A design of case-based decision making method by ES in educational business model

Young Jun Kim

***Abstract*—**A case-based reasoning has emerged as an alternative to rule-based techniques for the expert system design. This paper describes an application of case-based reasoning method to the expert system design for decision making. With user's requirements, the system searches a domain dependent case-base for a similar case. If there is one, the system uses it to strategic decision making with minimum user's interaction. When there is no similar case, the system uses cases in domain independent case-base to make a new strategic decision making. The strategies for retrieving the relevant cases, for building a rough solution, for repairing the rough solution, and for learning cases are also described. It is also shown how the system works by using an example in designing for management applications. The major contribution of this paper is educational business model of an expert system with case-based reasoning approach to decision making. Since many strategic decision making experts use their old experiences in various cases, this study can lead to more realistic solution for such expert system in business applications.

***Keywords*—**Business modeling, case-based reasoning, decision making, education, management information system.

I. Introduction

Human designers appear to solve design problems not by reasoning from the primitive components available in educational business applications. Recently, this work has led to substantial study on case-based design [11]. It has also led to the construction of a number of small, experimental case-based reasoning systems. Thus, it needs much experience and knowledge about design techniques and enough understanding of design requirements. Usually, The strategic decision making design is performed by experts who obtain information about the users' needs through interviewing, examining existing documentation and other traditional means. Currently, many strategic decision systems are used in real data processing areas by non-expert users. However, they mostly do not provide efficient functions or tools to design a strategic decision making.

So, they may need design experts to ask them for a design. There are several works in strategic decision making process [5]. However, none of them uses case-based reasoning (CBR) techniques. However, a naive designer is hard to identify the entities and relationships on business modeling. The design method based on a form model is more powerful and convenient than the entity relationship model for naive users, since the users do not need to define the conceptual schema for their enterprise. A CBR technique can be used to the automation of database design process because CBR is now ripe for application to business application design problems. This approach of the use of case-based technology for designing some strategic decision making in educational business model has made it aware of several important considerations.

First, design requirements in educational business model are very dynamical1y changed and added continuously. This is because design requirements usually cannot be completely formulated at the beginning stage of decision making design. Those are rather evolving. Second, a strategic decision making should be normalized to preserve data integrity and consistency [2]. Third, it should be general enough to design for various application domains. To achieve such generality, it must have domain independent knowledge for strategic decision making and learning capability for more efficient ways of extending domain dependent knowledge. The proposed expert system for strategic decision making (ESSDM) applies the two kinds of methods to design work. First, target strategic decision making is designed by using the similar case with a few modifications. Second, when there is no similar case, the target strategic decision making is designed by using only domain independent case-base, general normalization technique. For this approach, we propose a relational conceptual chart to represent the design method of users' and each case in the case-base for business executive manager. It describes an overall idea for applying a CBR technique to strategic decision making. After experimenting in a limited domain, this system can effectively process a design task on business modeling.

II. Strategic Decision Making

A. Related Studies

An application of CBR in design include well process design, conceptual design of office building, conceptual design of hydro-mechanical systems [21], design menu of a meal, and autoclave layout design [3]. There are several works in strategic decision making. None of them uses CBR techniques. Korczak proposed a strategic decision making system using a rule-base; database design knowledge which is divided into a declarative part and a procedural part [13]. Ruoff developed an expert system prototype that assists a database designer in defining a conceptual modeling rules and heuristics [19]. Dogac developed a generalized expert system for decision system that consists of an expert system for

Young Jun Kim is with the Economics & Commerce Division, Baekseok Culture University, Professor, 393, Anseo-dong, Dongnam-gu, Cheonan, Chungnam, 330-705, Korea (phone: 82-41-550-0459; fax: 82-41-550-0548; e-mail: yjkim@bscu.ac.kr).

generating methodologies for management design and an expert system for decision design [7]. Thus, the designer should identify the entities and relationships in educational business model. Choobineh proposed a form model and an expert database decision making that analyzes instances of the form model to derive a conceptual applications. The user paints the form on the screen, and the form definition system originates a conversation to capture the form.

This system derives an entity relationship diagram by analyzing a collection of decision forms. A number of other researchers have reported works in educational business model input for decision making. Bouzeghoub and Gardarin report the implementation of an expert system which uses natural language to drive a relational entity [5]. Eick reports a similar approach based in business applications inputs from different user groups and the conversation with the designer [8]. A designer is hard to identify the entities and relationships on business modeling.

