

데이터 모델 재사용을 위한 사례기반추론 프레임워크

이재식*·한재홍**

Case-Based Reasoning Framework for Data Model Reuse

Jae Sik Lee*·Jae Hong Han**

요 약

A data model is a diagram that describes the properties of different categories of data and the associations among them within a business or information system. In spite of its importance and usefulness, data modeling activity requires not only a lot of time and effort but also extensive experience and expertise. The data models for similar business areas are analogous to one another. Therefore, it is reasonable to reuse the already-developed data models if the target business area is similar to what we have already analyzed before. In this research, we develop a case-based reasoning system for data model reuse, which we shall call CB-DM Reuser (Case-Based Data Model Reuser). CB-DM Reuser consists of four subsystems: the graphic user interface to interact with end user, the data model management system to build new data model, the case base to store the past data models, and the knowledge base to store data modeling and data model reusing knowledge. We present the functionality of CB-DM Reuser and show how it works on real-life application.

Keyword : Data Model, Data Modeling, Software Reuse, Systems Analysis, Case-based Reasoning.

*아주대학교 경영대학 부교수

**아주대학교 대학원 경영학과 박사과정

1. 서론

정보 시스템 구축을 위한 전통적인 구조적 분석 기법은 프로세스에 대한 관점을 중심으로 시스템을 분석하는 기법이다. 그러나 관계형 데이터베이스의 성능 향상과 80년대 말 발표된 정보공학 방법론의 영향으로 프로세스에 대한 관점보다는 데이터 모델에 대한 중요성이 강조되고 있다. 구조적 분석 기법에서는 프로세스 중심의 자료흐름도(Data Flow Diagram)에 기반을 두고 자료흐름도에 종속적인 데이터 사전(Data Dictionary)을 작성하는 반면에 데이터 모델은 프로세스와는 독립적으로 데이터를 명세화 한 것이다. 데이터 모델은 데이터에 비해 상대적으로 변경이 많은 업무처리 프로세스에 종속되지 않기 때문에, 이를 바탕으로 분석한 시스템은 안정성(stability)과 신뢰성이 높다. 이러한 이유로 데이터 모델은 시스템 분석 과정에서의 산출물로서 뿐만 아니라 기업 및 조직에서 전사적 정보 자원 관리(Enterprise Information Resource Management)의 측면에서도 중요한 대상이 되고 있다[Martin,1990], [McFadden and Hoffer, 1991]

데이터 모델은 분석 대상이 되는 특정 분야 또는 조직이 갖고 있는 데이터와 업무 규칙(Business Rule)을 추상화하고 체계화하여 논리적으로 표현한 모델이다. 그러나 특정 분야의 데이터 모델을 만드는 데이터 모델링 과정에는 여러가지 어려움이 존재하고 있다. 인간은 본질적으로 업무 절차를 논리적으로 표현하거나 체계화하는 것에는 익숙하지만, 데이터를 논리적으로 표현하거나 체계화하는 것에는 익숙하지 않다. 그러므로 프로세스 모델 중심의 분석보다는 데이터 모델 중심의 분석에 많은 시간과 노

력이 요구된다[Flavin, 1981],[Barker, 1989]. 또한 현실적인 측면에서도 프로세스 모델에 익숙한 시스템 분석가들에 비해 데이터 모델에 익숙한 시스템 분석가들이 적다는 점이 데이터 모델링의 어려움을 가중시키고 있다[Fleming and von Halle, 1989].

본 연구에서는 데이터 모델링 과정에서의 시간과 노력을 줄이고 데이터 모델의 완전성을 지원하기 위한 하나의 대안으로, 과거의 데이터 모델을 재사용하여 데이터 모델링을 지원하는 사례기반추론(Case-Based Reasoning) 시스템을 제시한다. 본 연구에서 제시하는 시스템은 과거에 수립된 데이터 모델들을 사례 베이스에 저장하고 있으며, 사용자가 데이터 모델링을 수행하는 과정에서 시스템이 저장하고 있는 유사한 과거의 데이터 모델의 전부 또는 일부를 제공함으로써 데이터 모델링을 지원한다.

본 논문은 5절로 구성되어 있다. 다음 절에서는 본 연구와 관련된 주제들인 데이터 모델, 사례기반추론, 소프트웨어 재사용과 관련된 연구들을 살펴보고, 제 3절에서는 실제 시스템을 개발하기 위해 필요한 여러가지 고려 요소들과 데이터 모델 재사용을 위한 사례기반추론 시스템의 구성 요소들을 설명한다. 제 4절에서는 본 연구에서 개발한 시스템의 사용 예를 제시하고, 제 5절에서는 본 연구의 의의와 한계를 살펴보고 결론을 맺는다.

2. 관련 연구

2.1 데이터 모델

데이터 모델은 1976년 Chen에 의해 발표된 '개체관계 모델(Entity-Relationship Model)'

[Chen, 1976]을 근간으로 조직 내의 정보와 업무 규칙을 개체(Entity), 그들 간의 관계(Relationship), 그리고 개체의 속성(Attribute) 등으로 표현한 모델이다. Chen의 개체관계 모델은 두 가지 방향에서 큰 발전을 이루어 왔다. 첫째는 현실을 보다 많이 모델에 반영할 수 있도록 데이터 모델링에 사용되는 부호(Notation)를 확장하려는 노력이고[Teorey and Fry, 1982], [Storey, 1991], 둘째는 정보공학과 같은 시스템 개발 방법론과의 통합을 통하여 개체관계 모델의 적용 영역을 확장하고자 하는 노력이다 [Martin, 1990], [Finkelstein, 1992]. 이러한 두 가지 노력을 통하여 개체관계 모델은 데이터 모델의 주요한 구성 요소로 인정을 받게 되었다[McFadden and Hoffer, 1991], [Barker, 1989], [Hay, 1996]. 데이터 모델은 개체관계도(Entity-Relationship Diagram: ERD)와 이의 명세를 위한 사전(dictionary)으로 구성된다. ERD는 몇 가지의 부호를 사용하여 데이터를 표현하기 때문에 간결하고 이해하기 쉽다는 장점은 있으나, 세부적인 내용은 표현하지 못한다. 이에 대한 보완으로서 ERD에 대한 세부 사항을 텍스트로 기록하는 사전이 필요하다. 본 연구에서 데이터 모델의 작성은 ERD의 작성으로 국한시킨다.

개체는 해당 업무에서 데이터를 관리하고 유지하여야 할 대상이 되는 사람의 역할, 장소, 사물, 사건 그리고 개념 등을 말한다. 개체를 파악하는 가장 일반적인 기준은 그 개체에 대한 데이터를 저장 관리하여야 할 필요성의 유무이다. ERD에서 개체는 사각형으로 나타내며, 개체 이름은 사각형 안에 표기한다. ERD에서 표기하지 못하는 개체 정의, 동의어 등과 같은 개체의 특성은 별도로 문서화한다.

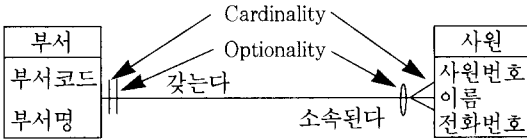
관계는 두 개체 사이의 연관성을 나타내는 논리적인 연결(link)이다. 관계는 많은 경우에 조

직 내의 업무 규칙을 표현한다. 관계는 두 개체 사이를 연결하는 직선으로 나타내는데, 그 특성으로 옵션성(Optionality)과 카디널리티(Cardinality)를 표시하여야 한다. 옵션성은 두 개체 사이의 관계가 절대적이냐 선택적이냐를 표현하는 것이다. 옵션성을 표시하는 부호는 필수적 관계(Mandatory)를 나타내는 작은 직선과 선택적 관계(Optional)를 나타내는 작은 타원이 있다. 카디널리티는 두 개체 사이의 연관 정도를 나타낸다. 한 개체의 인스턴스 하나가 상대 개체의 인스턴스 몇 개와 연관성을 갖는지를 표현하는 것으로서 두 종류의 부호가 있다. 하나의 인스턴스와의 연관 관계는 짧은 직선으로 표시하며 복수의 인스턴스와의 연관 관계는 속칭 까마귀발(crow foot)로 표시한다. 관계에서의 카디널리티 유형은 1:1 관계, 1:M 관계, M:M 관계 등 3가지 유형이 있다.

속성은 개체를 구성하는 단위 데이터 항목으로서 업무에서 발생하는 사실 또는 정보를 묘사하는 가장 상세한 수준의 항목이다. 예를 들어 사원이라는 개체는 사원번호, 이름, 전화번호 등과 같은 속성으로 구성된다. 식별자(Identifier)라고 부르는 특수한 속성은 개체 인스턴스를 식별할 수 있는 속성을 말하며, 널(null) 값이 허용되지 않고, 그 값들이 유일(unique)하여야 한다. 속성에 대한 세부적인 내용은 속성의 정의, 속성의 동의어, 속성의 유일성(Uniqueness), 널 값의 허용 여부 등의 항목을 중심으로 문서화한다.

ERD는 위에서 설명한 개체, 관계, 속성 요소들을 이용하여 분석 영역의 데이터와 업무 규칙을 체계적으로 표현한 것이다. 예를 들어 <그림 2-1>의 ERD는 부서와 사원에 대한 데이터와 그들 사이의 업무 규칙을 보여주고 있다. 즉, 부서는 부서코드와 부서명이라는 단위 데이터 항목

으로 구성되며, 사원은 반드시 하나의 부서에 소속되어야 하며 부서에는 사원이 없거나 여러 명이 소속될 수 있다는 것을 표현하고 있다.



〈그림 2-1〉 ERD의 예

데이터 모델링은 해당 문제 영역의 데이터 모델을 만드는 과정으로서 일반적으로 다음과 같은 절차를 갖고 있다. 첫째, 문제 영역에서 개체들을 파악하고 둘째, 문제 영역의 업무 규칙을 통하여 개체들 사이의 관계들을 파악한다. 셋째, 각 개체의 식별자와 속성들을 파악하여 초기 데이터 모델을 완성한다. 넷째, 초기 데이터 모델을 정규화(Normalization)를 통하여 일반적으로 제 3 정규형(Third Normal Form)을 만족하는 데이터 모델을 완성한다. 시스템 분석가는 해당 문제 영역의 최종 사용자와의 인터뷰 및 각종 장표를 포함한 관련 서류의 분석을 통하여 데이터 모델을 구축하게 된다. 데이터 모델링 과정은 지식 공학 분야에서의 지식 획득 과정과 마찬가지로 데이터 관점에서의 업무 관련 지식을 사용자로부터 획득해 가는 과정이다. 그러므로 많은 시간과 노력이 요구된다.

