A Design of Expert System for Strategic Decision Making using CBR

YOUNG JUN KIM
Economics & Commerce Division
Baekseok Culture University
393, Anseo-dong, Dongnam-gu, Cheonan, Chungnam 330-705
KOREA
Email: yjkim@bscu.ac.kr

Abstract: Generally, designers appear to solve some problems not by reasoning from the primitive components available in the design domain, but by adapting the design cases of previously encountered artifacts. This paper describes an application of case-based reasoning method to the development of an expert system for strategic decision making. Instead of maintaining only domain specific knowledge for designing a target application, the proposed system, an expert system for strategic decision making (ES-SDM), relies on a case-base that consists of a domain dependent case-base and a domain independent case-base. Whenever a new strategy is designed, it is added to the domain dependent case-base. The strategies for retrieving the relevant cases, for building a rough solution and for learning cases are also described. This system is particularly useful for naive users and experts who use it as a design support tool.

Key-Words: - Strategic information systems, decision making, case-based reasoning, expert system, MIS.

1 Introduction

An applications of cased-based reasoning (CBR) techniques include story understanding, explanation reasoning, adaptive planning, learning and architectural design. The general task of design takes as input a specification of the constraints on the design of an artifact. Much of design activity in the real-world seems to be case-based. Recently, this work has led to substantial research on case-based design [9]. It has also led to the construction of a number of small, experimental case-based design systems. Traditionally, design has relied on human experience rather than mechanistic algorithms [10].

Thus, it needs much experience and knowledge about design techniques and enough understanding of design requirements. This is usually performed by experts who obtain information about the users' needs through interviewing, examining existing documentation and other traditional means. Many data base management system 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 project. There are several works in strategic decision making [4]. But, none of them uses CBR techniques. Most of research in the strategic decision making area is based on an entity-relationship model. However, a naive designer is hard to identify the entities and relationships in business. The strategic decision making design method is more powerful and convenient than the entity-relationship model for naive users, since the users do not need to define the conceptual model for business strategy [11].

This study approach of the use of case-based technology for designing a strategic decision making has made us aware of several important considerations. First, design requirements in the real world 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]. To achieve such generality, it must have domain independent knowledge for strategic decision making design and learning capability for more efficient ways of extending domain dependent knowledge.

The proposed expert system for strategic decision making (ES-SDM) applies the two kinds of methods to the decision making design. When there are exists a similar design case in domain dependent case-base, the target decision making is designed by using the similar case with a modifications in management modeling.
2 Previous Studies

Some related studies for decision making using CBR and for strategic decision making are briefly reviewed. But, to the best of our knowledge, no attempts have been made to exploit CBR as methodology for database design automation. Applications of CBR in design include weld-process design, conceptual design of office building [9], conceptual design of hydro-mechanical systems, design menu of a meal [3], and autoclave layout design [8].

There are several works in strategy decision making. None of them, however, uses CBR technique. Korczak proposed a strategy decision making system using a rule-base [11]; database design knowledge which is divided into a declarative part and a procedural part. Ruoff developed an expert system prototype that assists a database designer in defining a conceptual decision making modeling rules and heuristics. And, Dogac developed a generalized expert system for decision design that consists of an expert system for generating methodologies for database design for modeling the conceptual decision making. Therefore, the designer should identify the entities and relationships in the application domain. Choobineh proposed a form model and an expert decision design system that analyzes instances of the form model to derive a conceptual schema. The user paints the form on the screen, and the form definition system originates a conversation to capture [6].

A number of other researchers have reported works on the natural language input for decision making. Bouzeghoub report the implementation of an expert system which uses quasi natural language to drive a decision making [4]. Eick reports a similar approach based on the natural language inputs from different user groups and the conversation with the designer [7].

2.1 Decision making

It represents these requirements via some formal modeling technique such as strategy information system. To design 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 [3]. Human experts perform derivation of feasible physical structure, gross and fine placement, access path optimization, space requirements, performance prediction, and physical schema definition. It derives programs from database structures and business functions, form and report generation, and programming in the host language environment. Also this maintains the information for security, integrity, recovery, backup, authorization, auditing, and tuning. This tool performs restructuring, reorganization and application software conversion when requirement, policy, or strategy is changed.

The most important steps during the early stages of the strategy decision making are conceptual modeling and logical design. At the conceptual modeling step, a decision making designer analyzes the requirements of the operation of an organization and then defines the logical structure of a decision making to satisfy those requirements at the logical design. The decision making designer should identify the entities that the database is to contain, the relationships among those entities [3]. Traditionally, it has been thought to be easy for the designer to describe the design requirements of a decision making through input/output forms. It came to conclude that specifying the design requirements using printed report forms called form-oriented approach, is one of the most convenient methods for users [12]. In designing a logical decision making, all identified entities and relationships need to be transformed into relations for the relational model [5]. It integrates each data schema into an application data structure chart as a conceptual model. And the transforms the application data structure chart into a relational tool.