B. Strategic Decision Making Process

A requirement analysis consists of a high-level analysis of the operation of in educational business model. It represents these requirements via some formal modeling technique [4]. In this paper, we accept the requirement description by natural language rather than formal modeling techniques because naive users cannot use such techniques and cannot easily understand it. And, conceptual modeling to develops a strategic decision making, a conceptual model must be designed. This conceptual model reflects the entities and their relationships based on the requirements of the organization.

The most important steps during the early stages of the strategic decision making are conceptual modeling and logical schema design. At the conceptual modeling step, a designer analyzes the requirements of the operation of a management and then defines the logical structure to satisfy those requirements. The designer should identify the entities that the database is to contain, the relationships among those entities. Traditionally, it has been thought to be easy for the designer to describe the design requirements of a database through input forms by using natural language. We came to conclude that specifying the design requirements using printed report forms, is one of the most convenient methods for users. All identified entities and relationships need to be transformed into relations for the relational database model [6]. With this approach on business modeling, we view the process of designing strategic decision making as the following steps: 1) takes the requirements for each form in natural language, extracts the information for management modeling, normalizes the form business model to guarantee, and produces the form decision making, 2) integrate each form into an application decision making structure as a business model, 3) transforms the application data structure into a relational model as a logical model by using many cases.

III. Case-Based Reasoning Method

Some reasoning has traditionally involved writing and customizing rules or models to solve problems. However, there is another paradigm that has proven effective in many experimental and applied systems. Cases are indexed according to carefully chosen features that may resemble to rules and frequently, they are generalized to eliminate irrelevancies and to limit the scope of the search. Despite those similarities, there are several advantages of using CBR instead of rule-based reasoning [10]. There is no need to extract a complete and consistent set of rules referring to the domain knowledge but to gather a set of significant cases which are for more accessible than information embodying experiential knowledge.

A CBR system is much more flexible and easily modifiable by only changing the classification of eases compared with a rule-based system where major modifications could imply regenerating the design. There is no need for an additional effort to carefully rank the firing order of the rules. Rule-based expert systems have difficulty dealing with the constraints imposed by the uncertainty inherent in educational business model [23]. It also fails when applicable rules fired by the system are inconsistent with each other. Rule-based expert systems work well for well-defined problems which can be solved by algorithmic methods. The model uses reasoning to select and transform specific solutions to previous design problems to be appropriate as solutions for a new design problem. The issues associated with using this model for design include the identification of the necessary information about a design episode to reason about its applicability in a different context, the meaning of a similar design, and the transformation of the solution from the original context to the new context.

Although human designers appear to be good at using this type of analogy, it is difficult to automate it. As a process model, CBR involves several operations. One set of operations is recall relevant cases from case memory, select the most promising case, construct a solution for the new problem, test the solution, evaluate the results, and update case memory by storing the new case. The operations of most interest are centered on case retrieval, selection, and modification. The selection of a case among potential cases requires recognition of the relevance of each case and how close the case is to providing a solution to the design problem. The extremes in representing cases in memory are to represent each case in its entirety or breaking a case into pieces, there are various approaches, such decision making grouping. The most obvious way to index cases is to use appropriate features as indexes. The selection of a set of indexes can be fixed, which is not flexible, or the selection can be based on inductive-learning methods to identify predictive features or explanation techniques or to define a vocabulary associated with a case-oriented approach to problem solving. The more general features or shared features of various cases are searched first, eliminating large amounts of cases according to these shared features. There are examples in which cases were stored in entirety using redundant discriminate networks and in pieces [13]. A weighted count of matching features provides one way to select the best case. Some approaches to finding the best case are preference heuristics, decision making analysis [19].

When storing design episodes as cases, the content, as well as the organization, of a case must be considered. There is also the issue of storing the design solution or the operators used to produce the design description. The advantage to storing the design solution is that many existing cases can be used immediately, augmenting the geometric description with the

relevant functions, and so on. The alternative, strong the operators, allows the transformation to the new problem to be an execution of the old solution operators using the new problem statement. The use of CBR for design synthesis assumes that the design knowledge is represented in the form of design episodes, defining the type of knowledge needed. This paper describes an architecture and overall design flow of ESSDM proposes the relational conceptual chart formalism which is an internal representation scheme to describe the dependency structure among related attributes of the user' design requirements. An example for representing a form in university administration application is presented. Also it is describes the design using domain independent case-base, the design using domain dependent case-base, case retrieval, adapting cases, explanation, solution refinement, and case learning method.

IV. Design of ESSEM

A. ESSDM Structure

This system consists of some modules, and case-base. The modules are user interface, relational conceptual chart translator, case retriever, and decision making designer. The case-base is domain independent case-base and domain dependent case-base and Fig. 1 show the system structure.

