PERGAMON

Expert Systems with Applications 25 (2003) 369–385

Expert Systems with Applications

www.elsevier.com/locate/eswa

Design and implementation of new object-oriented rule base management system

Yao Tsung Lin*, S.S. Tseng, Chi-Feng Tsai

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

Abstract

In this work, a knowledge model, new object-oriented rule model (NORM), is proposed based on the concept of learning and thinking behaviors of human. It provides high maintainability and reusability through the object-oriented concept. There are four basic relations between knowledge concepts defined in NORM: Reference, Extension-of, Trigger and Acquire. These relations are helpful in describing the cooperation of the different knowledge concepts. In addition, we describe how to construct and maintain a knowledge base under this model. A NORM knowledge modeled rule base system platform, DRAMA, is also introduced and applied in this paper. In order to illustrate the capability of NORM knowledge model, a learning management system using DRAMA to infer the knowledge for selecting appropriate learning content for different student is designed and implemented.

Keywords: Rule base; Object-oriented; Knowledge base; Computer-assisted learning

1. Introduction

An expert system, which emulates the decision-making ability of a human expert, is a knowledge-based system (KBS) and has been widely applied in several domains. In rule-based expert system, the domain knowledge of experts is implicit in rules (Buchanan & Shortliffe, 1984; Roesner, 1988). In order to reduce the effort for constructing a rule base, the rule base can be built up with some kind of rule-based knowledge representation language, such as CLIPS or existing rule-based shells.

The quality of a knowledge base system (KBS) can be evaluated on two criteria: performance and maintainability. The performance criterion, which includes run-time efficiency, functionality, and so on, has been the major target of systems evaluation in KBS. On the other hand, the maintenance criterion includes knowledge understandability, reusability and modifiability (Lee & O'Keefe, 1996). For traditional software engineering, the object-oriented technology, which has been proved to achieve these goals effectively, should be effective for KBS as well.

Designing KBS is more difficult than designing traditional software because the model of knowledge is more complex and it should be updated more frequently. Hence, a systematic maintenance mechanism is needed. In this work, we propose a knowledge model, new object-oriented rule model (NORM), based on the concept of incremental learning and inference of human. It provides high maintainability and reusability through the object-oriented concept. There are four basic relations between knowledge concepts defined in NORM: Reference, Extension-of, Trigger and Acquire. These relations are helpful in describing the cooperation of the different knowledge concepts. In addition, we describe how to construct and maintain a knowledge base under this model.

DRAMA, a NORM knowledge modeled rule base system platform, which is developed by Coretech Inc. and Knowledge and Data Engineering Laboratory, is then introduced in Section 2. And a learning management system (LMS) using DRAMA to infer the knowledge for selecting appropriate learning content for different students is designed and implemented, which is also introduced in this paper.

2. Related work

2.1. Object-oriented technology

The object-oriented technology provides a way to analyze problem effectively. Although this technology is

* Corresponding author. Tel./fax: +886-3571212156658.
E-mail addresses: gis88801@cis.nctu.edu.tw (Y.T. Lin), sstseng@cis.nctu.edu.tw (S.S. Tseng).

0957-4174/03/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0957-4174(03)00064-2

independent of programming language, various languages that adapt this idea have been designed, e.g. C++, Smalltalk and so on (Booch, 1991). With those language tools, users can more easily focus on the problem itself without paying too much attention to the language syntax. In addition, some properties of the object-oriented technology, e.g. encapsulation, inheritance, dynamic binding, may improve the maintainability, reusability, and adaptability of software (Rumbaugh & Blaha, 1991).

Most knowledge systems exploit the object-oriented technology. Based on the object-oriented concepts, knowledge can be divided into some classes. Only the required classes are loaded for inference. Thus, the requirement of system resources can be reduced and the performance can be improved.

The knowledge representation schemes with properties of object-oriented technology are effective on the maintainability of KBS. The property of encapsulation means that only the interface can be used to access the functions or data within a class. Similarly, there is an interface to access the rules or data that are encapsulated in a class of knowledge. Because the details of the knowledge are hidden, this feature can benefit managing a large knowledge base. Based on inheritance, KBS provides the reusability. Moreover, the ability of dynamic binding makes knowledge representation more flexible.

2.2. Rule base system

Rule is a natural knowledge representation, in the form of the 'IF … Then…' structure and rule base system (RBS) is popular for real applications among expert systems. RBS consists of two components, inference engine and assertions. The assertions can be divided into a set of facts and a set of rules that can be fired by patterns in facts. The inference engine, an interpreter of an RBS, uses an iterative match-select-act cycling model. In act phase of the cycle, a fired rule may modify or generate some facts.

CLIPS, 1998, one of the most successful expert system shell, which allows a knowledge base to be partitioned into modules, provides a feature called *defmodule*, and provides a more explicit method for controlling the execution of a system. Each module is able to inference sequentially and independently by inference engine. Different domain knowledge can be placed in different modules created by defmodule functions. Logically, related rules and facts can be collected into one module, which provides better maintenance and performance.

RBS has many advantages (Reichgelt, 1991). The first is naturalness of expression since experts rely on rules rather than on textbook knowledge. The second is modularity that permits RBS easy to construct, to debug, and to maintain. Restricted syntax and ability of explanation are also the advantages of RBS. Although RBS is powerful enough in many applications, it has several disadvantages in maintenance and construction, e.g. the weak ability of incremental construction of knowledge (Lee & O'Keefe, 1996).

Accordingly, many researches aim to integrate object-oriented and rule-based programming paradigms to take advantage of OO technology. There are two paradigms on the integration of objects and rules: incorporating rules into objects and embedding objects into rules. Knowledge objects are an integration of the object-oriented paradigm with logic rules (Wu, 2000). Furthermore, many rule-base tools, which cooperate with OO technology, are developed, e.g. COOL (CLIPS object-oriented language) (CLIPS, 1998).

2.3. Frame-based system

Frame-based system is also combined with the OO concept and it provides a structured representation of an object or a class of object with frame (Fikes & Kehler, 1985). Frame is structural representing knowledge about a limited aspect of the world. Information is stored by slot in a frame. Therefore, there is a hierarchical structure between frames to describe the relations of super-class and member-of.

The frame-based system has several advantages, e.g. proper balance between expressive power and efficiency, ability of describing new things using comparison with known things. There are still some drawbacks with frame-based KR languages. The first is perils of inheritance, e.g. the inheritor cannot determine which inherited from super-class are unnecessary. The second is expressive limitations, e.g. an instance frame corresponds to only one class frame and is distinct from all other entities (Reichgelt, 1991).

2.4. OORBMS and drama model

In Tsai and Tseng (2002) and Wu (1999), the object-oriented rule base management system (OORBMS) was proposed based on Drama Model (DM) for constructing a reusable, sharable, and modifiable knowledge base. The model manages rules under object-oriented paradigm. Although DM provides with advantages as reusability, maintainability, etc. the model has several disadvantages as follows:

1. DM tries using theater to describe the inference process and scenario to describe knowledge in a concept. However, most of knowledge concepts aren't plainly represented with this model.
2. DM lacks the ability of describing the relationship between different scenarios because it is restricted by the inheritance concept under OOP.
3. DM cannot perform inexact reasoning.

3. Knowledge model

Recently, knowledge management has become increasingly popular (Choi & Lee, 2002; van Elst & Abecker, 2002; Souza & de Ferreira, 2002). Knowledge or expertise of experts in numerous domain should be extracted, managed and reused to improve the performance and reduce human resources needed for difficult tasks. In most cases, knowledge needs to be constructed incrementally no matter what type the knowledge is, and hence maintainability for KBS is very important since KBS needs to be updated frequently. According to the above considerations, the following features are important for knowledge maintenance and management. Moreover, a simple and clear knowledge model with these features is proposed in this work

Modularity. Modular knowledge elements can be used sequentially and independently by inference engine. Modular knowledge representation benefits the maintenance of a KBS because of its localizing the effects of specifying flows of information between modules.

Abstraction. Abstraction is an approach that helps us deal with complexity by emphasizing relevant characteristics and suppressing other details. In most knowledge-based applications, the details of knowledge are not cared about.

Reusability. Knowledge reusability provides the facility of using original knowledge to build new knowledge. The property of inheritance is useful for knowledge reusing, yet a mechanism to reduce the knowledge conflict is needed.

Sharability. Sharable knowledge can be used to build up applications on various platforms. In another aspect, different KBS can also cooperate through the knowledge sharing.

Uncertainty reasoning. Uncertainty is an integral part of the world. If the ability of inexact reasoning is integrated into knowledge representation, the representation will be more natural (Salzgeber, 1993).

In order to increase the reusability, sharability and satisfy modularity and abstraction for knowledge base, a new model, NORM, is proposed for managing rules under object-oriented paradigm.

3.1. Aerial view

Various kinds of knowledge are defined in psychology (Gagné, 1984, 1985); however expert system mainly deals with the procedural and declarative knowledge excluding motor skill, attitude, etc. Knowledge is constructed by lots of concept blocks, for example, the concept about identifying a bird, a fish and so on. By building ontology to connect different concept, a complete conceptual knowledge model to solve a problem can be built. According to how people learn knowledge and ponder, three major kinds of relationships are defined between knowledge concepts. Thus, we define a clear knowledge framework and build a corresponding KBS.

