



PERGAMON

Expert Systems with Applications 22 (2002) 235–248

Expert Systems  
with Applications

www.elsevier.com/locate/eswa

# Building a CAL Expert System based upon Two-phase Knowledge Acquisition

Chang-Jiun Tsai, S.S. Tseng\*

*Department of Computer and Information Science, National Chiao Tung University, Hsinchu 300, Taiwan, ROC*

## Abstract

With the fast growing and globally accepted e-learning technology, Computer-Assisted Learning (CAL) system, which can provide the individualized materials, becomes a matter of great importance. In this paper, we propose a CAL Expert System (CAL-ES), in which teachers can provide their teaching strategy as guidance to students' learning. Because the knowledge hierarchy and the knowledge itself are both important in CAL domain, and the teaching strategy can be considered as rule-format, Two-phase Knowledge Acquisition is proposed to acquire the knowledge hierarchy and the rule-based knowledge by the cooperation of teachers and knowledge engineers. The prototype of CAL-ES is constructed of the mathematical teaching materials. The CAL-ES is useful for student learning and is easy for teachers maintaining. © 2002 Elsevier Science Ltd. All rights reserved.

*Keywords:* Ontology; Computer-Assisted Learning; Knowledge Acquisition; Resource Description Framework

## 1. Introduction

As Internet usage becomes more popular over the world, e-learning system, e.g. online learning, employee training courses, e-book, etc. has been accepted globally. How to provide the optimum content for individualized learning is an important issue for e-learning system. Several e-learning systems including online teaching, online tutoring, e-book, etc. have been proposed in the past 10 years (Alessi & Trollip, 1991; Beishuizen & Stoutjesdijk, 1999; Chou, 1996; Hwang, 1998; Oakley, 1996; Sun & Chou, 1996). However, these systems have provided the same or similar materials based on some fixed strategy or fixed learning map for all online learners despite of their personalized features or learning status.

A Computer-Assisted Learning (CAL) system, a kind of e-learning system, based upon Object-Oriented Course Model (OOCM) (Su, Tseng, Tsai, & Cheng, 1999, 2000) have been proposed to provide individualized teaching materials for learners in accordance with their learning aptitudes and evaluation results. In this system, the original teaching materials are divided into several segments, which are called teaching objects, according to the instructional objectives defined by teachers or teaching material editors. Then, these teaching objects can be constructed dynamically in accordance with students' learning status,

when they are learning online. The construction algorithm of teaching materials can be considered as the teaching strategy of teachers. However, teaching strategy may differ with teachers owing to the differences on teaching materials and teaching experiences. Therefore, in this paper, we build up a CAL Expert System (CAL-ES), which can guide students to learn the online teaching materials according to the teaching strategy provided by teachers in rule-format, and the learning aptitudes and evaluation results of students.

The knowledge hierarchy and the relevant knowledge of CAL-ES can be considered as the learning map and the teaching strategy, respectively. The knowledge about how to guide students to learn is stored as a set of rules, the teaching strategy of different instructional objectives can be modularized, and there exist some directed links among the different knowledge objects (Wu & Cai, 2000), which form the knowledge hierarchy, so it is considered as the learning map.

Moreover, unfamiliarity with expert systems and computer technologies may have caused the difficulty in knowledge acquisition for teachers. Therefore, we propose the Two-phase Knowledge Acquisition (Tp-KA) to help teachers acquire the knowledge hierarchy and the relevant knowledge systematically and effectively. The first phase is to construct the knowledge hierarchy and the hierarchical grids. The second phase is to transform the knowledge hierarchy from lattice structure into tree structure and extract the rules with embedded meaning from the hierarchy grids. Based upon Tp-KA, teachers can transform their teaching

\* Corresponding author. Tel.: +886-3-573-1966; fax: +886-3-572-1490.  
E-mail address: sstseng@cis.nctu.edu.tw (S.S. Tseng).

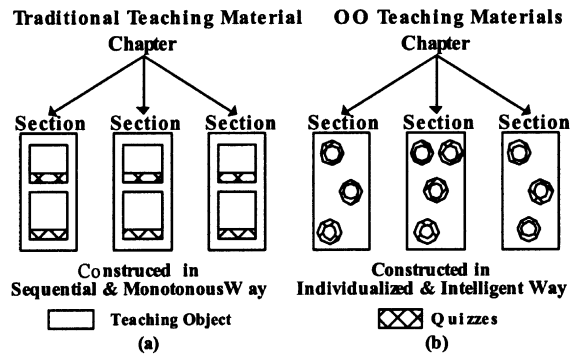


Fig. 1. Traditional and OO teaching materials.

strategy into guiding rules stored in CAL-ES, and students can learn individualized online teaching materials by following the instructions. The prototype of CAL-ES is constructed on a basis of the domain about mathematical teaching materials of senior high school.

## 2. Related works and motivation

CAL systems, which can guide learners to study the materials, become more popular over the world. Therefore, how to design and construct CAL systems together with its teaching materials is of much concern. Several approaches, which can be used to provide the teaching materials for students learning, had been developed in the past 10 years (Alessi & Trollip, 1991; Beishuizen & Stoutjesdijk, 1999; Chou, 1996; Hwang, 1998; Oakley, 1996; Sun & Chou, 1996). Some of them (Hwang, 1998; Oakley, 1996; Sun & Chou, 1996) provide the evaluation mechanism to find out what instructional objectives were not achieved well. According to these evaluation results, the proper remedial teaching materials can be offered to the students, who have not learned well.

As shown in Fig. 1(a), the traditional materials usually arrange the content and quizzes in sequence monotonously. It means that all the students learn the same teaching materials sequentially without allowed to skip the subsection they have learned. In Fig. 1(b), OOCM (Su et al., 1999, 2000) can segment the original teaching materials into several segments, which are called teaching objects. These teaching objects contain the teaching content and quizzes based on the instructional objectives defined by teachers.

Based upon the OOCM, the teaching materials can be constructed dynamically by organizing the teaching objects according to learning map and students' learning aptitudes and evaluation results. Thus, the individualized teaching materials can be offered to each student for learning.

The learning map, which is defined by teachers or teaching materials editors, consists of some teaching objects and teaching strategy of teachers. The teaching materials can be provided for students to learn according to the learning map. In other words, the learning map can guide the learning

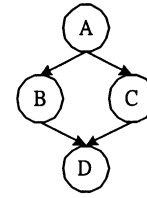


Fig. 2. An example of learning map.

process of students. Given four teaching objects, A, B, C, and D in the example of learning map shown in Fig. 2, A is the prerequisite knowledge of B and C, B and C are the prerequisite knowledge of D, and D is the end of the course. Based on this learning map, A is provided for students to learn first, and then either B or C is provided in accordance with the learning aptitudes and evaluation results of the students. Finally, D is provided for finishing this learning process.

As we know, most existing CAL systems guide students to learn under fixed strategy and fixed material construction algorithm. They rarely provide dynamical mechanism, a function allowing teachers to use their own teaching strategy or the existing teaching strategy defined by educational experts or senior teachers to guide the students in learning process. Therefore, our idea is to construct a rule-based CAL-ES, in which the teaching strategy of teachers is represented by rule-format. Teachers can easily encode their teaching strategy into rule-format to guide students learning. Moreover, the learning map and the teaching strategy, which are constructed by senior teachers or educational experts and stored in the Knowledge Base (KB) of CAL-ES as the knowledge hierarchy and the relevant knowledge, respectively, can be reused as a reference for junior teachers to improve their teaching skills.

One important function of CAL-ES is to provide a proper knowledge acquisition to assist the teachers, who are not familiar with computer and expert system technologies in acquiring their teaching strategy and store them into the KB. In addition, the knowledge hierarchy and the relevant knowledge are both important for constructing expert system in CAL domain, since the information about learning map and the teaching strategy is important for teachers to refine their designed teaching materials. Therefore, we also propose Tp-KA, which divides the processes of constructing the knowledge hierarchy and acquiring the knowledge into two phases, to knowledge engineers so as to assist teachers in processing the knowledge acquisition systematically and effectively.