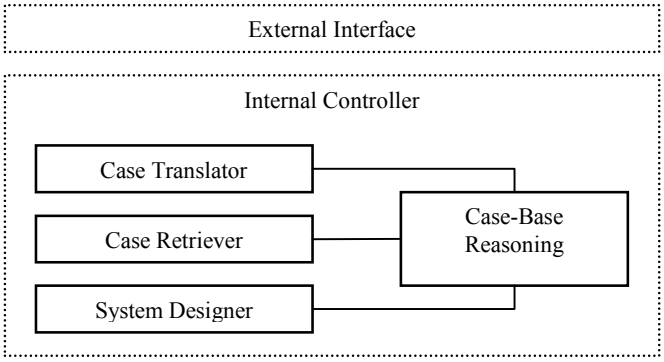

Fig. 1 an expert system structure for strategic decision making

The interface module provides interactive Q&A functions. It takes the users' requirements for strategic decision making, asks the user for specific information, accepts the answer, present the description. The interaction between the system and user are performed by using natural language sentences, from layout, and chart. The relational conceptual chart translator transforms natural language input sentences for specifications of this the requirements into a relational conceptual chart. Since natural language processing is beyond the scope of this study, we will not discuss the details of the relational conceptual chart translator in this paper.

The case retriever selects a similar case domain independent case-base or domain dependent case-base. This module performs the following functions: searches for the relevant case by navigating a hierarchical case tree in domain dependent case-base through user interaction, calculates similarity to select the relevant cases, maintains a list of relevant cases, presents the best similar case to the structure design module. The strategic decision making designer takes a relevant case and performs design actions described in the case. In design using domain dependent case-base, this module presents the content of selected from case to a user, such as from layout and from schema description. The user can request the modification of the form case to the system. In design using domain independent case-base, it performs the normalization process through iterative case matching and user interaction. The case-base consists of two case-bases; domain independent case-base and domain dependent case-base, domain independent case-base includes normalization cases and domain dependent case-base is a set of cases for strategic decision making.

B. ESSDM Flow

The system searches the similar design case by matching the design features with the description of cases in domain dependent case-base. The case retriever searches for relevant case by navigating the case to design a schema. However, if a similar case does not exist in domain dependent case-base, the system designs a schema by using domain independent case-base. In design using domain dependent cases, if the system needs to normalize the modified schema, the system uses domain independent case-base to normalize it. After the design process is completed, the new design result is saved into domain dependent case-base. It has been thought to be easy for users to describe design requirement of strategic decision making through input forms.

From our experience, we came to conclude that specifying the requirements printed report forms [9] is one of the most convenient methods for naive users. Mostly, requirement specification in strategic decision making is based on an entity relationship diagram. However, a naive user is hard to identify the entities and relationships in the enterprise. A user explains the design requirements and the dependency among the attributes in a form by using natural language sentences. A design requirements described in natural language sentences should be transformed into internal structures to be processed by a computer system. A relational conceptual chart consists of concept nodes and relation nodes.

V. System using Case-Base

A case is domain in domain independent and can be represented as a set of rules or cases. Since the domain dependent case-base is organized as a set of cases. The strategic decision making using domain independent case-base is processed by the flow in Fig. 2. Transformed relational conceptual chart is matched with the cases in domain independent case-base to normalize the requirements. The relational conceptual chart is normalized by applying the most relevant case. When a relational conceptual chart is normalized, the system transforms it into the higher normal form by applying the actions specified in each case. If the relational conceptual chart does not guarantee the third normalization forms, the system notices the reason to the user and feedback to relational conceptual chart translation step. And, the system accepts the modified requirements from user; the process does the same step with the relational conceptual charts iteratively until they are exactly

matched with cases which guarantee the third normalization forms for management applications.