3.1.1. Human learning

Learning is the most significant knowledge activity in our lives. A topic is required before people start the learning activity, for example, 'To learn how to identify a bird' is the topic before we learn what a bird is. Knowledge about the topic will be built after successfully studying about the topic as shown in Fig. 1.

3.1.2. Knowledge class

In this work, knowledge class (KC) is used to describe each concept. Learning is to study piece of knowledge, e.g. a domain concept, and to convert the knowledge into a KC. All the new knowledge is built upon the original knowledge according to educational psychology. In other words, learning is an activity to construct the relationship between different KC, as shown in following figure (Deese, 1965; Klausmeier, 1971)

3.1.2.1. Association. As we build domain knowledge inside our mind, association with existing knowledge is used to reduce the difficulty of learning. This kind of knowledge model is widely used in human knowledge processing. This relationship between domain concepts is seen as *reference*, i.e. to refer some existing knowledge Fig. 2.

3.1.2.2. Modification and extension. *Modification* of knowledge is also a similar activity. Efficient learning is absorbing the existing knowledge and experience from other people, but these knowledge contents may be modified or corrected according to user's experience or some new definitions of knowledge. On the other hand, *extension* is similar to *modification* except the knowledge can be not only overrided but also extended under extension relation.

3.1.3. Inferring

As shown in Fig. 3, when human gets facts through sensor, the facts will be inferred with a specific concept in a domain and other three concepts can be associated according to their relationships. However, people may not consider all relevant knowledge at the same time, since too much effort may be required to solve the problem. Some inference skills are widely used in human thoughts to improve the performance of knowledge inference.

3.1.3.1. Transference. Sometimes, a problem can be transformed to another problem according to some

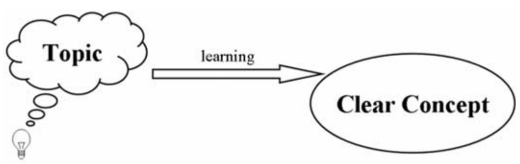

Fig. 1. The learning activity.

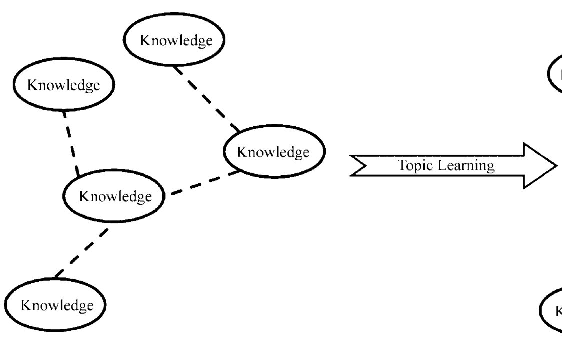

Fig. 2. Binding the new and existent knowledge in learning activity.

conditional judgment. For example, we may consider how to save water if we detect that climate will be drought. The transference is the activity of *triggering* thinking for another concept. On the other hand, a problem can be partitioned into some sub-problems when certain conditions are matched. For example, when a student is bad at mathematics, and then the knowledge of planning an extra mathematics course will be included; otherwise, the knowledge will not be included. This relation between two concepts is treated as *acquirement*.

3.1.3.2. Fact transform. In addition, the fact might have different name or meaning among concepts. For example, in different knowledge concepts, the fact, 'the temperature of the body', could be represented in adjective as 'fever' or in degrees centigrade as '39 °C'. So fact transformations may be attached to transference between two concepts.

3.2. New object-oriented rule model (NORM)

A knowledge model, NORM, is proposed according to the above ideas in this section. There are various subjects of domain knowledge in mind, but a knowledge system is often concerned with only one domain. However, a subject may contain various concepts.

Because rule is the natural and common representation of knowledge, rule is chosen to represent knowledge of each concept. As shown in Fig. 4, a rule base is defined as a container that deals with domain knowledge and contains various KCs; hence, related facts collected from real world can be used for inference within a knowledge class of corresponding concept.

3.2.1. Facts and fact-collection

The facts represent all kinds of appearance in real world and are used when inferring. During inference process, the rules use facts to obtain reasonable

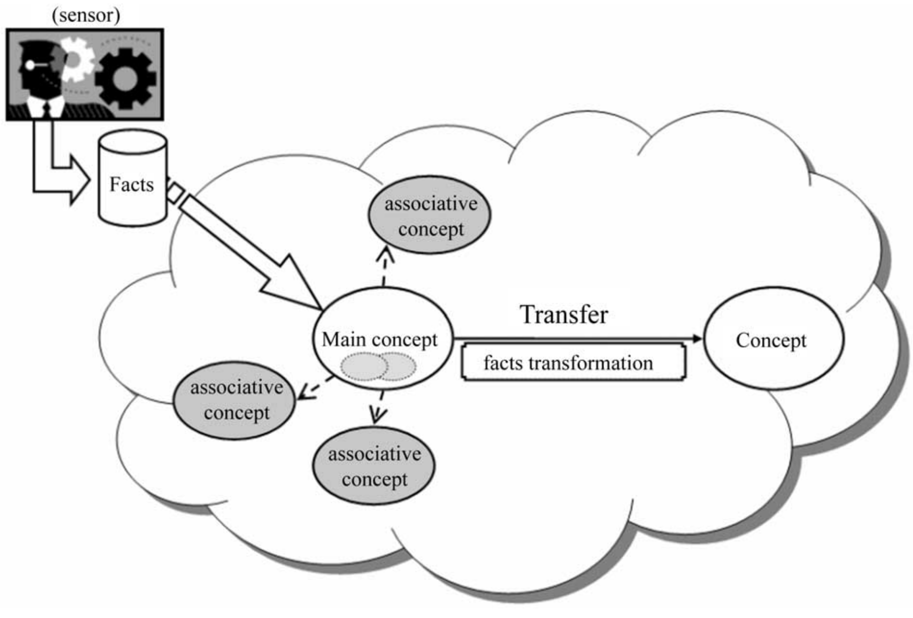

Fig. 3. The behavior of pondering over known information.

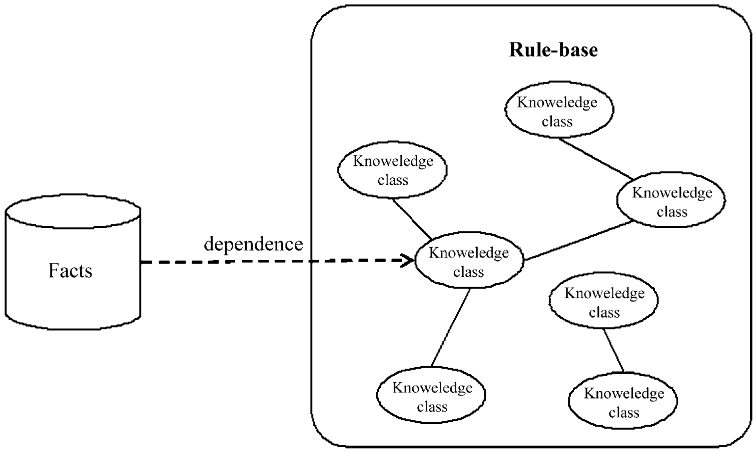

Fig. 4. New Object-oriented Rule Model (NORM).

conclusion. A fact consists of name, value, and possibility. A general expression for fact is as follows:

$$F : n = v(p)$$

where

n : the name of fact, which is used to identify a fact
v : value
p : possibility

The value of a fact could be any type including string, integer, float, date, Boolean value. If the value or type of a fact is unknown, it can be set as NULL. In order to support uncertainty reasoning, the possibility represents degree of belief of a fact. The possibility value is confined to the interval [0,1]. An activation of 1 is interpreted as 'highly positive', and zero as 'uncertain'.

3.2.1.1. Fact collection (FC). Fact collection (FC) is a set of facts and contains the meaningful facts for inferring. An FC performs as working memory and every inference process should own an independent FC. In other words, the FC is a temporary run-time component and will not be stored in a KBS.

3.2.2. Knowledge class

A KC represents a kind of concept. It consists of rules, relation with other KCs and fact declarations as shown in Fig. 5. After aggregating adequate facts in an FC, the facts could be inferred with a specific KC. During inferring, facts in an FC might be modified or generated. Finally, the conclusion could be drawn from the generated facts.

The fact declarations define which information is meaningful for a KC. There are two types of facts included in the facts declared, including the respondent facts and the required facts. The required facts are prerequisites for inferring under a concept, and on the other hand, the respondent facts are the interests of the conclusion. In other words, required fact is seen as input and the respondent fact as output.

A fact declaration consists of the name of fact and default value. If an FC does not contain some required facts before inferring, these facts should be initiated with the default value. On the other hand, if some respondent facts are not generated after inferring, these facts will be obtained with the default value as well. Thus, the fact declarations could be used to represent declarative knowledge.