Hay는 이러한 데이터 모델링 과정의 어려움을 극복하기 위한 하나의 대안으로서 유사한 문제 영역들에서 발견된 일반화되고 단순화된 데이터 모델 패턴들을 통하여 데이터 모델링 과정을 용이하게 하는 데이터 모델 패턴 기반 접근법을 제안하였다[Hay, 1996]. Hay의 제안은 유사한 분야의 샘플들을 이용함으로써 데이터 모델링 과정을 용이하게 하는 측면에서는 큰 기여를 하였으나 다음과 같은 측면에서 미진한 부

분이 존재한다. 첫째, 업무의 비정형화 정도가 높아질수록 일반화의 과정이 어렵고, 일반화된 모델의 의미가 약해진다. 예를 들어 기업의 업무 중 비정형화 정도가 높은 연구 개발 분야에서는 산업분야, 기업 규모, 기업 전략 등에 따라 일반화가 불가능하다고 할 수 있다. 둘째, 정형화 정도가 높은 업무 분야임에도 불구하고 데이터 모델의 일반화 정도에 따라 완전한 데이터 모델을 구축하기 위해서 요구되는 노력이 달라질 수 있다는 것이다. 즉, 일반화 정도가 높은 데이터 모델 패턴을 이용하면 적용 가능한 분야가 많아지지만 완전한 데이터 모델을 구축하기 위해 요구되는 노력이 많아지고, 그 반대로 좀더 세분화된 데이터 모델 패턴을 이용하면 모델링에 요구되는 노력이 적어지는 반면에 적용 분야가 좁아지는 상충 관계에 놓이게 된다.

2.2 사례 기반 추론

인공지능 기법의 하나인 사례 기반 추론은 기억 장치에서 현재의 문제와 유사한 과거의 문제를 찾고, 과거의 문제와 현재의 문제 간의 차이점을 분석하여 과거 문제의 해법을 현재의 문제에 알맞게 수정하여 문제를 풀어가는 기법이다 [Riesbeck and Schank, 1989]. 사례 기반 추론이 갖는 기본적인 아이디어는 인간이 과거의 문제를 해결하기 위해 사용한 해법을 수정하여 새로운 문제의 해결에 사용한다는 것이며, 이와 같은 문제 해결 과정의 재사용을 통하여 자동적인 학습이 가능해진다. 특히 사례 기반 시스템에서는 지식을 사례의 형태로 저장하기 때문에 기존의 인공지능 기법의 문제점으로 지적 되어온 지식 획득 병목현상의 문제를 완화할 수 있다는 장점이 있다.

Aamodt와 Plaza는 〈그림 2-2〉와 같이 사례 기

반 추론 과정을 4R이라고 부르는 4단계로 나누어 보았다[Aamodt and Plaza, 1996].

1) 검색(Retrieval)

검색은 입력된 문제의 속성들에 대하여 가장 잘 일치하는 과거의 사례를 찾는 것이다. 검색은 속성 확인, 탐색, 초기 일치, 선택의 순서로 이루어진다. 속성 확인에서는 입력 문제의 어떤 속성의 집합을 유사 사례 탐색에 사용할 것인지를 결정하여 탐색의 기준으로 사용한다. 초기 일치는 주어진 유사성의 기준을 넘는 사례들을 제시하는 것이다. 그리고 선택에서는 가장 유사한 사례를 선택한다.

2) 재사용(Reuse)

검색된 사례가 제공하는 해의 재사용에서는 두가지 측면을 고려하여야 한다. 첫째는 입력 사례와 과거 사례간의 차이이고, 둘째는 검색된 사례의 어떤 부분이 새 사례로 전이될 것인가이다. 재사용의 방법은 검색된 사례의 해를 그대로 복사해 사용하는 방법과 검색된 사례에서 해를 유도하는 방법을 재사용하여 해를 도출하는 방법으로 나눌 수 있다.

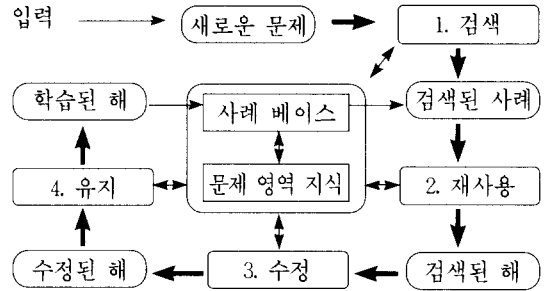
3) 수정(Revise)

재사용 단계에서 제공된 해가 입력문제에 적합하지 않았을 때에는 발생한 실패로부터 학습할 기회가 발생한다. 이러한 상태를 수정이라고 부르는데 이것에는 두가지 경우가 있다. 첫째는 재사용에 의해 제안된 해를 평가하여, 만약 성공적이면 성공으로부터 학습하는 경우이고 둘째는 영역 특성 지식을 사용하여 해를 수정하는 것이다.

4) 유지(Retain)

이 과정은 입력 문제에 대해 제안된 해를 지식으로 유지하기 위해 유용한 것들을 합치는 것이다. 제안된 해의 성공 또는 실패로부터 얻어

진 해에 대한 평가와 해에 가해진 수정에 의해 연쇄 작용이 학습으로 일어난다. 유지 과정에는 유지할 정보의 선택, 색인의 부여 여부, 사례 기억 장치에의 저장 방법 결정 등이 포함된다.



〈그림 2-2〉 Aamodt & Plaza의 사례 기반 추론 과정

2.3 소프트웨어 재사용

소프트웨어 재사용이 소프트웨어 위기를 극복할 수 있는 유력한 방안 중의 하나로 등장하면서 소프트웨어 공학의 중요한 분야로 대두되었다. 소프트웨어 재사용은 소프트웨어 개발시에 축적된 각종 산출물들을 신규 소프트웨어 개발에 가능한 한 많이 활용하고자 하는 체계적인 노력이다.

이와 같은 재사용의 원칙을 소프트웨어 개발 과정에 도입함으로써 개발 노력을 감소시키고, 생산성을 증대시킬 수 있으며, 소프트웨어 품질을 향상시키고 표준화를 통한 관리의 효율성을 가져다 줄 수 있다. 또한 소프트웨어 개발 비용의 절감과 개발기간의 단축, 검사 과정의 단순화와 같은 이점을 얻을 수 있다. 이러한 이점 때문에 소프트웨어 재사용은 소프트웨어 공학의 중요한 관심사로 대두되고 있으며, 현재까지 많은 연구들이 진행되고 있다. Prieto-Diaz는 최근까지 주장된 여러 이론들을 정리하여 기술적인 측면에서 소프트웨어 재사용 분야를 관점에

따라 <표 2-1>과 같이 여섯 가지로 분류하였다 [Prieto-Diaz, 1993].

<표 2-1> 소프트웨어 재사용의 분류

실체에 의한	영역에 의한	모드(Mode)에 의한	기술에 의한	의도에 의한	산출물에 의한
<ul style="list-style-type: none"> •아이디어, 개념 •인공물, 컴포넌트 •절차,기능 	<ul style="list-style-type: none"> •수직적 •수평적 	<ul style="list-style-type: none"> •계획적, 체계적 •비체계적, 편의주의적 	<ul style="list-style-type: none"> •합성적 •생성적 	<ul style="list-style-type: none"> •블랙박스 •화이트박스 	<ul style="list-style-type: none"> •소스코드 •설계 •명세 •객체 •텍스트 •아키텍처

소프트웨어 재사용 분야 중에서 가장 활발히 연구가 진행되었고, 좋은 성과를 보여 준 것은 라이브러리를 통한 소스 코드(source code)의 재사용이다. 소스 코드의 재사용은 Prieto-Diaz의 분류 체계에 따르면 컴포넌트, 수평적, 합성적 재사용이라는 특성을 갖고 있다. 그러나 전체 소프트웨어 개발 노력 중 소스 코드의 작성이 차지하는 비율은 5%에 불과하므로, 소스 코드의 재사용만으로는 소프트웨어 개발에서 높은 생산성을 기대하기 어렵다. 따라서 소스 코드의 재사용에 그치지 않고, 설계 및 분석 단계의 다양한 산출물에 대하여 적극적이고 광범위한 재사용을 고려하는 연구들이 시도되고 있다 [Huff et al., 1991], [Nakakoji,1991], [Soloderitsch, 1991], [Park, 1993], [Sindre et al., 1995]. 특히 분석 단계에서의 산출물은 다른 단계에서의 산출물에 비해 고 수준의 사용자 요구사항 표현이며, 환경의 변화로부터 가장 영향을 적게 받는 강력한 재사용 정보형태이다[Neighbors, 1989], [Prieto-Diaz, 1990]. 따라서 소프트웨어 개발에서는 분석 단계의 산출물을 재사용하여야 높은 생산성을 기대할 수 있다.

1970년 대에는 분석 정보의 재사용은 불가능한 것으로 간주되었다. 1970년 대 말 구조적 분

석을 지원하는 도식적 기법이 등장하였지만 분석 정보는 주로 수작업으로 문서에 표현되었기 때문에 분석 정보의 재사용은 실질적으로 불가능하였다. 그후 컴퓨터 기반의 분석 지원 도구와 CASE의 등장으로 각종 분석 정보를 컴퓨터로 생성하고 저장할 수 있게 됨에 따라 분석 정보의 재사용이 가능하게 되었다. 데이터 모델은 이러한 분석 정보의 하나로서 재사용의 대상이 되며, 특히 최근에는 데이터 모델링 과정에 CASE가 적극 활용됨에 따라 재사용이 가능한 환경에 놓이게 되었다.

데이터 모델을 포함한 분석 단계의 산출물을 재사용하기 위한 지금까지의 연구는 도메인 분석(Domain Analysis) 및 도메인 공학(Domain Engineering) 기법을 중심으로 수행되고 있다 [Prieto-Diaz, 1993]. 도메인 분석 및 도메인 공학 기법의 핵심은 사전에 동일한 도메인의 여러 시스템을 분석하여 공통점을 찾아 도메인 표준을 설정한 후, 새로운 소프트웨어가 요구 될 때 공통점과 차이점을 분석하여 공통적인 사항은 재사용하고, 차이점만 추가·삭제하여 빠른 시간 내에 좋은 품질의 소프트웨어를 경제적인 비용으로 개발하고자 하는데 있다[Neighbors, 1989].