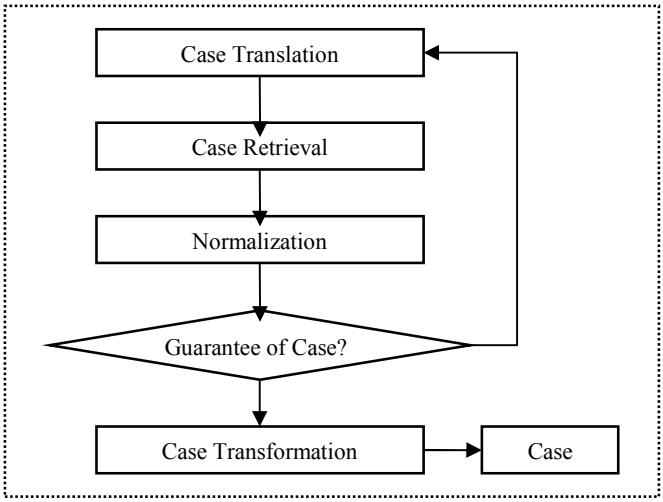

Fig. 2 a flow of domain dependent case-base

A. Domain Independent Case-Base

The domain independent case-bases are cases for normalizing the relational conceptual chart set. Each case consists of description, outcome, and actions. The description in a case is a relational conceptual chart that violates third normalization forms or guarantees the third normalization forms. The outcome is a normalized relational model as a logical decision making. The actions consist of a sequence of the action rules for normalization, such as an inquiry for more information, confirmation for third normalization forms, etc. A design case may contain several different types of description. In the domain independent case-base, currently there are only some cases as the following:

domain independent case-base 1:
many-to-many mapping with partial-key dependency
domain independent case-base 2:
many-to-many mapping with transitive dependency
domain independent case-base 3:
many-to-many mapping with full functional dependency
domain independent case-base 4:
one-to-one mapping with partial-key dependency
domain independent case-base 5:
one-to-many mapping with partial-key dependency
domain independent case-base 6:
one-to-one mapping with transitive dependency
domain independent case-base 7:
one-to-many mapping with transitive dependency
domain independent case-base 8:
one-to-one mapping with full functional dependency
domain independent case-base 9:
one-to-many mapping with full functional dependency

Fig. 3 a definition of some cases in domain independent case-base

The cases domain independent case-base 1 to domain independent case-base 3 is cases that violate the first normal form. If any relational conceptual chart can be matched with those cases, the system aware the chart violates the first normal form and takes actions specified in each case. The outcomes of the actions are revised charts which guarantee, at least, the first normal form. Similarly, the cases domain independent case-base 4 and domain independent case-base 5 are cases for the charts which may violate second normal form, and domain independent case-base 6 and domain independent case-base 7 are for third normalization forms. The domain independent case-base 8 and domain independent case-base 9 are the cases which guarantee third normalization forms. Therefore if any chart can be matched to those cases, the charts are guaranteed to satisfy third normalization forms. For the normalization, each relational conceptual chart is matched with each in domain independent case-base, and the case retriever selects the similar cases by similarity order. When a relational conceptual chart is normalized, the decision making designer module transforms it into the relational conceptual charts of higher normal form by using the actions specified in each case.

domain independent case-base 1
many-to-many mapping with partial-key dependency

DESCRIPTION:
[composite_key] -
(part) → [tentative_key:{key-set-1}]-
(1-M) → [attribute:{attr-set-3}]
(M-1) ← [attribute:{attr-set-4}]
(part)→[tentative_key:{key-set-2}]
(any1) → [attribute:{attr-set-1}]
(any2) ← [attribute:{attr-set-2}]

OUTCOME:
decompose the input relational conceptual chart into two relational conceptual charts
[tentative_key:
{key-set-1 U key-set-2 U inferred-key-set}]
(any1) → [attribute:{attr-set-1}]
(any2) ← [attribute:{attr-set-2}]
[tentative_key:{key-set-1 U inferred-key-set}] -
(1-M) → [attribute:{attr-set-3 - inferred-key-set}]
(M-1) ← [attribute:{attr-set-4 - inferred-key-set}]

ACTIONS:
• If it has more than two relationship subcharts,
- Process step-excess of requirement
• If it lacks some relationship subcharts
• Process step-insufficient requirement
• Feedback to get more information after instantiating the referents and relations

Fig.4 an example of case-base 1

For example, the content of case domain independent case-base 1 and 4 are shown in Fig. 4 and Fig. 5. The domain independent case-base 1 and 4 consist of third parts: description, outcome, and actions. The description of domain independent

case-base 1 and 4 represent case of a relational conceptual chart that violates the second normal form. The case transforms first normal form relational conceptual chart into second normal form relational conceptual charts. The constraints mean that they must satisfy the condition before processing the actions in the case. The actions are for transforming the violating relational conceptual chart into the relational conceptual charts in the outcome to guarantee the second normal form. After processing the actions, schema designer module instantiates the original symbols and produces the outcome that is decomposed the input relational conceptual chart into two relational conceptual charts.

domain independent case-base 4
one-to-one mapping with partial-key dependency

DESCRIPTION:
[composite_key] -
(part) → [tentative_key:{key-set-1}]-
(1-1) → [attribute:{attr-set-3}]
(1-1) ← [attribute:{attr-set-4}]
(part)→[tentative_key:{key-set-2}]-
(any1) → [attribute:{attr-set-1}]
(any2) ← [attribute:{attr-set-2}]

OUTCOME:
decompose the input relational conceptual chart into two relational conceptual charts
[tentative_key:{key-set-1 U key-set-2}]
(any1) → [attribute:{attr-set-1}]
(any2) ← [attribute:{attr-set-2}]
[tentative_key:{key-set-1}]
(1-1) → [attribute:{attr-set-3}]
(1-1) ← [attribute:{attr-set-4}]

ACTIONS:
• If attr-set-1 and 2 are different, process step-ida.
• If attr-set-3 and 4 are different, process step-ida.
• Produce the outcome.