2.2 Case-based reasoning

There is a paradigm that has proven effective in many experimental and applied systems: CBR remembers previous problems and either adapts their solutions or uses their outcomes to evaluate new cases. 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. 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 business application problems.
It is a model of design that directly uses design experience in the form of episodes rather than compiles and generalizes it. There are examples in which cases were stored in entirety using redundant discriminate networks and in pieces [8]. Parallel implementations use multiple can be simultaneously checked [12]. A weighted count of matching features provides one way to select the best case; however, this approach does not take into account that the case itself might determine the importance of a feature. Some approaches to finding the best case are preference heuristics, dimensional analysis, and dynamically changing weighted evaluation functions [1].

3 Structure of ES-SDM
An expert system for strategic decision making (ES-SDM) consists of third major modules, and two kinds of case-bases. The modules are case-bases, user interface, translator, case retriever. The case-bases are domain dependent case-base and domain independent case-base. The user interface module provides interactive question-answering functions. It takes the users’ requirements for strategy decision making, asks the user for specific information, accepts the answer, presents the schema description, and show the designed logical.

The interaction between the system and user are performed by using natural language sentences, from layout, and chart. The translator transforms natural language input sentences for the specifications of requirements. Since natural language processing is beyond the scope of this research, we will not discuss the details of the translator. The case retriever selects a similar case domain independent case-base or domain dependent case-base.

3.1 Representation of design
The 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 decision making 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. This includes normalization cases for guaranteeing the third normal forms. It is a set of cases for application database schema and it is well organized.

This system searches the similar design case by matching the design features with the description of cases in domain dependent case-base. This case-base 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. Traditionally, it has been thought to be easy for users to describe design requirement of strategy decision making through input/output forms. From our experience, we came to conclude that specifying the requirements printed report forms is one of the most convenient methods for naive users.

Mostly, requirement specification in strategy decision making is based on an entity-relationship diagram. However, a naive user is hard to identify the entities and relationships in the enterprise. In this approach, a user explains the design requirements and the dependency among the attributes in a form by using natural language sentences. Design requirements described in natural language sentences should be transformed into internal structures to be processed by a computer system. A relational conceptual graph is an internal representation scheme to describe the structure of dependency among related attributes. A relational
conceptual graph consists of concept nodes and relation nodes. In this paper, transforming the requirement sentences to relational conceptual graph is beyond the scope of this study because it requires ideal natural language processing techniques.

### 3.2 Domain independent/dependent case-base

Human experts have the design knowledge about how a database schema should be normalized. Such design knowledge is domain in domain independent and can be represented as a set of rules or cases. The strategy decision making using domain independent case-base is processed by the flow in next section. The relational conceptual graph is normalized by applying the most relevant case.

#### Fig. 2. Definition of some cases in domain independent case-base

- **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

In ES-SDM, the design task is performed as selecting the relevant previous cases, adapting it to users' requirements, repairing it for new solution, and indexing the new solution into a 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. A case adaptation is a process that builds a rough solution as merging the relevant cases. The new design is added to the domain dependent case-base as a new case.

The design requirements for a form are described by using pre-defined tabular form. 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 in Fig. 3. At the first, case retrieval step selects the relevant previous case in domain dependent case-base. Selected cases are adapted to users' requirements for generating the rough solution. The system briefly

---

**Fig. 3. A flow of ES-SDM by domain dependent case-base**
explains the rough solution to the user. 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.

4 System Design

The retrieval process tries to match the domain-case level first and the form-case level afterwards. After reduces its scope, the system searches the most similar from. The selected cases may not exactly match with the user's requirements. Adaptation rules are needed to find the gaps and fill the missing parts.

4.1 System design method

To generate a rough solution, the system processes the following two steps: ① merge the relevant forms - data structure chart selected by the case retriever, ② remove unnecessary components comparing the design requirements. First, the merging action is started by appending the necessary entities in partial matched form - data structure diagrams to the best form. The generated rough solution has all entities and attributes that are included in previous cases. However the system 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. The system briefly explains the overall design status to help the user to identify the design process, rough solution, design deficiency, and comments to aid the solution refinement.

In explaining the rough solution, the system describes the solution as tabular representation 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 graph that illustrates the relationships among attributes in that relation. 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. We have proposed an idea for the design using a domain independent case-base. If the user confirms the correctness of the final solution that meets the users' requirements completely, the system accepts the attribute of each field in the relation.

4.2 System case learning

Case-based learning algorithms 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-value representation, where features are either predictor or goal features.

A case learning method is distinguished by their processing method they focus on some parts of the case-based learning paradigm while deemphasizing others. To be explicit, all case-based algorithms have at least the following functions: ① pre-processor:

This prepares the input for processing, ② similarity:
This function assesses the similarity of a given case with the previously stored cases in the concept description, ③ prediction: 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. This is called similarity-based generalization. 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.
5 Conclusion
It is important to remember previous experiences for designing a database schema. This paper presents an alternate approach for developing an expert system for strategy decision making using CBR method. 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 schema for the application. During the process, the user can interact with the system to change the case for his 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 normal 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 normal forms. An important finding is that case search space is very small in domain independent case-base because we were able to get third normal forms by 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 technology, we expert to take advantage of recent development of CBR research community in artificial intelligence. The major contribution of this paper, we believe, is the modeling of an expert system with CBR approach to strategy decision making automation. This paper has only presented the overall implementation idea to develop an expert system for strategy decision making by using CBR approach. It is concentrating on the construction of the domain dependent case-base including enough design cases in management application.

References: