

이용자 시스템 설계를 위한 객체 지향 모델링 방법에 관한 연구

A Study on Object Oriented Modeling Methodology for User System Design

유양근 (Yang-Keung Yoo)*

목 차

- | | |
|-------------------------|--------------------------|
| 1. 서론 | 3.1 객체 지향의 개념 |
| 1.1 연구 목적 | 3.2 객체 지향 분석 방법 |
| 1.2 연구 범위 | 4. 이용자 시스템 객체 지향 모델링 방법론 |
| 2. 기존 데이터 모델의 문제점 분석 | 4.1 이용자 시스템의 개념 |
| 2.1 데이터 모델의 개념과 요건 | 4.2 이용자 시스템 분석 |
| 2.2 기존 데이터 모델의 특성 및 문제점 | 4.3 객체 모델링 |
| 3. 객체 지향 개발 방법론 | 5. 결론 및 제언 |

초 록

기존의 모델링 방법은 분석, 설계, 구현 단계에서 시스템이 수행해야 할 기능적 및 비기능적 요구를 완전하고 일관성 있게 반영할 수 있는 설계 절차와 명세서가 미흡하다. 이러한 문제점은 다양한 형태의 자료와 다양한 검색 방법을 요구하고 있는 문헌정보 시스템에서는 문제가 있을 것으로 본다. 따라서 본 논문에서는 OMT에서 제시되고 있는 방법을 부분적으로 개선하여 시스템 개발에 적용할 수 있도록 OMT를 변형한 객체 모델링을 도서관 이용자 시스템 요구사항을 대상으로 설계하였다.

ABSTRACT

In analyzing, designing, implementing the system, the traditional modeling method lacks design procedure and specification in reflecting the functional or nonfunctional requirements perfectly and consequently. These problems are unreliable in the Library and Information System since it requires a diversity of reference methodes based on various types of data. In this paper, we modify the method partially which the OMT proposes and apply it on system development. We also prepare the object oriented modeling modified from OMT type's modeling for Library User System Requirements.

* 강남대학교 사회과학대학 문헌정보학과 교수

■ 논문 접수일 : 1997년 11월 12일

I. 서 론

1.1 연구 목적

객체 지향 모델링 기술은 소프트웨어의 시스템의 대규모화, 복잡화, 멀티미디어 응용의 다양화 등, 새로운 응용분야에 모델링 기법으로 정착하고 있다. 이는 최근 명세변경에 따른 회피할 수 없는 보수비용의 증가, 분산개발에 따른 문제, 등 소프트웨어 개발과제를 해결해주는 기술이기 때문이다.¹⁾

기존 모델링에 의한 개발 방법은 문자 정보 이외의 다양한 형태의 정보나 멀티미디어 데이터베이스 활용시 여러 가지 문제가 발생하고 있다. 본 연구에서는 이러한 점을 감안하여 개발자와 이용자와의 의사소통의 원활을 기하고, 이용자의 요구를 충족시킬 수 있는 데이터베이스 설계를 위해 요구분석 사항을 객체 지향 방법으로 모델링한다.

문헌정보 데이터베이스의 이용자 시스템의 요구사항과 객체 지향 모델링 방법을 제안하는 목적은, 객체 지향 시스템 설계의 효율성을 극대화 시키고, 새로운 시스템 개발을 전제로 한 도서관 이용자 시스템의 요구 분석 모델링을 정형화 하는데 있다.

1.2 연구 범위

본 연구에서는 데이터베이스 설계를 위한 모델링의 일반적 방법론과 기존 모델링의 문

제점, 차세대 데이터베이스의 요구사항을 분석하고 데이터베이스 구축을 위한 객체 지향 모델링 기법을 도서관 이용자 시스템으로 한정하여 적용 하였다.

제안된 객체 지향 모델링의 적용범위는 다음과 같이 제한하였다.