Fig. 5 an example of case-base 4

When charts lack some of the relationship, ESSDM asks the user to confirm the insufficient information. When relationship chart exist more than two, it asks the user to confirm the exceeding information. To guarantee second normal form, it asks the user to confirm whether the partial-key dependency exists or not. When a relational conceptual chart has many-to-many relationship, it should be transformed into a relational conceptual chart that has one-to-one relationship because it violates the first normal form. It is possible to transform into first normal form by appending the candidate tentative-keys in non-key attributes to current tentative-key set. If there are no such attributes, ESSDM selects the attributes that have been used in other relational conceptual charts. And it asks the user to confirm whether it is possible to do that.

In case matching, our system calculates similarity score to measure relevancy. These relevant cases are presented to the schema designer module on order of the score. In case of partial matching, the decision making designer requests the user for more specific information. This information is appended to the relational conceptual chart, and the relational conceptual chart is matched with the domain independent case-base again. ESSDM maintains the matching history to identify what kinds of cases have ever been matched with the relational chart. After all relational conceptual charts are processed it produces a normalized model for target application as merging normalized relational chart set for all form. The final normalized relational conceptual chart set, as an application, is indexed to a domain dependent case-base to be used to design the similar decision making method later.

domain independent case-base 8
one-to-one mapping with full functional dependency

DESCRIPTION:
[tentative_key:{key-set}]-
(1-1) → [attribute:{attr-set-1}]
(1-1) ← [attribute:{attr-set-2}]

OUTCOME:
[key:{key-set}]-
(1-1) → [attribute:{attr-set-1}]

ACTIONS:
• Produce the outcome
• Save outcome into ncg-set after instantiating the referents and relations
• If attr-set-3 and 4 are different, process step-ida

Fig. 6 an example of case-base 8

domain independent case-base 9
one-to-many mapping with full functional dependency

DESCRIPTION:
[tentative_key:{key-set}]-
(1-1) → [attribute:{attr-set-3}]
(M-1) ← [attribute:{attr-set-4}]

OUTCOME:
[key:{key-set}]-
(M-1) → [attribute:{attr-set-3}]

ACTIONS:
• Produce the outcome

Fig. 7 an example of case-base 9

The merged relational conceptual chart has a partial key dependency, but this is incomplete relational chart because a relationship chart is omitted. The attribute lists of two charts are also different. A case retriever selects relevant case in order of similarity such as domain independent case-base. This case is not exactly matched, but partially matched. After merging and combining again, the merged relational conceptual chart is partially matched with domain independent case-base again. The case designer inquires of user for acquiring omitted attribute. The merged relational conceptual chart is matched

exactly with domain independent case-base again. The system asks the user to confirm whether partial-key dependency exists, because it is known that it exists in an attribute set. These decomposed relational charts are exactly matched with domain independent case-base, domain independent case-base is case for one-to-many mapping with functional dependency. This confirms to the user transitive dependency for guaranteeing the third normalization forms. The set of final normalized relational conceptual charts can be also considered as a final relational strategic decision making.

B. Domain Dependent Case-Base

The domain dependent case-base, case memory, is well organized as a hierarchical case tree. The relevant cases are retrieved by calculating the similarity to measure the relevancy of domain and form features. The rough solution is modified by repairing the missing or incorrect part by using a domain independent case-base to guarantee the third normalization forms. Specifying the design requirements is based on a report form. The design requirements for a form are described by using pre-defined tabular form [11]. From users' requirements, the system identifies two kinds of design features that consist of domain features and form features. In design using domain dependent case-base, the design task is processed by the flow. After generating a domain specific explanation of which parts of the requirement are missing in the rough solution, the system interacts with the user for repairing the deficient parts of the solution. The system normalizes the modifications by using domain independent case-base and merges them into the rough solution. All cases are hierarchically organized to form a partially ordered chart to help the system to reduce its search space. Linking the cases into a well-defined hierarchy will facilitate modify of the case-base through addition of a new case and case abstraction for business strategy.