## 3. CAL Expert System

As we know, rule-based expert system is to model the decision-making competence of human experts, in this case, considered as the ability of teachers to guide students learning in accordance with their learning aptitudes and

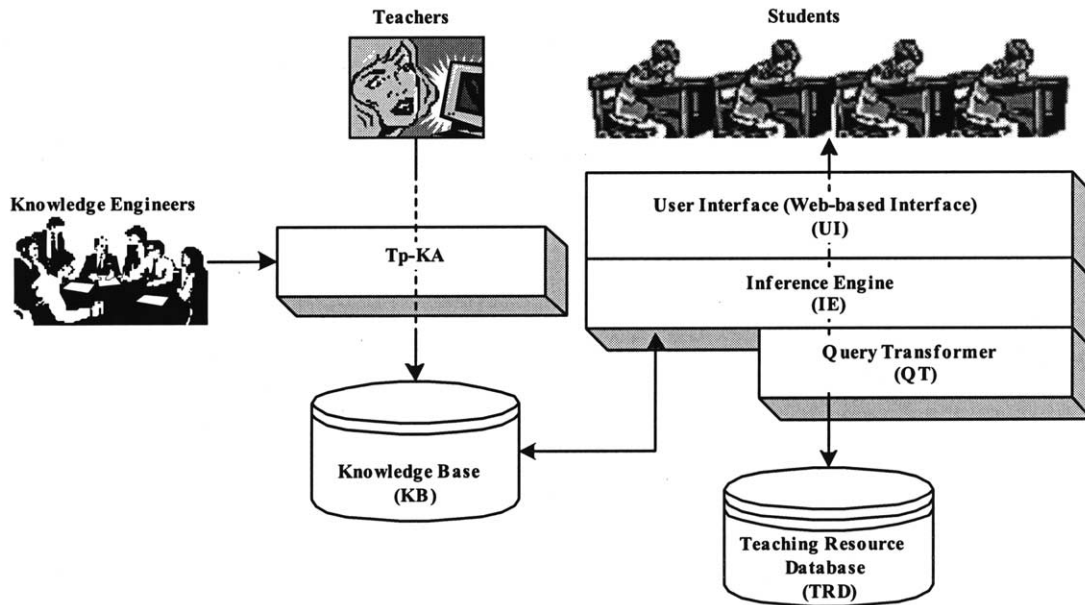


Fig. 3. The architecture of CAL-ES.

evaluation results. The reasoning processes of CAL-ES are modeling those of human experts in solving problems, which can be considered as the thinking processes of teachers on how to teach the students in accordance with their learning aptitudes. Thus, CAL-ES, which stores the rule-format teaching strategy to guide the learning processes of students, is designed as a kind of rule-based expert system.

Because teaching strategy is a set of rules, the concept of knowledge object, which is chunk of knowledge, is used to represent the knowledge of CAL-ES. The teaching strategy for different instructional objectives can be modularized, and some directed links existing among the different knowledge objects form the knowledge hierarchy can be considered as the learning map.

### 3.1. Architecture

Fig. 3 shows the architecture of CAL-ES composed of Tp-KA module, User Interface (UI), Inference Engine (IE), Query Transformer (QT), KB, and Teaching Resource Database (TRD).

UI is a web-based CAL system, which can provide proper teaching materials for students according to their learning aptitudes and evaluation results. KB contains learning map and teaching strategy that are stored as knowledge hierarchy and the relevant knowledge of KB. TRD stores the elements of teaching materials, the teaching objects. IE uses forward inference algorithm for inferring, which teaching objects are best fit for students according to their learning aptitudes and evaluation results of students. Then, QT is triggered to retrieve the required teaching objects from TRD, combine and organize the teaching objects into teaching materials, and display the teaching materials to students through UI. In

the following sections, QT, TRD and Tp-KA will be described more in detail.

### 3.2. Query Transformer and Teaching Resource Database

TRD stores the attributes and index of teaching objects, which are the set of XML-based documents, for managing and retrieving these teaching objects. The Document Type Definition (DTD) describes the structure of XML-based documents. It also includes the information about what elements must be presented, what their attributes are, and how they can be structured with links between each other. The corresponding DTD of XML-based document is shown as follows.

```

<!DOCTYPE DocumentObject [
  <!ELEMENT DocumentObject (DOIndex, CombineSet,
    Relationship)>
  <!ELEMENT DOIndex (#PCDATA)>
  <!ELEMENT CombineSet (#PCDATA)>
    <!ATTLIST CombineSet Member NMTOKENS
      #REQUIRED>
  <!ELEMENT Relationship (#PCDATA)>
    <!ATTLIST Relationship Link NMTOKENS
      #IMPLIED>
]

```

The `<DocumentObject>` tag consists of `<DOIndex>`, `<CombineSet>`, and `<Relationship>` tags. The `<DOIndex>` tag is the index of the document object. If this document is a combination of several other document objects, all indexes of these document objects are indicated in the `Member` attribute of the `<CombineSet>` tag as several tokens separated by space token, and a new index for the new combined

document object is indicated in the  $\langle DOIndex \rangle$  tag. If the document object has a directed link with another document object, the index of this document object is indicated in the  $Link$  attribute of the  $\langle Relationship \rangle$  tag.

To illustrate the definition and constructing operations of teaching materials, we use the following six teaching objects about the fundamental trigonometric function in Examples 3.1–3.4.

- $D_1$ : The teaching object of sine and cosine definition.
- $D_2$ : The teaching object of tangent and cotangent definition.
- $D_3$ : The teaching object of secant and cosecant definition.
- $D_4$ : The teaching object of basic Example 1.
- $D_5$ : The teaching object of basic Example 2.
- $D_6$ : The reference data.

**Example 3.1.** The teaching object  $D_1$ , mentioned earlier, can be defined as follows according to the DTD of teaching object.

```

<!--The teaching object of sine and cosine definition-->
<DocumentObject>
  <DOIndex Member = "1">Definition of sine & cosine</DOIndex>
  <CombineSet Member = ""></CombineSet>
  <Relationship Link = ""></Relationship>
</DocumentObject>

```

When QT receives the inferring results of IE, it transforms the inferring results into standard SQL commands, and then retrieves the needed teaching objects from TRD. After retrieving the teaching objects, QT executes *Combine*, *Organize* and *Show* operations to construct the online teaching materials. The operations of QT and the algorithm for processing the teaching objects can be added, modified and improved according to practical demand for processing the XML-based documents. The following algorithm is proposed to construct the teaching materials.

#### QT Execution Algorithm

*Input*: Inferring results of IE.

*Output*: Online teaching materials showed for students learning.

*Step1*: According to the inference results, retrieve the required teaching objects from TRD.

*Step2*: Execute *Combine* operation to combine some teaching objects into one larger teaching object.

*Step3*: Execute *Organize* operation to establish the links among the teaching objects.

*Step4*: Execute *Show* operation to display the online teaching materials on UI for students learning.

The three operations of QT are defined as follows:

#### ● Combine

*Combine* operation is to join together two or more XML document objects into a new XML document object. It is expressed as  $Combine(Doc\_Index_{List}, Doc\_Index_{New})$ , where  $Doc\_Index_{List}$  indicates the token list of the index of XML document objects, in which tokens are separated by space token, and  $New\_Doc\_Index$  indicates the new index of combined XML document object.

**Example 3.2.** This example shows how the three teaching objects  $D_1$ ,  $D_2$  and  $D_3$  are combined into the teaching object  $D_7$ , which are the teaching materials about definition of fundamental trigonometric function.

```

<!--The command is Combine(1 2 3, 7)-->
<!--The teaching object of trigonometric function definition-->
<DocumentObject>
  <DOIndex Member = "7">Definition</DOIndex>
  <CombineSet Member = "1 2 3">
    1. Definition of sine & cosine,
    2. Definition of tangent & cotangent,
    3. Definition of secant & cosecant,
  </CombineSet>
  <Relationship Link = ""></Relationship>
</DocumentObject>

```

#### ● Organize

*Organize* operation is to arrange two or more XML document objects into online teaching materials by establishing the directed links among the teaching objects. It is expressed as  $Organize(Ancestor\_Index, Descendant\_Index)$ , where  $Ancestor\_Index$  and  $Descendant\_Index$  indicates the index of the ancestor and the descendant document object.