3.2.3. Rule

A rule is the basic knowledge element in a rule-base system. The general formulation of a rule is shown as

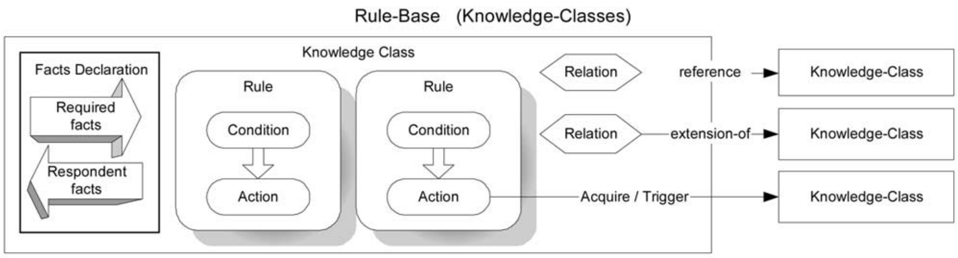

Fig. 5. The knowledge class in a Rule-Base.

follows:

R : IF c THEN a (CF $= \mu$), t, w

where

c: condition part of a rule
a : action part of a rule
μ : certainty factor of a rule
t : threshold
w : weight

3.2.3.1. Weight. The weight property allows the user to assign the priority to a rule. The rule with the highest priority will be fired first. The weight value should be an integer. If unspecified, the weight value for a rule defaults to zero.

3.2.3.2. Certainty-factor (CF). In order to support uncertainty reasoning, the certainty factor model, which was first used in the medical expert system MYCIN (Shortliffe & Buchanan, 1975), is adopted. In CF model, the certainty factor decides the degree of belief of a rule in matching phase and its value is confined to the interval $[-1,1]$.

3.2.3.3. Condition. A condition is a Boolean expression, which are the criteria for a piece of knowledge. Various operators can be used in the expression such as arithmetic operator, Boolean operator, etc. In rule matching phase, the result of the Boolean expression is evaluated, i.e. estimating the degree of confidence of a rule. The value is affected by several factors including logical operation and possibility of used facts. Finally, the degree of confidence of a rule has to be multiplied by CF of the rule (Giarratano & Riley, 1989). However, a rule is fired only when the degree exceeds a user-defined threshold t. For example,

F_1 : color = 'red' (0.9)
R_1 : IF color = 'red' THEN a, (CF = 0.8), 0.2, 0

Then the result of evaluating reliability is $09 \times 0.8 = 0.72$ and R_1 will be fired since 0.72 is larger than the threshold t, 0.2.

3.2.3.4. Action. An action represents the effect when the criterion of a rule is matched. The action of a rule should be one of following four types:

Assignment. This action is to assign value to fact or to generate a new fact. Before assigning value to a fact, the possibility of the new value is considered first, which is the result of the minimal possibilities of facts in condition expression multiplying the CF of the matched rule. The assignment is executed only if the new possibility given to assigned fact is equal or higher than current possibility of the fact, and the possibility of assigned fact will be modified as new possibility, too. For example, if the reliability of a rule is 0.8, and its action is to assign some value to a fact whose possibility is 0.9, the action will not perform. On the other hand, if the objective is a fact whose possibility is 0.7, the Assignment action will be completed successfully.

Trigger. The conditional transferences are divided into two kinds of actions: *Trigger* and *Acquire*. In Trigger relationship, it triggers another KC with current facts as knowledge transfer. In other words, the remnant knowledge in original KC should not be considered. During inferring, present inference process of the FC aborts, and a new inference process will start with the triggered KC.

Acquire. The second action of transference is Acquire that represents the acquirement relation. After Acquire process, the original inference process will continue and only facts predefined in the acquired KC will be carried back. At the same time, the possibility of these returned facts is multiplied by CF of the fired rule.

3.2.4. Relation

The relationships between KCs are divided into two kinds—dynamic and static. The relationships mentioned including Trigger relation and Acquire relation are dynamic because they are activated conditionally in the action part of a rule.

Two new relations, including *Reference* and *Extension-of*, will be defined as static relations. These two relations are designed according to the natural of building knowledge of human. Since a KC may refer several KCs, the topology of all KCs is a directed graph.

3.2.4.1. References. Reference is used to represent the associations between different concepts. Through the Reference relation, the knowledge contained in referred KC is regarded as the base knowledge and it will be taken into consideration together with the knowledge defined in the KC. On the other words, Reference can be thought as an unconditional acquire relation between KCs.

For example, as shown in Fig. 6, suppose we learn 'wild goose' via the some features of 'goose' and the property, flyable, of 'swallow'. Before considering whether the present facts indicate 'wild goose', the inference process first considers whether these facts conduct the property of 'flyable' under concept of 'swallow' and other properties under 'goose'. Thus, the initial facts could be automatically generated. Therefore, Reference relations should be declared between KC of 'wild goose' to KCs of 'goose' and 'swallow'.

3.2.4.2. Extension-of. As shown in Fig. 7, Extension-of is different from the Reference relation, the Extension-of relation makes the new KC to include all the knowledge contents of an existent KC. The activities of Extension-of relation include extension and modification. Therefore, it must support the overriding mechanism, including the overriding of fact and rule. For example, in Fig. 7, if a KC_B is extension of KC_A, and then KC_B will own respondent

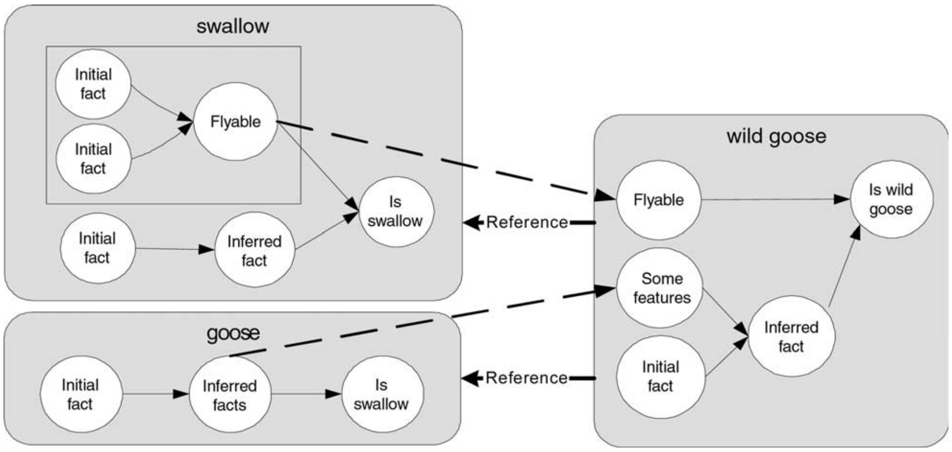

Fig. 6. A Reference relation example.

facts and required facts that KC_A owns. However, if there is a duplicate definition of fact in KC_B, the type and value of the fact will be based on the definition in KC_B. Overriding of rules in NORM is different from that of facts, which is defined as logical overriding. In logical overriding, if the rules in KC_A have the same action with KC_B, e.g. to assign value to the same fact, the action of KC_B will be taken instead of that of KC_A.

Finally, the relationships between KCs are not necessarily accurate and there may be some uncertainty of the fact declarations and the rule assertions in the relations. Relations can be asserted a certainty factor to reduce the degree of belief of default facts and rules in the referred KCs. The detail of this process will be discussed in Section 4.

3.2.5. Transformer

The transformer is used to transform the facts between two KCs, because the fact might be expressed in different measures. For example, the 'temperature' may be measured in Fahrenheit or Celsius for different knowledge concepts. Therefore, the transformers may be attached to the relations between KCs.

3.2.6. Rule-base

In this model, a Rule-Base (RB) records various knowledge concepts in a specific domain and each KC in the RB represents different concept of the domain knowledge.

In addition, RB is a unit of knowledge exchange and the meta-data of KCs supply relevant information for knowledge reuse, e.g. author, purpose, and so on.

3.2.6.1. Inferring. In cognitive structure of human (Collins & Quillian, 1969; Kintsch, 1970; Tulving, 1983; Tulving & Thomson, 1973), there is a complex mechanism to map perceived facts to the concept of long time memory, and use the knowledge of the concept for solving problems. However, the ideal mechanism can not be easily implemented. In NORM knowledge model, a KC that contains the control knowledge, which is the knowledge about considering which kind of knowledge should be used to solve problem, must be specified before using the knowledge in NORM. For example, in the knowledge system about medical diagnosis, a KC contains the control knowledge of determining which type of diagnosis KC, e.g. KC for Internal Medicine or Surgery, should be used.