[Tracz, 1994]. 그러나 지금까지의 도메인 분석 및 도메인 공학에 관한 연구는 주로 분석 절차와 분석 정보의 표현 기법 등에 관한 연구에 집중되어 왔으며[Tracz and Coglianese, 1991], 실질적으로 분석 정보를 재사용하는데 필요한 기법에 관한 연구는 매우 부족한 상태이다[김지홍과 송영재, 1995]. 또한 도메인 분석 및 도메인 공학에서는 도메인 표준을 설정하기 위해 동일 도메인에 대한 사전 분석 절차가 반드시 필요한데, 이러한 절차는 지식 공학에서의 지식 획득

과정과 매우 유사하며 지식 획득 과정이 지니고 있는 어려움 또한 동일하게 지니고 있다[Curtis et al., 1988], [Devanbu et al., 1991], [김영길, 1992], [Neighbors, 1992].

지식 기반 소프트웨어 재사용은 소프트웨어 개발의 각종 산출물의 재사용을 위해 지식 공학 기법을 응용하는 소프트웨어 공학의 한 분야이다[Lowery and Duran, 1989]. 현재까지의 지식 기반 소프트웨어 재사용에 관한 연구의 대부분은 정형화된 명세를 통한 기존 프로그램을 재사용하여 새로운 프로그램을 합성하는 합성적 재사용 분야에 집중되고 있다. 합성적 재사용 분야에서의 주요한 과제는 원하는 소프트웨어 부품에 대한 검색과 소프트웨어 부품의 수정 그리고 소프트웨어 부품들의 결합이다. 이러한 과제에 지식 기반 도구를 활용하고자 하는 노력이 지식 기반 프로그램 합성이다. 지식 기반 프로그램 합성 분야의 연구는 규칙 기반 추론 기법을 이용한 연구가 대부분을 차지 하고 있으며 [McDowell and Cassell, 1989], [Luqi and McDowell, 1991], [Soloderitsch, 1991], 최근에는 주어진 명세의 유사성을 파악하여 사례 기반 추론을 수행하는 연구가 시작되고 있다. 성백균의 연구에서는 객체 지향 언어인 C++로 구현된 소프트웨어 부품을 효과적으로 재사용하기 위해서 사례 기반 추론 기법을 이용하였다. 그는 이 연구에서 규칙 기반 추론이 처리하지 못하는 모호한 명세를 처리할 수 있고, 유사한 부품을 검색하여 새로운 부품으로 수정하여 사용할 수 있으며, 과거의 실패 경험을 기억하여 재사용이 곤란한 부품의 검색을 피할 수 있는 사례 기반 추론 모델을 제시하였다[성백균, 1994].

지식 기반 소프트웨어 재사용에는 합성적 재사용 분야 이외에 분석 단계의 산출물을 재사용하여 명세를 획득하고자 하는 지식 기반 명세

획득 분야가 있다. 이와 관련된 연구로는 도메인 분석의 결과인 도메인 모델을 재사용하고자 하는 DRACO[Neighbors, 1989]와 유사 분석(Analogy Analysis)을 통한 도메인 요약(Domain Abstraction)의 재사용을 제안한 연구[Maiden and Sutcliffe, 1992]가 있다. 그러나 이 연구들에 대해서는, 사전에 문제 도메인의 일반화된 모델 또는 요약을 만들기 위한 과정이 매우 어렵다는 점과 재사용을 위한 구체적인 기법들이 제시되지 않고 있다는 점이 문제점으로 지적되고 있다 [Park, 1993].

3. 데이터 모델의 재사용을 위한 사례기반추론 시스템(CB-DM Reuser)의 개발

본 연구에서는 데이터 모델의 재사용을 위한 사례기반추론 시스템인 CB-DM Reuser(Case Based Data Model Reuser)를 개발하였다. CB-DM Reuser는 과거에 완성된 데이터 모델을 사례 베이스에 저장하여 놓고 새로운 문제 영역에 대한 데이터 모델링을 수행하는 사용자에게 과거의 유사한 데이터 모델을 제공하고, 제공된 데이터 모델과 문제 영역의 특성과의 차이점을 분석하여 이를 보완하기 위한 데이터 모델의 일부를 제공함으로써 데이터 모델링 과정을 지원하는 시스템이다. 본 절에서는 이러한 기능을 갖는 CB-DM Reuser의 개발 시 고려 사항과 시스템의 구조 그리고 수행 절차를 설명한다.

3.1 CB-DM Reuser개발 시 고려 사항

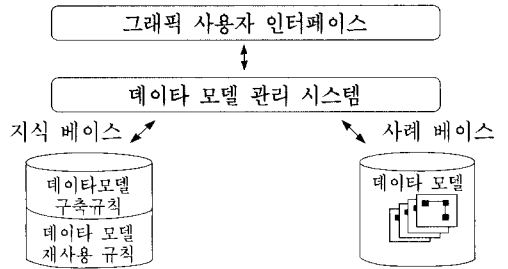
CB-DM Reuser의 개발에는 다음 사항들이 고

려되었다. 첫째, 본 시스템을 사용하는 사용자들에 대한 고려이다. 데이터 모델링 과정은 문제 영역에 속해 있는 사용자들의 지식과 관련 문서들을 조직화하여 몇 가지 부호로 구성된 그래픽 형식의 모델을 만드는 과정이다. 이러한 점에서 데이터 모델링에 참여하는 사용자와 분석가는 텍스트 형식의 지식과 문헌들을 그래픽 형식의 부호로 전환할 수 있는 지식을 갖고 있어야 한다. 따라서 본 시스템의 사용자는 데이터 모델의 문법적 지식(Syntactic Knowledge)을 갖고 있는 분석가 또는 전문 사용자(Power User)라고 가정한다. 둘째는 사용자 인터페이스에 대한 고려로서 데이터 모델링을 위한 그래픽 환경의 단일 인터페이스를 제공하여야 한다. 이러한 인터페이스는 사용자에게 데이터 모델 재사용에 따른 별도의 부담을 주지 않고, 친숙한 그래픽 환경의 다이어그램 도구를 통하여 데이터 모델링을 지원한다. 따라서 본 시스템의 입력과 출력은 그래픽 형식의 다이어그램을 이용하도록 한다. 셋째, 본 시스템은 데이터 모델링을 자동화하는 것이 아니라 지원하는 것을 목표로 하고 있다. 데이터 모델링이란 여러 가지 형식으로 분산되어 있고 구조화 되지 않은 지식들을 일정한 형식으로 구조화하는 지식 기반 행위이다. 이러한 점을 고려하여 볼 때, 데이터 모델링을 자동화하는 것은 현실적으로 거의 불가능하다. 따라서 본 시스템은 유사한 과거의 데이터 모델을 대안으로 제시하여 모델링 과정을 지원할 뿐이며, 최종 결과는 궁극적으로 사용자의 판단에 맡긴다. 넷째, 데이터 모델의 재사용을 위해 사전에 준비하는 시간과 노력을 최소화시킨다. 사례 베이스에 사례를 추가하기 위한 사전 준비가 복잡해지면 데이터 모델의 재사용에 따른 많은 이점들이 없어지게 된다. 따라서 본 시스템은 그래픽 형식의 데이터 모델을, 사용자가 텍스트

형식으로 변환하는 별도의 작업 없이, 자동으로 사례 베이스에 추가하도록 한다.

3.2 CB-DM Reuser의 구조

CB-DM Reuser는 <그림 3-1>과 같이 크게 그래픽 사용자 인터페이스, 데이터 모델 관리 시스템, 사례 베이스 그리고 지식 베이스로 구성되어 있다.



<그림 3-1> CB-DM Reuser의 구조

1) 그래픽 사용자 인터페이스

CB-DM Reuser의 사용자 인터페이스는 데이터 모델을 그릴 수 있는 그래픽 환경을 제공하므로, 다이어그램으로 시스템에의 입출력이 가능하다. 즉, 사용자는 과거의 데이터 모델을 재사용하여 데이터 모델을 완성하고자 할 때, 현재까지 스크린상에서 작성한 데이터 모델을 그대로 입력으로 사용하며, 시스템은 검색된 데이터 모델을 사용자에게 다이어그램 형식으로 출력하여 보여준다. 이러한 그래픽 환경의 사용자 인터페이스는 그 자체 만으로도 다이어그램밍 도구의 기능을 갖고 있으므로, 극단적으로는 시스템의 지원 없이도 데이터 모델을 구축할 수 있다.

2) 데이터 모델 관리 시스템

CB-DM Reuser의 데이터 모델 관리 시스템은 사례 베이스와 지식 베이스를 이용하여 유사한

데이터 모델을 검색하고, 대안을 제시하고, 수정하는 추론 기능과 CB-DM Reuser의 수행 과정을 조정하는 관리 기능을 갖는다. 또한 사용자 인터페이스를 통해 입력된 다이어그램을 추론 및 저장을 위하여 텍스트 형식으로 변환하고, 반대로 사용자에게 결과를 보여주기 위해 텍스트 형식을 그래픽 형식으로 변환하는 기능을 포함하고 있다.

3) 사례 베이스

CB-DM Reuser의 사례 베이스는 과거의 데이터 모델을 저장하는 장소이다. 그래픽 형식의 데이터 모델은 효율적인 저장과 사례 기반 추론의 기호 처리(symbolic processing)를 위해 관계형 데이터베이스 기반의 사례 베이스에 저장되어 있다. 데이터 모델의 사례 표현 예는 <부록 1>에 첨부하였다. 사례 베이스에 저장된 데이터 모델은 사용자의 요구에 따라 데이터 모델 관리 시스템의 추론 과정을 거쳐 새로운 문제 영역에 대한 해로 제공된다.

4) 지식 베이스

CB-DM Reuser의 지식 베이스는 ERD의 문법적 지식과 데이터 모델 관리 시스템의 추론 과정에 필요한 각종 지식을 저장하는 장소이다. 지식 베이스는 크게 데이터 모델 구축 규칙과 데이터 모델 재사용 규칙으로 나누어진다. 데이터 모델 구축 규칙은 사용자 또는 시스템에 의해 생성되는 데이터 모델의 정확성과 일관성을 보장하기 위한 규칙이다. 데이터 모델 재사용 규칙은 사례 베이스 내의 유사한 데이터 모델을 검색하여 새로운 데이터 모델을 생성하기 위한 규칙이다. 데이터 모델 재사용 규칙은 유사용어 색인, 검색 규칙, 재사용 규칙, 그리고 수정 규칙으로 구성되어 있다.