**Example 3.3.** The directed link between the teaching objects  $D_7$  and  $D_4$  is constructed by using *Organize* operator, where the  $D_7$  is the ancestor and the  $D_4$  is the descendant.

```

<!--The command is Organize(7, 4)-->
<!--The relationship between the definition and basic example-->
<DocumentObject>
  <DOIndex Member = "7">Definition</DOIndex>
  <CombineSet Member = "1 2 3">
    1. Definition of sine & cosine,
    2. Definition of tangent & cotangent,
    3. Definition of secant & cosecant,
  </CombineSet>
  <Relationship Link = "4">Basic Example 1</Relationship>
</DocumentObject>

```

#### ● Show

*Show* operation is to display the XML document objects

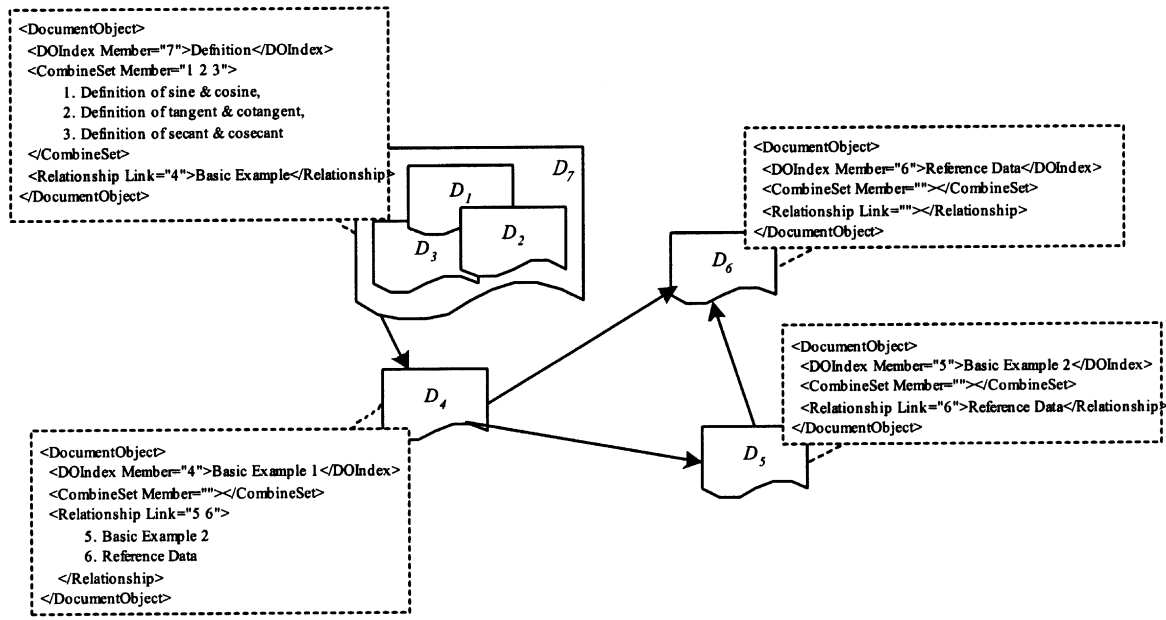


Fig. 4. The teaching material after executing the above sample rule.

in UI, i.e. browser, by using eXtensible StyleSheet Language (XSL). It is expressed as  $Show(Doc\_Index)$ , where  $Doc\_Index$  indicates the index of XML document teaching materials.

**Example 3.4.** After IE inference processes, assuming the condition part of the following sample rule is satisfied, the system executes the algorithm of QT to construct the online teaching materials. The result is shown in Fig. 4.

if condition part is satisfied, then  
 Combine(1 2 3, 7),  
 Organize(7, 4), Organize(4, 5), Organize(4, 6), Organize(5, 6), and  
 Show(7)

#### 4. Two-phase Knowledge Acquisition

Knowledge acquisition is a difficult, but important issue in constructing an expert system, especially for those, who are not familiar with expert system and computer technologies. In CAL-ES, the learning map is stored in the KB as knowledge hierarchy and the teaching strategy as knowledge. However, learning map cannot be shown to teachers easily, since the knowledge representation of CAL-ES is rule-format. That makes it difficult for teachers to know the structure of overall teaching materials. So, it is also difficult for teachers to refine teaching strategy and reorganize the structure of teaching materials. Thus, the concept of ontology is applied here to show the knowledge hierarchy of teachers' teaching strategy. Since knowledge engineers need a systematical knowledge

acquisition to interview the teachers to transfer their teaching strategy into rule-format, it seems to be necessary to provide a systematical and effective knowledge acquisition for teachers and knowledge engineers in CAL-ES.

As shown in Fig. 5, Tp-KA acquires the knowledge hierarchy and constructs the hierarchical grids in the first phase, and then transforms the knowledge hierarchy into maintainable and traceable format and extracts guiding rules from the hierarchical grids in the second phase. The algorithm of Tp-KA is as follows:

##### *Tp-KA Algorithm*

Input: The teaching domain know-how.

Output: The KB containing the learning map and the guiding rules.

##### Phase1

Step1: Execute *Constructing Knowledge Hierarchy (CKH) Algorithm* to construct the knowledge hierarchy.

Step2: Execute *Hierarchical Repertory Grids Analysis (H-RGA) Algorithm* to construct and fill up the hierarchical repertory grids according to the knowledge hierarchy constructed in Step1.

##### Phase2

Step1: Execute *Lattice to Tree (L2T) Algorithm* to transform the ontological lattice into ontological tree.

Step2: Execute *Embedded Meaning Capturing and Uncertainty Deciding (EMCUD) Algorithm* to extract guiding rules from the hierarchical repertory grids.

Step3: Store the ontological tree and the meaning-embedded rules into KB of CAL-ES.

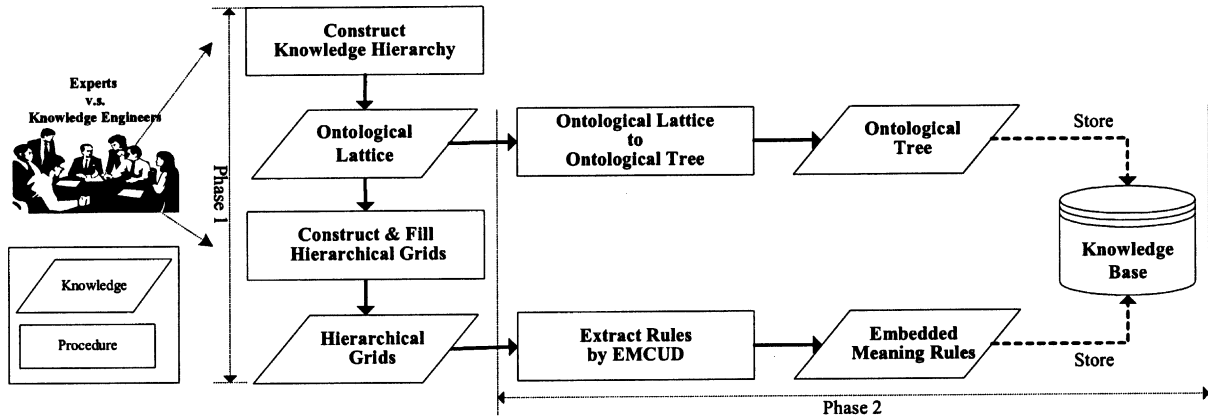


Fig. 5. The flow of Tp-KA.

In the first phase, by interviewing the senior teachers and educational experts, knowledge engineers can construct the knowledge hierarchy in lattice structure (Venter, Oosthuizen, & Roos, 1997), called ontological lattice. And then the hierarchical grids can be constructed and filled up according to the obtained knowledge hierarchy. In the second phase, the ontological lattice is transformed into tree structure, called ontological tree, for teachers to trace and refine the learning map easily, since the information of knowledge hierarchy is easier to be understood in tree structure than in lattice structure. In addition, the guiding rules with embedded meanings, the teaching strategy, can be extracted from the hierarchical grids obtained in the first phase. Finally, the KB containing the ontological tree and the meaning-embedded rules can be obtained in the end of Tp-KA.