3.2.7. Inference process

The inference process with the model is described as follows. The first step is to select a rule-base. Because a knowledge system cannot contain all types of domain knowledge, specifying a knowledge domain, e.g. internal medicine diagnosis, travel planning and so on, is necessary before inferring. The second step is to collect the facts and specify a KC containing the corresponding control knowledge for the problem to be solved. According to the specified KC, the inference engine will perform

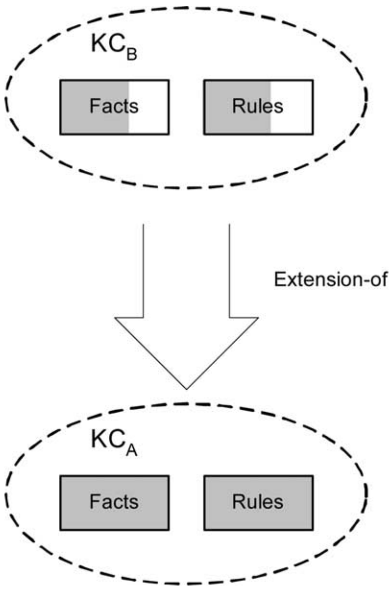

Fig. 7. The Reference relation and the Extension-of relation.

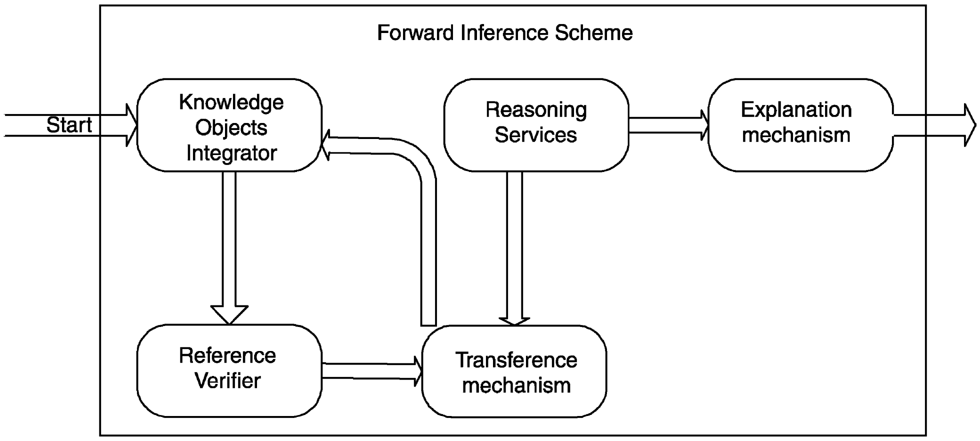

Fig. 8. The forward relation-based inference scheme.

the reasoning process. Finally, interesting information can be obtained from final fact value. Furthermore, the order of fired rules and causal relationship between those rules can be retrieved for explanation mechanism.

3.3. Relation-based inference mechanism

In order to deal with the various relationships under NORM, the relation-based inference mechanism is proposed. A forward relation-based inference mechanism shown in Fig. 8 includes following five modules.

3.3.1. Knowledge class integrator

This module integrates the rules and fact declarations through the Extension-of relations between KCs. Before inferring, it rewrites the action part of integrated rules and adjusts the certainty-factor value of these rules according to the Extension-of relation declaration. Similarly, it also combines the fact declarations of KCs.

This module also creates the relation tables about the interaction between rules and facts, including what facts are used in condition part of a rule and what facts or KCs are affected by the action part of a rule. The tables can help to increase the efficiency of the rule matching in reasoning.

3.3.2. Transference Mechanism

This mechanism mainly performs the Trigger or Acquire during reasoning process. An FC is KC-dependent to a KC if the FC is inferred with the KC. This module performs transference with changing the KC-dependence of an FC. In other words, it causes the FC to be KC-dependent to another KC, and restarts the inference process. As shown in Fig. 9, for Trigger action, the original inference process will be terminated. Unlike Trigger, the action of Acquire copies the current FC to begin a new inference process with the target KC. After the new process, facts are returned to original FC according to the fact declarations in the target KC, and the original inference process will continue.

3.3.2.1. Transformer. The transformer consists of two parts, *TO* and *FROM*, and performs in transference mechanism. Before the transference, the specific facts were assigned new value according to the TO part of a transformer declaration. If there is a fact that owns the same name as the assertion of *Source-Fact*, the fact will be replaced or removed before the new inference process. Besides, for the Acquire action, the values of facts will be responded, and the facts will be changed according to operations defined in the part of FROM. For example, in Fig. 10, while the action executes, the fact F of an FC is first converted into the fact C and then removed. After the action, the fact C will be transformed to F as well.

3.3.3. Reference verifier

This module deals with the Reference relations between KCs. When an inference process initiated, Reference

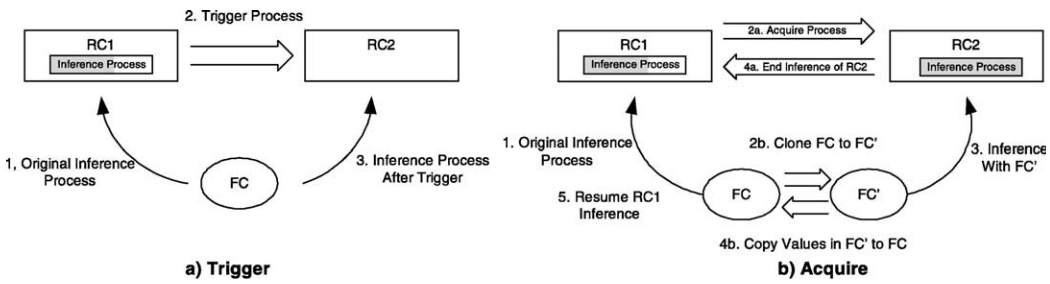

Fig. 9. The Trigger action and Acquire action.

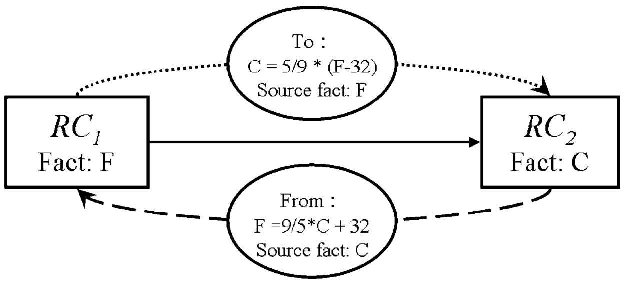

Fig. 10. A transformer example.

Verifier verifies the prerequisite of referenced KC and includes all the rules and facts of the KC through Reference relation. Included rules and facts will be used as a part of the knowledge to be processed during the inference process.

3.3.4. Reasoning service

This module is used to do the actual inference process within the rules and facts from previous mechanisms. The rules will be matched according to the given facts, rule actions except the transference actions will be taken, and new fact value is assigned or generated. All the above steps will be recorded for explaining the inference process by Explanation Mechanism.

3.3.5. Explanation mechanism

This module arranges conclusion in systematic form and provides the ability of explaining the conclusion. The conclusion is represented in three parts: the list of facts that are modified or generated during inferring, the cause relation between fired rules and facts, and the order of all fired rules. Thus, the inferring result can be explained.

4. Modeling a knowledge base

Modeling a knowledge base (Choi & Lee, 2002; Rosca & Wild, 2002) contains several processes, construction, maintenance, reuse and refinement. In the life cycle of KBS, KB maintenance and refinement repeat recurrently. In this section, the methodologies of modeling a knowledge base under NORM will be described.

4.1. Construction

The first process of modeling a knowledge base is construction, i.e. transforming the domain knowledge of experts into knowledge representation format of NORM. In this section, a construction procedure is proposed to construct the knowledge systematically. In knowledge base construction, it is assumed that no prior knowledge of the similar domain exists, and Extension-of relation will not be used in construction process. The construction procedure is divided into the following six steps.

1 Select a knowledge domain to be modeled. Before designing a knowledge base, the domain of the KB must be first selected. If a large system is built, the domain of the system may be divided into several sub-domains.

2 Identify concepts in the domain and model the concepts. This phase is to analyze what concepts are contained in one domain, similar to the use-case analysis in OOA/OOD. A concept in knowledge base is used to solve a problem as use-case.

In cognitive psychologist, the knowledge can be divided into three categories: declarative, procedural and strategic (Anderson, 1995; Glaser, 1987). Declarative knowledge is used to judge if the present facts correspond to things that the concept represents, and finally the result is obtained from the value of facts, e.g., deciding whether an entity is a bird according to the facts about its features.

Procedural knowledge contains the discrete steps or actions to be taken and the available alternatives to perform a given task. Thus a procedural concept is based on the visible facts to proceed planning for the concept, e.g. how to fix a bicycle. A plan may be generated from this kind of knowledge to solve a problem.

Strategic knowledge is used to decide course of action and regards the interrelationships and interdependencies among concepts. Strategic knowledge consists of reasoning strategy and control rules (Kintsch, 1970). In NORM, the control rules decide which KC will be used.

Fig. 11 is an example to show the relations betweens the KCs containing one of the three types of knowledge. Procedural KC may acquire result inferring by other procedural or declarative KCs, or trigger a control KC. The Control KC can decide which KC will be used with existing facts. However, an inference process can start with any type of KC.

