관계형 데이터모델의 확장을 위한 객체지향 설계의 활용 : *H중공업 사례를 중심으로*

김유일*), 신용철**)

Abstract

Database management systems are now used in almost every organization to maintain and utilize important collections of information. Relational DBMSs have a firm theoretical base. However, it lacks important features needed for representing certain critical aspects of an entity. One alternatives to remedy this problem is to adopt the object-oriented approach to the data model development process.

In this paper, we tried to show the advantages of object-oriented data model design process for the development of extended relational database schema needed for the integrated plant information system at *H. Heavy Industry Co.*

1. 서론

대부분의 기업체는 발전된 오늘날의 정보기술에 힘입어 데이터베이스를 의사결정을 지원하는 가장 기초적인 도구로 사용하고 있다. 경영관리의 측면에서 요구되는 조직내 여러 하위시스템간의 통합은 데이터베이스를 그 기반으로 요구하고 있다. 최근 20여년간 가장 많이 사용되고 있는 관계형 데이터베이스 관리시스템은 자료를 테이블 형태의 릴레이션으로 나타낸 간단한 구조를 가지고 있어 사용자들의 접근과 사용이 용이하다. 또한 이산수학에 기초한 확고한 이론적 기반과 SQL과 같은 질의어를 통한 다양한 사용자 지원 기능은 관계형 데이터베이스의 유용성을 더해 주고 있다.

그러나 데이터모델을 실제로 설계함에 있어서 관계형 데이터베이스는 여러 가지 문제점들을 안고 있다. 그 문제점을 해결하기 위한 노력은 객체지향 데이터베이스에서 해결책을 얻으려는 데까지 이르렀다. 객체지향 데이터모델이 관계형만큼 상용화되기에는 아직도 많은 문제점이 있으나, 관계형에 비해 개념적으로 우월한 면이 자주 지적되고 있다. 따라서 관계형 데이터베이스 시스템에서 객체지향 개념을 어떻게 지원할 수 있을까하는 것에 대한 노력이 뒤따르는 것은 당연하다. 이는 실제 사례에서 더욱 절실히 요구되고 있다.

본 연구는 H중공업의 사례에서 데이터모델의 확장에 대한 필요성을 발견하여 관계형 데이터모델을 구축하기 전에 객체지향 데이터모델을 설계해 봄으로써 데이터모델의 확장 방향을 확인하고, 관계형 데이터베이스 관리시스템 기반에서도 지원할 수 있는 방법을 모색하고자 한다.

2. 이론적 배경

2.1 관계형 데이터모델

관계형 데이터베이스는 테이블형태로 표시되는 릴레이션들의 집합으로 이루어진다. 따라서 관계형 데이터모델은 표현하고자 하는 모든 실체를 테이블 형태로 추상화하게 된다. 그러나 시스템 내에 표현하고자 하는 모든 실체들이 테이블의 행과 열의 구조 속에 들어가는 데는 한계가 있으며, 이는 관계형 이론에 기반한 데이터모델의 설계에 있어 표현력의 결핍으로 나타나게 된다.

관계형 데이터모델이 가지는 한계점으로는, 첫째로 복잡하고 중첩된 구조를 가진 실체를 모델링하기에는 너무 단순하며, 둘째로는 정수나 문자, 문자열 등과 같은 기본적인 데이터형만 지원하므로 프로그램에서 볼 수 있는 일반적인 형태의 데이터를 사용할 수 없다는 것이다. 셋째는 일반화와 집단화와 같은 의미적 측면을 나타낼 방법이 없고, 넷째로는 응용프로그램 구현 시에 사용되는 프로그램 언어와 데이터베이스 언어의 데이터 구조와 데이터모델이 많이 달라서 임피던스 불일치의 문제가 있다는 것이다. 이외에도 계산 위주의 응용에 부적합한 성능에 관한 문제나 장기 트랜잭션, 버전 등에 대한 개념을 지원하지 못하는 점들이 추가로 지적될 수 있다.