● 유사용어 색인

사례 베이스에 저장된 데이터 모델들은 유사

한 구조가 있을 수 있으므로 그 들간의 관계를 규정하여 두는 것이 효과적인 검색을 가능하게 할 것이다. 하지만, 데이터 모델들의 구조는 매우 다양하기 때문에 그 들간의 관계를 일관되게 규정하는 것은 거의 불가능하다. 그러므로 본 연구에서는 개별 요소인 개체와 속성들의 관계를 규정하는 데에 초점을 맞추었다. 즉 사례 베이스에 저장된 개체들과 속성들 중에서 유사한 것들을 연결하는 유사용어 색인을 <표 3-1>과 같은 테이블로 유지한다. 동의어란 하나의 데이터 모델에 사용된 개체나 속성의 이름과 비슷한 뜻의 단어들을 지칭하는 것으로서, 동의어들은 하나의 특정 데이터 모델에 부속된다. 한편, 하나의 데이터 모델에 사용된 개체나 속성의 이름이 다른 데이터 모델에 사용된 개체나 속성의 이름과 비슷한 뜻을 가진다면, 이들은 유사용어라고 지칭한다. 예를 들어 <그림 3-2>와 같이 첫번째 사례의 '사원' 개체는 '직원', '종업원' 이라는 동의어를 갖고 있고, 두번째 사례의 '판매원' 개체는 '영업직사원' 이라는 동의어를 갖고 있다. 이 때에 '사원' 과 '판매원' 을 유사용어로 정의함으로써 기존의 각 데이터 모델의 동의어 집합을 수정하지 않고 두 데이터 모델의 개체 이름간의 관계를 설정할 수 있는 것이다. 따라서 새로운 사례가 사례 베이스에 추가되면, 이 테이블들은 갱신될 수 있다

<표 3-1> 유사용어 색인의 예

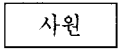
a) 유사 개체 색인

개체 이름	유사 개체
주문	판매
주문	거래
주문	영업

b) 유사 속성 색인

속성 이름	유사 속성
고객명	고객 이름
고객명	고객 이름
고객명	이름

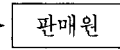
Case 1 :
데이터 모델 A



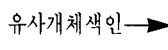
동의어: 직원
종업원

개체 이름	유사개체
사원	판매원
판매원	사원

Case 2 :
데이터 모델 B



동의어: 영업직사원

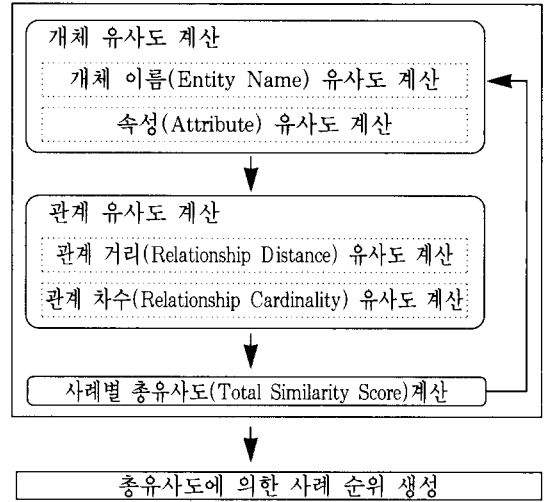


<그림 3-2> 개체의 동의어와 유사 개체 색인

● 검색 규칙

검색 규칙은 문제와 가장 유사한 데이터 모델을 사례 베이스로부터 검색하기 위한 일련의 절차와 규칙으로 구성되어 있다. <그림 3-3>과 같이 검색 규칙은 문제와 사례간의 개체 유사도 계산 및 관계 유사도 계산, 사례별 총유사도 계산, 그리고 총유사도에 의한 사례들의 순위 생성 과정으로 구성된다. 개체 유사도는 개체 이름(Entity Name)과 그 속성의 유사도 계산을 통하여 구해진다. 개체 이름 유사도는 주어진 문제의 개체 이름과 특정 사례의 개체 이름, 동의어, 그리고 유사용어 색인 중의 하나인 유사 개체와의 비교를 통하여 계산되는데, <표 3-2>의 유사도 값 매트릭스(Similarity Value Matrix)가 적용된다. 속성 유사도는 주어진 문제의 개체 속성 이름과 그 속성이 식별자인지 아닌지 등과 같은 속성의 특성을 특정 사례의 개체 속성 이름과 개체 속성 특성과 비교하여 계산되어진다. 즉, <표 3-3>의 속성 이름 유사도 값 매트릭스를 이용하여 속성 이름 유사도 값을 구하고, <그림 3-4>의 속성 특성 유사도 값 계산 규칙을 적용

하여 속성 특성 유사도 값을 구한 후 이들의 값을 더하여 속성 유사도를 계산한다.



<그림 3-3> CB-DM Reuser의 검색 규칙

<표 3-2> 개체 이름 유사도 값 매트릭스

사례 문제	개체 이름과 일치	개체 동의어와 일치	유사개체와 일치
개체 이름	1	0.8	0.6

<표 3-3> 속성 이름 유사도 값 매트릭스

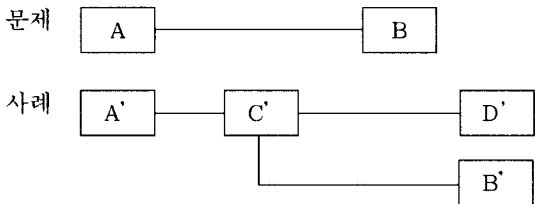
사례 문제	속성 이름과 일치	속성 동의어와 일치	유사속성과 일치
속성 이름	1	0.8	0.6

```
IF(문제 개체의 속성_식별자=Y AND 사례 개체의 속성_식별자 =Y) OR (문제 개체의 속성_파생=Y AND 사례 개체의 속성_파생=Y) THEN 속성 특성 유사도 값=0.2
```

<그림 3-4> 속성 특성 유사도 값 계산 규칙

관계 유사도는 두 개체 사이의 거리(Distance)와

차수(Cardinality)에 대한 유사도 값을 구하여 계산한다. 따라서 주어진 문제에 나타난 관계에 연결된 두 개체에 대하여 특정 사례안에 유사한 개체들이 존재하여야만 관계 유사도를 구할 수 있다. 관계 거리 유사도 값은 문제에 나타난 관계에 연결된 두 개체가 검색된 사례에서 어느 정도의 거리를 갖느냐에 따라 결정되어진다. 관계 거리는 두 개체 사이에 존재하는 관계의 수 즉, 관계를 나타내는 직선의 개수로 결정되며, 두 개체 사이에 여러 개의 관계 거리가 존재하면 가장 짧은 거리를 선택하게 된다. 예를 들어 <그림 3-5>와 같이 A, B 두 개체 사이의 관계에 대응되는 사례를 보면, A'와 B' 사이에 관계 거리는 2와 3이 존재하는데 이 중에서 가장 짧은 거리인 2가 관계 거리로 선택된다.



<그림 3-5> 관계 거리의 예

관계 거리 유사도 값은 구해진 관계 거리에 따라 <표 3-4>의 관계 거리 유사도 값 매트릭스를 적용하여 구한다. 관계 차수 유사도 값은 문제의 관계에 대한 카디널리티와 옵션칼리티에 따라 <그림 3-6>의 관계 차수 유사도 값 계산 규칙을 적용하여 계산된다.

<표 3-4> 관계 거리 유사도 값 매트릭스

관계 거리	1	2	3
관계거리 유사도 값	1	0.8	0.6

```

IF 문제 left cardinality = 사례 left cardinality
THEN
    관계 차수 유사도 = 0.25
IF 문제 right cardinality = 사례 right
cardinality THEN
    관계 차수 유사도 = 관계 차수 유사도 + 0.25
IF 문제 left optionality = 사례 left optionality
THEN
    관계 차수 유사도 = 관계 차수 유사도 + 0.25
IF 문제 right optionality = 사례 right
optionality THEN
    관계 차수 유사도 = 관계 차수 유사도 + 0.25
  
```

<그림 3-6> 관계 차수 유사도 값 계산 규칙

사례별 총유사도(TSS: Total Similarity Score)는 앞에서 계산된 개체 유사도와 관계 유사도에 동일한 가중치를 부여하여 계산되어진다. 다음 <식 3-1>은 사례별 총유사도를 계산하기 위한 공식을 보여주고 있다.

<식 3-1> 사례별 총유사도 계산 공식

$$TSS = \sum_{i=1}^L E_i + \sum_{j=1}^M R_j \quad i = 1, 2, \dots, L; \quad j = 1, 2, \dots, M$$

$$E_i = EN_i + A_i \quad i = 1, 2, \dots, L$$

$$R_j = RD_j + RC_j \quad j = 1, 2, \dots, M$$

$$A_i = \sum_{k=1}^N (AN_k + AP_k) / n, \quad i = 1, 2, \dots, L$$

E_i : 개체 i 의 유사도
 R_j : 관계 j 의 유사도
 EN_i : 개체 i 의 이름 유사도 값
 A_i : 개체 i 의 속성 유사도
 RD_j : 관계 j 의 거리 유사도 값
 RC_j : 관계 j 의 차수 유사도 값

AN_k : 속성 k의 이름 유사도 값
 AP_k : 속성 k의 특성 유사도 값

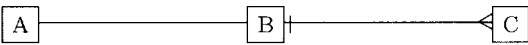
L : 문제의 개체 수
 M : 문제의 관계 수
 N : 문제 개체의 속성 수

총유사도에 의한 사례 순위 생성은 각 사례별 총유사도를 내림차순으로 정렬하여 총유사도가 가장 큰 사례 즉, 문제와 가장 유사한 사례를 선택하는 과정이다.