첫째, 문헌정보 전산시스템의 기본 모형을 구축하는데 절대 요소인 이용자 시스템을 분석한 요구사항에 한정하였으며,

둘째, 요구사항 기술을 표현한 다이어그램은 OMT에서 제시한 기본 모델링을 토대로 본 연구에서 작성된 분석 테이블을 사용하여 작성된 개략적 모델과 통합 객체모델링을 제안하였으며, 또한 객체모델 다이어그램(Object-Model Diagram)과 데이터사전(Data Dictionary)을 제시하였다

2. 기존 데이터 모델의 문제점 분석

2.1 데이터 모델의 개념과 요건

(1) 데이터 모델의 개념

데이터 모델은 데이터 타입, 관계성, 데이터에 관한 제약조건 등에 관한 것들을 기술하는 개념적인 도구이다. 데이터 모델은 사용자가 데이터를 감지하는 방법에 가까운 개념을 제공해 주는 개념적 데이터 모델(conceptual data model)과 데이터가 컴퓨터 내에서 조직되는 방법에 크게 무관하지 않으면서도 사용자가 이해할 수 있는 개념을 제공하는 구현 데

1) B. Stroustrup, "What is Object-Oriented Programming?", IEEE Software, Vol. 5, No. 3, 1988, pp. 10-20.

이터 모델 (implementation data model)로 나눈다. 개념 데이터 모델의 대표적인 모델은 ER 모델, Functional 모델 등을 들 수 있고, 구현 데이터 모델의 대표적인 모델은 관계형 모델, 계층형 모델 등을 들 수 있다. 데이터베이스 설계는 먼저 현실 세계를 개략적으로 기술한 개념적 모델을 이용하고 이 개념적 스키마를 논리 스키마(schema)²⁾로 전환한다.

따라서 데이터 모델은 실제의 물리적 저장 장치 위에 저장된 데이터베이스의 구조를 사용자 입장에서 표현한 논리적 구조라고 정의할 수 있다.³⁾

데이터를 효과적으로 조작할 수 있는 방법을 결정하기 위해서는 중요한 데이터의 특징들을 이해해야 하며 이런 특징들은 컴퓨터에 의해서 표현될 수 있어야 한다. 따라서 필요로 하는 데이터에 관련된 적절한 양의 의미를 파악하는 것이 무엇보다 중요하다. 데이터를 구상화(visualizing)하고 구조화(structuring)하는 주요한 방법중에 하나가 추상화(abstraction)기법을 이용하는 것이다.⁴⁾ 데이터 모델은 추상적 개념으로 조직된 구조를 가지며, 이 구조에서 허용되는 여러 가지 연산을 통해 데이터를 검색 또는 조작시키고, 데이터베이스 상

태의 변화, 데이터베이스에 저장된 데이터 값과 현실 세계의 값이 일치하기 위한 제약조건을 가져야 모델로서 역할을 할 수 있다.

(2) 데이터 모델의 요건

데이터 모델은 현실을 나타내는 도구로 설계자는 데이터 모델을 이용하여 현실의 표현인 스키마를 작성하며, 사용자는 데이터 모델을 통하여 데이터베이스 구조를 이해하게 된다. 그러므로 데이터 모델이 실세계를 잘 표현할 수 있어야만 정확하고 일관성 있으며 공유 가능하고 유연성 있는 데이터베이스가 구축될 수 있다. 이를 위하여 데이터 모델은 다음과 같은 성질을 가져야 한다.⁵⁾

1) 표현력 (Expressive)

많은 다양한 개념을 사용하면 실세계를 더욱 잘 표현할 수 있다. 그러므로 풍부한 데이터 모델은 또한 표현력도 매우 강하여 실세계를 정확하게 표현할 수 있을 뿐만 아니라 모델의 관리에도 용이하다.

2) 단순성 (Simplicity)

데이터 모델은 단순해야 그 모델을 사용한 스키마를 설계자와 사용자가 쉽게 이해할 수 있다. 표현력과 단순성은 상관관계가 있으므로 적절한 수준에서 조절하여 표현력과 단순

2) 스키마란 현실 세계의 특정한 한 부분의 표현으로서 특정 데이터 모델을 이용해서 만들어진 다. 스키마에 의해 정의된 데이터 구조에 맞는 데이터의 집합을 인스턴스(instance)라 하며 이는 시간에 따라 변하는 다이내믹한 특성을 갖고 있다. 각 스키마는 여러 인스턴스를 가질 수 있으며 특정 시간에 데이터베이스의 상태는 그 당시의 인스턴스들에 의해 결정된다.

3) Smith, J.M., and Smith, C.P., "Database Abstractions : Aggregation and generalization", ACM TODS, vol.2, 1977, pp.105-133.

4) D. Tschritzis and F. Lochovsky, "Data Models", Prentice-HALL, Englewood Cliffs, N. Y. 1982. pp.105.