Domain features are organization, management resource, and use of the form. Every domain case node in memory will be an index for available designed form cases. Domain case nodes with same features are abstracted as a higher level abstraction node. Every domain case node in memory will be an index for all form cases. Below the domain case hierarchy, there are form case hierarchies. Since this system assumes form oriented approach to schema design, a well-organized form case hierarchy is very important. Each domain case can have many form cases. Form features, which describe the structure of the form, are entities, attributes, the form type, utilization degree that represents the number of references of the case, and the form name. Form case nodes with the most similar features are grouped as a higher level abstraction node. The lowest part of the domain dependent case-base hierarchy consists of form data structure charts as instances of the design case. Each form data structure chart represents a set of normalized relational model conceptual charts for the entities and attributes appeared in the form.

C. Selecting and Managing Process

After reduces its scope into the form case level, the system searches the most similar from-data structure chart. For example, feature ‘entity’ has larger weight value than feature ‘attribute’, since ‘entity’ as a conceptual component is more important than ‘attribute’ as an element of a form. This system is still investigating for a more mechanical method to assign weigh to each feature. It is planning to define weight after constructing a sufficiently large case-base. The system keeps form data structure charts in the order of similarity value. It ignores the form data structure charts of which similarity is less than some threshold value. The selected cases may not exactly match with the user's requirements. Adaptation rules are needed to find the gaps and to fill the missing parts. To generate a rough solution, the system processes the following two steps: 1) merge the relevant form data structure charts selected by the case retriever, 2) remove unnecessary components comparing the design requirements. The merging action is started by appending the necessary entities in partial matched form data structure charts to the best form data structure chart.

For example, to design the schema for a form ‘financial manager schedule list’, the system appends the entity ‘business’ in a partially matched form data structure chart ‘office catalog’ to the best form data structure chart ‘financial manager registration list’. Next, the system removes the entities or attributes that have not been referenced in users' requirements. The generated rough solution has all entities and attributes that are included in previous cases. The system, however, does not have the entities and attributes that do not exist in the previous cases. Furthermore, it does not consider a correctness of the relationships among attributes, either.

D. Showing and Learning Process

This system describes the solution as tabular representation, such as Fig. 8, that consists of relations and description. Each relation may be an entity or a relationship in entity relationship model. A description is a relational conceptual chart that illustrates the relationships among attributes in that relation. An example of a design status report is shown in Fig. 8. After generating a domain specific explanation for the missing parts of the requirements in a rough solution, the system interacts with the used for repairing the missing parts of the solution. The system normalizes the modifications by using domain independent case-base and merges them into the rough solution.

This system has proposed an idea for the design using a domain independent case-base in educational business model. If the user confirms the correctness of the final solution that meet the users' requirements completely, the system accepts the attribute of each field in the relation, such as data type, length, range, value constraints, default-value, and null. A case-based learning algorithm input a sequence of training cases and output a concept description, which can be used to generate predictions of goal feature values for subsequently presented cases. The primary component of the concept description is a case-base, but almost all case-based learning algorithms maintain additional related information for the purpose of generating accurate predictions. Current case-based learning algorithms assume that case is described using a feature representation, where features are either predictor or goal features. Case-based learning algorithms are distinguished by their processing

method they focus on some parts of the case-based learning paradigm while deemphasizing others for management model.

An example (*design using domain dependent CBR*)
From: financial manager(MS) schedule list/semester
Design process:
- selected cases: 9 (similarity 70%~85%)
- best form data structure diagram
  : financial manager registration list
- removal: 8 entities, 12 attributes

Rough solution:
- structure: 10 entities, 17 attributes
- relations and description

Design deficiency:
- dummy entities: finance
- missed attributes: office, management, day-time

Comments:
- level of rough solution: excellent
- for complete design, it should describe the relationship between key attributes and 3 missed attributes.

An example **(***design using domain independent CBR)*
From: investor(NY) schedule list/semester
Design process:
- selected cases: 8 (similarity 70%~90%)
- best form data structure diagram
  : investor registration list
- removal: 5 entities, 13 attributes

Rough solution:
- structure: 14 entities, 20 attributes
- relations and description

Design deficiency:
- dummy entities: investment
- missed attributes: stocks, banking

Comments:
- level of rough solution: excellent
- for complete design, it should describe the relationship between key attributes and 2 missed attributes.