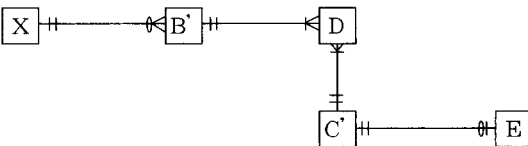
●재사용 규칙

검색 규칙에 의해 문제와 가장 유사한 데이터 모델이 검색되면, 이에 재사용 규칙을 적용하여 문제에 대한 초기 데이터 모델을 생성한다. 재사용 규칙에 의한 초기 데이터 모델의 생성은 문제 데이터 모델과 검색된 데이터 모델을 결합(merge)하여 만드는데, 개체 이름은 문제 데이터 모델의 것을 사용하고, 관계의 이름 및 특성은 검색된 사례의 것을 사용한다. 개체의 속성이나 동의어는 문제 데이터 모델의 것과 검색된 사례의 것의 합집합이 초기 데이터 모델의 것이 된다. <그림 3-7>은 재사용 규칙을 적용하여 초기 데이터 모델을 만드는 과정의 예를 보여 주고 있다.

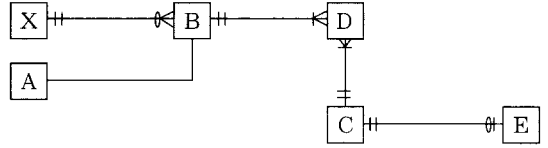
a) 문제 데이터 모델



b) 사례의 데이터 모델



c) 초기 데이터 모델

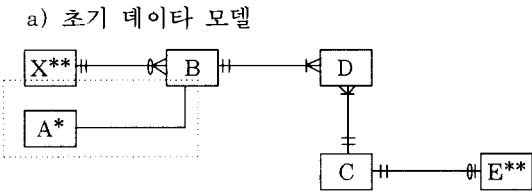


<그림 3-7> 재사용 규칙 적용을 통한 초기 데이터 모델 생성의 예

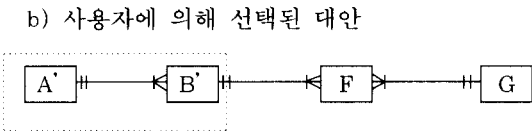
●수정 규칙

재사용 규칙에 의하여 생성된 초기 데이터 모델이 주어진 문제 상황에 만족되지 않을 때, 일련의 수정 규칙을 적용하여 이를 수정한다. 수정은 데이터 모델 중의 일부분인 특정 개체 또는 관계를 대상으로 이루어지는데, 수정이 필요한 부분에 대해서 시스템이 수정 대안들을 제시하고, 사용자가 대안 중의 하나를 선택하면 시스템이 재사용 규칙을 적용하여 데이터 모델을 수정한다. 수정 규칙은 수정이 필요한 부분을 누가 선정하느냐에 따라서 시스템 주도의 수정 규칙과 사용자 주도의 수정 규칙으로 나뉜다. 시스템 주도의 수정 규칙은 시스템 스스로 문제 데이터 모델과 초기 데이터 모델 사이의 차이점을 분석하여 수정이 필요한 부분을 찾아냄으로써 수정이 수행되는 규칙이고, 사용자 주도의 수정 규칙은 언제든지 사용자가 데이터 모델의 만족스럽지 못한 부분에 대하여 수정을 요구하면 시스템이 이를 수정하는 규칙이다. 수정 대안들은 일정 수준 이상의 유사도를 갖는 개체 또는 관계들로부터 생성되는데, 시스템 주도의 수정 규칙에서는 초기 데이터 모델을 만들 때에 이미 계산된 유사도들을 사용하는 반면, 사용자 주도의 수정 규칙에서는 유사도들을 다시 계산한다. 사용자 주도의 수정에서는 사용자가 개체 또는 관계를 변경하거나 추가한 후에 시스템에게 수정을 요구할 수 있는데, 그러한 개체 또는

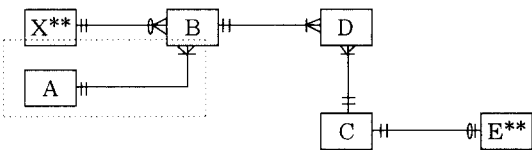
관계에 대해서는 유사도가 계산되어 있지 않다. 그러므로 사용자 주도의 수정 규칙에서는 유사도들을 다시 계산하는 것이다. <그림 3-8>은 <그림 3-7 c)>의 초기 데이터 모델이 시스템 주도의 수정 규칙을 통하여 새로운 모델로 변화되는 것을 보여 주고 있다.



- 문제를 만족시키지 않는 개체로서 시스템 주도의 수정 규칙의 대상
- 문제와 관계 없이 사례 베이스에 저장된 데이터 모델에 의하여 추가된 개체



c) 수정규칙과 재사용 규칙에 의해 생성된 데이터 모델



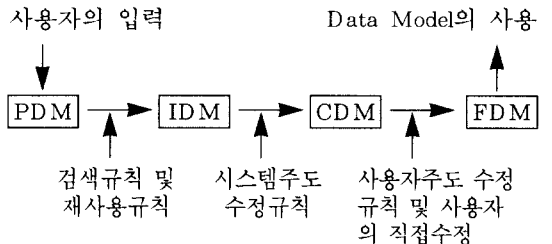
- 문제와 관계 없이 사례 베이스에 저장된 데이터 모델에 의하여 추가된 개체

<그림 3-8> 수정 규칙에 의한 새로운 해의 생성

3.3 CB-DM Reuser의 수행 절차

CB-DM Reuser에 의하여 작성되는 데이터 모델은 <그림 3-9>와 같이 그 완성도에 따라 분

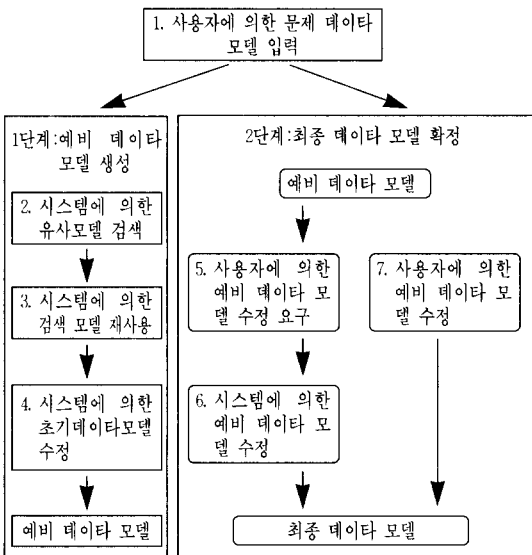
계 데이터 모델(Problem Data Model : PDM), 초기 데이터 모델(Initial Data Model : IDM), 예비 데이터 모델(Candidate Data Model : CDM), 그리고 최종 데이터 모델(Final Data Model : FDM)로 부르는데, CB-DM Reuser가 수행되는 절차는 크게 두 단계로 나누어 볼 수 있다. 첫번째 단계는 사용자에게 의해 작성되는 PDM을 기반으로 검색 규칙과 재사용 규칙을 적용하여 IDM을 생성하고, 이 IDM에 시스템 주도의 수정 규칙을 적용하여 CDM을 작성하는 단계이다. 두번째 단계는 CDM에 대하여 사용자 주도의 수정 규칙을 적용하거나 사용자에게 의한 직접적인 수정을 통하여 FDM을 완성하는 단계이다. 첫번째 단계는 시스템에 의해 자동적으로 수행되며, 두번째 단계는 사용자의 의도에 따라 사용자와 시스템의 상호작용을 통하여 수행된다. CB-DM Reuser의 수행 절차는 항상 첫번째 단계와 두번째 단계가 순서대로 수행되어지는 것은 아니다. 사용자가 문제 데이터 모델의 부분적인 수정만을 원하는 경우 첫번째 단계를 생략하고, 두번째 단계를 바로 수행할 수도 있다. <그림 3-10>은 CB-DM Reuser의 세부 수행 절차를 보여주고 있다.



<그림 3-9> CB-DM Reuser에 의해 작성되는 데이터 모델 명칭

1) 사용자에게 의한 PDM의 입력

사용자는 그래픽 환경의 사용자 인터페이스를 통하여 해당 문제 영역에 대한 데이터 모델을 입력한다. 이 때 입력되는 데이터 모델은 사용자의 모델링 경험과 기술 그리고 모델링을 위한 노력 정도에 따라 매우 다양한 형태로 존재할 수 있다. 즉, 문제 영역의 가장 기본이 되는 몇 개의 개체만이 존재하는 데이터 모델로부터 완전한 데이터 모델에 이르기까지 다양한 데이터 모델이 입력되어 질 수 있다. 입력되는 데이터 모델이 구체화될수록 보다 유사한 데이터 모델이 검색될 수 있는 확률은 높아진다.



<그림 3-10> CB-DM Reuser의 세부 수행 절차

2) 시스템에 의한 유사 모델의 검색

PDM이 입력되면 데이터 모델 관리 시스템은 지식 베이스의 검색 규칙을 적용하여 사례 베이스로부터 가장 유사한 데이터 모델을 검색한다. 이때 검색된 데이터 모델은 다른 사례에 비해 상대적으로 PDM과 유사성이 높을

뿐이다. 따라서 검색된 데이터 모델이 사용자가 원하는 것과 같지 않을 수 있다. 이러한 경우 사용자는 입력되는 PDM에 보다 많은 정보를 표현할 필요가 있다.

3) 시스템에 의한 유사 모델의 재사용

검색된 데이터 모델은 지식 베이스의 재사용 규칙을 적용하여 사용자에게 의해 입력된 PDM과 결합되어 IDM을 생성한다.

4) 시스템에 의한 IDM의 수정

생성된 IDM이 입력된 PDM을 충분히 만족시키지 못하면, 다른 사례에서 유사한 개체 또는 관계를 검색하여 IDM을 수정한다. 이 과정을 통하여 시스템은 CDM을 완성하고, 제 2단계인 사용자에게 의한 최종 데이터 모델 확정 단계로 넘어간다. 제 1단계를 생략한 경우에는 제 2단계의 수정 대상은 PDM이 된다.

5) 사용자에게 의한 CDM 수정 요구

사용자는 CDM에 대하여 전체 모델이 아닌 모델의 일부분을 선택하고 시스템에 수정을 요구한다.

6) 시스템에 의한 CDM의 수정

사용자의 수정 요구에 의하여 모델의 일부분에 대하여 검색 규칙과 재사용 규칙을 적용하여 CDM을 수정하게 된다. 이 과정은 사용자의 의도에 따라 반복적으로 수행할 수 있다.

7) 사용자에게 의한 CDM의 수정

시스템 또는 시스템과 사용자의 상호 작용에 의해 최종적으로 수정된 CDM은 그 자체만으로는 완전하지 않을 수가 있다. 따라서 완전한 데이터 모델을 작성하기 위해서는 문제 영역의 범위와 특정 업무 규칙 등을 근거로 사용자의 판단에 의해 CDM을 수정하여야 한다.