2.2 객체지향 설계의 활용

이러한 한계점들을 극복하기 위해 관계형 데이터모델을 확장하는데 대한 연구가 많이 진행되고 있다. 그 확장의 방법 중 하나가 객체지향 개념을 지원할 수 있게 하는 것이다. 즉, 릴레이션의 열값이 시스템에서 정의한 자료형뿐 아니라 사용자가 정의한 릴레이션의 튜플이 될 수 있도록 허용하고, 다음으로 객체지향 개념인 캡슐화를 지원할 수 있게 확장함으로써 중첩 릴레이션을 통한 표현력의 증대와 하나의 릴레이션이 그 튜플의 상태와 작동을 캡슐화할 수 있도록 하자는 것이다. 또 한가지는 클래스 계층과 같이 릴레이션의 계층을 지원함으로써 관계형의 독립적인 릴레이션의 집합에 비해 확장성을 높이고, 릴레이션 간의 의미관계를 명확히 하자는 것이다.

이렇게 확장된 데이터모델은 다름 아닌 객체지향 데이터모델인 셈이다. 객체를 그 관리 대상으로 하는 데이터베이스 시스템을 객체지향 데이터베이

*) 부산대학교 경영학과 교수
**)부산대학교 경영학과 석사과정

스 시스템이라 한다면, 객체지향 데이터모델이란 데이터를 모델링하기 위해 유용한 객체지향 개념을 잘 모아서 정의해 놓은 집합이라 할 수 있다. 객체지향 시스템은 그 내재적인 특성상 매우 바람직한 방향으로 시스템 개발과정을 이끌어 가게 되는데 그 내재적인 특성은 데이터 추상화, 객체 캡슐화, 계승, 그리고 다형성 등을 의미한다. 이를 통해서 소프트웨어 엔지니어링에서 강하게 요구되는 추상화, 모듈성, 응집력(cohesion), 정보 은닉과 같은 원칙들이 자연스럽게 지원된다.

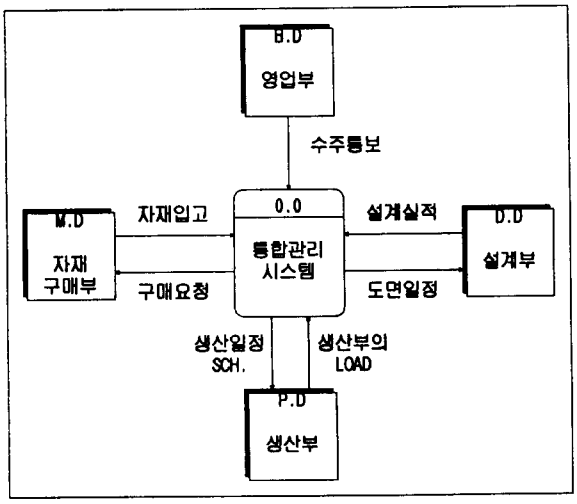
3. 사례 분석

3.1 사례기업의 소개

사례기업은 발전소, 화학공장, 제철공장과 같은 플랜트를 주문 받아 생산하는 기업으로 수주에서 납품까지의 일관적 정보관리시스템과 효율적 임원정보시스템 및 환경변화에 대한 유연한 대응체제를 구축하기 위해 영업, 생산, 설계, 자재관리를 통합하는 관리시스템을 개발하는 중에 있다. 이 시스템은 생산 부서의 설비 및 장치관련 데이터, 자재 부서의 자재관련 데이터, 그리고 설계 부서의 도면관련 데이터를 기반으로 하게 된다. 이들은 모두 관계형 데이터베이스에 기초하여 개발되며, 영업 부서에서 수주가 발생할 때 데이터베이스로부터 통합일정 수립을 위한 기초자료가 추출되는 메커니즘으로 움직인다.

3.2 시스템 개발의 도메인

(그림 1)에 나타난 Level 0 DFD는 사례기업이 목표로 하는 통합관리시스템과 관련 부서의 역할을



(그림 1) Level 0 DFD

잘 보여주고 있다. 사례기업은 자재구매부, 설계부, 생산부, 영업부의 각 기능부문별로 이미 과거 실적에 기초한 데이터가 확보되어 있으며, 약간의 수정을 통해서 새로 개발 중인 Oracle기반의 관계형 데이터베이스로 흡수가 가능한 상황이다.

