

## Formalized Entity Extraction Methodology for Changeable Business Requirements

NAMGYU KIM\*, SANGWON LEE AND SONGCHUN MOON

\**Department of Business Information Technology*

*Kookmin University*

*Seoul, 136-702 Korea*

*Department of Management Engineering*

*Korea Advanced Institute of Science and Technology*

*Seoul, 305-701 Korea*

Without a formal methodology extracting entities from business descriptions, a business requirement in the real world cannot be abstracted correctly into an entity-relationship schema. Once core entities are discovered, we can obtain an *Entity-Relationship Diagram* (ERD) by inserting relationships between/among the relevant entities and by aggregating some attributes into one of the entities or relationships. There have been so many studies on formal guidelines for extracting entities from business descriptions. Most of them adopt a knowledge-based approach which consults a knowledge base to recommend entity candidates. However, the knowledge-based approach usually fails to construct the most appropriate ERD for a given business domain. The approach performs the entity extraction on the stiff premise that an object would be classified as an entity if it happen to be classified as an entity once or more in past applications. The previous studies did not consider the flexibility in the object classification that even the same object could be regarded as either an entity or an attribute according to the various concerns of field workers. In this paper, we discuss some limitations of the previous researches on object classification and propose a new methodology for flexible entity extraction. To evaluate the practicality of the devised methodology, we developed a tool for the methodology and performed a case study on option trading applications with the tool.

**Keyword:** database design methodology, enterprise data modeling, entity-relationship model, requirements analysis, database design automation

### 1. INTRODUCTION

Information loss usually occurs during the data modeling process because a database schema is generated by the abstraction of the real-world. The abstraction [1] emphasizes essential objects in business affairs and reduces or excludes other information of little relevance. In the case of the *Entity-Relationship Model* (ERM) [2], for instance, an *Entity-Relationship Diagram* (ERD) should classify only core objects as entities, treat less important ones as attributes, and exclude every irrelevant information in order to be regarded as a high quality one. Therefore, even an expert data modeler would fail to obtain an appropriate ERD unless he or she has a clear perception of field worker's requirements for a specific business application. Unfortunately, user's requirements might differ from the modeler's perception of the requirements due to a discrepancy between

---

Received February 8, 2006; revised July 19 & August 31, 2006; accepted September 7, 2006.  
Communicated by Ming-Syan Chen.

their backgrounds. An ERD generated from the distorted perception cannot perfectly reflect the original business affairs. This has motivated studies on *Automated Database Design* (ADD) systems which enable a field worker to perform data modeling without reliance on an expert modeler.

It is already known that conceptual design phase is hard to automate owing to its artistic characteristics. As the second best policy, ADD systems for this phase requested a field worker to prepare a list of entities as an initial input and attempted to validate it by use of their embedded knowledge. Although knowledge-based ADD systems can discover most of omitted entities, some omissions would remain undiscovered if there is a deficiency of knowledge. This can be demonstrated with the ERD in Fig. 1. The ERD needs further modification because SCHOOL\_FEE is determined by STUDENT's DEPARTMENT rather than by STUDENT. In a well-refined ERD, DEPARTMENT should be classified as a separate entity which possesses SCHOOL\_FEE as its attribute. If DEPARTMENT has been classified as an entity in previous practices and this information is recorded as knowledge, the system could recommend DEPARTMENT as an entity which owns SCHOOL\_FEE as its attribute. If there is no knowledge about a history of DEPARTMENT, however, a knowledge-based system cannot amend the ERD in Fig. 1. It implies that the correctness of ERD produced by knowledge-based systems could fluctuate according to the quality of knowledge.

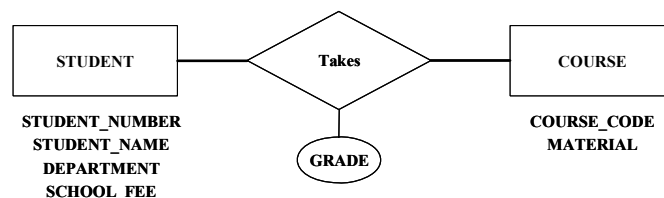


Fig. 1. Inappropriate ERD caused by omitted entity.

Moreover, the knowledge-based approach did not consider the fact that even the same object could be regarded as either an entity in some cases or an attribute in the other cases according to the various concerns of field workers. Let us revisit the ERD in Fig. 1. Assume that SCHOOL\_FEE is not mentioned in the business description because the field worker has no concern for it. In this case, DEPARTMENT needs not to be entitized because it does not lead any additional information. Therefore, entitization of DEPARTMENT based on the system's stale knowledge would generate an absurd ERD which does not accord with the given business description. To obtain an appropriate ERD for a specific application, we should extract entities on the basis of a given business description rather than system's rigid knowledge accumulated from past experiences.

In this paper, we propose a new methodology for entity extraction which enables a field worker to perform conceptual data modeling without any reliance on an expert modeler. The proposed methodology could generate an appropriate ERD for given requirements because it does not use any other information except the business descriptions during the process of entity extraction. However, the entity extraction could not be performed correctly without a kind of preprocessing for semantic redundancy resolution because initial business descriptions written by field workers may contain lots of redun-

dant descriptions about data objects. For this reason, we devise a new graphical model so called *Semantic Association Model* (SAM) as a tool for detecting and resolving the semantic redundancy in business descriptions. Although the SAM is originated from the traditional *Simple Binary Data Model* (SBDM) [3-6], it can be used to schematize correlations among three or more objects while the traditional SBDM can handle only two objects at once. The proposed methodology includes follows: (1) syntactic rules for writing initial business descriptions, (2) semantic rules for refining initial business descriptions, (3) conversion rules for schematizing objects and associations among them in the business descriptions into SAM, (4) mechanisms for detecting and resolving semantic redundancy in the SAM, and (5) algorithms for extracting entities from the SAM.

The remainder of this paper is organized as follows. In section 2, we review previous researches on entity extraction and ADD systems for conceptual database design. The proposed methodology with detailed explanation is presented in section 3. Section 4 presents a case study of the methodology and its tool on option trading applications. Finally, section 5 concludes this paper.

## 2. RELATED WORKS

Few studies on the formal methodology for conceptual data modeling have been found in the literature while so many previous studies on logical and physical phases have accomplished much contribution. There has been an approach [7] to formulate an ERD from data-intensive source codes. However, most approaches for the conceptual modeling have begun their modeling process with analysis of users' requirements. Among the approaches, [8] has been regarded as the most dominant one. It provided guidelines for the entity extraction according to the principle of ERM theory. There are two assertions as follows. Firstly, an object can be classified as an entity if it has descriptive information for itself. Secondly, if a descriptor is multi-valued, the descriptor can be classified as an entity even though it does not have any other descriptors. Most human database designers usually extract entities on the basis of the two assertions.

Contrary to [8] which provided rather abstract guidelines for the entity extraction, [9] presented a statistical mechanism so called *Attribute Synthesis Method* (ASM). To discover entities, ASM picks out objects from given tasks and then performs statistical manipulation on the objects through several phases. During the statistical manipulation, ASM calculates values of three indices such as *Usage Across Tasks* (UAT), *Usage with Other Data* (UOD), and *Usage Ratio* (UR). With the indices, ASM can roughly choose entity candidates. An object has the higher possibility for being classified as an entity if it has higher *UAT*, higher *UOD*, and lower *UR*. ASM however does not define the thresholds for the indices clearly so it makes it ambiguous to determine whether a value of an index is high or low. Unfortunately, the quality of ERD would vary with users' arbitrary selection of the threshold.