4. CB-DM Reuser의 사용 예시

4.1 사례의 개요

본 연구에서 사용하고자 하는 사례는 A 사무기기 상사의 거래 관리 업무이다. A 사무기기 상사는 컴퓨터, 팩시밀리, 복사기 등 사무기기의 판매 및 대여와 관련 소모품을 판매하는 회사인데, 수작업으로 수행하여 온 거래 내역과 대금 청구에 대한 관리를 데이터베이스 구축을 통하여 전산화하고자 한다. <그림 4-1>은 현 거래 관리 분야의 업무 규칙을 정리하여 보여주고 있다.

A 사무기기 상사에서는 관련 제품들을 여러 납품 업체로부터 공급 받고 있는데, 각 제품은 하나의 납품 업체로부터 공급 받고, 하나의 납품 업체는 여러 제품을 공급한다. 회사에서는 고객들에게 한번에 여러 제품을 대여할 수 있으며 1달 단위로 대여료를 청구 한다. 이를 위해 제품별 대여내역을 기록한다. 제품 판매는 제품이 고가인 관계로 한번에 하나의 제품이 판매 되는 것이 일반적이다. 그러나 예외적으로 여러개의 제품이 한번에 판매되기도 한다. 사무기기 소모품 판매는 한번에 여러 종류의 소모품이 여러개 판매되며, 소모품 또한 여러 고객들에게 판매되어진다. 판매에 따른 청구는 대여에 의한 청구와 병합하여 처리된다.

<그림 4-1> A 사무기기 상사의 거래 관리 업무 규칙

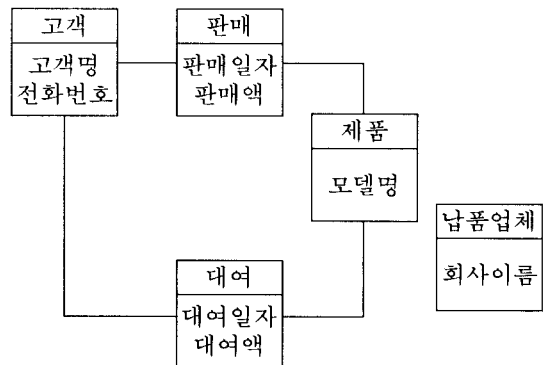
4.2 CB-DM Reuser에 의한 데이터 모델링 과정

CB-DM Reuser는 현재 사례 베이스에 8개의 사례를 저장하고 있다. 각 사례들은 다양한 업종의 거래 관리, 주문 관리, 계약 관리 분야에 대한 데이터 모델들이다. 사례 베이스에 저장되어 있는 데이터 모델은 <부록 2>에 첨부되어 있다. CB-DM Reuser에 의한 A 사무기기 상사의 데이터 모델링 과정을 CB-DM Reuser의 수행 절차에 따라 설명한다.

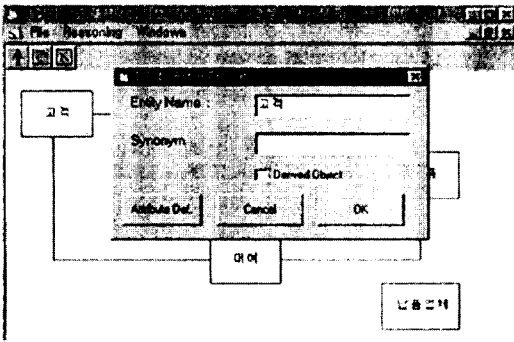
1) 사용자에 의한 PDM의 입력

A 사무기기 상사의 업무 규칙을 통하여 <그림 4-2>와 같이 중요한 개체와 관계들을 파악할 수 있다. 사용자의 데이터 모델링 경험과 지식 그리고 문제 영역에 대한 이해 정도에 따라 파악되는 개체와 관계들은 달라질 수 있다. <그림 4-3>은 이 단계에서 파악된 개체와 관계 그리고 개체의 속성들을 CB-DM Reuser의 그래픽 사용자 인터페이스를 통하여 입력하는 것을 보여주고 있다.

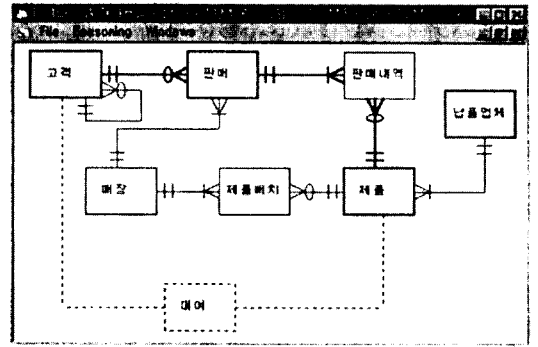
A 사무기기 상사 거래관리



<그림 4-2> A 사무기기 상사의 PDM



〈그림 4-3〉 PDM의 입력



〈그림 4-4〉 재사용 규칙을 통한 A 사무기기 상사의 IDM 생성

2) 시스템에 의한 유사 모델의 검색

CB-DM Reuser는 A 사무기기 상사의 PDM과 가장 유사한 사례로서 'K 할인 판매점 판매관리'의 데이터 모델(부록 2의 사례 4)을 검색한다. 〈표 4-1〉은 검색 규칙에 의해 생성된 총유사도 계산 결과이다.

〈표 4-1〉 총유사도 계산 결과

사례	모델 이름	총유사도	개체유사도	관계유사도
4	K할인판매점 판매관리	8.7	6.9	1.8
5	SS보험사 생명보험 계약관리	6.3	4.3	2.0
2	S가구 대리점 주문관리	6.3	4.5	1.8
3	DW도매 주류상사 주문관리	6.0	4.2	1.8
1	YG서적대여점 대여관리	4.5	3.5	1.0
6	VW비디오대여점 대여관리	4.5	3.5	1.0
7	JC카드사 거래관리	4.3	3.3	1.0
8	GS기공소 주문관리	4.1	3.1	1.0

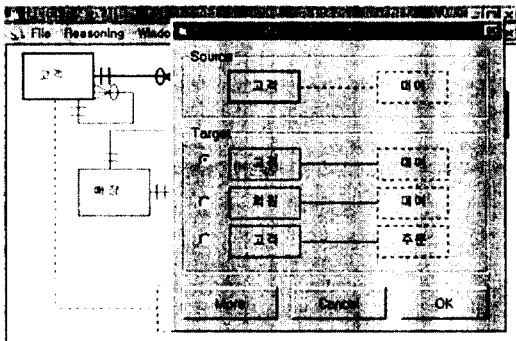
3) 시스템에 의한 유사 모델의 재사용

검색된 데이터 모델인 'K 할인 판매점 판매관리' 모델이 재사용 규칙에 의하여 PDM과 결합되어 〈그림 4-4〉와 같은 IDM을 생성한다.

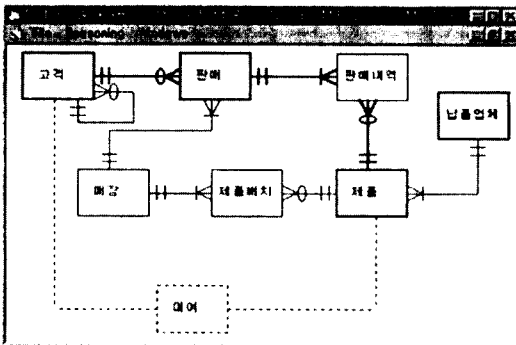
4) 시스템에 의한 IDM의 수정

CB-DM Reuser는 IDM과 PDM의 차이점을 분석하여 〈그림 4-4〉와 같이 사용자에게 그 차이점을 보여준다. 즉, '대여,' '고객-대여,' '대여-제품'과 같이 PDM에는 있으나, IDM에서 유사한 개체 또는 관계를 찾지 못한 경우에는 점선으로 보여주고, '고객,' '판매,' '제품,' '납품업체,' '고객-판매,' '판매-제품'과 같이 유사한 개체와 관계를 찾은 경우에는 굵은 실선으로 나타내고, '매장,' '판매내역,' '제품배치'와 '매장-제품배치,' '제품배치-제품,' '제품-납품내역,' '판매-매장,' '고객-고객'과 같이 PDM과 무관하게 검색된 사례로부터 추가된 개체 또는 관계는 가는 실선으로 나타낸다. 시스템 주도의 수정은 점선으로 나타나는 '대여,' '고객-대여,' '대여-제품'을 중심으로 이루어지게 된다. 〈그림 4-5〉는 수정 대상이 되는 '고객-대여'에 대하여 시스템이 제안하는 대안들이다. 이 대안들은 앞의 검색 규칙을 이용하여 생성된 개체 유사도와 관계 유사도를 근거로 하여 유사도가 높은 순서대로 만들어진다. 여기에서는 사례 6의 '고객-대여'와 사례 1의 '회원-대여' 그리고 사례 2의 '고객

—주문’ 등이 대안으로 제시된다. 제안된 대안들 중 사용자에게 의해 하나가 선택되면 <그림 4-6>과 같이 CDM이 생성된다. 동일한 과정을 통해 ‘대여—제품’에 대하여 사례 6의 ‘대여—대여내역—서적’과 사례 1의 ‘대여—대여내역—비디오’가 대안으로 제시되고, 사용자의 선택으로 CDM이 변경된다.



<그림 4-5> 수정 규칙에 의한 대안 생성

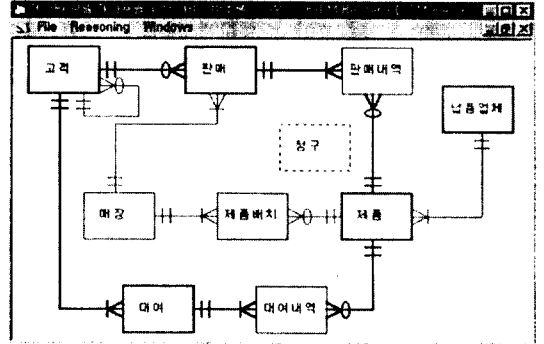


<그림 4-6> CDM의 생성

5) 사용자에게 의한 CDM의 수정 요구

사용자는 CDM의 개체 또는 관계를 수정 또는 추가하고, 필요한 경우 시스템에게 수정을 요구할 수 있다. <그림 4-7>은 추가로 파악된

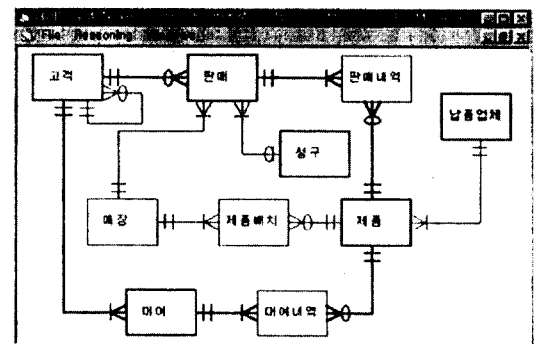
‘청구’를 추가하고 시스템에게 수정을 요구하는 것을 보여주고 있다.