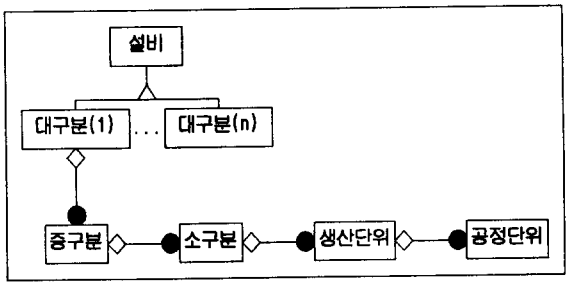
통합관리시스템은 특정 설비에 대한 수주가 발생했을 때, 자재구매부로부터의 자재정보와 설계부로부터의 도면정보 그리고 생산부로부터의 유효공수에 대한 정보를 바탕으로 구매요청과 도면 발행 일정, 그리고 대일정계획, 중일정계획, 소일정계획으로 이어지는 생산일정계획을 제시하는 시스템인 된다.

문제가 되는 것은 주문생산 설비의 특성상 각 설비에 대한 사양을 발주자가 결정하므로 소요도면과 소요자재에 대한 완전한 표준을 사전에 준비할 수 없다는 것이다. 특정 설비가 수주된 후에야 수주설비에 대한 도면이 작성되고, 그 도면을 근거로 소요자재가 결정될 수밖에 없다. 그렇지만 통합관리의 측면에서는 관리의 대상이 될 기초자료가 요구되므로, 과거의 실적을 토대로 기준이 될만한 데이터를 중심으로 수주공사의 사양에 따라 필요한 정보를 어느 정도 산출해 낼 수 있는 데이터베이스를 구축하기로 하였다. 본 연구의 도메인은 통합관리시스템의 기반인 표준 데이터베이스의 구축으로 한정된다.

3.3 시스템개발의 진행방향

통합관리시스템은 하나의 독립된 실체로서의 설비를 그 관리대상으로 한다. 즉, 한 설비에 대해서 그 설비는 어떤 하위장치들이 모여서 이루어지며, 그 각 각의 장치를 만들어 내기 위해 필요한 도면과 소요자재의 조달을 통합 관리하여 세부적인 생산일정 계획의 수립으로까지 연계하겠다는 것이다. 그러므로 시스템의 개발은 각 설비를 관리가능단위로 분해하는 것으로부터 출발한다. 설비를 관리가능단위로 분해하는 것은 설비를 구성하는 각 장치에 대해 실체성을 부여함은 물론 분해의 정도에 따라 해당 단위를 대·중·소일정계획에 대응시키게 된다는 점에서 큰 의미를 가진다.

OMT로 표현된 설비 클래스의 모습은 (그림 2)와 같다. 하나의 설비는 여러 개의 장치들로 구성되고, 그 장치들은 다시 더 작은 장치들의 집합으



(그림 2) 설비 클래스

로 구성된다. 이에 착안하여 우선 설비마다 고유의 분류코드를 부여한 후, 각 설비를 유형별로 크게 대분류하고, 이를 다시 중분류, 소분류하였다. 장치별로 소분류된 모든 단위들은 다섯 자리의 코드를 가지게 된다. 이렇게 소분류된 장치단위는 구체적인 도면정보와 자재정보에 유기적으로 연결될 수 있는 생산단위로 다시 나누어지는데, 이 생산단위는 자재부서에서 구매하여 조달할 자재와 설계부서에서 발행하게 될 모든 도면의 기준단위가 된다. 또 생산단위는 생산현장에서의 공정관리가 가능한 활동수준으로서의 공정단위로 한 번 더 나누어지게 되며, 이 때 생산단위와 공정단위는 별도의 코드를 각 각 부여받게 된다.

4. 데이터모델의 개발

4.1 객체지향 데이터모델의 설계

표준 데이터베이스는 객체로서의 '설비'를 데이터베이스화하는 것이다. 설비는 설비의 상태를 나타내는 여러 속성과 설비의 동작을 나타내는 오퍼레이션으로 구성된다. 설비의 속성은 분류코드와 설비명, 물량, 공수, 제작기간, 설계부서, 생산부서 납기, 도면, 자재 등을 포함하며 설비의 오퍼레이션은 설비명 변경, 물량 가중치 변경, 제작기간 가중치 조정 등과 같은 데이터 조작을 위한 것과 물량 결정, 장치 쉬프팅 등과 같은 클래스 고유의 오퍼레이션으로 나누어 진다. 이 중에서 데이터베이스 관리시스템이 기본적으로 필드값의 수정과 삽입·삭제와 같은 데이터 조작 기능을 제공하므로 그 부분은 따로 고려하지 않아도 된다. 설계된 설비 클래스의 데이터모델이 (그림 3)에 예시되어 있다. OMT의 표기법을 기본으로 하되, 중첩클래스는 화살표를 이용하여 표현하였다. 밑줄로 강조된 속성들은 표준 데이터베이스의 목적에 부합하여 백분율 가중치를 사용하게 된다.