4.1. Ontology representation by RDF

As we know, Resource Description Framework (RDF) is a resource description manner recommended by W3C. The major difference between RDF and Extensible Markup

Language (XML) is that RDF has a unique representation, but XML may have various representations for the same object. This unique property is useful for depicting the domain-specific ontology, and for exchanging the information of ontology. For example, RDF is used to depict the framework of the semantic web, which can be considered as a kind of ontology (Decker, Mitra, & Melnik, 2000).

Fig. 6 shows the RDF schema, which is used to depict the ontological lattice and the ontological tree of CAL-ES. In RDF, *rdfs:Class* indicates the object is a class, *rdfs:Resource*, which indicates the resource, is the subclass of *rdfs:Class*, and the relationship between superclass and subclass is indicated as *rdfs:subClassOf*. *rdfs:Property*, which indicates the property of some class, is also the subclass of *rdfs:Class*, with the range and the domain. If some object is a type of class or property, then this relationship is indicated as *rdf:type*. The relationship between the object, which is a type of property, and its range is indicated as *rdfs:range*, and the relationship between this object and its domain is indicated as *rdfs:domain*.

According to the RDF schema shown in Fig. 6, *sw:Ontology*, which is a type of class and is a subclass of

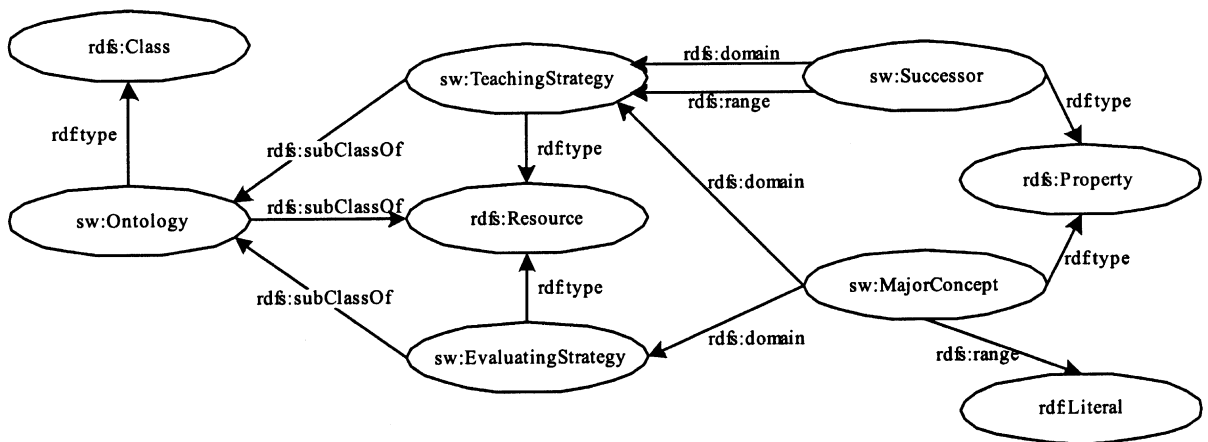


Fig. 6. The RDF Schema in CAL-ES.

resource, contains two subclasses, *sw:TeachingStrategy* and *sw:EvaluationStrategy*. These two are the type of resources and indicate the teaching strategy and evaluation strategy, e.g. guiding rules or evaluation rules created by teacher. In *sw:TeachingStrategy*, there are two properties, *sw:Successor* and *sw:MajorConcept*. *sw:Successor* indicates the successor of the node on the ontology, and thus the range and domain of *sw:Successor* are both *sw:TeachingStrategy*. *sw:MajorConcept* indicates the major concept of the teaching strategy or evaluation strategy, and thus the range is the literal defined as *rdf:literal*, and the domain is *sw:TeachingStrategy* or *sw:EvaluationStrategy*.

#### 4.2. Knowledge construction in the first phase

In most cases, when teachers guide students to work on the teaching materials, the students may think of some problems, which need to be solved by teachers. Different teachers may deal with these problems by different strategies, but some of these strategies will have something in common, which can be divided into some chunks of knowledge (Tsai, Tseng, & Wu, 1999), called knowledge objects, and modularized to be reused by teachers for guiding students in learning. It is useful to manage these teaching strategies. These related knowledge objects are connected by some directed links to show the causal knowledge.

Here, we need to model the knowledge about teaching strategy of teachers in an acyclic conceptual map or semantic network, since the directed links among knowledge objects do not contain any cycle when inferring, if do, will cause the infinite inferring problem. However, the general conceptual map or semantic network does not exclude the existence of cyclic relationships. Thus, we adopted ontological lattice, a kind of Directed Acyclic Graph (DAG), where the nodes indicate the knowledge objects and the directed links among the nodes indicate the causal knowledge.

To construct the ontological lattice for some problem domains, cooperation between domain experts and knowledge engineers is required. The following CKH Algorithm can guide the constructing process of knowledge hierarchy.

##### CKH Algorithm

Input: The teaching domain know-how.

Output: The learning map in ontological lattice format.

Step1: List all elementary knowledge objects according to the instructional objects of teaching materials.

Step2: While the knowledge hierarchy is not completed, ask the domain experts to proceed the following substeps:

Step2.1: List the meta-knowledge objects based on some of built knowledge objects.

Step2.2: Establish the directed links among the meta-knowledge objects and the built knowledge objects.

**Example 4.1.** In this example, we want to construct the ontological lattice of teaching materials by giving funda-

mental trigonometric function as an example. According to CKH Algorithm, the following 12 knowledge objects are listed in the sequence. First, list (*KO*<sub>10</sub>, *KO*<sub>11</sub>, *KO*<sub>12</sub>), second, list (*KO*<sub>6</sub>, *KO*<sub>7</sub>, *KO*<sub>8</sub>), third, list (*KO*<sub>2</sub>, *KO*<sub>3</sub>, *KO*<sub>4</sub>, *KO*<sub>5</sub>), and finally list (*KO*<sub>1</sub>).

Teaching strategy:

*KO*<sub>1</sub>: [*Definition I*] The definition course of acute angle trigonometric.

*KO*<sub>2</sub>: [*Formula I*] The formula course of acute-angle trigonometric.

*KO*<sub>3</sub>: [*Exercise I*] The exercises of fundamental definition and formula.

*KO*<sub>4</sub>: [*Formula I (Supplement)*] The supplementary course of formula.

*KO*<sub>5</sub>: [*Applied Example I*] The simple practical example of trigonometric.

*KO*<sub>6</sub>: [*Definition II*] The definition course of generalized trigonometric.

*KO*<sub>7</sub>: [*Formula II*] The formula course of law of sines and cosines.

*KO*<sub>8</sub>: [*Applied Example II*] The advanced practical example of triangulation.

*KO*<sub>9</sub>: [*End\_Course*] The test of teaching materials.

Evaluation strategy:

*KO*<sub>10</sub>: [*Learning Grade*] The grade evaluation of the quizzes following the course.

*KO*<sub>11</sub>: [*Network Learning Performance*] The performances evaluation of online learners.

*KO*<sub>12</sub>: [*Interview*] The interview evaluation between online tutor and student through an online chat room.

Fig. 7(a) shows the sample ontological lattice of the learning map of fundamental trigonometric function. Fig. 7(b), which contains the teaching materials of Definition I, Formula I, and Exercise I, is a part of learning map shown in Fig. 7(a). The elliptical knowledge objects are the teaching strategies worked out by teachers, and the rectangular knowledge objects are the evaluation strategies, which are the elementary knowledge objects.