There have been many studies [10-15] on automating the phase of conceptual database design. Most of them required a field worker to prepare an entity list as an initial input and attempted to refine it according to the systems' knowledge. Among the studies, *View Creation System* (VCS) [16] is reputed as the dominant one and it has invoked many succeeding studies [17-19]. In VCS, a user submits an entity list, and then VCS

discovers omitted entities by performing lexical analysis and by consulting its knowledge base. However, the lexical analysis cannot discover some omitted entities when the relevant attributes do not share any common morpheme even though they are semantically correlated. On the contrary, the limitations of the knowledge-based modification were already discussed in section 1. It implies that, therefore, we need any other way to design a database without excessive reliance on prior knowledge about object classification.

### 3. REQUIREMENTS-DRIVEN ENTITY EXTRACTION

We propose *Requirements-driven Entity Extraction Methodology* (REEM) as a new entity extraction methodology for changeable business environments. The overall architecture of REEM and its detailed explanations are presented in this section. The methodology consists of five steps as shown in Fig. 2.

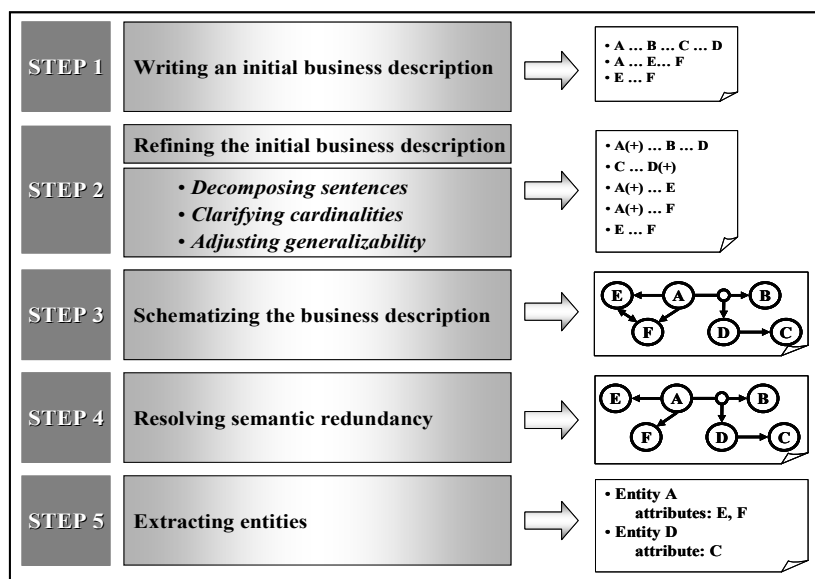


Fig. 2. Overall architecture of REEM.

In STEP 1, a field worker prepares an *Initial Business Description* (IBD) according to some syntactic rules. The IBD is then refined in STEP 2 after going through three internal processes such as *sentence decomposition*, *cardinality clarification*, and *generalizability adjustment*. In STEP 3, the *Refined Business Description* (RBD) is schematized into SAM and STEP 4 generates CSAM by resolving semantic redundancies on the original SAM. Finally, entity extraction is performed in STEP 5.

#### 3.1 Syntactic Rules for Writing Initial Business Description (STEP 1)

A user should write an IBD according to the following syntactic sentence rules.

**Sentence Rule 1** *Capitalization of object's name*: Every object should be capitalized in a business description while other words should be written in lower-case letters.

The rule is established for the purpose of simplifying the object extraction from sentences. Of course, the object extraction can be performed by using a noun dictionary devised in the literature of natural language processing. In this paper, instead of utilizing the contribution of previous studies on natural language processing, we simply employ Sentence Rule 1 to make our methodology and system to concentrate on entity extraction.

**Sentence Rule 2** *Uniqueness and consistency of object's name*: The name of each object should be unique and consistent through the entire business description.

The rule implies that a business description should not contain any homonyms or synonyms of an object's name. That is to say, a single name cannot be used for indicating two or more objects, and neither can two or more names be assigned to the same concept. Homonyms and synonyms should be detected and removed because they might cause ambiguity during the process of semantic acquisition from the business description. The solutions to manage the homonyms and synonyms have been studied in another research literature [20]. To simplify further discussion of this paper, we just exclude homonyms and synonyms by Sentence Rule.

**Sentence Rule 3** *Singularization of object's name*: The name of every object should be written in the singular form.

Mapping cardinality among each object, which is closely related to the singular or plural form of objects in sentences, is core information for our methodology to perform object classification. The system developed for the methodology should accept and validate the cardinality cautiously because the quality of object classification will be critically affected by the correctness of the acquired mapping cardinalities among the inputted objects. To prevent user's mistakes in perceiving and inputting mapping cardinalities, the system allows the user to input cardinalities after an entire business description has been inputted. This implies that distinction between the singular and plural forms of objects in an IBD does not make any difference to further processes of ERD generation. Therefore, we ordain users to write every object in a singular form. The system can acquire the information about the cardinality later through user interactions.

**Sentence Rule 4** *Parallelism among objects*: Multiple sentences of a similar pattern can be compressed into one sentence by using commas.

This rule is established for promoting the convenience of input process and enhancing the readability of a business description. For instance, suppose that there are five objects such as *A*, *B*, *C*, *D*, and *E*. Suppose again that a user wants to enter three sentences: "*A verb B adverb E*," "*A verb C adverb E*," and "*A verb D adverb E*." With Sentence Rule 4, the user can compress the three sentences into one like "*A verb B, C, D adverb E*." The IBD written in conformance with the four rules will be refined in STEP 2.

### 3.2 Refining the Initial Business Description

Because an IBD usually contains ambiguous descriptions, it should be refined before being schematized into SAM. The refining process consists of *sentence decomposition*, *cardinality clarification*, and *generalizability adjustment*. The sentence decomposition can be again divided into syntactic decomposition which decomposes a compressed sentence by commas and semantic decomposition which decomposes a sentence according to semantic dependencies between/among objects.

#### 3.2.1 Syntactic decomposition using parallelism

A sentence compressed by Sentence Rule 3 should be decomposed into original sentences to describe exact semantics among objects. The decomposition is accomplished by performing the reverse process of compression. Let us examine the sentence, “*A, B verb C, D, E adverb F.*” It will be decomposed into the following six sentences each of which contains three objects such as  $(A, C, F)$ ,  $(A, D, F)$ ,  $(A, E, F)$ ,  $(B, C, F)$ ,  $(B, D, F)$ , and  $(B, E, F)$ . Let us consider another case about a real business rule. Sentence 1-1 will be decomposed into six sentences, and Sentence 1-2 is one of them. Every verb is written in a singular form according to Sentence Rule 3.

**Sentence 1-1:** an UNDER\_STUDENT, GRADUATE\_STUDENT is given a GRADE, SEAT, NUMBER in the enrolled COURSE.

**Sentence 1-2:** an UNDER\_STUDENT is given a GRADE in the enrolled COURSE.

#### 3.2.2 Semantic decomposition using mapping cardinality

Another type of sentence refinement is still needed for decomposing every sentence into semantically atomic ones. A sentence can be regarded as an atomic one if it describes minimal unit of a business affair and cannot be decomposed further without information loss. The modification of a non-atomic sentence into atomic ones can be accomplished by separating less relevant objects from the source sentence. Let us examine Sentence 2-1. Although it seems to be describing an atomic business affair about ROOM, STUDENT, and SEX, more careful observation could convince us that it can be decomposed into Sentences 2-2 and 2-3 without any information loss.