(그림 3)에 나타난 설비 데이터모델을 살펴보면, 우선 설비클래스의 의미론적 특성을 알 수 있다. 즉, 기본적으로 하나의 설비는 해당 설비의 여러 유형들이 일반화(generalization)된 형태를 가진다. 그러나 유형별로 대구분된 이후로는 해당 설비의 장치나 생산단위 또는 작업공정을 기준으로 관리가능 수준으로의 세분화가 이루어 지는데, 이 과정은 집단화(aggregation) 관계로 표현될 수 있다. 또 하나는 생산단위에 대한 도면과 자재의 관계로, 이들은 모두 생산단위의 도면과 자재군이라는 속성에 대응하는 중첩 클래스가 된다.

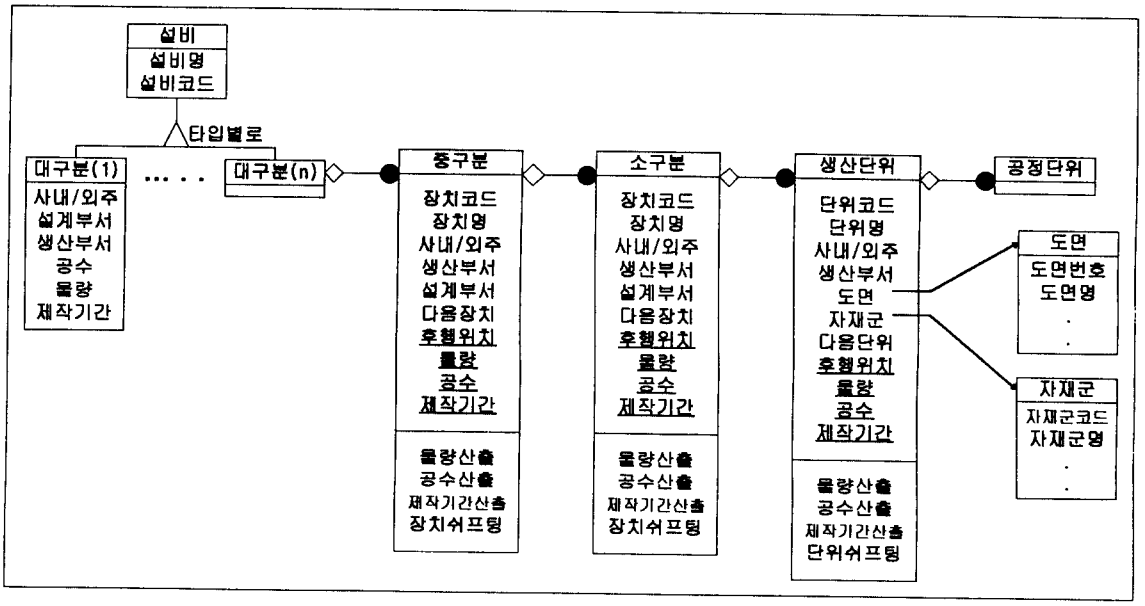
관계형 데이터모델에 대한 확장의 필요성은 확인된 몇 가지의 오퍼레이션을 어떻게 처리할 것인가에서 발생한다. 가장 핵심이 되는 장치쉬프팅 오퍼레이션을 예로 들기로 하자. 표준 데이터베이스는 한 설비에 관한 모든 관련 장치에 대해 구성되며 표준을 위한 가중치의 부여는 상위계층의 장치를 기준으로 하는 구성비로 결정된다. 따라서 수주시에 납기와 설비 규모에 대한 정보가 주어지면 표준적인 일정계획을 산정할 수가 있다. 그러나 주문설비의 특성상 수주단위가 다양하여 소구분장치만을 수주했다거나, 중구분장치 중에서 몇 개만을 수주했다고 할 경우 표준 데이터베이스의 가중치는 완전한 구성비일 수가 없다. 이를 위한 오퍼레이션이 바로 장치쉬프팅으로 수주된 부분만을 포함하는 하위장치 구성비를 즉각 제시하는 오퍼레이션이다. 또한 생산공정의 스케쥴링을 위해 제외된 장치의 후행장치를 선행장치에 연결하는 기능도 포함된다.