Based upon the RDF schema defined in Section 4.1, the following shows how the node of the ontological lattice, *Definition I*, is described in RDF document.

```

<Ontology>
  <TeachingStrategy>
    <rdf:type resource = "Hyperlinks of teaching strategy about Definition I">
      <Successor>
        <rdf:type resource = "Hyperlinks of Formula I">
          </Successor>
          <MajorConcept>
            <rdf:type resource = "Definition I">
              </MajorConcept>
            </TeachingStrategy>
          </Ontology>

```

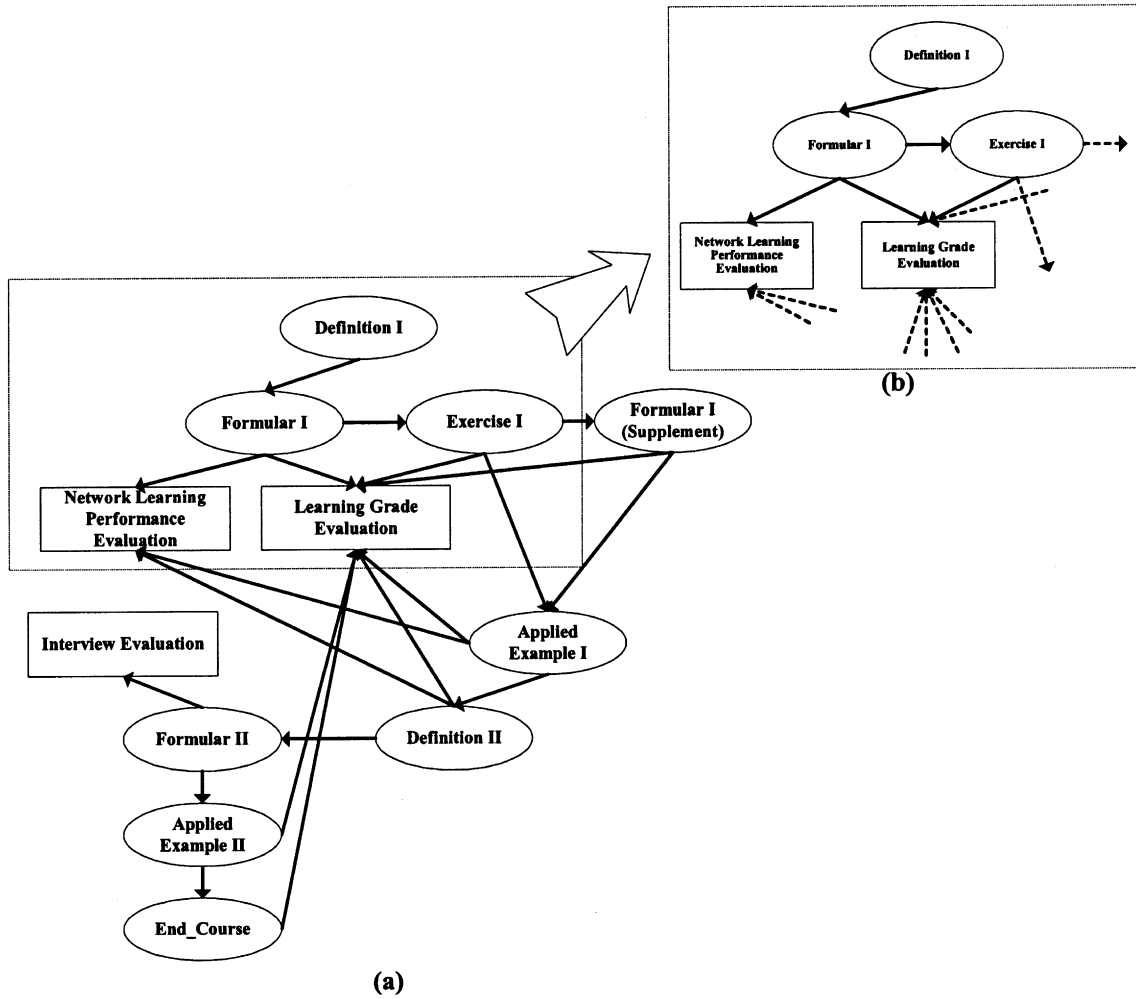


Fig. 7. Ontological lattice of teaching strategy about trigonometric function.

According to the obtained knowledge hierarchy, many kinds of knowledge acquisition approaches, grid-driven procedure, menu-driven procedure, table-driven procedure, and question/answer procedure (Boose, 1989; Boose & Gaines, 1988; Hwang & Tseng, 1990; Jeng & Chen, 1997; Marcus, 1988) may be applied to transform expertise into KB. In this paper, we use the multiple-level grid-driven procedure to acquire and generate the rule-format knowledge.

In the knowledge hierarchy in ontological lattice format as shown in Fig. 7(a), each node of the ontological lattice indicates a knowledge object in accordance with characteristics of problem domain. The ancestor node can be considered as the meta-knowledge of the descendant node. For each knowledge object, a repertory grid is used to describe its properties, with the objects as column headers and attributes as row headers of the grid. The hierarchical relationship between the ancestor grid and the descendant grid is determined by the multiple-level grids based upon the ontological lattice. Thus, the following H-RGA Algorithm is proposed to construct and fill up the hierarchical multiple grids according to the knowledge hierarchy built in the first stage of the first phase in the cooperation of experts with

knowledge engineers. Based upon H-RGA, the row and the column of each repertory grid consist of an attribute set and an object set, respectively, which are used to elicit knowledge from experts.

*H-RGA Algorithm*

- Input: The learning map in ontological lattice format.
- Output: The hierarchical grids.
- Step1: Visit the lattice according to *Depth First Search* algorithm.
- Step2: Derive objects of grid from the experts according to the major concept of the visited node.
- Step3: Elicit attributes from the experts, in which some attributes are corresponding with the child node of the ontological lattice and can be used to build the directed link between two nodes.
- Step4: Fill in the values of the derived pair of [object, attribute] of the grid.
- Step5: If there exists any unvisited node, then go to Step1.
- Step6: Stop.

Step2 and Step3 are to construct the multiple hierarchical



Table 1  
An example of Definition I grid

	Definition	Basic_Example_1	Basic_Example_2	Next_Section
Prerequisite	X	Definition	Definition	Definition
Major Concept	Trigonometric Definition	Trigonometric Example	Trigonometric Example	Formula
Difficult Level	1	1	2	0

grids according to the knowledge hierarchy of problem domain. Step4 is to fill in the value of the pair of [object, attribute] of the grids.

**Example 4.2.** For the ontological lattice shown in Fig. 7(b), the following five grids are constructed by H-RGA Algorithm.

The *Definition I* grid as shown in Table 1 contains four objects: Definition, Basic\_Example\_1, Basic\_Example\_2 and Next\_Section, and three attributes: Prerequisite, Major Concept and Difficult Level.

The *Formula I* grid as shown in Table 2 contains four objects: Basic\_Formula, Sum\_of\_Product, Last\_Section and Next\_Section, and five attributes: Prerequisite, Major Concept, Difficult Level, Learning Grade and Network Learning Performance. This grid and Learning Grade grid are related to each other by Learning Grade attribute, and the Network Learning Performance grid by Network Learning Performance attribute.

The *Exercise I* grid as shown in Table 3 contains five objects: Definition, Formula, Integration, Next\_Section\_1 and Next\_Section\_2, and four attributes: Prerequisite, Major Concept, Difficult Level and Learning Grade. This grid and Learning Grade grid are related to each other by Learning Grade attribute.

The *Learning Grade Evaluation* grid as shown in Table 4 contains five objects: Poor, Not Good, Average, Good and Excellent, and three attributes: Upper Bound, Lower Bound and Understanding Level.

The *Network Learning Performance Evaluation* grid as shown in Table 5 contains three objects: Non-Active, Average and Active, and four attributes: Interaction, Effective Learning Time, Effective Quizzing Time and Session Time.

#### 4.3. Knowledge hierarchy transformation and knowledge extraction in the second phase

The second phase contains two procedures: L2T Algorithm and EMCUD Algorithm (Hwang & Tseng, 1990).