**Sentence 2-1:** a ROOM is assigned to a STUDENT in consideration of his or her SEX.

**Sentence 2-2:** a ROOM is assigned to a STUDENT.

**Sentence 2-3:** a STUDENT is distinguished by SEX.

With regard to database schema, all information in Sentence 2-1 can be deduced by composing Sentences 2-2 and 2-3. It results from the fact that one STUDENT can have only one value of ROOM and one value of SEX. It implies that three objects in Sentence 2-1 need not to be mentioned together in one sentence because the correlation between ROOM and SEX can be deduced by the association between ROOM and STUDENT and the other one between STUDENT and SEX. The influence of SEX on arranging ROOM should be implemented by application program rather than database schema. To formally

discuss sentence decomposition using mapping cardinality, we first need to establish a definition for *Semantic Dependency* (SD) among data objects.

**Definition 1** *Semantic dependency*: Let  $X_1$  be an object or a combination of objects and  $X_2$  be another object. If a value of  $X_2$  is always determined uniquely by a value of  $X_1$ , we say that  $X_2$  is semantically dependent on  $X_1$  and depict it as  $SD X_1 \rightarrow X_2$ .  $\square$

Although SD and *Functional Dependency* (FD) are defined on the similar theoretical bases, they clear difference in coverage of target objects. Contrary to FD which depicts dependencies only among attributes in the same relation, SD deals with dependencies among every object before it is classified into an entity or an attribute. A rule for decomposing sentence using mapping cardinality is defined in Rule 1.

**Rule 1** *Sentence decomposition using cardinality*: Suppose that a sentence  $S$  contains  $n$  objects,  $a_1, a_2, \dots,$  and  $a_n$  (for  $n \geq 3$ ). A union of  $n$  objects is represented as  $D$ .  $X_1$  and  $X_2$  are subsets of  $D$ . If  $SD X_1 \rightarrow X_2$  is true,  $S$  should be decomposed into two sentences such that one contains objects in  $\{X_1 \cup X_2\}$  and the other contains objects in  $\{D - X_2\}$ .  $\square$

**Table 1. SDs in sentence 2-1 and their effects on sentence decomposition.**

Detected SD	Objects in Decomposed Sentences	
	New Sentence	Remained Sentence
STUDENT $\rightarrow$ ROOM	STUDENT, ROOM	STUDENT, SEX
ROOM $\rightarrow$ SEX	ROOM, SEX	ROOM, STUDENT
STUDENT $\rightarrow$ SEX	STUDENT, SEX	STUDENT, ROOM

Discovered SDs in Sentence 2-1 and the result of the decomposition are summarized in Table 1. The results would vary with users' different views on which SD is more essential and critical. A sentence containing four or more objects can be decomposed in the similar manner. A sentence is said to be atomic if it cannot be semantically decomposed any more. For instance, there is no SD in any pair of three objects in Sentence 3-1. The only SD in the sentence is  $SD (STUDENT, COURSE) \rightarrow GRADE$ . The sentence is semantically atomic because each of the three objects is participating in the SD as either a determinant or a dependent. Therefore, the sentence cannot be decomposed without information loss.

**Sentence 3-1:** STUDENT gains GRADE in the enrolled COURSE.

### 3.2.3 Cardinality clarification

To complete an RBD, a user should clarify mapping cardinalities between/among objects in each sentence. For instance, let us examine Sentence 4-1 which contains only one semantic dependency,  $SD COURSE \rightarrow PROFESSOR$ . The SD implies that an instance of COURSE can determine the unique instance of PROFESSOR while the opposite is not true. In this case, we attach a ( $\rightarrow$ ) symbol to prior to PROFESSOR. It results Sentence 4-1 to be converted into Sentence 4-2. In general, we attach a ( $\rightarrow$ ) symbol to prior to each object if the object appears in the dependent side of any SD.

**Sentence 4-1:** a PROFESSOR teaches a COURSE.

**Sentence 4-2:** a ( $\rightarrow$ )PROFESSOR teaches a COURSE.

**Sentence 5-1:** a STUDENT gains a GRADE in the enrolled COURSE.

**Sentence 5-2:** a STUDENT gains a ( $\rightarrow$ )GRADE in the enrolled COURSE.

Now, let us discuss cardinality clarification of Sentence 5-1. The sentence is identical to Sentence 3-1 and we have already verified that it is semantically atomic. The only SD detected in the sentence is  $SD (STUDENT, COURSE) \rightarrow GRADE$ . Note that only GRADE appears in the dependent side of the SD. The result modifies Sentence 5-1 into Sentence 5-2 by attaching a ( $\rightarrow$ ) symbol to prior to GRADE. The semantic dependency in Sentence 5-2 does not invoke any ambiguity. The only possible interpretation is that GRADE is semantically dependent on the combination of STUDENT and COURSE. GRADE is not semantically dependent on either one of the other two objects. Otherwise, the sentence may have been decomposed into multiple ones by Rule 1. The cardinality information clarified here plays a core role in STEP 3.

### 3.2.4 Generalizability adjustment

The last process in STEP 2 is to construct generalization hierarchies between objects. To do this, a user should examine every sentence which contains a pattern of “*OBJ\_A is OBJ\_B*” or “*OBJ\_C is classified into OBJ\_D and OBJ\_E.*” When the patterns are detected, the user should determine whether any association related to the *OBJ\_A* (*OBJ\_D* and *OBJ\_E* in the latter pattern) can be more generalized or not. The generalizability adjustment process can be understood easily with the following example.

**Sentence 6-1:** a ( $\rightarrow$ )GRADUATE\_STUDENT is a ( $\rightarrow$ )STUDENT.

**Sentence 6-2:** a GRADUATE\_STUDENT gains a ( $\rightarrow$ )GRADE in the enrolled COURSE.

**Sentence 6-3:** a STUDENT gains a ( $\rightarrow$ )GRADE in the enrolled COURSE.

Sentence 6-1 forms a generalization hierarchy between GRADUATE\_STUDENT and STUDENT. This invokes a need for further examination of some sentences which contain GRADUATE\_STUDENT. A user determines whether GRADUATE\_STUDENT can be replaced with STUDENT in the sentences. Let us suppose that GRADUATE\_STUDENT in Sentence 6-2 can be replaced with STUDENT. As a result, Sentence 6-2 is converted into Sentence 6-3.

### 3.3 Schematizing the Refined Business Description into SAM

In subsection 3.2, we discussed three processes for refining an IBD. In STEP 3, the RBD is converted into SAM which is a newly proposed data model. The SAM enhances expressivity of traditional SBDM by introducing some new components. Contrary to traditional approaches which have considered SBDM and ERM to be mutually exclusive and competitive ones, we regarded SAM as an intermediate tool between business descriptions and ERM. SAM is also used to resolve semantic redundancies in STEP 4. Nodes and links comprise main components of SAM. Each node represents each object in an RBD. By connecting nodes, each link could represent one association among the

**Table 2. Links between two nodes.**

	Sentence in an RBD	Card.	SAM Representation
1	a (→)STUDENT has a (→)STUDENT_NUMBER	1 : 1	
2	a (→)PROFESSOR teaches a COURSE	1 : N	
3	a STUDENT enrolls a COURSE	M : N	
4	a (→)GRADUATE_STUDENT is a (→)STUDENT	1 : 1	

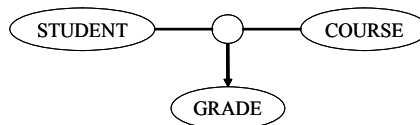


Fig. 3. Usage of a pseudo node.

corresponding objects. The direction of a link is determined by the mapping cardinality of an association. Table 2 summarizes the method to schematize associations between two objects according to their mapping cardinalities.