<그림 4-7> 사용자에게 의한 개체의 추가 및 CDM의 수정 요구

6) 시스템에 의한 CDM의 수정

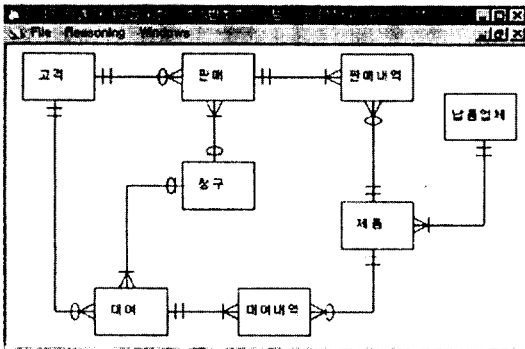
CB-DM Reuser는 ‘청구’와 이와 관련된 잠재적인 관계에 대한 유사한 사례를 검색하기 위해 검색 규칙을 적용하여 총유사도를 다시 계산한다. 계산 결과, 사례 3의 ‘청구’와 ‘주문—청구’가 가장 유사한 사례로 선택되고, <그림 4-8>과 같이 CDM은 수정된다.



<그림 4-8> CDM의 수정

7) 사용자에게 의한 CDM의 수정

CB-DM Reuser는 기본적으로 사례 베이스에 저장된 과거의 데이터 모델을 근거로 사용자의 데이터 모델링을 지원하는 시스템이므로 CB-DM Reuser가 제공한 CDM내의 어떤 개체의 속성이나 관계의 특성이 사용자가 모델링하고자 하는 문제를 올바르게 표현하지 못하는 경우도 있다. 따라서 사용자는 다이어그램으로 직접 확인이 가능한 개체와 관계뿐만 아니라 개체 속성과 관계 특성이 현재 문제를 정확히 반영하였는지를 검토하여야 한다. 만일 'A 사무기기 거래관리'에서는 대어를 한 번도 하지 않아도 고객으로 간주한다면 <그림 4-8>의 '고객-대어'의 특성은 올바르게 표현되어 있지 않다. 또한 '매장'과 '매장-제품배치'와 같이 현재의 문제와 관계없는 개체나 관계를 제시하기도 한다. 이런 부분에 대한 최종 결정은 사용자의 판단에 맡긴다. 즉, 사용자는 제시된 CDM에 대하여 개체나 관계의 추가와 삭제 그리고 개체 속성 및 관계의 특성에 대한 갱신 작업을 통하여 데이터 모델을 완성한다. 여기에서는 '고객-대어'에 대한 관계 특성을 변경하고, '대어-청구'를 추가하고, '매장'과 '제품배치' 그리고 연관된 관계들을 삭제하여 <그림 4-9>와 같이 'A 사무기기 거래관리'의 FDM을 완성한다.



<그림 4-9> 완성된 'A 사무기기 거래관리'의 FDM

5. 결 론

본 연구에서는 인공지능 기법 중의 하나인 사례기반추론을 이용하여 과거의 데이터 모델을 재사용하여 데이터 모델링을 지원하는 시스템인 CB-DM Reuser를 개발하였다. 최근 들어 기업에서는 중요한 자원인 데이터를 구조화하고자 데이터 모델을 만드는 데에 많은 시간과 노력을 소요하고 있으나, 인간이 본질적으로 데이터를 표현하는 방법에 서툴고, 데이터 모델링 전문가가 현실적으로 적다는 점 때문에 큰 어려움을 겪고 있다. 따라서 유사한 조직과 기능들에서는 요구되는 데이터와 업무 규칙들이 비슷하다는 점을 고려할 때, 과거의 유사한 데이터 모델을 재사용하고자 하는 노력은 타당하다고 할 수 있다. 또한 소프트웨어 재사용 분야와 데이터 모델링 분야에 대한 최근까지 연구들에서 데이터 모델을 적극적으로 재사용하는 구체적인 기법에 관한 연구가 매우 부족하다는 점을 고려할 때, 본 연구는 적극적으로 데이터를 재사용하는데 필요한 체계적인 기법을 제시하였다는 점에서 큰 의의가 있다 하겠다. CB-DM Reuser를 사용하면 데이터 모델링 과정에서 시간과 노력을 줄이는 등 어려움을 상당 부분 완화할 수 있을 것으로 기대한다. 특히, 시스템이 제공하는 유사한 과거의 사례들과 수정 규칙들은 전문가들이 모델링 과정에서 제공하는 적절한 가이드 라인과 같은 역할을 대신할 수 있을 것이다.

본 연구에서 개발한 시스템인 CB-DM Reuser는 아직 실용성 측면과 성능 측면에서 한계점을 가지고 있다. CB-DM Reuser의 사용자는 데이터 모델에 대한 전문적인 지식을 소유하고 있어

야 한다. 그러므로 CB-DM Reuser의 실용성을 판정하기 위해서는 여러 명의 전문 사용자들 대상으로 하여 CB-DM Reuser를 사용할 때와 사용하지 않을 때에 그들의 데이터 모델링 작업에 어떠한 변화가 있는지를 관찰하여야 한다. 하지만, 본 연구에서는 이러한 실험을 수행하지 못하였다. 그러므로 본 연구에서는 실용성에 대한 검증은 향후 연구로 남겨두고, 단지 데이터 모델의 재사용을 사례기반 추론으로 가능하게 하였다는 점을 공헌으로 부각시키고자 한다. 두 번째의 한계점은 CB-DM Reuser의 성능 측면인데 첫째, CB-DM Reuser는 개체와 속성에 사용된 용어들의 의미론적으로 이해하지는 못하고 있다. 동일한 개념 또는 사물을 표현하더라도 용어는 사람에 따라 다르게 사용할 수 있다. 예를 들어 비디오를 비디오 자체로 표현하는 사람도 있고, 상품으로 표현하는 사람도 있을 수 있다. 이런 경우에 비디오와 상품을 유사한 개념으로 이해할 수 있어야 하나, 본 연구에서는 이런 점을 사용자의 지식과 상식에 의존하여 유사용어 색인을 갱신·유지함으로써 해결하고 있다. 유사용어들 간에도 유사의 정도에 따라 유사도 값이 달라질 것이다. 용어의 이해 기능이 보완되고 유사용어들의 유사 정도가 계층적으로 파악된다면 보다 현실적으로 데이터 모델링을 지원할 수 있을 것이다. 둘째, 사례기반추론의 학습 기능에 의한 사례의 추가 기능을 고려하지 않았다. 단순히 하나의 다이어그램을 사례로 추가할 수 있으나, 사례간의 유사성을 고려한 색인이 생성되지 않기 때문에 각 사례들이 개별적으로 저장될 뿐이다. 따라서 사례 베이스를 용이하게 확장하는데 걸림돌이 되고 있다. 셋째, 사례 베이스에 다양한 분야의 많은 사례가 저장될 때, 검색 과정의 효율성을 보장하기 위한 저장 방식을 고려하지 못하였다. 유사한 사례를 찾기 위

한 검색 속도 자체가 데이터 모델링의 주요 요인이 될 수는 없으나, 사례의 저장 방식의 개선이 데이터 모델링의 효율성을 높이는데 기여할 수는 있을 것이다.

이러한 성능 측면의 한계점을 극복하기 위해서는 개체와 속성의 용어 이해와 사례기반추론의 학습 기능을 이용한 유사용어 식별규칙, 그리고 효율적인 검색을 위한 사례 베이스 저장 방식에 관한 연구가 추가로 필요하다. 또한 데이터 모델을 재사용하는 기법을 확장하여 데이터와 프로세스를 하나로 통합하는 객체 모델(Object Model)의 재사용 기법에 관한 연구도 필요하다. 객체 모델은 객체 지향 소프트웨어 개발 전과정의 산출물을 재사용하는데 가장 기본이 되는 모델이다. 객체 모델의 재사용은 단순히 분석 단계만이 아닌 분석, 설계, 구축의 전 단계를 지원할 수 있기 때문에 재사용의 파급 효과는 매우 클 것으로 기대된다. 현재 저자들은 사례 기반 추론을 이용하여 객체 모델을 재사용하는 기법에 관한 연구를 진행하고 있다.

참고문헌

- 김영길, 소프트웨어 재사용을 위한 객체지향 클래스 라이브러리에서의 정보 검색, 박사학위 논문, 전자계산학과, 중앙대학교, 1992
- 김지홍, 송영재, "도메인 분석정보 재사용을 위한 처리기의 설계 및 구현," 정보처리논문지, 제 2권 제 4호 (1995), 499-508
- 성백균, 소프트웨어 재사용을 위한 사례기반의 추론 모델, 박사학위논문, 전자계산학과, 중앙대학교, 1994