Table 2  
An example of Formula I grid

	Basic_Formula	Sum_of_Product	Last_Section	Next_Section
Prerequisite	Definition	Basic_Formula	Definition	Formula
Major Concept	Trigonometric Formula	Trigonometric Formula	Trigonometric Formula	Exercise
Difficult Level	1	1	0	0
Learning Grade	Good	Excellent	Not Good	Good
Network Learning Performance	Average	Active	Non-Active	Average

L2T Algorithm is to transform the knowledge representation from ontological lattice into ontological tree. EMCUD Algorithm is to extract rules with embedded meaning from the hierarchically grids constructed in the first phase.

The existence of the cross-links among knowledge objects in lattice representation seems to have made it difficult for teachers to trace and refine the completed ontological lattice. On the contrary, the tree representation, which can show the hierarchical information in more easily and clearly way than lattice representation, is useful for teachers to verify and refine the learning map. Therefore, we propose an algorithm for transformation, L2T Algorithm, to transform the knowledge representation from ontological lattice into ontological tree.

#### L2T Algorithm

Input: The learning map in ontological lattice format.

Output: The learning map in ontological tree format.

Step1: Visit the lattice according to *Depth First Search* Algorithm, and set the visited node a *visited-flag*.

Step2: If the node has set the *visited-flag*, then duplicate the subtree with the visited node as root, and construct relationship from the parent node of the visited node to the new duplicated tree.

Step3: If there exists any unvisited node, then go to Step1.

Step4: Stop.

**Example 4.3.** The ontological lattice shown in Fig. 7(b) can be transformed by using L2T Algorithm into the ontological tree shown in Fig. 8.

It seems that the ontological tree needs enormous space to store the duplicated nodes. Generally speaking, the knowledge object does not need to duplicate itself, but only a change of its index. When user gives the conditions (learning aptitudes and evaluation results) for some inference results, the related knowledge objects are initialized and copied into the working memory. Then the conditions

Table 3  
An example of Exercise I grid

	Definition	Formula	Integration	Next_Section_1	Next_Section_2
Prerequisite	Definition	Formula	Definition, Formula and Exercise	Integration	Integration
Major Concept	Trigonometric Exercise	Trigonometric Exercise	Trigonometric Exercise	Applied Example	Formula I Supplement
Difficult Level	1	1	2	2	2
Learning Grade	Not Good	Average	Good	Average	Poor

given by users would be dynamically bound into these knowledge objects and starts the inference process. Therefore, logically, both ontological lattice and ontological tree need the same system resource, except the index file of knowledge object.

The other procedure of the second phase is to extract rules with embedded meaning from the hierarchical grids built in the first phase. For extracting the rules with embedded meaning, EMCUD knowledge acquisition (Hwang & Tseng, 1990) based on Personal Construct Theory (Kelly, 1955) is used. The tools or approaches to extract rules from the hierarchical grids can be substituted according to the characteristics of problem domain, but should be improved by following the concept of hierarchical grids. The detailed process of EMCUD is described in Appendix A.

Each meaning-embedded rule extracted by EMCUD has a Certainty Factor (CF), which is based upon a general repertory-grid-analysis method. For the inference process of each rule, the result may be affected by the rules in child knowledge objects. In other words, the CF of the inferred rule may be affected by the CFs of the rules in child knowledge objects. In this paper, a new formula of calculating CF based upon hierarchical grids is defined as follows.

Formula 1

$$CF'(R_i) = \frac{CF(R_i)}{n} \left( (n - m) + \sum_{R_j \in M} CF(R_j) \right),$$

where  $CF'(R_i)$  indicates the new CF of  $R_i$  based upon hierarchical grids,  $n$  indicates the number of attributes of  $R_i$ ,  $m$  indicates the number of attributes needed to infer another rules, denoted as  $R_j$ , to determine the attribute value,  $M$  is the set of  $R_j$ , and  $CF(R_j)$  indicates the CF of  $R_j$ .

In Formula 1,  $CF(R_i)/n$  means the average CF of each attribute of  $R_i$ ,  $n - m$  means the number of attributes, where the attribute value is given by actual state, and

Table 4  
An example of Learning Grade Evaluation grid

	Poor	Not Good	Average	Good	Excellent
Upper Bound	50	60	70	80	100
Lower Bound	0	51	61	71	81
Understanding Level	1	1	2	2	3

$\sum_{R_j \in M} CF(R_j)$  means sum of CFs of  $R_j$  in  $M$ , which are a set of rules to be inferred for  $m$  attribute values of  $R_i$ . The CF formula of hierarchical grids is the general model. When  $m$  is equal to 0 and  $M$  is the null set, the formula is specific as original CF formula shown in Appendix A.

**Example 4.4.** Assume there are two rules as followings

- $R_1$ : if  $att_1 = A$  and  $att_2 = B$  and  $att_3 = C$  then  $D$
- $R_2$ : if  $att_4 = E$  then  $att_2 = B$

The value of  $att_2$  of  $R_1$  needs to be determined by inferring the rule  $R_2$ .  $R_1$  and  $R_2$  are located at the ancestor and descendant knowledge objects, respectively. The facts of  $att_1$ ,  $att_3$ , and  $att_4$  are given as  $A$ ,  $C$ , and  $E$ . Assume the original  $CF(R_1)$  and  $CF(R_2)$  are equal to 0.9 and 0.8, respectively. Then, the new CF of  $R_1$  based upon hierarchical grids is as follows.

$$CF'(R_1) = \frac{0.9}{3} (3 - 1 + 0.8) = 0.84.$$

**Example 4.5.** Based upon the new CF calculating formula of hierarchically grids and original EMCUD, the rules with embedded meaning can be derived from the Definition I, Formula I, and Exercise I grids. For example, by interacting with teachers, the second column of Definition I grid may derive the following two meaning-embedded rules.

- /\*The teaching object  $D_1$  to  $D_7$  is as shown in Fig. 4.\*/*
- $R_1$  (CF = 1.0) if (Major Concept = Trigonometric Definition)  $\wedge$  (Difficult Level = 1)
- then Combine(1 2 3, 7),
- Organize(7, 4), Organize(4, 5), Organize(4, 6),
- Organize(5, 6), and
- Show(7)

Table 5  
An example of Network Learning Performance Evaluation grid

	Non-Active	Average	Active
Interaction	Infrequent	Frequent	Frequent
Effective Learning Time	Short	Average	Average
Effective Quizzing Time	X	Average	Short
Session Time	Long	Average	Short

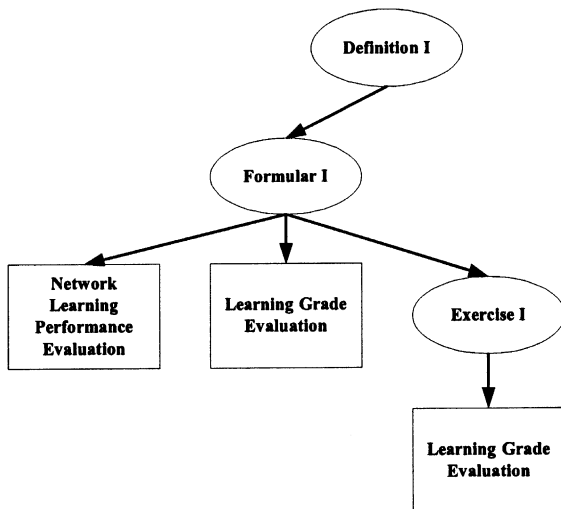


Fig. 8. The ontological tree transformed from Fig. 7(b).

/\*The teaching object  $D_8$  is the advanced explanations about the definition.\*/

$R_2$ (CF = 0.5) if (Major Concept = Trigonometric Definition)  $\wedge$   $\sim$ (Difficult Level = 1)  
 then Combine(1 2 3, 7),  
 Organize(7, 4), Organize(4, 5), Organize(4, 8),  
 Organize(5, 8), and  
 Show(7)

The second column of Exercise I grid may derive the following two meaning-embedded rules.

/\*The teaching object  $D_9$  is the basic exercise of trigonometric function definition.\*/

$R_3$ (CF = 1.0) if (Prerequisite = Definition)  $\wedge$  (Major Concept = Trigonometric Formula)  $\wedge$   
 (Difficult Level = 1)  $\wedge$  (Learning Grade = Not Good)  
 then Show(9).