5) Batini, C., Ceri, S. and Navathe, S., Conceptual Database Design. : An Entity-Relationship Approach, The Benjamin/Cumming Publishing, 1992. pp.15-54.

성이라는 상반된 모델이 통일된 방법으로 실세계를 나타내도록 해야 한다.

3) 최소성 (Minimality)

어떤 개념도 다른 개념을 합성해서 만들 수 없어야 한다. 즉, 한 가지 개념은 그 개념을 통해서만 표현할 수 있어 설계된 모델이 통일된 방법으로 실세계를 나타내도록 해야 한다.

4) 정형성 (Formality)

데이터 모델을 사용해서 만들어진 스키마는 데이터의 정형화된 명세를 나타낼 수 있어야 한다. 즉, 모든 개념은 유일하고 간결하고 잘 정의된 해석을 가져야 한다.

5) 편리성 (Convenience)

데이터 모델은 사용하기 쉬어야만 초보자들도 쉽게 데이터 모델을 이해하고 작성을 용이하게 할 수 있다. 모델링을 위한 그래픽 도구는 모델 사용을 용이하게 하는 수단이 된다.

2.2 기존 데이터 모델의 특성 및 문제점

(1) 개체관계 모델 (ER : Entity-Relationship Model)

실세계에 존재하는 사물의 유형과 그들 상호간의 관계를 가지고 시스템을 표현하기 위한 개념적 모델로서 개체관계 모델은 개체들과 그들간의 관계성, 그리고 그들이 가지고 있는 속성들로서 구성되어 있다.⁶⁾ ER모델은 단계적이고 계층적인 하향식 접근에 의한 모델링을 수행함으로써 데이터베이스 설계자가 보다 용이한 방법으로 데이터베이스 스키마를

개발할 수 있도록 한다. ER 모델링의 기본적인 절차는 다음과 같다.

- 1) 개체를 찾아낸다
- 2) 개체들간의 관계성을 찾아낸다
- 3) ER 다이어그램을 작성한다
- 4) ER 다이어그램에 있는 개체와 관계의 속성을 결정한다
- 5) 키를 찾아낸다

ER 다이어그램을 작성한 후 관계 스키마인 릴레이션으로 변환시켜줌으로써 DBMS특성에 맞는 논리적 데이터 모델을 만들어 준다. 개체들과 관계성들은 변환 규칙에 의해 릴레이션으로 변환된다. 변환된 릴레이션은 릴레이션의 속성간에 존재하는 종속을 고려하여 릴레이션을 분해하는 정규화 과정을 거쳐 데이터베이스 구조를 최적화 시킨다. 논리적 데이터 모델은 데이터베이스의 물리적 구조로 바로 사상(Mapping)될 수 있어야 한다. ER 모델은 효과적인 도형화 기법 (ER Diagram)을 통하여, 사용자와 분석자 간의 의사 소통을 용이하게 하고, 간단하게 개체와 그들간의 관계성을 표현함으로써 데이터베이스의 개념적 설계를 위해 가장 광범위하게 사용하는 기법이다. 이러한 유용성에도 불구하고, ER 모델은 크고, 복잡한 시스템을 모델링하기에는 엔티티와 관계성에 대한 서브 구조가 부족하다. 즉, 뷰 통합을 위해 일반화 같은 추상화 개념이 제공되어야 하며, 널 속성 값을 갖는 데이터 무결성을 위해 관계성이 널 값을 허용하는지 아닌지에 대한 정의가 요구된다.

6) Chen, P., " The Entity-Relationship Model : Toward a Unified View of Data ", ACM Trans. Database System, Vol. 1, No.1, 1976. pp. 9-36.

(2) 확장 개체관계 모델(EER:Extended Entity-Relationship Model)

ER 모델의 단점을 개선하기 위해 연구된 EER 모형의 구조는 개체와 속성(property)으로 되어 있으며 선택(Optional) 관계성, 서브집합(Subset), 일반화 개념을 추가하였다. EER 모델을 이용하여 관계형 데이터베이스를 설계하기 위한 LRDM(Logical Relational Design Methodology) 방법론이 제안되었다.⁷⁾ EER 모델은 실제 데이터 요구에 대한 사항을 정확하게 잡아내어 설계자의 능력을 향상시키고, 세부적인 의미를 파악할 수 있을뿐 아니라 일반화와 같은 추상화 개념을 지원한다. 그러나 EER 모델에서 일반화는 동일한 속성을 갖는 개체들을 통합하는 개념만을 제공할 뿐 객체 지향 프로그래밍에서의 일반화 처럼 상속의 개념을 지원하지는 못한다. 이 모형의 특징은 시스템이 정의하는 기본키를 사용한다는 것이며 EER 모델의 개체유형(Entity Type)은 다음과 같이 세 가지 유형이 있다.