A bidirectional link in type 1 is used for schematizing mutual dependency between two objects. In type 2, a unidirectional link is used for schematizing dependency from an object to the other one. The link starts from an object without (→) symbol and directs toward the other one. A non-directional link in type 3 is used for indicating no dependency between two objects. The last one in type 4 is used for describing generalization hierarchy. When three or more objects are associated, a new component so called *pseudo node* is connected to each of them and indicates a dependency between an object and a combination of all the others. The usage of a pseudo node can be understood easily with Sentence 7-1 (Fig. 3).

**Sentence 7-1:** a STUDENT gains a (→)GRADE in the enrolled COURSE.

An incoming link into an object implies that the object is dependent on a combination of all the other objects. There is only one object, GRADE, which has an incoming link in Fig. 3. Therefore, there is only one SD,  $SD (STUDENT, COURSE) \rightarrow GRADE$  among the three objects. The incoming link into the node for GRADE cannot be interpreted as the notification of either  $SD STUDENT \rightarrow GRADE$  or  $SD COURSE \rightarrow GRADE$ . If so, the sentence may have been decomposed in the phase of sentence decomposition. In this way, an association among three or more nodes can be schematized clearly with a pseudo node. When a pseudo node is frequently used, SAM can be simplified by making two or more associations share the pseudo node. An SAM in Fig. 4 (a) schematizes two associations, one among *A*, *B*, and *C* and the other among *A*, *B*, and *D*. Both of the two pseudo nodes in Fig. 4 (a) are connected to both of *A* and *B*. The SAM may become more



(a) Original representation.

(b) Simplified representation.

Fig. 4. Merging multiple pseudo nodes into one.

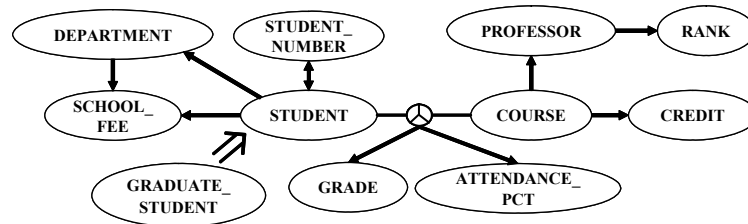


Fig. 5. Schematizing associations in the RBD into SAM.

difficult to read if too many pseudo nodes are used. In such a case, readability can be enhanced by integrating the pseudo nodes into one (Fig. 4 (b)). Note that the SAM in Fig. 4 (b) does not represent an association among four objects, but it is only a concise representation for two associations each of which is connected to three objects.

The SAM's components such as nodes, pseudo nodes, and links can be explained all together with an SAM in Fig. 5 which is formulated from Sentences 8-1 through 8-10.

**Sentence 8-1:** a ( $\rightarrow$ )GRADUATE\_STUDENT is a ( $\rightarrow$ )STUDENT.

**Sentence 8-2:** a STUDENT gains a ( $\rightarrow$ )GRADE in the enrolled COURSE.

**Sentence 8-3:** the ( $\rightarrow$ )ATTENDANCE\_PCT of a STUDENT in the enrolled COURSE is recorded.

**Sentence 8-4:** a COURSE has a ( $\rightarrow$ )CREDIT.

**Sentence 8-5:** a ( $\rightarrow$ )PROFESSOR teaches a COURSE.

**Sentence 8-6:** a ( $\rightarrow$ )RANK is given to a PROFESSOR.

**Sentence 8-7:** a ( $\rightarrow$ )STUDENT has unique ( $\rightarrow$ )STUDENT\_NUMBER.

**Sentence 8-8:** a STUDENT belongs to only one ( $\rightarrow$ )DEPARTMENT.

**Sentence 8-9:** a ( $\rightarrow$ )SCHOOL\_FEE varies according to a DEPARTMENT.

**Sentence 8-10:** a ( $\rightarrow$ )SCHOOL\_FEE is imposed on a STUDENT.

### 3.4 Semantic Redundancy Resolution

The SAM in Fig. 5 contains redundant information due to the duplicate descriptions on business affairs in the IBD. To achieve compactness of SAM by resolving redundant semantic dependencies, we should detect and remove duplicate paths between every pair of nodes. The original SAM can be converted into *Compact SAM* (CSAM) in STEP 4 after the semantic redundancies are removed. In advance to further discussion on a method to detect semantic redundancy, we introduce formal definitions of *semantic path* and *duplicate semantic path*.

**Definition 2** *Semantic path*: Let us suppose that there are two or more nodes and the two of them are  $Nstart$  and  $Nfinish$ . We say that a semantic path is formed from  $Nstart$  to  $Nfinish$  if there is a directional route from  $Nstart$  to  $Nfinish$ . The semantic path from  $Nstart$  to  $Nfinish$  is depicted as  $SPATH(Nstart, Nfinish)$ , and it is defined as an ordered set of objects which have been visited during the traverse.  $\square$

**Definition 3** *Duplicate semantic path*: Let us suppose that there are two or more semantic paths from  $Nstart$  to  $Nfinish$ . We say that there is a duplicate semantic path from  $Nstart$  to  $Nfinish$  if the following condition is satisfied for any  $SPATH_i$  and  $SPATH_j$  (for  $i \neq j$ ), when  $SPATH_i$  and  $SPATH_j$  are the semantic paths from  $Nstart$  to  $Nfinish$ :

$$\{SPATH_i - \{Nstart, Nfinish\}\} \cap \{SPATH_j - \{Nstart, Nfinish\}\} = \emptyset. \quad \square$$

Definition 3 asserts that duplicate semantic paths share the same starting node and the same finishing node but they do not share any node except the two nodes. This can be understood with the SPATHs detected in Fig. 5. Among the 13 SPATHs in Fig. 5, only 4 SPATHs which have the possibility of creating a duplicate semantic path are listed here.

1.  $SPATH(STUDENT\_NUMBER, SCHOOL\_FEE) = \{STUDENT\_NUMBER, STUDENT, SCHOOL\_FEE\}$
2.  $SPATH(STUDENT\_NUMBER, SCHOOL\_FEE) = \{STUDENT\_NUMBER, STUDENT, DEPARTMENT, SCHOOL\_FEE\}$
3.  $SPATH(STUDENT, SCHOOL\_FEE) = \{STUDENT, SCHOOL\_FEE\}$
4.  $SPATH(STUDENT, SCHOOL\_FEE) = \{STUDENT, DEPARTMENT, SCHOOL\_FEE\}$

SPATHs 1 and 2 have the same starting node and the same finishing node. However, they are not regarded as duplicate semantic paths from  $STUDENT\_NUMBER$  to  $SCHOOL\_FEE$  because they share not only the starting and finishing nodes but also another node,  $STUDENT$ . On the contrary, SPATHs 3 and 4 form duplicate semantic paths from  $STUDENT$  to  $SCHOOL\_FEE$  because no node is shared by the two SPATHs except the starting and finishing nodes. It implies that the SAM contains redundant information about the semantic association between  $STUDENT$  and  $SCHOOL\_FEE$ . To resolve the redundancy, any link in either of the two SPATHs should be eliminated.