/\*The teaching object  $D_{10}$  is the advanced exercise of trigonometric function definition.\*/

$R_4$ (CF = 0.8) if (Prerequisite = Definition)  $\wedge$  (Major Concept = Trigonometric Formula)  $\wedge$   
 (Difficult Level = 1)  $\wedge$   $\sim$ (Learning Grade = Not Good)  
 then Show(10).

The third column of Learning Grade Evaluation grid may derive the following two meaning-embedded rules.

$R_5$ (CF = 1.0) if (Learning Grade  $\leq$  60)  $\wedge$  (Learning Grade  $\geq$  51)  $\wedge$   
 (Understanding Level = 1)  
 then return(Not Good).

$R_6$ (CF = 0.6) if (Learning Grade  $\leq$  60)  $\wedge$  (Learning Grade  $\geq$  51)  $\wedge$   
 $\sim$  (Understanding Level = 1)

then return(Not Good).

The CF values of  $R_3$  and  $R_4$  are influenced by  $R_5$  or  $R_6$ . If the further inference of  $R_3$  is to infer  $R_6$ , then the new CF value of  $R_3$  is calculated as follows.

$$CF'(R_3) = \frac{1.0}{3}(3 - 1 + 0.6) \cong 0.87.$$

The Learning Grade attribute and the Network Learning Performance attribute of Formula I grid should proceed a further inference of the meaning-embedded rules derived from Learning Grade Evaluation grid and Network Learning Performance Evaluation grid, respectively. And the like, the Learning Grade attribute of Exercise I grid should infer the meaning-embedded rules derived from Learning Grade Evaluation grid. Therefore, teachers can construct the teaching strategy KB by filling up these grids and answering the questions asked by EMCUD. Moreover, according to the teaching strategy acquired from teachers, CAL-ES can show the teaching materials to students for learning adaptively.

For each inference process, the CF of rule can be used as an indication of the degree of supporting the guidance for students. In other words, the low CF of the rule means the low degree of executing the consequence part of this rule. The system can set the threshold to prune some inference results with low CF.

In CAL-ES, based upon rule-based technology, the teaching strategy can be expanded, modified, and reused easily. Senior teachers can construct the ontological lattice of teaching strategy, since they may know more about how to guide the students in learning processes, and about the hierarchy of teaching materials. Thus, the ontological lattice constructed by senior teachers can be used for teaching not only by themselves, but also by junior teachers to get to know about the skills and the hierarchy of teaching materials. The junior teachers can start the knowledge acquisition for their own teaching strategies from the second phase of Tp-KA. In other words, the ontological lattice constructed by senior teachers can be considered as the template of teaching strategies.

## 5. Implementation

As we know, online learning system is becoming popular and usually of importance for both formal and informal education. Right now, we are building an online learning system, CAL-ES, to be applied to the mathematics education of senior high schools in Taiwan. The prototype of CAL-ES is built under Windows 2000 Server, IIS web server, and ASP programming language. The CAL-ES now has three topics of teaching materials, fundamental concept of mathematics, fundamental trigonometric function, and fundamental algebra. There are about 100 teaching

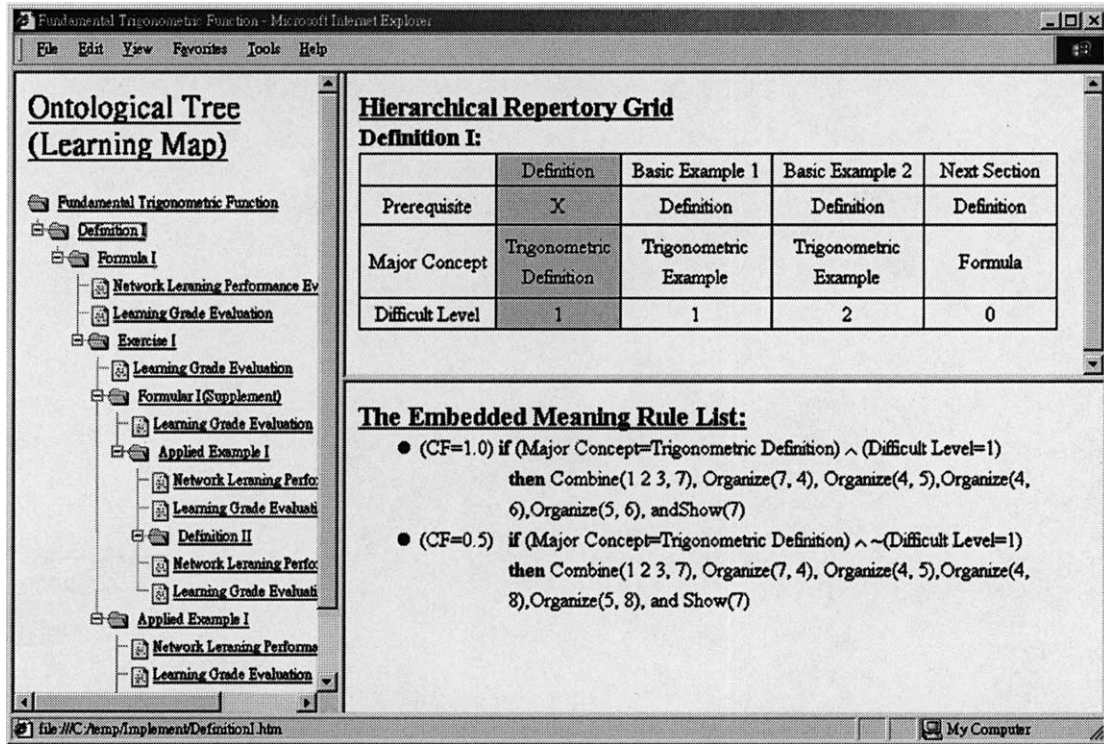


Fig. 9. The management window of CAL-ES.

objects and 300 rules as the teaching strategy in these materials. Students learn the online teaching materials and teachers edit and manage the learning map and the teaching strategy through Microsoft IE browser.

Base on the architecture of CAL-ES shown in Fig. 3 and the flow of Tp-KA shown in Figs. 5 and 9 show the management window of CAL-ES. The left frame shows the ontological tree of Fundamental Trigonometric Function, the top frame shows the hierarchical repertory grid of Definition I, and the bottom frame shows the list of the rules with embedded meaning corresponding with the grid shown in the top frame. By tracing the ontological tree (learning map), teachers can find out the interested grid and refine the corresponding teaching strategy, i.e. the rules with embedded meaning.

**6. Conclusion**

In this work, we proposed a CAL-ES and the corresponding knowledge acquisition, Tp-KA. CAL-ES can guide student to learn the online materials according to the learning map and the teaching strategy provided by teachers. The KB of CAL-ES stores the learning map and teaching strategy, which are constructed by using Tp-KA to interview senior teaching or educational experts, and can be reused by junior teachers to improve their teaching skills. Tp-KA contains two phases, in which the first phase consists of CKH Algorithm and H-RGA Algorithm, and the second phase consists of L2T Algorithm and EMCUD Algorithm.

Based upon the concept of hierarchical grids, the new formula of calculating CF is proposed in EMCUD Algorithm. About the knowledge representation of CAL-ES, the RDF is used to depict the ontological lattice and ontological tree, the knowledge hierarchy of CAL-ES. The prototype of CAL-ES was built on a basis of applying to mathematics of senior high school, which contains three topics of teaching materials, fundamental concept of mathematics, fundamental trigonometric function, and fundamental algebra.

Tp-KA is not only useful for teachers and knowledge engineers to construct the learning map and teaching strategy in CAL-ES, but also useful for the other domains, which exist the knowledge hierarchy and rule-based knowledge. In the future, we will develop some similarity algorithm to evaluate and compare two or more ontology, and develop the merging and separating algorithm for these ontological structures to assist teachers in refining the design of the online teaching materials.

**Acknowledgements**

This work was partially supported by the National Science Council of the Republic of China under Grant No. NSC89-2511-S-009-023 and NSC90-2521-S-009-004.