- 1) 독립된 존재로서의 개체인 핵심 개체유형(Kernel Entity Type)
- 2) 다른 개체를 기술하거나 성격지어 주는 개체로서 특성 개체유형(Characteristic Entity Type)

- 3) 두 개 이상의 개체 사이의 N : M의 관계를 나타내 주는 관련 개체유형(Associative Entity Type)

3. 객체 지향 개발 방법론

3.1 객체 지향의 개념

객체 지향적 개발을 위해 객체 지향 개념을 나타내는 도식적 표기법을 객체 지향 모델링이라고 한다. 객체 지향 접근법은 먼저 적용업무 영역으로부터 객체를 식별하고 이에 절차적 요소를 결합함으로써 요구사항이 변화할 때 객체 지향 소프트웨어의 적절한 지원으로 해결할 수 있다. 이는 단일 문제 상황에 대한 특별한 기능적 요구사항이 아닌 적용업무 자체의 근간이 되는 구조(underlying framework)에 근거하기 때문이다.⁸⁾ 따라서 객체 지향 모델링은 실제 세계에 관한 여러 사실 또는 사물에 대하여 주어진 문제에 객체 지향의 여러 개념을 사용하여 분석 및 설계를 가능하게 하는 새로운 모델링 기법이라 할 수 있다.⁹⁾ 객체 지향 모델링은 문제이해, 응용 전문가와의 의사소통, 정보 조직체 모델링, 문서준비, 그리고 프로그램 및 데이터베이스 설계에 유용

7) Toby, J. T., Yang, D., James and P. Fry, "A Logical Design Methodology for Relational Database Using the Extended Entity-Relationship Model", Computer Survey, Vol. 18, No. 2, 1986. pp. 197-222.

8) J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen, "Object-Oriented Modeling and Design", Prentice-Hall International, 1991. pp. 215.

9) I. Jacobson, M. Christerson, P. Jonsson, G. Overgaard, "Object-Oriented Software Engineering : A Use Case Driven Approach", ACM, Inc. 1992. pp. 43-62.

〈표 1〉 객체 지향 용어와 개념

기본용어	객체, 클래스(/인스턴스), 속성, 메시지, 메소드, 관계(관련, is-a, part-of)
기본개념	상태, 추상화, 캡슐화, 상속
응용개념	추상클래스, 다중상속, MIX-IN, 다형화, 동적결합, 병행성, 통신

하다고 본다.¹⁰⁾

(1) 객체 지향의 일반적인 개념

객체 지향 시스템의 기본적인 구조는 다음 〈표 1〉과 같이 객체, 클래스, 속성, 메시지, 메소드, 관계의 6가지 용어에서 설명할 수 있다.

Coad와 Yourdon은 객체 지향은 객체(Object), 클래스(Classification), 상속성(Inheritance)이 조직화된 것으로서 최근에는 통신(Communication)까지 연결하여 조직화하여야 한다고 정의를 내렸다.¹¹⁾ 객체 지향 접근방법에 요구되는 특징이 정확히 무엇인가에 대해서는 많은 논란이 있지만,¹²⁾ 일반적으로 객체의 특성으로 다음과 같이 식별성(identity), 클래스화(classification), 다형성(polymorphism), 상속성(inheritance)의 네 가지 측면에서 표현된다.¹³⁾

1) 식별성

객체(object)로 표현되는 데이터들이 각각 독립적인 개체(entity)로서 다른 개체와 구분되는 것을 말한다. 객체는 파일 시스템의 파일처럼 물리적이거나, 멀티프로세싱 운영체제에서 policy scheduling 같은 개념적일 수 있으며 각 객체는 본질적으로 사람이 생각할 수 있는 실체 즉, 식별성(identity)¹⁴⁾을 가지고 있다.