### 3.5 Entity Extraction

All information about data objects and their associations are already contained in the CSAM. This enables STEP 5 of REEM to extract entities from the CSAM with only one simple rule. This subsection is devoted to present some lemmas for this rule. On the basis of the philosophy of ERM, we consider that an object is important enough to be classified as an entity if and only if it leads descriptive information or it has multi-valued property. According to this philosophy, REEM extracts entities by analyzing links on the CSAM. Nodes for unclassified objects and their possible adjacent link types are enumerated in Table 3. In the table, a node marked with  $U$  represents an unclassified object while a node  $E$  or  $A$  represents that its object has been classified as an entity or an attribute respectively.

**Table 3. Possible link types adjacent to unclassified objects.**

Type	SAM representation	Type	SAM representation	Type	SAM representation
1	$\textcircled{U} \rightarrow \textcircled{A}$	5	$\textcircled{U} \rightarrow \textcircled{A}$	9	$\textcircled{U} \leftrightarrow \textcircled{A}$
2	$\textcircled{U} \rightarrow \textcircled{E}$	6	$\textcircled{U} \leftrightarrow \textcircled{E}$	10	$\textcircled{U} \leftarrow \textcircled{A}$
3	$\textcircled{U} \leftarrow \textcircled{A}$	7	$\textcircled{U} \leftarrow \textcircled{A}$	11	$\textcircled{U} \Rightarrow \textcircled{E}$
4	$\textcircled{U} \leftarrow \textcircled{E}$	8	$\textcircled{U} \text{---} \textcircled{E}$	12	$\textcircled{U} \Leftarrow \textcircled{E}$

Links in types 7, 9, and 10 cannot exist because a generalization hierarchy and a many-to-many association can be established only between entities. Type 3 is another case which cannot exist in CSAM. If the node  $U$  in type 3 is assumed to be an entity, there will be an anomaly that the entity cannot uniquely determine its attribute. If a node  $U$  is representing an attribute, this represents an association between two attributes in the same entity. Note that any association between attributes opens duplicate semantic paths from their entity to one of the attributes (Lemma 1). No duplicate path is allowed to exist in a CSAM, so a CSAM cannot contain any association between attributes. Therefore, type 3 cannot exist in a CSAM regardless of the node  $U$ 's classification.

**Lemma 1** If an unclassified node is connected to a unidirectional or a bidirectional link and the opposite side of the link is already classified as an attribute, the unclassified node should be classified as an entity.

*Proof:* Let us assume that there is a semantic dependency from a node  $a_i$  to a node  $a_j$ . And assume that objects represented by  $a_i$  and  $a_j$  are classified as attributes of an entity  $E$ . By the  $SD$   $a_i \rightarrow a_j$ , we can find the following semantic path:  $SPATH(a_i, a_j) = \{a_i, a_j\}$ . Because every entity can uniquely determine all of its attributes, we can find additional semantic paths:  $SPATH(E, a_i) = \{E, a_i\}$  and  $SPATH(E, a_j) = \{E, a_j\}$ . By combining  $SPATH(E, a_i)$  and  $SPATH(a_i, a_j)$ , we can derive another semantic path,  $SPATH(E, a_j) = \{E, a_i, a_j\}$ . It implies that there are duplicate semantic paths from  $E$  to  $a_j$ :  $E \rightarrow a_j$  and  $E \rightarrow a_i \rightarrow a_j$ . This contradicts CSAM's assumption that there is no duplicate semantic path on it. Therefore, both sides of a unidirectional or a bidirectional link cannot be simultaneously classified as attributes.  $\square$

Now, let us analyze various types of links in Table 3 excluding the types 3, 7, 9, and 10. Firstly, we can find seed objects which can be immediately classified as entities. A sufficient condition for entitizing unclassified objects is presented in Rule 2.

**Lemma 2** If an unclassified node is connected by a double or non-directional link, the node could be classified as an entity immediately. Note that a generalization hierarchy and a many-to-many association can be established only between entities. See the entitizations of types 8, 11, and 12.

**Lemma 3** If an unclassified node is connected to a unidirectional or bidirectional link and the opposite side of the link is classified as an attribute, the unclassified node should be an entity. If not, this contradicts Lemma 1. See the entitizations of types 1 and 5.

**Lemma 4** Let us assume that a node  $E$  is representing an entity while a node  $U$  is unclassified. If there is a link like  $U \rightarrow E$  in a CSAM, the object represented by the node  $U$  should be classified as an entity. Unless, there will be an anomaly that an entity cannot uniquely determine its attribute. See the entitization of a type 2.

**Rule 2** *Immediate entitization of an unclassified object:* By Lemmas 2 through 4, a sufficient condition for entitizing an unclassified object  $U$  is as follows:

there is a link like  $U \Leftarrow E, U \Rightarrow E, U - E, U \rightarrow A, U \leftrightarrow A, \text{ or } U \rightarrow E.$  □

Now, let us establish conditions for classifying an unclassified object as an attribute. An unclassified node in type 4 or 6 can be classified as either an entity or an attribute. In fact, its classification is determined by its terminality. We say that a node is terminal one if it is connected by only one link. If the node  $U$  in type 4 is terminal, it has no reason to be classified as an entity. We classify the node  $U$  as an attribute of the object connected to the other side of the link. Similarly, the node  $U$  in type 6 needs not to be classified as an entity if it is terminal. Therefore, it could be classified as an attribute too. In conclusion, an unclassified node  $U$  in type 4 or 6 is classified as an attribute if it is terminal (Rule 3).

**Rule 3** *Attributization of some terminal nodes:* A necessary and sufficient condition for attributizing an unclassified object  $U$  is as follows:

the object  $U$  is terminal and it is not connected to a link like  $U \Leftarrow E, U \Rightarrow E, U - E, U \rightarrow A, U \leftrightarrow A, \text{ or } U \rightarrow E.$  □

We have so far discussed every type of links in Table 3 except non-terminal nodes in types 4 and 6. If a node  $U$  is non-terminal, it certainly has at least one more link in addition to the one appears in the table. If the additional link is one of the links in types 1, 2, 5, 8, 11, and 12, the node  $U$  may have been already classified as an entity. If it has not been classified yet, the additional links would be in types 4 and 6 (Fig. 6). In the every case of Fig. 6, the node  $U$  should be classified as an entity (Lemma 5). If not, there will be an anomaly that one attribute is owned by two different entities simultaneously.

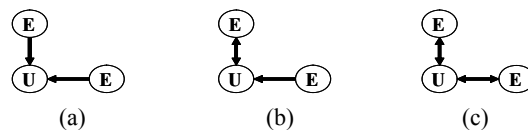


Fig. 6. Possible links adjacent to a non-terminal node  $U$  in types 4 and 6.

**Lemma 5** Let us suppose that a node  $E$  is representing an entity-classified object while a node  $U$  is an unclassified one. If  $U$  is a non-terminal one and there is a link like  $U \leftarrow E$  or  $U \leftrightarrow E$ , the node  $U$  should be classified as an entity.

By integrating Lemma 5 and Rules 2 and 3, we can derive the rule of thumb for object classification (Rule 4). Let us assume a CSAM which can be obtained from the SAM

in Fig. 5 by removing the direct association between STUDENT and SCHOOL\_FEE. By applying Rule 4 to the CSAM, REEM can extract DEPARTMENT, STUDENT, COURSE, and PROFESSOR as entities.