Fig. 8 examples of a design process

Assessment may involve explicit encoding and dynamic computation, most practical case-based learning similarity functions find a compromise along the continuum between these extremes. This function inputs the similarity assessments and generates a prediction for the value of the given case's goal feature. A case learning in domain dependent case-base consists of two steps: adding a new case and reorganize the case hierarchy by case abstraction. To add a new case into the domain dependent case-base, the system identifies the indexing terms for the added or modified domain features and form features. Then, it modifies the abstraction hierarchy for reflecting the identified indexing terms. New abstractions are formed when a number of cases are discovered to share a common set of features. The common features are used as indices to the original cases. The starting point for similarity judgments is the nearest neighbor algorithm. This algorithm searches through every case in memory, applies a similarity metric, and returns the case that is most similar to the new case. The nearest neighbor algorithm assumes that a case will be represented as a set of features. The similarity metric of this algorithm simply counts the number of features that a new case and a stored case have in common: Forming new abstractions simply on the basis of shared features is not a very good technique. A good index for CBR is distinctive but not unique. The most useful features to use for indexing are the features shared by many instances in memory as a whole, but by only a few of the instances.

VI. Evaluation of System

A. Hypotheses

It is time to pursue research for the output in case-based reason phase. For the purpose of evaluating this system, it will establish the hypotheses as follow:

- Hypothesis 1: ESSDM might be more efficient than the exiting CBR system.

- Hypothesis 2: ESSDM might be more efficient than the exiting ES system.

- Hypothesis 3: This system efficiency might be higher in the expert's group than the beginner's group.

For these hypothesis tests, it measured the time required in strategic decision making of how to use the exiting model and how to use beginner's group and expert's group. This means that the users choose the system instead of the existing model. As the expert's group is more experienced in modeling than the beginner's group, the improvement was more salient in the former group. Summarizing the result and analyzing the hypotheses. Table 1 was analyzed overall. It means that the time required for strategic decision making model was remarkably reduced compared to the existing model. Therefore, hypothesis 1 can be accepted. It can safely say that the strategic decision making model given in this system was more efficient than the existing CBR system. In other words, according to the result acquired by both beginner's and expert's group, this system properly supported the process of modeling in both groups regardless of the user's experience and knowledge level. Moreover, the time required for strategic decision making model by this system was reduced in both groups. For this reason, hypothesis 2 can be accepted. This means that strategic decision making model is more efficient than the existing ES system in the beginner's group as well as the expert's group.

Table 1 comparing exiting model and ESSDM

measure	exiting model(CBR)	ESSDM
time(mean)	41.8	17.4
s.deviation	3.73	1.45
comparing	100	41.6

Table 2 is the efficiency comparison between beginner's group and expert's group. This table means that the time required for using the strategic decision making model was

remarkably reduced in the expert's group than in the beginner's group since the former group has more experience in strategic decision making models than that of the latter group. So the user with good experience for decision making uses this system, it is expected that he/she would be able to bring about considerable productivity in business strategic decision. Thus hypothesis 3 can be accepted. The model efficiency is higher in the expert's group than the beginner's group.

Table 2 comparing beginner's and expert's group

measure	beginner's group		expert's group	
	exiting model	ESSDM	exiting model	ESSDM
time(mean)	51.6	23.3	32.3	9.3
s.deviation	4.02	1.72	2.73	1.01
comparing	100	45.2	100	30.3

B. Significance

The merits produced from the result of verifying hypotheses are discussed as follow. The merits of this system bring about the effects of reducing the effort and cost, which are expected on account of shortening the time required for strategic decision making in educational business modeling. This system, offering graphic user interface which expresses the outputs in system, enables strategic decision making modeling to be easier than exiting CBR system and ES system. At last, as this system allows the users to avoid the problems of strategic knowledge acquisition, it helps them to develop easily the new models. Namely strategic decision making by this system reduces decision making time and effort, consequently improving productivity. It is expected that the natural strategic decision making of this system can enhance management quality and maintain the efficiency of business management.

VII. Conclusion

This study presents an alternate approach for developing an expert system for strategic decision making using CBR methodology. This system proposed a relational conceptual chart formalism to represent the entities appeared in user's design requirement and the relations among them. The system uses a domain dependent case-base to find a case which is similar to the user's application. If there is a similar case, the system uses it to make a new strategic decision making method for the business model application. During the operation, the user can interact with the system to change the case for the purpose. Whenever the system changes the schema in the case, it needs to use a domain independent case-base to ensure that the newly changed schema should satisfy third normalization forms. When the system fails to find a similar case, it designs a new one using the domain independent case-base.