2) 클래스화

클래스화(classification)란 하나 이상의 유사한 객체들을 묶어서 하나의 공통된 특성을 표현한 것으로서 데이터 구조와 행위를 가진 객체가 한 클래스로 그룹되어짐을 의미하고, 클래스는 객체들의 집합으로 정의될 수 있다.¹⁵⁾ 각 객체는 그 클래스의 인스턴스(instance)라 말할 수 있으며 한 클래스에 속한 모든 인스턴스들은 속성 이름들과 연산(operation)들

10) J. Martin, J. Odell, "Object-Oriented Analysis & Design", Prentice-Hall, 1992, pp. 29-42.

11) P. Coad, E. Yourdon, "Object-Oriented Analysis", Prentice-Hall, 1990, pp. 1-7.

12) G. Booch, "Object-Oriented Development", IEEE Trans. Software Engineering, Vol. SE-12, No. 2, February, 1986, pp. 211.

13) B. Meng, "Object-Oriented Programing", MacWorld, Vol. 7, No. 1, January, 1990, pp. 174-183.

14) 객체의 식별성(identity) : 객체 지향 시스템에서는 객체가 시스템을 구성하는 기본 단위가 된다. 모든 객체(object)들은 다른 객체들과 구별할 수 있는 이름을 갖는다. 이름은 제한된 영역을 가져야 하며, 시스템 설계시 각 객체들의 관계를 고려해야 한다. 즉, 이름은 객체들이 가지고 있는 하나의 값(value)이나 속성(attribute)을 의미한다.

15) 클래스(class)는 공통된 특성과 행위를 갖는 객체들을 표현한 것으로, 클래스는 객체 고유의 상태를 저장하는 메모리 영역인 인스턴스(instance) 변수와 이를 조작하는 연산의 정의들로 구성된다.

을 공유하고, 각 인스턴스는 서로 다른 속성 값을 가질 수 있다.

3) 다형성

다형성(polymorphism)은 다른 클래스들 내에서 같은 이름의 오퍼레이션들이 다르게 정의될 수 있음에 기인한다. 따라서 어떤 오퍼레이션은 컴파일 또는 실행 시간에 그 해당 클래스가 결정됨에 따라 해당 클래스에 정의된 내용대로 행동하게 됨을 의미한다. 오퍼레이션¹⁶⁾은 객체가 수행하는 입출력, 또는 데이터에 대한 연산 및 변형에 관한 작업으로 정의될 수 있으며 이는 클래스에서 메소드(method)로 표현된다.¹⁷⁾

4) 상속성(inheritance)

계층적 관계에 기반한 클래스들 사이에 속성과 오퍼레이션의 공유를 말한다. 한 클래스는 여러 하위클래스들로 나뉠 수 있다. 각 하위클래스는 자신의 상위클래스의 속성과 오퍼레이션을 상속하고 자신만의 고유한 속성을 갖는다.¹⁸⁾

(2) 객체 지향적 개발의 의미

객체 지향적 개발은 현실세계의 복잡한 사

실들의 추상화에 기초한, 소프트웨어에 대한 새로운 사고방식을 의미한다.¹⁹⁾ 객체 지향적 개발의 본질은 실제 프로그래밍을 통한 구현 보다는 오히려 응용분야에서 객체 지향에 따른 여러 개념들의 식별 및 조직화를 강조하게 되며 객체 지향적 개발의 의미는 다음과 같다.²⁰⁾

첫째, 프로그래밍언어를 이용한 구현이 아니라 개념을 모델링하는 것이다. 객체 지향적 개발은 마지막 단계까지, 프로그램 언어를 통한 구현과는 독립적인 개념적 프로세스다. 객체 지향적 개발은 근본적으로 사고의 새로운 방법이지 프로그래밍 기술이 아니다. 객체 지향적 개발을 통해 개발자와 사용자는 추상적 개념을 명확하게 표현할 수 있으며, 의사소통을 가능하게 한다.

둘째, 객체 지향적 개발은 먼저 적용업무 영역으로부터 객체를 식별하고 이에 절차적 요소를 결합함으로써 요구사항이 변화할 때 객체 지향 소프트웨어의 적절한 지원으로 해결할 수 있다.²¹⁾ 객체 지향적 개발을 위한 방법론은, 응용분야의 개념들을 모델링하고 시스템

16) operation과 method는 특별히 구별하여야 할 경우를 제외하고는 서로 혼용해서 사용된다.