**Rule 4** *Rule of thumb for object classification:* If and only if a node in a CSAM is connected to only one link and the link is a bidirectional or an incoming one, the node is classified as an attribute and all other nodes are classified as entities.  $\square$

## 4. CASE STUDY FOR OPTION TRADING APPLICATIONS

### 4.1 Experiment Design

In this section, we analyze the practicality of the proposed modeling methodology. There would be three ways to show the practicality of a new methodology. The first method is an analytical one which provides theoretical foundation and formal derivation for a methodology. However, in the case of a conceptual modeling methodology, this method would not be the best choice of estimating the practicality of the methodology because any database schema cannot be regarded as the one and only correct answer to a certain business affair. Owing to this artistic aspect of the data modeling, we do not adopt the analytic method to verify the practicality of our methodology.

The second method would be to show the relative superiority of a new methodology by comparing it with other methodologies. But, a quite difference in assumptions between VCS and REEM makes it difficult to show the superiority of REEM by this method. REEM performs the entity extraction process while field workers are responsible for the entity extraction in the case of VCS. In the case of the VCS modeling, a difference in the amount of users' knowledge about data modeling would bring about a difference in the quality of the outcome. Moreover, while REEM can perform the modeling independently of knowledge base, the outcome quality of VCS would vary according to the contents of knowledge base. It implies that the simple comparison between the outcomes of VCS and REEM is not admirable because the outcome quality of VCS would be determined by the contents of knowledge base, not by the methodology. So, we decided to adopt another substitute method to verify the practicality of REEM.

The last substitute is to develop a tool for the methodology and apply the tool to real business affairs. Even though this method does not provide theoretical foundations, it is widely used for empirical tests in various research areas. We have developed the prototype of the tool which adopts REEM as its modeling methodology. The prototype dubbed as *Business Data Modeler* (BizData) was implemented with Microsoft Visual Studio 6.0 and operates on Windows XP computers. To evaluate the practicality of BizData, we performed an experiment on database design for option trading applications. Business descriptions appeared in textbooks [21, 22] are inappropriate for the sources of the experiment, because they are too well-formed as compared with the one used in the real business field. For a more realistic experiment, we used a business description on option trading applications which is written by field workers in the domain.

## 4.2 Experiments

### 4.2.1 Initial business description on option trading applications

Business rules of the target application could be summarized as follows. Option commodities are derived from listed individual stocks. To submit buy-or-sell orders for the option commodities, every customer should open one or more accounts at security companies. On accepting the customer's order, the security company submits corresponding quota to a stock exchange market. When a contract is concluded between bid quotations and ask quotations, contracted options are accumulated in a customer's account in a form of unsettled contracts. Each unsettled contracts can be cleared by a settlement on an expiration date of the target option. The IBD used in this experiment covers six parts of option trading applications, which are Opening Accounts, Option Commodities and Individual Stocks, Orders and Quotations, Managing Unsettled Contracts, Margin Deposit, and Settlement in Stock Exchange. BizData loads the IBD in a batch. The IBD loaded by the Business Description Loader module appears in the upper box of Fig. 7. When a user presses the Parse button, the names of behaviors and objects in each sentence are extracted and listed in the lower box. The comment in the bottom of the window notifies that 114 data objects appear 230 times in 70 sentences of the IBD.

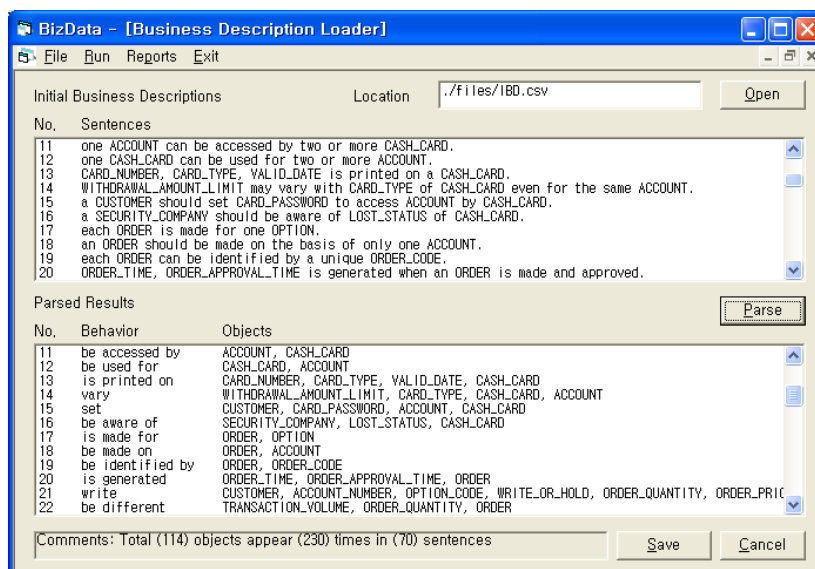


Fig. 7. Loading the IBD into BizData.

### 4.2.2 Refined business description on option trading systems

Syntactic Decomposer, the first sub-module for refining an IBD, decomposes the compressed sentences containing commas into multiple ones. In Fig. 8, Sentence 44 in the IBD is decomposed into two sentences during the syntactic decomposition process. The upper box of Fig. 8 shows the list of the compressed sentences in the IBD. As soon

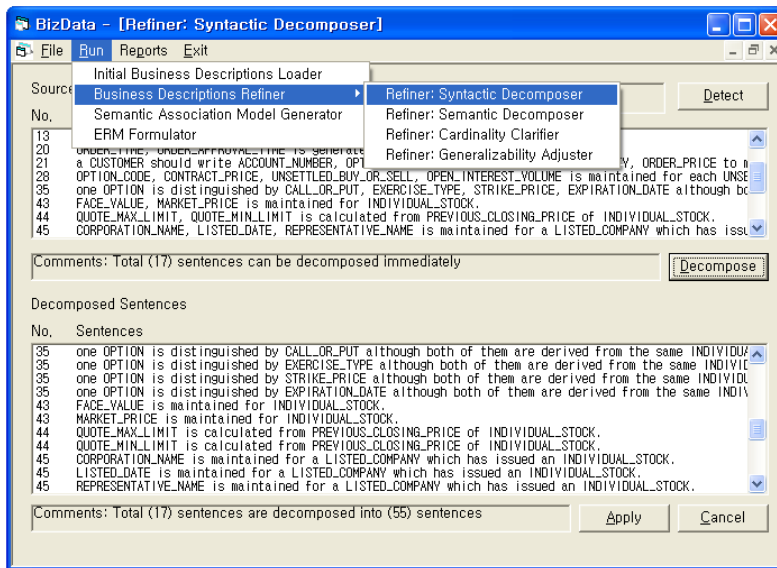


Fig. 8. Syntactic decomposition of the compressed sentences.

