To Develop an Item Generating System with Analyzing the Problem Model in the Inter-Correlated Knowledge

Rita Kuo, Wei-Peng Lien[1], Maiga Chang[2], Jia-Sheng Heh

Dept. of Electronic Engineering, Chung-Yuan Christian Univ., Taiwan, R.O.C

[1]Dept. of Visual Communication Design, Shu-Te University, Taiwan, R.O.C

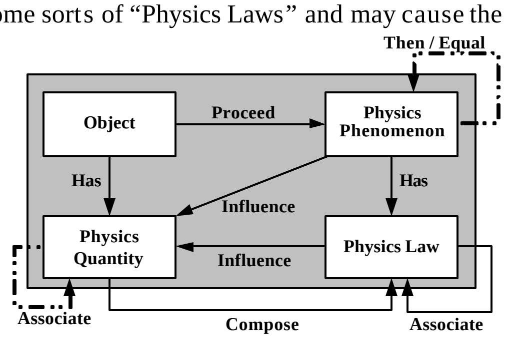
[2]Dept. of Special Education, Chung-Yuan Christian Univ., Taiwan, R.O.C

rita@mcsl.ice.cycu.edu.tw, apinh@hotmail.com, maiga@ms2.hinet.net, jsheh@ice.cycu.edu.tw

Abstract: This paper proposes two problem structures for generating basic problems from the Inter-Correlated Knowledge. The first problem structure is *Problem Graph*, which expresses the relationship among concepts in the problem. Another problem structure, *Problem Matrix*, is used for manipulating those concepts and relations in the *Problem Graph*. Using these definitions, the problem model for the Inter-Correlated Knowledge could be divided into three essential parts, *Sub-Problem Flag Attribute*, *Given Attributes* and *Unknown Attributes*, for constructing a problem. According to this analysis, *Proposition Table* and *Problem Template* also become an important data structure in the process of problem construction. The system architecture for item generating is proposed in this paper and takes a physics problem as demonstration.

1. Knowledge Map and Inter-Correlated Knowledge

Knowledge Map, the knowledge structure used in the paper, comes from the idea of *Concept Map* and *Schema*. (Kuo et al. 2002) In order to achieve the objectives, Item Generating and Problem Solving, with *Knowledge Map*, a *Knowledge Object* is proposed to replace the simple concept node exists in *Concept Map* for constructing a *Knowledge Map*. (Tung, 2002) There exists at least one *Core Knowledge Object* for representing the central idea of specific domain which composed by several *Knowledge Objects*. (Fig. 1a) shows a simple example for Physics that consists of four *Knowledge Objects*, "Object", "Physics Phenomenon", "Physics Law", and "Physics Quantity." The *Core Knowledge Object* selected here by us in this example is "Physics Phenomenon." As (Fig. 1a) shown below, there are numerous relation types among *Knowledge Objects*. (Clifford, 1981) For example, a "Physics Phenomenon" is happened based on some sorts of "Physics Laws" and may cause the value of its "Physics Quantity" to change.



(a)

⇔

Proposition	Relation Type
[OBJ]<Proceed>[PHE]	PW
[PHE]<Has>[LAW]	PW
[PHE]<Influence>[QTY]	INF
[LAW]<Influence>[QTY]	INF
[OBJ]<Has>[QTY]	PW
[QTY]$_{1\sim n}$<Compose>[LAW]	PW
[LAW]$_{1\sim n}$<Associate >[LAW]$_0$	ASO
[PHE]$_2$<Then>[PHE]$_1$	SQN
[PHE]$_1$<Equal>[PHE]$_2$	EQL
[QTY]$_1$<Associate>[QTY]$_2$	ASO

(b)

Figure 1: Inter-Correlated Knowledge example and its relation analysis in Physics

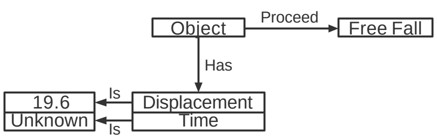
Knowledge could be classified into three categories: *Grammatical Knowledge*, *Positioning Knowledge*, and *Inter-Correlated Knowledge*. (Kuo, 2002) Ideas and design in this paper will focus on the architecture in *Inter-Correlated Knowledge*, which pays much attention on the manipulation of inter-correlation among knowledge as in Physics, Chemistry, and Mathematics. By using the definition of *Knowledge Objects* in specific knowledge domain, Section 2 models the problem structure in *Inter-Correlated Knowledge* and designs two essential data structure for describing the problem architecture.