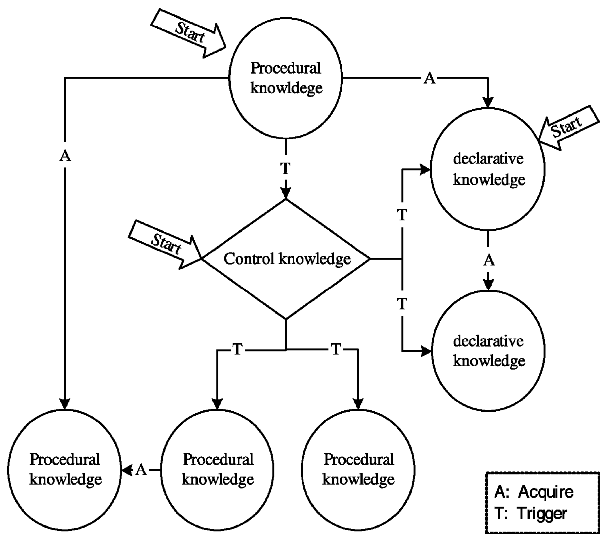

Fig. 11. The cooperation of KCs with different types of knowledge.

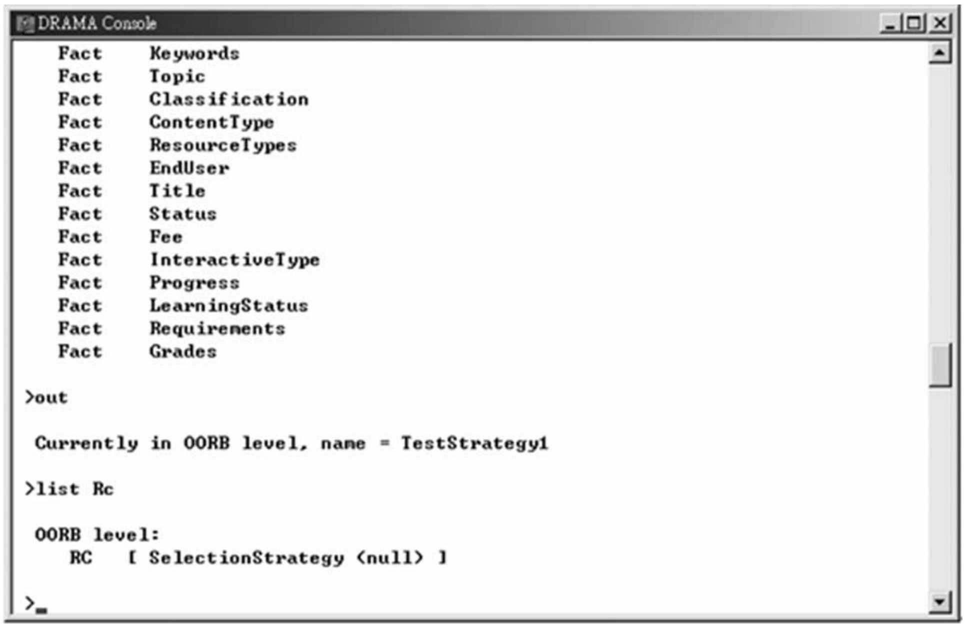

Fig. 12. DRAMA console.

In this step, the type of concept to be modeled must be decided, and each concept must be mutual exclusive from each other.

3 Identify the relationships among concepts. Next, according to the exclusive relation of concepts, the type of their knowledge relation in NORM model must be found according to following basis. The concept with generalization is defined as Reference relation; the concept with causal relationship is defined as Trigger relation; at last, through further analysis, the sub problem or sub concept can be defined as Acquire relation.

4 Identify the features of each concept. In this step, according to perception of experts, the features that affect each concept will be defined, and the facts in each concept will be used in designing corresponding KC Figs. 12–14.

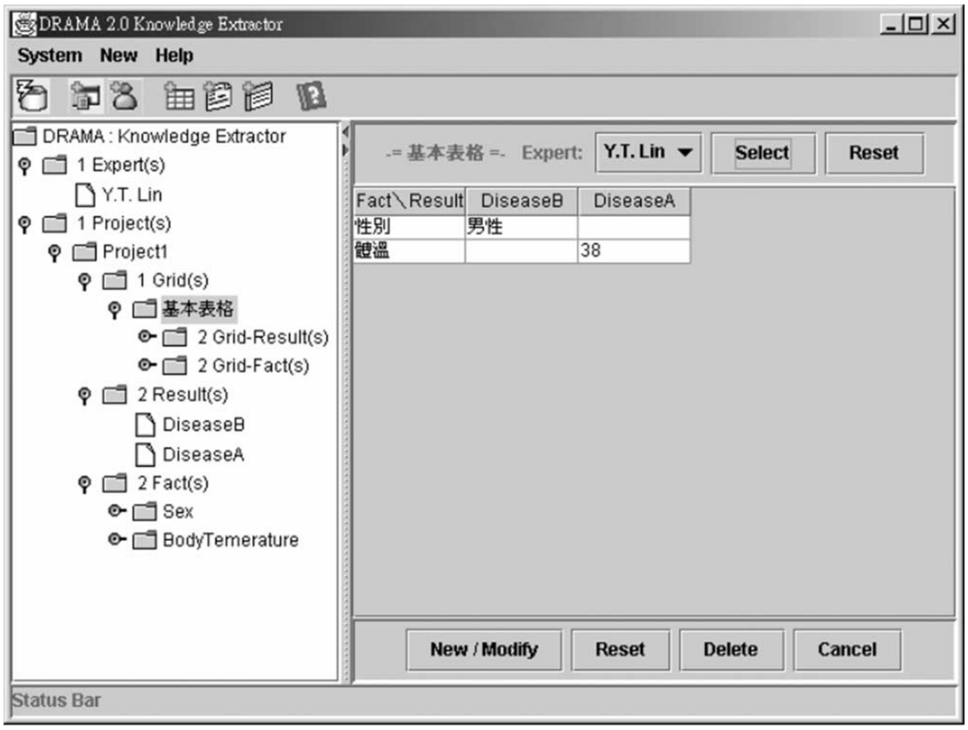

Fig. 13. DRAMA knowledge extractor.

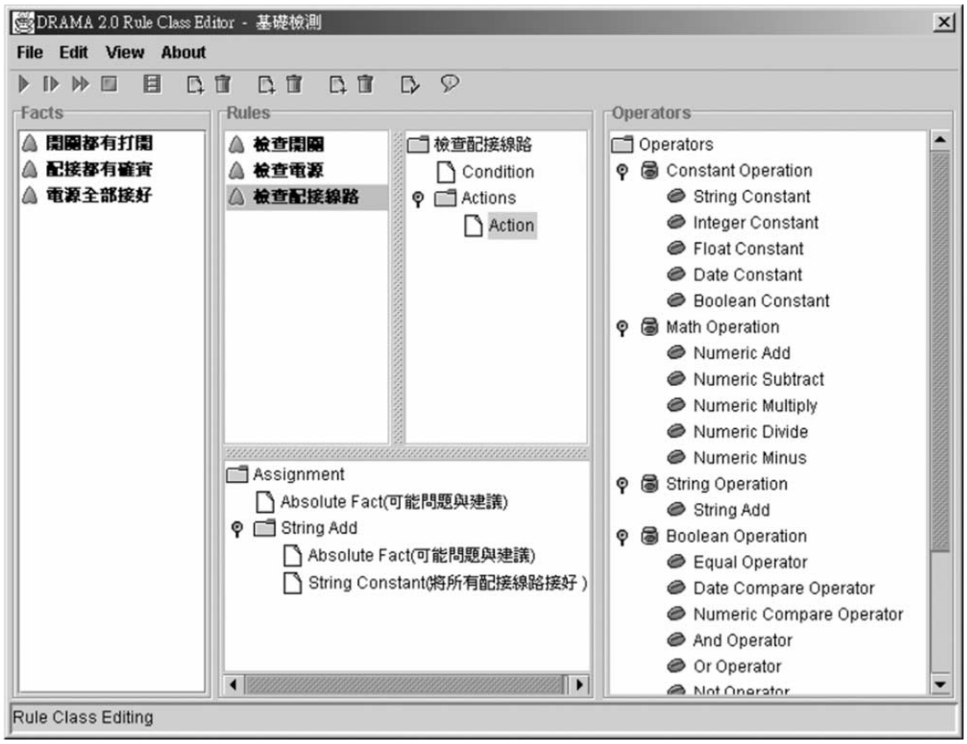

Fig. 14. DRAMA rule editor.

Facts can be divided into two categories, respondent facts and required facts. Respondent facts possess the function of output, which means all of the relevant features generated through inferring the basic information, can be categorized as respondent facts. On the other hand, all necessary basic information for inferring with KC is a type of required facts.

5 Design the transformer. When a KB is constructed from several KCs, the transformer may be needed between KCs to transform useful facts. A transformer should be designed if the format requirements of cognominal feature facts between two KCs are different. The transformer will be assigned to the relation between KCs except Extension-of relation.

6 Acquire knowledge of each concept. Because rules are chosen to represent knowledge of each concept in NORM, the knowledge should be transformed into rule form. According to the relations between KCs analyzed in previous steps, this step acquires the knowledge of experts about each KC. The acquisition process for one KC can rely on some developed KA methodology such as repertory grid. However, the rules dealing with Trigger and Acquire relation between KCs should be asserted.