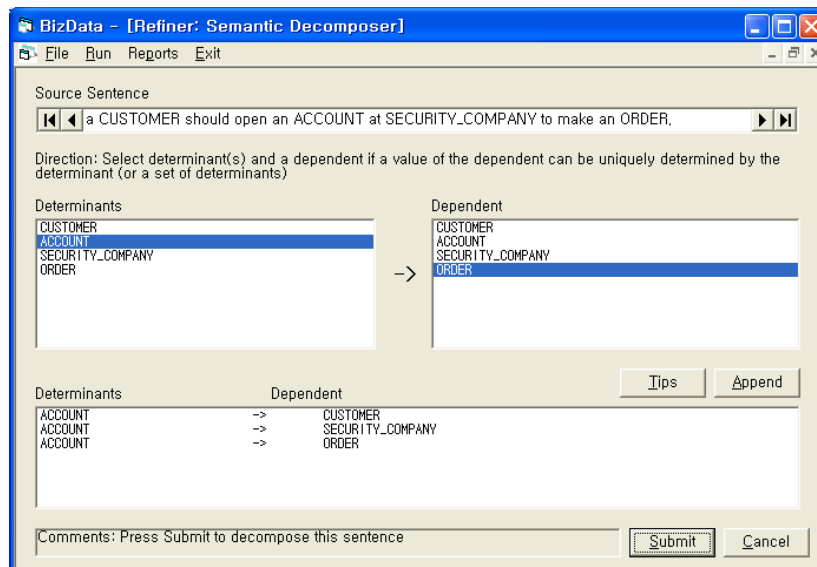


Fig. 9. Semantic dependency acquisition.

as the user presses the Decompose button, the decomposed results appear in the lower box. This process is automatically executed with one click and without any user interaction.

In order that Semantic Decomposer, the second sub-module for the refinement of IBD, may decompose sentences, BizData should be aware of all the semantic dependencies among data objects. The dialog window for acquiring semantic dependencies from

the user appears in Fig. 9. The window shows that the user can submit several dependencies in the same sentence at a time. The source sentence will be decomposed into several atomic sentences according to the submitted dependency. Among the 108 IBD sentences, only 62 sentences are selected as the target for semantic decomposition because each of them contains three or more objects. The 62 sentences are decomposed into 134 ones during this process. After 49 redundant sentences are removed, 131 sentences are finally left in the IBD.

The third sub-module for the IBD refinement, Cardinality Clarifier, acquires mapping cardinalities among objects through a dialogue window which is similar to the window used in the semantic decomposition process. Only 39 dialogues are performed because cardinalities in the other sentences have been already discovered during the semantic decomposition process. The last sub-module, Generalizability Adjuster (Fig. 10), reorganizes the associations among objects according to the generalization hierarchy. Two sentences building up the generalization hierarchy are found. One of them, which contains `INDIVIDUAL_STOCK` and `UNDERLYING_ASSET`, appears in the upper box of Fig. 10. Fig. 10 shows the adjustment process for 13 sentences in which `INDIVIDUAL_STOCK` might be replaced with `UNDERLYING_ASSET`. Among the 13 sentences, the user agrees that `INDIVIDUAL_STOCK` can be substituted with `UNDERLYING_ASSET` in 5 sentences. The IBD after the generalization adjustment process is renamed RBD. The RBD with 131 sentences is used as the input for BizData's SAM creation.

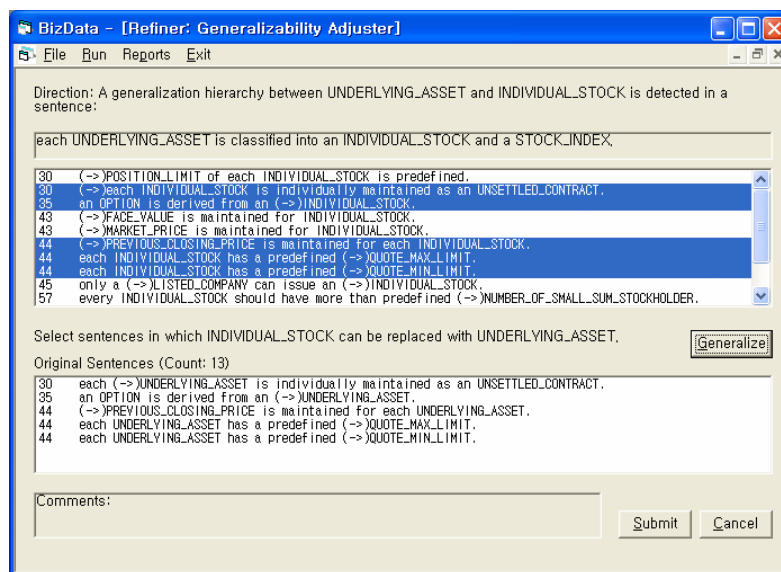


Fig. 10. Generalizability adjustment.

#### 4.2.3 Compact semantic association diagram on option trading applications

The RBD is transformed into the graphical model by SAM generator. This transformation does not need any user interaction and so can be executed immediately with only one click. We manually schematized the result in Fig. 11. This initial SAM contains





for ERWin or Rational Rose, which are the predominant tools for the various phases of logical and physical modeling. The performance of BizData can be evaluated from the two aspects of convenience and correctness, which is analyzed in this section.

To evaluate the practicality of the tool, not only the correctness of the result but also the convenience of the usage should be analyzed. Clearly, the number of user interactions for exchanging business information would increase as the size of target enterprise gets larger. Therefore, to fairly analyze the convenience of BizData, we measure *Rate of User Interaction* (RUI) which is the ratio of the number of user interactions to the number of data objects. Totally 103 user interactions have been invoked during the whole process of the modeling. Because the number of objects is 114, the value of RUI would be 90%. Among the interactions, 60% of them occurred during the semantic decomposition process. It implies that comparatively many interactions are needed for decomposing complex sentences into atomic ones. The 38% of the interactions appeared during the cardinality acquisition. This overhead could be reduced if we develop more efficient algorithms for reusing the information acquired in the semantic decomposition process. The interactions during the sentence reorganization by the generalizability adjustment were only 2% of the whole interactions, which would be a negligible level.

Data modeling is often regarded as being in the domain of an art rather than in the domain of a science. It implies that any database schema cannot be regarded as the only correct answer to a certain business affair and many appropriate results may exist according to modeler's concerns. Hence, to analyze the correctness of the experimental output, we should select the contrastive ERM first which is appropriate enough to be regarded as one of the correct answers. We assumed that the ERM formulated by a professional modeler from the same business description in subsection 4.2.1 is a correct one and evaluated the correctness of BizData by comparing the ERM with the one generated from BizData. As a performance index for the correctness, *Rate of Inappropriate Modeling* (RIM) is used. This index measures the difference between the human modeler and BizData in object classifications or attribute ownerships. There are two kinds of RIM, *RIM-Classification* (RIM-C) and *RIM-Ownership* (RIM-O). RIM-C measures the difference between BizData and the human modeler in determining the type (*i.e.* an entity or an attribute) of data objects. On the other hand, RIM-O measures the difference between BizData and the human modeler in determining the owner (*i.e.* an entity or a relationship) of attributes. The expert modeler agreed fully with the results of BizData's object classification. That is to say, the modeler agreed that 114 data objects should be classified into 22 entities and 92 attributes. In this experiment, therefore, the value of RIM-C is 0%. However, he disagreed slightly with the ownership of 4 attributes which are represented in an italic style in Fig. 13. While BizData recognizes the owners of the 4 attributes as entities, the human modeler regards the attributes as being owned by relationships. Therefore, the value of RIM-O calculated in this experiment is 4.3%. Excessive decomposition during the semantic decomposition process is thought to be the main cause of the inappropriate ownerships. Judging from the 0% of RIM-C and less than 5% of RIM-O, we can say that the quality of ERM formulated by BizData is as good as the one designed by the expert modeler.