**Appendix A**

EMCUD (Hwang & Tseng, 1990) is able to derive the

embedded meanings of knowledge from the existing multiple repertory grids by interacting with experts and to guide experts to decide the certainty degree of each rule with embedded meaning. The Attribute-Ordering Table (AOT) (Hwang & Tseng, 1990), which is used to record the importance of each attribute to each object, is employed to capture the embedded meanings of the resulting grids. The value of each AOT entry, a pair of attribute and object, may be label 'X', 'D' or an integer. X means that there is no relationship between the attribute and the object. D means that the attribute dominates the object. The integer is used to represent the important degree of the attribute to the object. Larger integer implies the attribute is more important to the object.

According to AOT, the original rules generate some rules with embedded meaning, and the CF of each of them, which is between 0 and 1, could be determined to indicate the degree of supporting the inference result. A higher CF means the result is more reliable. In addition, after deriving the meaning-embedded rules, experts can add extra rules to illustrate some special situations in each knowledge object, since the repertory grid analysis does not cover all kinds of situations. The following steps describe the process of EMCUD.

#### Algorithm of EMCUD

Input: The hierarchical grids.

Output: The guiding rules with embedded meaning.

Step1: Build the corresponding AOT with each grid of the hierarchical multiple grids.

Step2: Generate the possible rules with embedded meaning.

Step3: Select the accepted rules with embedded meaning through the interaction with experts.

Step4: Generate the CF of the rules with embedded meaning.

Step5: Stop.

A Certainty-Sequence (CS) value represents the decreasing degree of certainty for a meaning-embedded rule, which is generated by negating some predicates of its original rule. CS of the rules with embedded meaning is calculated by using the following formula.

$$CS(R_i) = \text{SUM}(\text{AOT}[\text{attribute}_k, \text{object}_i]),$$

where  $\text{attribute}_k$  belongs to the attribute set of  $R_i$ , and object is the object of  $R_i$ .

**Example 1.** Assume the rules with embedded meaning listed below are generated from the rule,  $R_a$ : if  $\text{att}_1 = A$  and  $\text{att}_2 = B$  and  $\text{att}_3 = C$  then  $\text{goal} = \text{obj}_1$ , where the corresponding AOT entries are  $[2(\text{att}_1, \text{obj}_1), 1(\text{att}_2, \text{obj}_1), D(\text{att}_3, \text{obj}_1)]$ .

$$R_1: \text{if } \sim(\text{att}_1 = A) \text{ and } \text{att}_2 = B \text{ and } \text{att}_3 = C \text{ then } \text{goal} = \text{obj}_1$$

$$R_2: \text{if } \text{att}_1 = A \text{ and } \sim(\text{att}_2 = B) \text{ and } \text{att}_3 = C \text{ then } \text{goal} = \text{obj}_1$$

$$R_3: \text{if } \sim(\text{att}_1 = A) \text{ and } \sim(\text{att}_2 = B) \text{ and } \text{att}_3 = C \text{ then } \text{goal} = \text{obj}_1$$

$$\text{So the } CS(R_1) \text{ is } \text{AOT}[\text{att}_1, \text{obj}_1] + \text{AOT}[\text{att}_2, \text{obj}_1] = 2 + 1 = 3.$$

To decide the CF of the meaning-embedded rules, the upper bound of CF,  $UB(R_a)$  and the lower bound of CF,  $LB(R_a)$  must be determined first, where  $R_a$  is the original rule for these meaning-embedded rules.  $UB(R_a)$  is the CF of the original rule, since it is impossible for any meaning-embedded rule to have a greater CF than its original rule.  $LB(R_a)$  is determined by comparing the meaning-embedded rule with maximum CS value and the original rule. Four comparison levels suggested by the experts are 'confirm', 'strongly support', 'support' and 'may support', which are mapped to 1.0, 0.8, 0.6 and 0.4, respectively. CF of the meaning-embedded rule,  $R_i$ , is then generated by the following formula:

$$CF(R_i) = UB(R_a) - \{CS(R_i)/\text{MAX}(CS_i)\{UB(R_a) - LB(R_a)\}\},$$

where  $\text{MAX}(CS_i)$  is the maximum CS value of the meaning-embedded rules generated from the original  $R_a$ .

**Example 2.** For the example  $R_a$  and its generated meaning-embedded rules,  $R_1$ ,  $R_2$ , and  $R_3$ , the  $\text{MAX}(CS_i)$  for  $R_1$ ,  $R_2$ , and  $R_3$  is  $CS_3$ .

$Q_1$ : Which degree support  $R_a$ ? /\*To get the  $UB(R_a)$ \*/

Expert: Confirm. /\* $UB(R_a)$  is 1.0\*/

$Q_2$ : Which degree support  $R_3$ ? /\*To get the  $LB(R_a)$ \*/

Expert: May Support. /\* $LB(R_a)$  is 0.4\*/

Then the  $CF(R_1)$ ,  $CF(R_2)$ , and  $CF(R_3)$  are as follows.

$$CF(R_1) = 1.0 - (2/3)(1.0 - 0.4) = 0.6$$

$$CF(R_2) = 1.0 - (1/3)(1.0 - 0.4) = 0.8$$

$$CF(R_3) = 1.0 - (3/3)(1.0 - 0.4) = 0.4$$

## References

- Alessi, S. M., & Trollip, S. R. (1991). *Computer-based instruction: Methods and development*, (2nd ed.). Englewood Cliffs, NJ: Prentice-Hall.
- Beishuizen, J. J., & Stoutjesdijk, E. T. (1999). Study strategies in a computer-assisted study environment. *International Journal of Learning and Instruction*, 9, 281–301.
- Boose, J. H. (1989). A survey of knowledge acquisition techniques and tools. *Knowledge Acquisition*, 1, 3–37.
- Boose, J. H., & Gaines, B. R. (1988). *Knowledge acquisition tools for expert system*, San Diego, CA: Academic Press.

- Chou, C. (1996). A computer logging method for collecting use-reported inputs during formative evaluation of computer network-assisted distance learning. *Proceedings of ED-Media'96*.
- Decker, S., Mitra, R., & Melnik, S. (2000). Framework for the semantic web. *IEEE Internet Computing*, 4 (6), 68–73.
- Hwang, G. J. (1998). A tutoring strategy supporting system for distance learning on computer networks. *IEEE Transactions on Education*, 41 (4), 9.
- Hwang, G. J., & Tseng, S. S. (1990). EMCUD: A knowledge acquisition method, which captures embedded meanings under uncertainty. *International Journal of Man Machine Studies*, 33, 431–451.
- Jeng, B., & Chen, M. C. (1997). An object-oriented approach to knowledge acquisition. *Journal of Information Science and Engineering*, 13, 563–583.
- Kelly, G. A. (1955). *The psychology of personal constructs*, Vol. 1. New York: W.W. Norton.
- Marcus, S. (1988). *Automating knowledge acquisition for expert systems*. Norwell: Kluwer Academic Publishers.
- Oakley, B. (1996). A virtual classroom approach to teaching circuit analysis. *IEEE Transactions on Education*, 39 (3), 287–296.
- Su, G. H., Tseng, S. S., Tsai, C. J., & Cheng, J. R. (1999). Building an object-oriented and individualized learning environment on the WWW. *Proceedings of ICCE'99* (pp. 728–735).
- Su, G. H., Tseng, S. S., Tsai, C. J., & Cheng, J. R. (2000). Implementation of an object-oriented learning environment based on XML. *Proceedings of ICCE'00* (Vol. 2, pp. 1120–1127).
- Sun, C. T., & Chou, C. (1996). Experiencing CORAL: Design and implementation of distance cooperative learning. *IEEE Transactions on Education*, 39 (3), 357–366.
- Tsai, C. J., Tseng, S. S., & Wu, Y. C. (1999). A new architecture of object-oriented rule base management system. *Proceedings of International Conference on ToolsAsia'99*, Beijing, China.
- Venter, F. J., Oosthuizen, G. D., & Roos, J. D. (1997). Knowledge discovery in databases using lattices. *Expert Systems with Applications*, 13 (4), 259–264.
- Wu, X., & Cai, K. (2000). Knowledge object modeling. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 30 (2), 96–107.