In order to avoid redundant design of the rules, the knowledge of a KC can be acquired if the KC is the top of relationship hierarchy between KCs, i.e. it does not refer other KCs.

4.2. Maintenance and reuse

There are some differences between maintenance and reuse of existing knowledge in NORM. Maintenance means the modifier is the originator of a KBS, but reuse means that someone else uses existing KC and modifies it. Therefore, reusing an existing RB could be proceeded by building Extension-of relation.

Understanding an existing rule-base is the prerequisite to reuse or maintain it, which means user has to know the domain problem solved in the rule base, the concepts of KCs contained in the rule base, and the declarations of each fact in KCs. Thus, the process could be proceeded as follows.

1. Analyze the relationship of the new concept with original KCs. In order to add a new concept to a rule-base, the relationships between new concept and original KCs must be known. In most cases, Reference is used to describe the relation between two KCs, which cooperates to solve a problem. Extension-of may be used if one KC is a modification of another KC and they have similar concept or solve the similar problem.
2. Identify the facts of the new KC. According to the Extension-of or Reference relation, the key facts of new KC could be identified. In addition, the new concept may use features that are not declared in the referred KCs, and those feature facts should be declared in new KC.
3. Check conflict of fact definitions and design the transformers. The names of facts in two KCs should be unified. For example, if a KC use 'fever' to express a rise in the temperature of the body, the other KC should

not use 'pyrexia' to express the same concept. However, if needed, the transformer can be designed according to type of fact value and the meaning of facts.

4. 4 Acquire knowledge of the new concept. The step is similar to Step 6 in Section 4.1.

4.3. Refinement

Knowledge acquisition can be divided into two phases, initial phase and refinement phase, in which the initial knowledge base is refined to produce a high performance system (Ginsberg, Weiss, & Politakis, 1988; Kingston, 2001). In this phase, the knowledge base should be corrected through a debug process and the relationships between KCs may be refined, e.g. the common concept of KCs can be extracted into an independent KC.

5. System implementation

In this secstion, a rule base platform, DRAMA, designed based on NORM knowledge model is first introduced. Then, a CAL prototype system to solve the problem of selecting learning content using DRAMA is also introduced in this section.

5.1. DRAMA

DRAMA, a NORM based rule base platform, is a product of Coretech Inc (DRAMA, 2003), Taiwan, which is developed in cooperation with Knowledge and Data Engineering Laboratory (KDB Lab.) of National Chiao Tung University, Taiwan. DRAMA is implemented using Java, and it includes DRAMA Server, DRAMA Console, DRAMA Knowledge Extractor, DRAMA Rule Editor.

DRAMA Server is implemented to manage rule bases, which is used to contain and process knowledge, and provide rule base services. NORM-modeled knowledge can be contained in DRAMA Server and inferred according to user given facts. DRAMA Console is a command mode interface for user to access DRAMA Server.

DRAMA Knowledge Extractor is implemented by repertory grid mechanism (Hwang & Tseng, 1990; Tsai & Tseng, 2002), a knowledge acquisition mechanism, to extract and retrieve knowledge from experts. The extracted knowledge will be transformed into NORM rules which will be used in DRAMA Server.

For the knowledge already defined in rule format, DRAMA Rule Editor with a GUI interface is provided for editing NORM KC and rules. Differ from traditional rule base building tools, DRAMA Rule Editor is a user friendly GUI with drag and drop operations.

Also, Application Programming Interface (API) to access DRAMA server is also provided for developing DRAMA integrated systems. In the prototype system introduced in Section 5.2, the API is used to integrate DRAMA with SCORM (Sharable Content Object Reference Model) (Sharable Content Object Reference Model, 2003) compliant LMS.

5.2. Prototype system

In CAL systems and researches (Beishuizen & Stoutjesdijk, 1999; Chou, 1996), Adaptive Learning is an important issue to be solved, and selecting appropriate learning content for different students is an important feature in Adaptive Learning. For different students in different learning situations, teachers want to provide different learning content to students to improve their learning performance. Therefore, processing teaching strategy which contains the knowledge about selecting learning content is important in CAL systems. However, traditional computer technologies like database query, which only select information according to some criteria of data instead of considering all of the factors influence learning achievement, is not suitable for expressing the knowledge of teachers to select learning content. Hence, KBS is used in these systems for learning content selection purpose.

As shown in Fig. 15, a learning content selection system, which used to select appropriate learning content for students, consists of three components, including learning strategy, student profile/records, and learning object. Each of these components should be managed by a specific system. In order to create, store, reuse and manage learning content, a Learning content management system (LCMS) is required. KBS is required for managing and processing the learning strategies, and a teaching platform is also required for monitoring and recording students' behaviors as learning profile.

Students are always different according to their learning achievements and learning behaviors even they study the same learning content. In order to improve their learning performance, teachers should prepare different learning content for different students, for example, teachers should provide easier learning content for the student with lower learning achievement. However, it is tedious and time consuming for a teacher to prepare different learning content for all students, and a systematic and efficient mechanism to help selecting appropriate learning content for students is required. In our experiment, NORM knowledge model and DRAMA, is used in designing and implementing KBS for a CAL system to select appropriate learning content for learner, and solve the problem for teachers to prepare the learning content for different students.

In the following sections, the experiment including the usage of NORM rule base is introduced. First, the learning achievement of student and corresponding features is introduced, and then the meta data of the learning content, which contains the information and properties of learning content, is also designed. Finally, the platform to use

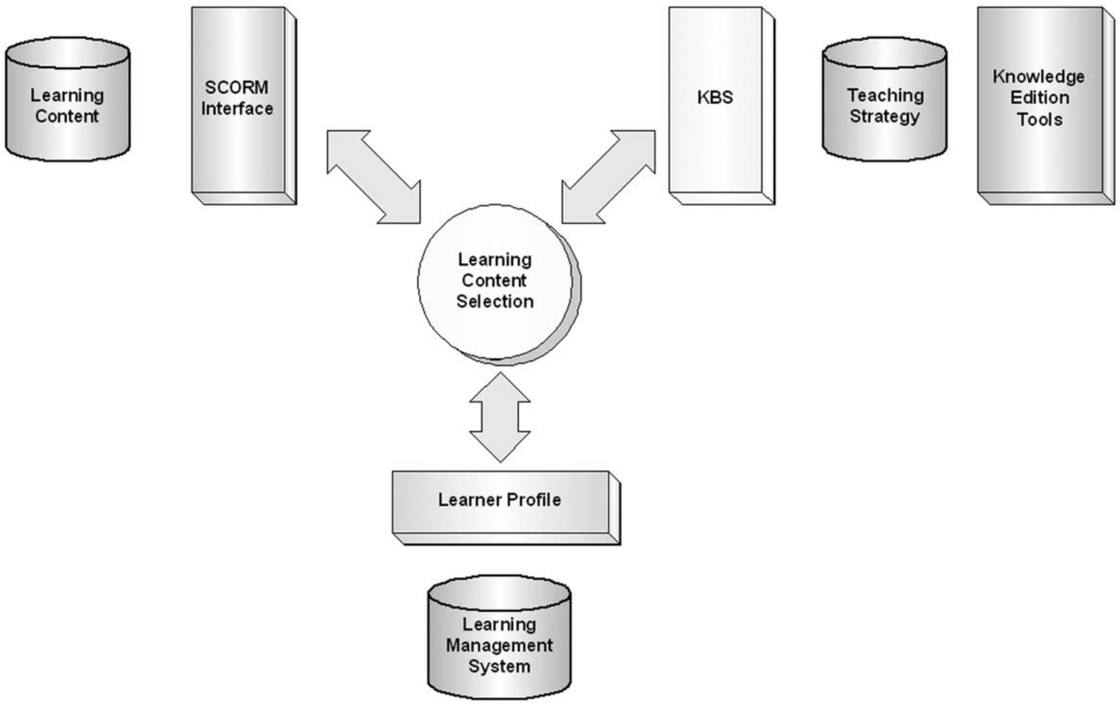

Fig. 15. Components for learning content selection system.

NORM rule base to manage and process teaching strategy edited by teacher for selecting learning content is described.

Design student profile format and KC template. According to previous studies of CAL (Beishuizen & Stoutjesdijk, 1999; Chou, 1996), students' learning activities and corresponding learning achievements are important to find appropriate learning material for student to learn; for example, if a student is not good in mathematics according to the grades in exams, learning content about basic mathematics theorems should be included when we plan the topics for this student to learn; otherwise, these basic learning content should not be included.

In our prototype, following attributes are included in the learning profile of each student to represent his/her learning achievement of a learning topic:

Topic: The topic of the course to be recorded, for example, Mathematic, English, etc.
Grades: The grades got of the corresponding course or learning topic.
Progress: The progress of a course or a learning topic, maybe represent in percentage.
Learning status: The learning status of a student in the corresponding course or learning topic, for example, study hard or normal.