17) 하나의 연산자나 함수가 여러 자료형의 객체에 적용될 수 있음을 의미하며, 다형성은 각 객체가 동일한 메시지에 대하여 자신이 속한 클래스에 적합한 방식으로 반응하는 기능을 제공한다.

18) 상속이란 이미 정의된 클래스(상위클래스)를 이용하여 새로운 클래스(하위클래스)를 정의할 때, 하위클래스가 상위클래스의 자료형, 연산 등 모든 특성을 계승받는다를 것을 말한다. 따라서 계승의 개념을 이용하면 클래스 정의 자체를 수정하지 않고도 유사한 성질을 갖는 클래스들을 쉽게 작성할 수 있다.

19) I. Jacobson, M. Christerson, P. Jonsson, G. Overgaard, op. cit., pp. 70-80.

20) J. Rumbaugh et all, op. cit., pp. 4-7.

21) 기존의 기능 지향적 방법론은 시스템의 기능을 상술(specifying)하고 분해(decomposition)를 강조하여 목표 시스템 구현에 직접적인 방식으로 이용되고 있으나 요구사항이 변화하는 경우 대량의 재구조화를 요구하는 단점이 있기 때문에 이러한 문제를 해결하기 위하여 최근객체 지향 방법론을 택하는 사례가 늘고 있는 추세이다.

을 설계하는 기간에도 모델 구현에 관한 세부 사항을 추가할 수 있다.

셋째, 객체 모델링 기술 방법론을 도입하여 사용할 수 있다. 객체 모델링 기술은 시스템을 구축하기 위해 객체 모델(object model), 동적 모델(dynamic model), 기능적 모델(functional model)을 사용한다. 객체 모델은 시스템 내의 객체의 정적 구조와 각 구조 사이의 관계를 묘사하며 객체 다이어그램(object diagram)으로 설계하는데 사용하며, 동적 모델은 시간의 경과에 따른 시스템의 변화를 묘사하고, 시스템의 제어에 관한 측면을 명확하게 구현하는데 사용된다. 이 모델은 상태도(state diagram)를 포함한다. 기능적 모델은 시스템 내의 데이터 값의 변화를 설명하는 자료흐름도(DFD : data flow diagram)를 사용한다.

(3) 객체 지향 접근 방법

소프트웨어의 생산성을 향상시키는 기법으로는 재이용 기술이 기대되고 있지만, 생각처럼 진보하지 않는 것이 사실이다. 특히 소프트웨어 개발 생명주기의 상위단계에 있어서 재이용 및 명세 변경에 따른 보수비용의 증가도 회피할 수 없는 과제이며 최근 주목 받고 있는 분산개발에 적용되지 못하고 있다.²²⁾ 객체 지향은 아래와 같은 특징을 가지고 있기 때문에 이러한 소프트웨어 개발현상의 과제를 해결해주는 최신 기법이라 할 수 있다.²³⁾

1) 추상화

추상화(Abstraction)는 사건의 발생에 따라 변하는 개체의 사건중심적 속성보다 개체의

본질적인 측면에 관심을 갖는다. 시스템 개발에서 이것은 객체가 어떻게 구현되어야 하는가를 결정하기 전에, 객체가 무엇이고 무엇을 하는가에 중점을 둔다는 것을 의미한다.

2) 캡슐화

캡슐화(Encapsulation)는 다른 객체들이 접근할 수 없는 객체 내부의 구현에 관한 세부 사항과 다른 객체가 접근할 수 있는 객체의 외부적인 측면을 분리하는 것이다. 자료 구조와 행위를 결합하는 능력은 캡슐화를 더 명확하게 하고, 자료 구조와 행위를 구분하는 기존의 프로그래밍 언어보다 강력하게 한다.

3) 공유

자료구조와 행위의 상속은 여러 비슷한 하위 클래스 사이에서 일반적인 구조가 공유(Sharing)될 수 있게 한다. 객체 지향적 개발은 응용내에서 정보가 공유되게 할 뿐만 아니라 미래의 프로젝트에 설계와 코드를 재사용할 수 있게 한다.

4) 객체구조

프로시저(procedure) 구조가 아니라 객체 구조(Object Structure)를 강조한다. 객체 지향적 기술은 객체가 어떻게 사용되는가보다는 객체가 무엇인가를 명확히 설명하는 것을 강조한다.