It may need lots of interactions with the user in this step to ensure a schema in third normalization forms. An important finding is that case search space is very small in domain independent case-base because we were able to get third normalization forms using only some cases. Whenever the design process is completed, the new design result is saved into domain dependent case-base for case learning. For many fundamental questions such as case learning, case matching, case memory reorganization which is essential for CBR method. The major contribution of this paper is the modeling of an expert system with CBR approach to strategic decision making automation.

Since many strategic decision making experts use their old experiences in various cases, this study can lead to more realistic solution for such expert system. This study has only presented the overall implementation idea to develop an expert system for strategic decision making by using CBR approach. This system is concentrating on the construction of the domain dependent case-base including enough design cases on business modeling. To complement the weak points of this paper, further study is needed in particular for setting the value of the parameters, such as the weight of each feature and threshold values for deciding the relevancy of selected cases, modifying the matching function for more efficient retrieval of domain dependent case-base, analyzing the performance of this system for management applications.

References

[1] Ashley, D., Assessing Similarities among Cases: A Position Paper, *Proceedings of a Workshop on Case-based Reasoning,* 2009, pp.102-118.

[2] Atre, S., *Data Base: Structured Techniques for Design, Performances, and Management*, John Wiley & Sons, 2000.

[3] Batini, C. and M. Lenzerizio, A Methodology for Data Schema Integration in the Entity Relationship Model, *IEEE Transactions on Software Engineering,* 2004, pp.720-734.

[4] Beamon, B. M., "Measuring Supply Chain Performance," *International Journal of Operation Management*, 2004, pp.275-292.

[5] Bouzeghoub, M. and G. Gardarin, The Design of an Expert System for Database Design, *Proceedings International Workshop New Applications of Databases*, 2003.

[6] Briand, H., H. Habrias, J. Hue and Y. Simon, Expert System for Translating an Entity-Relationship Diagram into Databases, *Proceedings 4th International Conference E-R Approach,* 2005.

[7] Dogac, A., B. Yuruten and S. Spacapietra, A Generalized Expert System for Database Design, *IEEE Transactions on Software Engineering,* Vol.35, No.2, 2009, pp.567-580.

[8] Eick, C., From Natural Requirements Language to Good Database Definition – A Database Design Methodology, *Proceedings International Conference Decision Making,* 2009, pp.301-321.

[9] Feret, M. and J. Glasgow, Case-Based Resoning in Model-Based Diagnosis, *Proceedings of 17th International Conference on Applications of Artificial Intelligence,* 2002, pp.237-245.

[10] Goel, A. and B. Chandrasekaran, Use of Device Models in Adaptation of Design Cases, *Proceedings of the Second DARPA Case-Based Reasoning Workshop,* 2005, pp.52-62.

[11] Goel, A. and B. Chandrasekaran, A Task Structure for Case-Based Design, *Proceedings IEEE International Conference on Systems, Man, and Cybernetics,* 2006, pp.145-162.

[12] Gordon, S., Front-End Analysis for Expert System Design, *Proceedings of the Human Factors Society 35th Annual Meetings,* 2007, pp.176-189.

[13] Korczak, J., L. Maciaszek and G. Stafford, Knowledge Base for Database Design*, Proceedings International Symposium on Database Systems for Advanced Applications,* 2007, pp.54-70.

[14] Leake, D., ACCEPTER: A Program for Dynamic Similarity Assessment in Case-Based Explanation, *Proceedings of Case-Based Reasoning Workshop*, 2007, pp.34-50.

[15] Lebwitz, M., Generalization from Natural Language Text, *Cognitive Science*, Vol.7, 1983.

[16] *Project Management Series,* Ernst & Young Navigator System Series, Release 3.0, 1995, pp.3-213.

[17] Pu, P. and M. Reschberger, Case-based Assembly Planning, *Proceedings of Case-Based Reasoning Workshop,* 2008, pp.210-220.
[18] Ran, S., "A Modeling for Services Discovery with QOS," *ACM SIGECOM Exchanges*, Vol. 4, No. 1, March, 2003.
[19] Ruoff, K., CODES: A Database Expert System Prototype, *Proceedings of the 20th Conference on Artificial Intelligence Applications,* 2006, pp.410-422.
[20] Simoudis, E., Using Case-Based Retrieval for Customer Technical Support, *IEEE Expert,* 2007, pp.78-91.
[21] Sowa, J., *Conceptual Structures: Information Processing in Mind and Machine,* Addison-Wesley Publishing Company, 1998.
[22] Tausworth, C., "The Work Breakdown Structure in Software Project Management," *The Journal of Systems and Software*, Vol.1, No.3, 2005.
[23] Tsichritzis, D. and F. Lochovsky, *Data model,* Prentice-Hall, 2000.