Hence, a student's learning profile can be thought as a set of records to represent as the learning history of the student.

Design SCORM corresponding data format. According to the definition of SCORM Metadata, many information can be contained in the metadata for LCMS to understand and manage the learning content. For the LCMS system in this work, SCORM metadata is used for managing system imported learning content and finding appropriate learning content for student. However, not all the information contained in SCORM metadata is useful for learning content managing and retrieving, and following information is selected as managing information for our LCMS.

Title: The title of the learning content.
Keywords: The keywords of the learning content.
Version: The version of the learning content, useful to track the evaluation of the learning content.
Status: The status of the learning content, which maybe Draft, Final, etc.
Content type: The content type of the data included in the learning content, which may be the data format of the learning content.
Requirements: The technical requirements to view the learning content, for example, Browser, Operating System, etc.
Interactive type: The type of interaction between student and the learning content.
Interactive level: The level of interaction between student and the learning content.
Learning resource type: The type of learning resource contained in the learning content.
End user: The type of the end user to use the learning content.

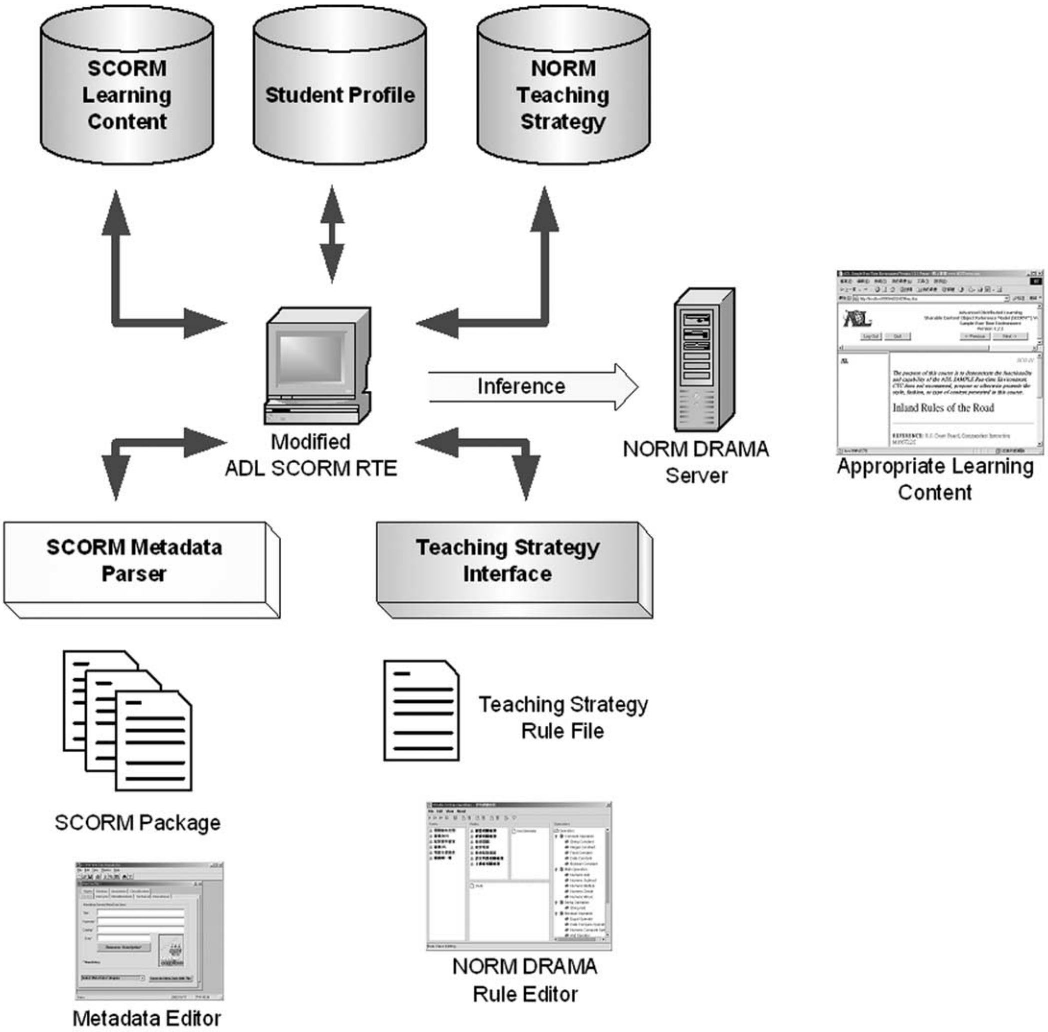

Fig. 16. The architecture of the prototype LMS system.

Fee: Indicate if fee is required to use the learning content.
Classification: The classification of the learning content.

As SCORM learning content needed to be managed, the above information contained in the MANIFEST file of the SCORM learning object will be retrieved and stored into LCMS managing mechanism, and provide learning object searching, exchanging, and planning functionalities.

Find a teaching domain and collect learning content. In our experiment, we select high school mathematic as the teaching domain, and learning content about high school mathematic are stored in the system and ready to provide to users of the system.

Design the architecture. In order to provide learning content selection service based on teacher-defined strategy, the architecture of a prototype system is designed as shown in following figure

In this architecture, we use ADL SCORM Sample RTE (Sharable Content Object Reference Model (2003))0 as the basic architecture to build an LCMS. ADL SCORM Sample RTE is a basic LMS provided by ADL which satisfies SCORM RTE 2.0 standard. In SCORM Sample RTE, administrators can import SCORM packaged courses for learners of the system to study, and learners can register courses to start learning. However, currently there is no information for learners to understand what included in a course, how difficult the course is, etc. As the number of courses grows in this RTE, there will be a problem for learners to find appropriate course to study.

The metadata contained in SCORM packaging courses may include information about the course, which will be useful for learners to understand and select the course. Since the metadata of SCORM courses is formatted in XML, a SCORM Metadata Parser is implemented in the prototype to extract the meta information. On the other hand,

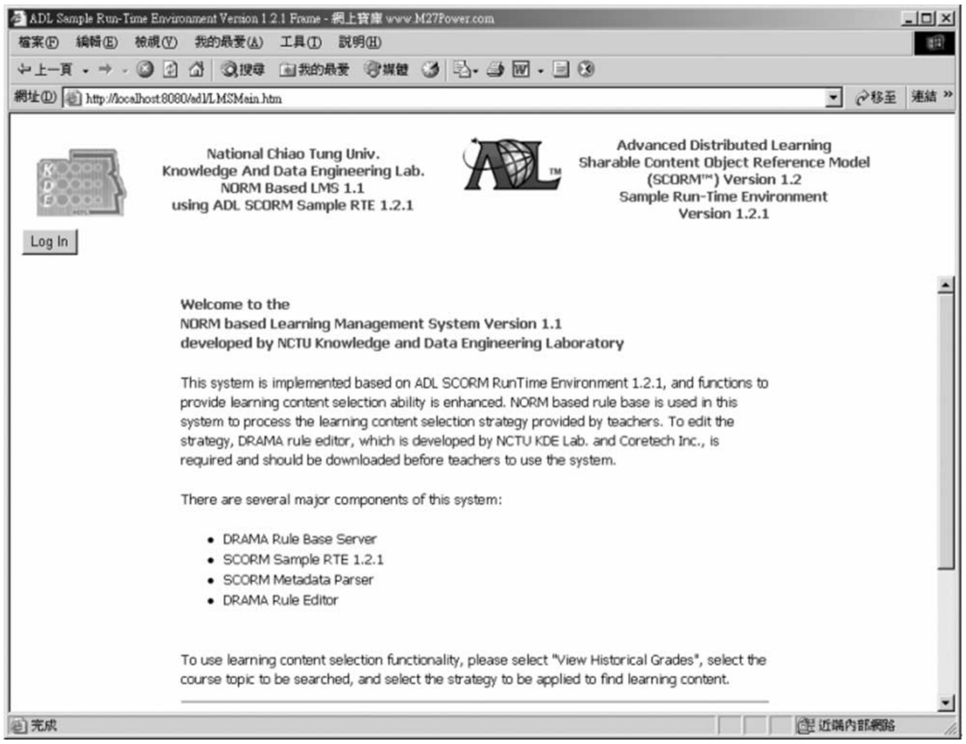

Fig. 17. Login page of the NORM based learning management system.