- : Foundational Issues, Methodological Variations, and System Approaches, *Artificial Intelligence Communications*, Vol. 7, No. 1 (1996), 9-13
- Barker, R., *CASE*Method: Entity Relationship Modeling*, Addison-Wesley, Workingham, 1989
- Chen, P., "The Entity Relationship Model: Toward a Unified View of Data," *ACM Trans. on Database Systems*, March (1976), 9-36
- Curtis, B., H. Krasner and N. Iscoe, "A Field Study of the Software Design Process for Large Systems," *Communications of the ACM*, Vol. 31, No.11 (1988), 1268-1287
- Devanbu, P., R. J. Brachman, P. G. Delfridge and B. W. Ballard, "LassIE: A Knowledge-Based Software Information System," *Communications of the ACM*, Vol. 34, No. 5 (1991), 35-49
- Finkelstein, C., *Information Engineering : Strategic Systems Development*, Addison-Wesley, Workingham, 1992
- Flavin, M., *Fundamental Concepts of Information Modeling*, Yourdon Press, NJ, 1981
- Fleming, C. C. and B. von Halle, *Handbook of Relational Database Design*, Addison-Wesley, Workingham, 1989
- Goodman, M., "CBR in Battle Planning," *Proceeding of Case-Based Reasoning Workshop (DARPA)*, Morgan-Kaufmann, San Mateo CA., (1989), 264-269
- Hay, D. C., *Data Model Patterns : Conventions of Thought*, Dorset House Publishing, NJ, 1996
- Huff, K. E., R. Thomson and J. W. Gish, "The Role of Understanding and Adaptation in Software Reuse Scenarios," *Proceedings of 4th Annual Workshop on Software Reuse*, 1991
- Kass, A., "Adaptation-Based Explanations: Extending Script/Frame Theory to Handle Novel Input," *Proceeding of the 11th International Joint Conference on AI*, Detroit, MI., Morgan-Kaufmann, San Mateo CA., (1989), 141-147
- Lowery, M. and R. Duran, "Knowledge-based Software Engineering," *The Handbook of Artificial Intelligence*, Vol. 4, Addison-Wesley, 1989
- Luqi and J. McDowell, "Software Reuse in Specification-Based Prototyping," *Proceedings of 4th Annual Workshop on Software Reuse*, 1991
- Maiden, N. and A. Sutcliffe, "Exploiting Reusable Specification through Analogy," *Communication of the ACM*, Vol. 35, No. 5 (1992), 55-64
- Martin, J., *Information Engineering*, Prentice-Hall, Englewood Cliffs, NJ, 1990
- McDowell, R. and K. Cassell, "The RLF Librarian: A Reusability Librarian Based on Cooperating Knowledge-Based Systems," *Proceedings of 4th Annual Knowledge-Based Software Assistant Conference*, September, 1989
- McFadden, F. R. and J. A. Hoffer, *Database Management*, Benjamin/Cummings, Redwood City, CA, 1991
- Nakakoji, K., "Software Reuse in Integrated,

Domain-Oriented, Knowledge-Based Design Environment," *Proceedings of 4th Annual Workshop on Software Reuse*, 1991

Neighbors, J. M., "Draco: A Method for Engineering Reusable Software System," *Software Reusability*, Vol. 1 (1989), 303-314

Neighbors, J. M., "The Commercial Applications of Domain Analysis," *Proceedings of 5th Annual Workshop on Software Reuse*, 1992

Park, S., "Software Requirement Text Reuse," *Proceedings of 6th Annual Workshop on Software Reuse*, 1993

Prieto-Diaz, R., "Domain Analysis: An Introduction," *ACM SIGSOFT Software Engineering Notes*, Vol. 15, No. 2 (1990), 47-54

Prieto-Diaz, R., "Status Report: Software Reusability," *IEEE Software*, May (1993), 61-66

Riesbeck, C. K. and R. L. Schank, *Inside Case-based Reasoning*, Hillsdale, N.J. Lawrence Erlbaum Associates, 1989

Sindre, G., R. Conradi and E. Karlsson, "The REBOOT Approach to Software Reuse," *Journal of Systems and Software*, Vol. 30 (1995), 201-212

Soloderitsch, J., "An Organon: Intelligent Reuse of Software Assets and Domain Knowledge," *Proceedings of 4th Annual Workshop on Software Reuse*, 1991

Storey, V. C., "Relational Database Design Based on The Entity-Relationship Model," *Data and Knowledge Engineering*, Vol. 7 (1991), 47-83

Teorey, T. J. and J. P. Fry, *Design of Database Structures*, Prentice-Hall, Englewood Cliffs, NJ, 1982

Tracz, W., "Reuse State of the Art and State of the Practice Report Card," *3rd International Conference on Software Reuse*, IEEE Computer Society, (1994), 194-198

Tracz, W. and L. Coglianese, "An Outline for a Domain Specific Software Architecture Engineering Process," *Proceedings of 4th Annual Workshop on Software Reuse*, 1991

〈부록 1〉 데이터 모델의 사례 표현 예

본문 제 2.1절 〈그림 2-1〉의 ERD를 CB-DM Reuser의 사례베이스에 저장하기 위해서는 아래와 같이 표현한다.

a) 데이터 모델

모델 번호	모델 이름
1	ERD의 예

b) 개체

개체 번호	개체 이름	모델 번호
E101	부서	1
E102	사원	1

c) 관계

관계	좌측	좌측	좌측	우측	우측	우측	모델
번호	관계	카드	옵서	관계	카드	옵서	번호
	이름	날리	날리	이름	날리	날리	
R101	소속된다	M	0	갖는다	1	1	1

c) 개체-관계

개체 번호	관계번호	위 치
E101	R101	Left
E102	R101	Right

d) 개체 동의어

동의어번호	동 의 어	개체번호
ES101	직원	E102
ES102	종업원	E102

e) 속 성

속성번호	속성이름	식별자	파생속성	개체번호
A101	부서코드	Y	N	E101
A102	부서명	N	N	E101
A103	사원번호	Y	N	E102
A104	사원명	N	N	E102
A105	전화번호	N	N	E102

f) 속성 동의어

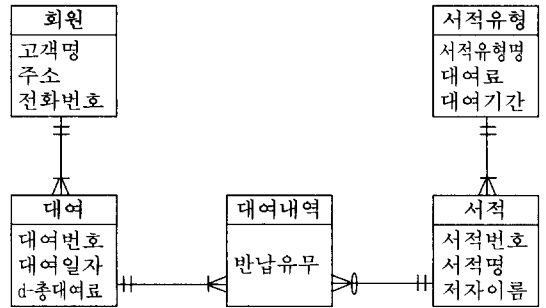
동의어번호	동 의 어	속성번호
AS101	부서이름	A102
AS102	사원코드	A103
AS103	사원이름	A104
AS104	이름	A104
AS105	연락처	A105

<부록 2> CB-DM Reuser의 사례 베이스에 저장되어 있는 데이터 모델

편의상 관계의 이름과 개체 및 속성에 대한 특성(동의어 등)은 생략하였음.

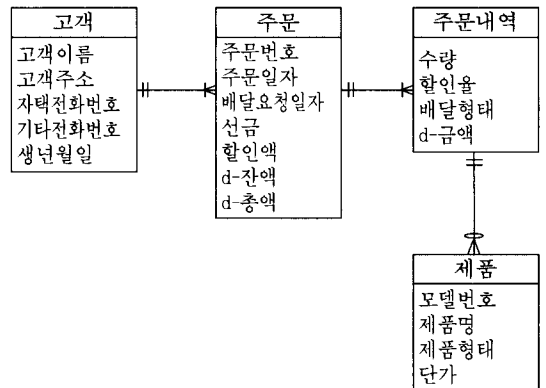
1) 사례1

YG서적대여점 대여관리



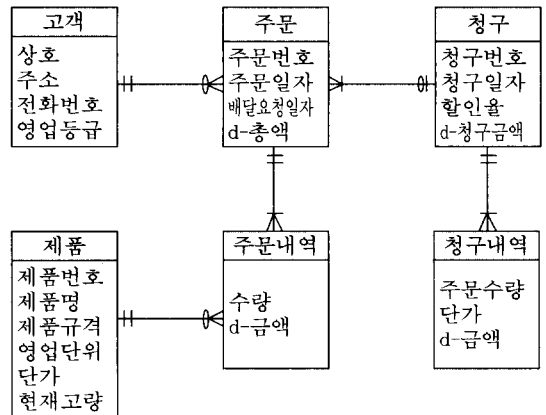
2) 사례 2

S가구 대리점 주문관리



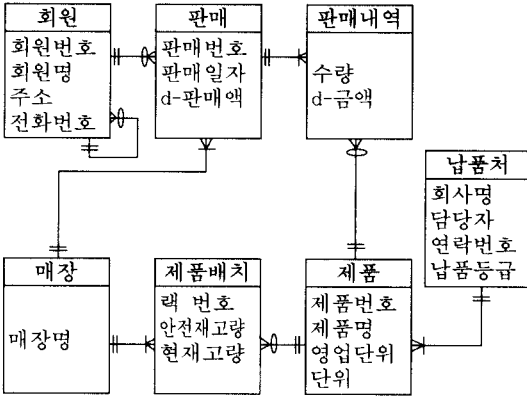
3) 사례 3

DW도매 주류상사 주문관리



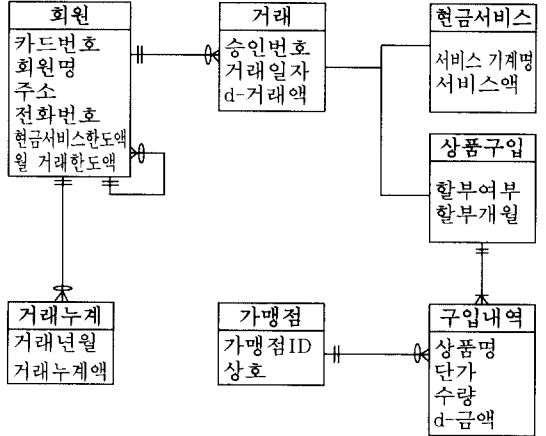
4) 사례 4

K할인 판매점 판매관리



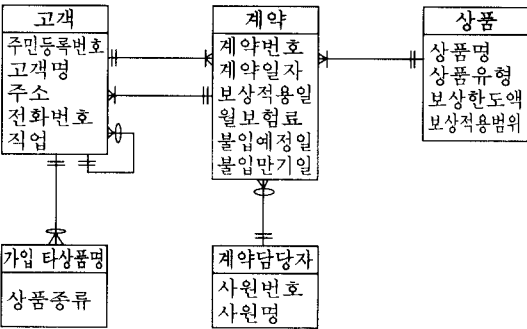
7) 사례 7

J카드사 거래관리



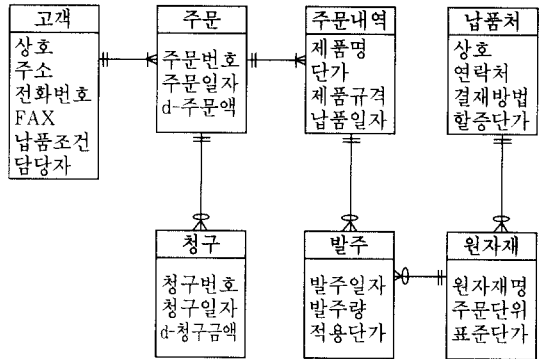
5) 사례 5

SS보험사의 생명보험 계약관리



8) 사례 8

GS기공소 주문관리



6) 사례 6

VW비디오 대여점 대여관리