통합관리시스템은 표준 데이터베이스로부터 제시된 자료를 기초로 도면일정과 구매요청 및 생산일정을 작성하여 해당 부서에 제시하는 시스템이다. 그러나 발생할 모든 수주의 경우의 수만큼 데이터베이스를 구축하여 수주 발생시 적절한 표준값을 제공한다는 것은 불가능하므로, 확장된 데이터베이스 스키마를 기반으로 요구되는 오퍼레이션 기능을 포함하도록 데이터베이스 관리시스템을 물리적으로 확장함으로써 표준 데이터베이스 시스템을 확립하고자 한다.

4.2 데이터모델의 변환과 확장

데이터모델의 변환은 기본적으로 클래스를 릴레이션으로 바꾸는 과정이 된다. 이때 한 가지 고려해야 할 것은 일반화와 집단화를 통해 논리적으로 명확하게 구성된 객체지향 데이터모델의 본질을 어떻게 최대한으로 살리면서 변환할 것인가 하는 것이다. 테이블 수의 증가와 일부 데이터의 중복을 감수함으로써 의미적인 측면을 살릴 수 있다. 설비 데이터모델의 관계형 데이터베이스를 위한 릴레이션으로의 변환은 최소한 제3정규형을 만족하는 릴레이션으로의 변환을 원칙으로 한다.

설비와 대구분(1…n)간의 일반화 관계에 대해서는 대구분클래스의 애트리뷰트가 많고 설비클래스의 애트리뷰트는 설비명과 같이 식별을 위한 기본정보만 포함되므로, 상위클래스를 하위클래스에 포함시켜 한 테이블로 구성한다. 나머지 집단화 관계에 대해서는 객체들 각각에 대해 하나의 테이블을 구성하는 것이 가장 이상적일 것이다. 테이블 수를 줄여서 성능의 향상과 저장공간의 절약을 꾀할 수도 있겠으나, 테이블을 따로 구성함으로써 데이터모델 설계의 본질을 살림은 물론 주문설비라는



(그림 3) 설비 데이터모델

특수성에 비추어 사후적인 확장의 여지를 남겨 두는 것이 바람직하다. 이는 사실상 일반화 관계는 동일한 한 객체에 대한 관계의 표현이고, 집단화 관계는 분명히 별개의 객체가 집단을 이룬다는 특수한 연관관계에 의해 발생한다는 것을 생각한다면 당연한 결과일 수도 있다. 따라서 (그림 3)의 모델 중 자재클래스를 제외할 경우 모두 13개의 테이블로 변환이 가능하다. 이러한 결과 역시도 객체지향 데이터모델로의 설계과정을 통해 얻어진 효과의 하나인 것이다.

사례기업 데이터모델의 확장문제는 관계형 데이터베이스 관리시스템만으로는 해결될 수 없는 것이므로, 적절한 응용프로그램을 데이터베이스를 기반으로 실행함으로써 실현 가능하게 될 것이다. 이는 데이터베이스 관리시스템에 대한 구조적 확장이며, 설계기법과 구현기법의 불일치로 인한 변칙(anomaly)인 셈이다. 본 연구에서는 PC 기반의 관계형 데이터베이스 시스템인 Microsoft Access 2.0과 Visual Basic 4.0을 이용하여 프로토타입을 개발하고 있다.

5. 기대효과

사례기업이 객체지향 데이터베이스 시스템을 구축하려는 것이 아님에도 불구하고 확장된 관계형 데이터모델을 개발하기 위해 객체지향 데이터모델의 설계를 시도한 것은 확장의 준거를 명확히 하자는 것이다. 왜냐하면, 관계형 데이터모델 확장의 목표가 바로 객체지향 데이터모델이기 때문이다. 시스템을 개발하거나 개선함에 있어 문제가 되는 것 중 하나가 기존의 시스템의 재사용과 여러 연구개발팀의 최종산출물을 통합하는 문제일 것이다. 객체지향 개념은 내재적 특성에 비추어 매우 바람직한 방향으로 시스템 개발의 과정을 이끌어 가게된다. 그러한 내재적 특성은 데이터 추상화, 객체 캡슐화, 계승, 그리고 다형성 등을 말한다. 이를 통해서 소프트웨어 엔지니어링에서 요구되는 추상화, 모듈성, 모듈 내 응집력(cohesion), 정보 은닉과 같은 원칙들이 자연스럽게 지원된다.