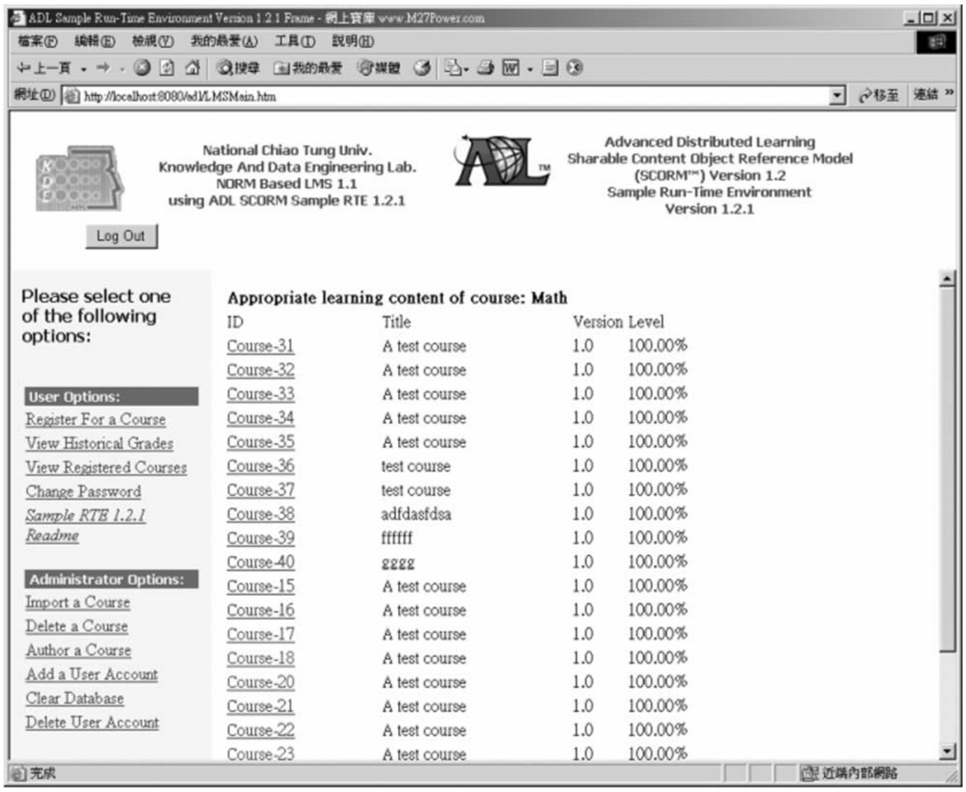

Fig. 18. Selecting learning content by inferring knowledge in DRAMA rule base.

the metadata for SCORM courses is generated using SCORM Meta-Data Generator Pro 1.2.0.

As we have mentioned, the selection of learning content considered not only the course information, but also the profile and learning history of learners must be considered. For this purpose, we also design a learner profile input interface for teachers to input students grades in each field, and will be used when learners trying to find appropriate learning content according to some learning content selection strategy defined by teachers. Figs. 16–18

In this prototype system, the learning content selection strategy (teaching strategy) is defined using NORM DRAMA rule editor as a rule file, and we designed a new function in the RTE for teachers to import new teaching strategy from rule file. When learners try to find appropriate learning content for him/her to study in some fields, they can use an imported teaching strategy and then select suitable learning content according to the meta information of courses and learner's learning profile. In order to process the knowledge included in teaching strategy, a NORM DRAMA Server is installed on the server, and when the RTE trying to process the teaching strategy, the prototype system will connect the NORM DRAMA Server and give corresponding facts for the server to infer. The result of the inference process will be used to select the learning content.

Currently, the server is hosted in http://e-learning.nctu.edu.tw/norm with high school mathematic learning material, and expect to be extended to all fields of high school education. Following are some snapshots of the NORM based LMS.

6. Conclusion

In this work, we first review some previous rule-based system and OO technology and discuss some requirements for maintaining a KBS. For the requirements, we propose NORM (New Object-oriented Rule Model) under OO concept according to the behavior of human learning. In NORM, every knowledge concept is represented by a KC and three kinds of relationships between KCs are defined to represent the cooperation of KCs. Reference relation represents the association of two different KCs if the KCs have common piece of knowledge. This relation is helpful for using original knowledge to construct new knowledge. Extension-of relation is used to extend or modify the KC constructed by other people. The facility of Extension-of is useful for knowledge sharing and exchange. Transference relation including Trigger and Acquire is used to represent the interaction of different KCs. Moreover, the knowledge reusability, sharability and encapsulation in NORM can be confirmed with the three properties of OO technique, encapsulation, inheritance and dynamic binding.

DRAMA, a NORM knowledge modeled rule base system platform, which is developed by Coretech Inc, Taiwan, is also introduced and applied in this paper. And a LMS using DRAMA to infer the knowledge for selecting appropriate learning content for different student is designed and implemented, which is also introduced in this paper and applied to high school education.

Acknowledgements

This work was partially supported by MOE Program for Promoting Academic Excellence of Universities under grant number 89-E-FA04-1-4, High Confidence Information Systems.

References

Anderson, J. R. (1995). *Cognitive psychology and its implications*. New York: W.H. Freeman.

Beishuizen, J. J., & Stoutjesdijk, E. T. (1999). Study strategies in a computer assisted study environment. *International Journal of Learning and Instruction*, *9*, 281–301.

Booch, G. (1991). *Object-oriented design with applications* (2nd ed). San Francisco, CA: Benjamin/Cummings.

Buchanan, B. G., & Shortliffe, E. H. (1984). *Rule-based expert systems*. London: Addison-Wesley.

Choi, B., & Lee, H. (2002). Knowledge management strategy and its link to knowledge creation process. *Expert Systems with Applications*, *23*(3), 173–187.

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*.

CLIPS (1998). CLIPS reference manual volume I basic programming guide. Software Technology Branch, Lyndon B. Johnson Space Center, Version 6.10.

Collins, A. M., & Quillian, M. R. (1969). Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*.

Deese, J. (1965). *The structure of associations in language and thought*. Baltimore: Johns Hopkins Press.

DRAMA (2003). Coretech Inc, http://www.ctknow.com.tw/rule.htm

Van Elst, L., & Abecker, A. (2002). Ontologies for information management: balancing formality, stability, and sharing scope. *Expert Systems with Applications*, *23*(4), 357–366.

Fikes, R., & Kehler, T. (1985). The role of frame-based representation in reasoning. *Communications of the ACM*, *28*(9), 904–920.

Gagné, R. M. (1985). *The conditions of learning and theory of instruction*. New York: Holt, Rinehart and Winston.

Gagné, R. M. (1984). Learning outcomes and their effects. *American Psychologist*, *39*, 377–385.

Giarratano, J., & Riley, G. (1989). *Expert systems: Principle and programming*. Boston: PWS Publishing Company, pp. 63–102, see also pp. 509–513.

Ginsberg, A., Weiss, S. M., & Politakis, P. (1988). Automatic knowledge base refinement for classification systems. *Artificial Intelligence*, *35*(2), 197–226.

Glaser, R. (1987). Thoughts on expertise. In C. Schooler, & K. W. Schaie (Eds.), *Cognitive functioning and social structure over the life course* (pp. 81–94). Norwood, NJ: Ablex Publishing Corporation.

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.

Kingston, J. (2001). High performance knowledge bases: four approaches to knowledge acquisition, representation and reasoning for workaround planning. *Expert Systems with Applications*, *23*(4), 181–190.

Kintsch, W. (1970). Models for free recall and recognition. In D. A. Norman (Ed.), *Models of human memory*. New York: Academic Press.

Klausmeier, H. J. (1971). Cognitive operations in concept learning. *Educational Psychologies*.

Lee, S., & O'Keefe, R. M. (1996). The effect of knowledge representation Scheme on maintainability of knowledge-based systems. *IEEE Transactions on Knowledge and Data Engineering*, *8*(1), 173–178.

Reichgelt, H. (1991). *Knowledge representation, an AI perspective*. Norwood, NJ: Ablex Publishing Corporation.

Roesner, H. (1988). Expert systems for commercial use. *Artificial Intelligence and Expert Systems*, 35–59.

Rosca, D., & Wild, C. (2002). Towards a flexible deployment of business rules. *Expert Systems with Applications*, *23*(4), 385–394.

Rumbaugh, J., Blaha, M., et al. (1991). *Object-oriented modeling and design*. New Jersey: Prentice Hall.

Salzgeber, M. J., et al. (1993). Managing uncertainty in CLIPS: a system level approach. *Proceedings of the 6th Florida Artificial Intelligence Research Symposium, Florida AI Research Society*.

Sharable Content Object Reference Model (2003). Advanced Distributed Learning, http://www.adlnet.org/index.cfm

Shortliffe, E. H., & Buchanan, B. G. (1975). A model of inexact reasoning in medicine. *Mathematical Biosciences*, *23*, 351–379.

Souza, M. A. F., & de Ferreira, M. A. G. V (2002). Designing reusable rule-based architectures with design patterns. *Expert Systems with Applications*, *23*(4), 395–403.

Tsai, C.-J., & Tseng, S. S. (2002). Building a CAL expert system based upon two-phase knowledge acquisition. *Expert Systems with Applications*, *22*(3), 235–248.

Tulving, E. (1983). *Elements of episodic memory*. Oxford: Oxford University Press.

Tulving, E., & Thomson, D. M. (1973). Encoding specificity and retrieval processes in episodic memory. *Psychological Review*, *80*, 352–373.

Wu, X. (2000). Knowledge object modeling. *IEEE Transactions on System. Man, and Cybernetics-Part a: Systems and Humans*, *30*(2).

Wu, Y. C (1999). An approach to object-oriented rule base management system. Master Thesis, Department of Computer and Information Science, National Chiao Tung University.