solving*, Springer-Verlag Berlin Heidelberg, 1992, pp. 193 – 206

[12] A. Okada, H. Tarumi and Y. Kambayashi (2000), "Real-time quiz functions for dynamic group guidance in distance learning systems," *Proceedings of the First International Conference on Web Information Systems Engineering, 2000*, Vol.2, pp.188-195

[13] A. Shimano and H. Kuramae(2000), "Design and construction of educational computer system using self-maintenance system for files and user identification agent," *Proceedings of the 9th IEEE International Workshop on Robot and Human Interactive Communication, 2000. RO-MAN 2000*, pp.23-28

[14] Tung, Ta-Hsien (2002), "Apply Knowledge Map to Develop Physics Problem-Solving System," Thesis of Master, Dep. Information and Computer Engineering, Chung Yuan Christian University, Chung-Li, Taiwan, R.O.C., Jan, 2002