사례에서 관계형 데이터베이스 스키마의 확장을 위해 객체지향 데이터모델을 사용함으로써 얻어지는 효익은 다음과 같다. 첫째, 객체지향 데이터모델의 강력한 표현력을 기반으로 데이터모델의 의미적 측면이 논리적으로 깔끔하게 표현되어 릴레이션으로 변환되었다. 둘째, 개발팀을 달리하는 도면과 자재관련 데이터베이스 스키마도 설비클래스 내에 그대로 용해되어 논의하게 됨으로써 기개발된 산출물에 대한 통합의 효과를 얻을 수 있었다. 이는 객체를 중심한 철저한 모듈성과 응집력에 기초하여 얻어진 효익이다. 셋째는 생산일정계획과의 연계성에 관한 것이다. 생산일정 중에서 대일정계획은 장치소구분 클래스에 대해서, 중일정계획은 생산단위에 대해서, 소일정계획은 공정단위에 대해서 이루어지게 된다. 확장된 데이터베이스 시스템을 통해 각 계획들이 해당 릴레이션에 주목하여 계획을 수립할 수 있게 된다.

당장의 효과는 아니지만 사례기업의 경우 표준 데이터베이스 외에도 실적공사 데이터베이스가 유지되고 해당 데이터베이스에 대한 새로운 버전들이 계속 개발되면, 장기적으로는 객체지향 데이터베이스 시스템으로의 전환이 예상될 수 있다. 따라서 의미에서 객체지향 데이터모델의 설계는 상당히 미래지향적인 의미를 가지고 있다.

6. 결론

이제까지 사례를 통하여 관계형 데이터베이스 스키마의 확장을 위한 객체지향 데이터모델의 활용에 대해 살펴보았다. 정보기술의 발달로 관리해야 할 데이터의 유형이 다양해지고, 데이터 자체의 크기도 커진 상태에서 관계형 데이터모델의 한계는 곳곳에서 지적되고 있다. 사례기업의 경우 객체지향 시스템을 기반으로 하지는 않으나, 데이터의 특성상 시스템 개발 도구로서 객체지향 접근법을 사용한 작은 예가 된다.

사례기업의 통합관리시스템이 아직도 개발 중에 있으므로 이상에 소개된 객체지향 기법의 도입은 데이터베이스 구축뿐 아니라 시스템 전체의 설계와 구현에 있어서도 기여를 하게 될 것으로 기대된다. 객체지향 기법의 본질상 실체적 특성을 가진 모든 실체를 하나의 객체로 인식하게 되므로 견적, 자재구매요청, 도면일정 등과 같은 객체들을 추가할 경우에도 이제까지 설계된 데이터모델들은 수정 없이 그대로 하나의 객체로서 시스템 내에 존재하게 되어 개발의 일관성을 유지하게 될 것이다.

그러나 객체지향 데이터베이스 연구의 진행방향에 발맞추어 본 사례에 있어서도 객체들간의 존재 종속성이나, 설비별 데이터의 버전 관리, 그리고 제시된 설비 클래스와 다른 구조를 갖는 예외 설비의 처리 문제 등에 대해서도 지속적인 연구가 뒤따라야 할 것이다.

참고문헌

이석호 外 공역, *객체지향 데이터베이스시스템*, 홍릉과학출판사, 서울, 1995.

W. Kim, *MODERN DATABASE SYSTEMS: The Object Model, Interoperability, and Beyond*, ACM Press, NY., 1995.

J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, & W. Lorensen, *OBJECT-ORIENTED MODELING AND DESIGN*, Prentice Hall, New York, 1991.

Pi-Sheng Deng & C. L. Fuhr, "Using an object-oriented approach to the development of a relational database application system", Information & Management, Vol. 29(1995), pp.107~121.

S. B. Navethe, "Evolution of Data Modeling for Databases", Communications of The ACM, Vol. 35, No. 9(1992), pp.112~123.

A. Silberschatz, M. Stonebraker, & J. Ullman, "Database Systems: Achievements and Opportunities", Communications of The ACM, Vol. 34, No. 10(1991), pp.110~120.