2. Problem Structure

Problem solving has already been discussed for a long time since 1910. (Deway, 1910; Polya, 1965; Deek et al. 1999; Marshall, 1995; Cheng et al. 2001; Tung, 2002; Kuo et al. 2002; Hsu et al. 2002; Kuo, 2002) Most of the problems can be divided into several sub-problems for simplifying the process of problem solving. These kinds of problems are called complex problems. On the contrary, problems cannot be separated into smaller ones are called simple problems, or basic problems. For designing the problem model of basic problems in this paper, two major problem structures are proposed, *Problem Graph* and *Problem Matrix*.

Problem Graph

To illustrate a problem, a graph composed with proposition, which is defined as an integration of concepts and relations (Gagné et al., 1993). This paper defines a *Problem Graph* as a set of proposition for describing a problem. Take (Fig. 2) as example. There are six concepts ("Object", "Free Fall", "Displacement", "Time", "19.6", "unknown") exist in the problem graph. (Object <proceeds> Free Fall), (Object <has> Displacement), (Object <has> Time), (Displacement <is> 19.6), and (Time <is> Unknown) are five propositions existed in this problem (angle brackets expresses the relationship between two concepts). These propositions can be transformed to the related proposition matrix as shown in the right side of (Fig. 2). The relations within the proposition can be ten types of relationships proposed in the "The Frame Game" developed by (Clifford, 1981) as shown in (Tab. 1).



⇔

	Object	Free Fall	Displacement	Time	19.6	Unknown
Proceed	1	1	0	0	0	0
Has	1	0	1	0	0	0
Has	1	0	0	1	0	0
Is	0	0	1	0	1	0
Is	0	0	0	1	0	1

Figure 2: Example of Problem Graph and its related proposition matrix

Code	Pattern	Code	Pattern
HRC	Superordinates / Subordinates / Coordinates Pattern	*OPS*	Opposites Pattern
SQN	Sequence Pattern	*SML*	Similarity Pattern
PW	Parts-Whole Pattern	*CE*	Cause-Effect Pattern
ASO	Associates Pattern	*INF*	Influence Pattern
EQL	Equals Pattern	*ANG*	Analogy Pattern

Table 1: Ten Relation Types of Relationships in Proposition

Problem Matrix

Although a problem can be illustrated by *Problem Graph*, inter-correlative knowledge still needs a data structure to store and present the relations among *manipulating* concepts. For this purpose, this paper proposes a *Problem Matrix* to extract the manipulating relations among manipulating concepts. Take Physics for example, most of the problems focus on manipulating relations among Physics Quantity. (Fig. 3) shows a *Problem Matrix* for representing the Free-Fall Problem. Two Physics Laws in the first column presents the relationships among four major Physics Quantities described in the first row.

.

	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: Example of Problem Matrix in Free-Fall Problem

	Corresponding Concepts	Example of Physics Problem
Basic Problem Model	*Sub-Problem Flag Attribute*	*Free Fall* Phenomena
	Given Attributes	Given Value of *Distance* and *Velocity* Attributes
	Unknown Attributes	Ask for *Time* Attribute

Figure 4: Example of Physics Problem in Basic Problem

To construct a standard information-processing framework, a problem is constructed by some states: one end state, one starting state, and several intermediate states. (Gagné et al. 1993) The initial state of the problem has a goal and the goal is blocked for lack of resources (variables, or concepts in this paper) of information. (Kahney, 1993) This paper simplifies the study and takes focus on the structure of basic inter-correlated knowledge problem. For this purpose, a basic inter-correlated knowledge problem can be defined as a problem with only one *Core Knowledge Object*. Using this definition, a basic problem model can be separated into *Sub-Problem Flag Attribute, Given Attributes* and *Unknown Attributes*. *Sub-Problem Flag Attribute* is presented by *Core Knowledge Object*, and other knowledge objects refer to *Given* and *Unknown Attributes*. (Fig. 4) shows a "Free Fall" example with two *Given Attributes – distance* and *velocity*, and one *Unknown Attribute – time*.