In this section, we analyzed the practicality of our methodology and the devised system from the viewpoints of convenience and correctness. A few user interactions were required during the modeling process with BizData and the quality of the final result was

not quite different from that of a human modeler. There could be an assertion that it would be better for the system to be equipped with knowledge base for simplifying the process of writing business descriptions and reducing the number of user interactions. However, a knowledge-based tool has a clear limitation that the quality of its output would be affected according to the contents of knowledge base accumulated during the learning process, although acceptable results are expected in the most cases. Moreover, the knowledge-based system cannot operate until a certain level of knowledge for the target business application is accumulated. To overcome the limits, we have attempted to devise the modeling methodology and the automation tool which can perform conceptual modeling on the basis of business descriptions instead of the knowledge base. However, it is clear that a cautious utilization of the knowledge base to the restricted part of data modeling can offer more convenience to users. Therefore, the future studies need to investigate an efficient method to utilize the knowledge base.

## 5. CONCLUSION

Without a formal methodology for extracting entities from given business descriptions, business requirements in real world cannot be abstracted correctly into an entity-relationship schema. Once the entities are discovered, the whole ERD could be obtained simply by aggregating some attributes into one of the extracted entities and by inserting relationships between/among the relevant entities. Most of traditional knowledge-based entity extraction methodologies have common limitations that they cannot be applied to changeable business environments. The limitations come from the fact that the knowledge acquired from the past practices is not valid any longer if the business requirements have changed. In this paper, we proposed a formal methodology so called REEM for extracting entities from given business descriptions. REEM is expected to generate the appropriate ERD for given business requirements because its process is dependent only on the given business requirements instead of past practices in the similar business domain.

In spite of continual efforts to provide automated modeling tools over the past couple of decades, most tools are commonly suited only for well-trained data modelers. In addition, the tools cannot be applied to changeable business environments because they are heavily dependent on stale knowledge acquired from past experiences. To overcome these limitations, we proposed a new automated database design system called BizData which adopts REEM as the design methodology. The only input needed for BizData is an enterprise-wide business description in which entities and attributes appear together without any distinction. BizData performs its modeling process on the basis of information gathered from the business description, not from past knowledge. Because the current prototype of BizData deals only with formulating an ERM from a business description, the utility of BizData can be maximized when it is used as a front-end tool for ERWin or Rational Rose, which are the predominant tools for performing the various phases of logical and physical modeling. The results of our experiments imply that an enterprise could save much time and cost by minimizing the number of meetings for constructing an enterprise-wide database because the field workers can perform data modeling without any reliance on professional data modelers.

## REFERENCES

1. C. Batini, S. Ceri, and S. Navathe, *Conceptual Database Design: An Entity-Relationship Approach*, The Benjamin/Cummings Publishing, California, 1992.
2. P. P. Chen, "The entity-relationship model – Toward a unified view of data," *ACM Transactions on Database Systems*, Vol. 1, 1976, pp. 9-36.
3. J. R. Abrial, "Data semantics," *Data Base Management*, 1974, pp. 1-60.
4. R. Hull and R. King, "Semantic database modeling: survey, applications, and research issues," *ACM Computing Surveys*, Vol. 19, 1987, pp. 201-260.
5. J. Peckham and F. Maryanski, "Semantic data models," *ACM Computing Surveys*, Vol. 20, 1988, pp. 153-189.
6. J. M. Smith and D. C. P. Smith, "Database abstractions: association," *Communications of the ACM*, Vol. 20, 1977, pp. 405-413.
7. H. Yang and W. C. Chu, "Acquisition of entity relationship models for maintenance – Dealing with data intensive programs in a transformation system," *Journal of Information Science and Engineering*, Vol. 15, 1999, pp. 173-198.
8. T. J. Teorey, D. Yang, and J. P. Fry, "A logical design methodology for relational databases using the extended entity-relationship model," *ACM Computing Surveys*, Vol. 18, 1986, pp. 197-222.
9. T. J. Teorey and J. P. Fry, *Design of Database Structures*, Prentice-Hall, New Jersey, 1982.
10. M. Bouzeghoub, G. Gardarin, and E. Metais, "Database design tools: an expert system approach," in *Proceedings of the 11th International Conference on Very Large Data Bases*, 1985, pp. 82-95.
11. J. Choobinch, M. V. Mannino, J. F. Nunamaker, and B. R. Konsynski, "An expert database design system based on analysis of forms," *IEEE Transactions on Software Engineering*, Vol. 14, 1988, pp. 242-253.
12. A. Dogac, B. Yuruten, and S. Spaccapietra, "A generalized expert system for database design," *IEEE Transactions on Software Engineering*, Vol. 15, 1989, pp. 479-491.
13. S. A. Noah and M. Williams, "Exploring and validating the contributions of real-world knowledge to the diagnostic performance of automated database design tools," in *Proceedings of the 5th IEEE International Conference on Automated Software Engineering*, 2000, pp. 177-185.
14. P. Shoval and M. E. Chaime, "ADDS: a system for automatic database schema design based on the binary-relationship model," *Data and Knowledge Engineering*, Vol. 2, 1987, pp. 123-144.
15. V. C. Storey and R. C. Goldstein, "Knowledge-based approaches to database design," *MIS Quarterly*, Vol. 17, 1993, pp. 25-46.
16. V. C. Storey, *View Creation: An Expert System for Database Design*, International Center for Information Technologies, Washington, D. C., 1988.
17. V. C. Storey, R. H. L. Chiang, D. Dey, R. C. Goldstein, and S. Sundaresan, "Database design with common sense business reasoning and learning," *ACM Transactions on Database Systems*, Vol. 22, 1997, pp. 471-512.
18. V. C. Storey, R. C. Goldstein, and H. Ullrich, "Naive semantics to support automated database design," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 14,

- 2002, pp. 1-12.
19. Y. Wand, V. C. Storey, and R. Weber, "An ontological analysis of the relationship construct in conceptual modeling," *ACM Transactions on Database Systems*, Vol. 24, 1999, pp. 494-528.
  20. S. C. S. Chen, M. Yu, Y. Y. Yao, S. Y. Hwang, and B. P. Lin, "VDBMS: a solution to integrating heterogeneous data sources," *Journal of Information Science and Engineering*, Vol. 13, 1997, pp. 585-603.
  21. R. Ramakrishnan and J. Gehrke, *Database Management Systems*, 3rd ed., McGraw-Hill Companies, New York, 2003.
  22. L. Silverston, *The Data Model Resource Book*, revised ed., John Wiley & Sons, New York, 2001.



**Namgyu Kim** received the B.S. degree in Computer Engineering from Seoul National University in 1998 and Ph.D. degree in Management Engineering from Korea Advanced Institute of Science and Technology (KAIST) in 2007. He has been working for Kookmin University since then. His current research interests include enterprise data modeling and data mining.



**Sangwon Lee** received the B.S. degree in Mathematics from Hanyang University, Korea in 1995 and M.B.A. degree in Management Information Systems from Korea Advanced Institute of Science and Technology (KAIST) in 2002. He is a Ph.D. candidate in Management Engineering of KAIST. His current research interests include enterprise data modeling, data warehousing and XML.



**Songchun Moon** received his Ph.D. degree in Computer Science from the University of Illinois at Urbana-Champaign in 1985. He has been working for Korea Advanced Institute of Science and Technology (KAIST) since then. He has developed a multi-user relational database management system, IM, which is the first prototype in Korea in 1990 and a distributed database management system, DIME, first ever in Korea in 1992. His research interests include enterprise data modeling, security, privacy, piracy, and data warehousing.