3. Problem Construction

As the basic problem model analyzed in (Fig. 4), the first step to construct a problem is to define the problem domain and given attributes. To reach this goal, the setting of attributes in the problem can be extracted from the attributes in *Concept Schema* to the manipulating attributes and relations in *Problem Matrix*. (Fig. 5) shows an example and the words in italic type in the second row display the setting values. The *Given Attributes* are "vi", "ti", "tf", and "a" and the *Unknown Attribute* is "vf". Besides using *Problem Matrix*, other two data structures are also used for constructing a readable problem by human. They are *Proposition Table* and *Problem Template*.

	di	df	vi	vf	ti	tf	a
			4	*unknown*	*2*	*36*	*9.8*
vf = vi + a * (tf - ti)	0	0	1	1	1	1	1
df = di + vi * (tf - ti) + 0.5 * a * (tf - ti) ^ 2	1	1	1	0	1	1	1
vf ^ 2 = vi ^ 2 + 2 * a * (df - di)	1	1	1	1	0	0	1

Figure 5: Problem Matrix

Proposition Table

For constructing meaningful sentence, *Proposition Table* stores a table for transforming sentence and proposition. This table comes from the analysis of proposition/relation in *Knowledge Objects* as shown in (Fig. 1a). Seven relations with normal line present relations among *Knowledge Objects* and two dotted lines present relations among sub-problems. All these relations in *Proposition Table* are shown in (Fig. 6). Three major attributes are designed in *Proposition Table*: *proposition id*, *syntax pattern* and *proposition format*. *Syntax pattern* stores syntax problem may use with its related *proposition id*. Correspondingly, each syntactic rule in *syntax pattern* can be transformed into proposition form in *proposition format*, identifying the relation and concepts in the sentence. A Physics example is also shown in the white block of (Fig. 6).

	Proposition ID	Syntax Pattern	Proposition Format
Proposition Transform Unit	Relation Type with identify no.	Syntactic rule	Relation Type Concept Set
Example in Physics [Fig. 1a]	PW-1	[Object] <Proceed> [Phenomena]	Relation Type: PW Concept 1: Object in Physics Concept 2: Phenomena in Physics

Figure 6: Proposition Transform Unit with Physics Example

Model	Parameter	Available Proposition	Selected Proposition in Physics Example	Example in Physics
Number of Sub-Problems	1			
Number of Unknowns	1			
Basic Problem Model	Sub-Problem Flag Attribute	PW-1	Relation Type: PW Concept1: Object in Physics Concept2: Phenomena in Physics	[Object] <proceed> [Free Fall]
	Given Attributes	EQL-1 EQL-2	Relation Type: EQL Concept 1: Quantity in Physics Concept 2: Value	[Displacement] <is> [19.6]
	Unknown Attributes	PWQ-1	Relation Type: PW Concept 1: Object in Physics Concept 2: Quantity in Physics	

Figure 7: Example of Problem Template

Problem Template

Problem Template is designed for constructing a complete problem. According to the analysis of problem model, a problem template of basic problem should consist of the *Problem Frame* (which border style is double-line) and its corresponding parameters, including *Number of Sub-Problems*, and *Number of Unknowns*. The left side of (Fig. 7) demonstrates an example of basic problem template. The template stores the number of sub-problems and the number of unknowns in the beginning. The lower part with double-line border is the problem frame of this template. Each parameter of the basic problem model has its available propositions indicated by their *Proposition ID* and can be referred to the *Proposition Table*. The right part of (Fig. 7) with dotted-line is an example of Physics problem.

4. Item Generating System and Conclusions

According to the analysis in previous Section, (Fig. 8) shows the architecture of an *Item Generating System* designed in this paper. After selecting the concept of the problem, the *Item Generating System* uses *Unknown Designer* to construct *Problem Matrix* and decides the given and unknown attributes, which needs information from *Declarative Knowledge* in *Knowledge Map*. The next step, *Problem Constructor*, uses the settings in *Problem Matrix* and decides which template is suitable from *Problem Template*. To construct a readable problem, each proposition in *Problem Template* should find the proper one from *Proposition Table*. (Fig. 9) shows a final result of the *Item Generating System* in Physics example: "In sky, a raindrop does Free Fall. The initial velocity is 4 m/s. Acceleration is 9.8 m/s^2. The final time is 36 s. The initial time is 2 s. Ask for the final value of velocity." The *Problem Matrix* of the example is also demonstrated in (Fig. 5) with given attributes, "vi", "ti", "tf", "a" and unknown attribute, "vf".