5) 시너지(Synergy)

식별성, 클래스화, 다형성, 그리고 상속성은 객체 지향적 언어의 주된 특징이다. 각 개념들은 독립적으로 사용될 수 있으나, 함께 사용되어 각 개념의 효과를 합한 것 이상으로 상승적

22) J. Rumbaugh et al, op. cit., pp. 7-9.

23) J. Martin, J. Odell, op. cit., pp. 15-28.

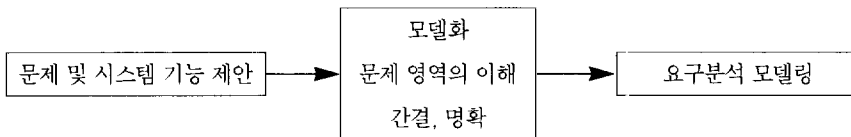
인 효과를 내며,²⁴⁾ 소프트웨어 개발자에게 객체의 본질적인 속성을 이해할 수 있게 하고 객체의 의미와 기능을 더욱 주의깊게 사고하도록 한다.²⁵⁾

3.2 객체 지향 분석 방법

〈그림 1〉은 이용자 시스템 분석단계를 나타낸 것이다. 요구모델의 정의는 분석 목적에 따라 다소 다르나²⁶⁾ 실세계(real world)의 이해를 근거로 삼아 시스템 기능을 확립하는 것을 목적으로 한 경우에는 실세계의 모델화를 말한다. 요구 분석 모델링은 문제 영역의 이해를 깊이하고 무엇을 시스템으로써 실현할까를 명확하게 하는 작업이다. 요구모델을 표현할 때는, 시스템 대상의 정확한 파악이 중요하다. 일반적으로 객체 지향에서는 대상의 정적인 구조에 관해서는 객체 모델, 대상에 있어서 이벤트의 실행순서(동적인 행동)에 관해서는 동적 모델, 대상에 있어서 자료 변환(기능)에 관

해서는 기능 모델을 표현한다. Meyer는 소프트웨어가 가지는 데이터와 그 데이터에 관한 처리를 비교할 때 데이터 자체의 변경보다는 데이터에 관한 처리 방법의 변화가 소프트웨어의 변경 요인이 되고 있다고 말하였다. 따라서 기존의 기능 중심의 분해에 의한 설계 방법보다는 데이터 중심의 시스템 분석 방법인 객체 지향적 설계방법이 소프트웨어의 생산성 향상과 유지 보수를 위한 효율적인 방법이라고 말할 수 있다.

객체 지향적 설계방법이 기존 설계방법에 비해 특별히 인정받는 이유는 문제 해결을 위해 다루어야 할 현실 세계의 객체들을 컴퓨터로 사상시켜, 소프트웨어 객체들이 현실세계의 객체들의 행동을 반영하고 해결할 수 있다는 점이다. 또한 상속(inheritance)과 정보은닉(information hiding)이란 특성을 통하여 소프트웨어 시스템의 장기적인 제반 사항에 대한 많은 변화 요인에 용이하게 적응할 수 있는 능력을 제공함으로써 소프트웨어의 재사용

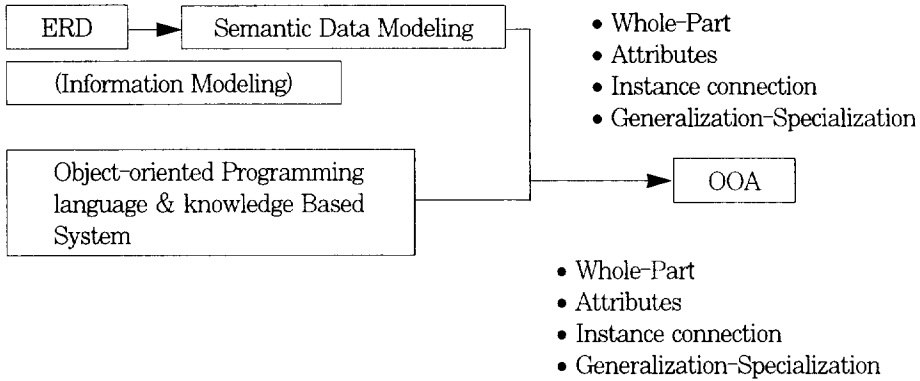


〈그림 1〉 이용자 시스템 분석단계

24) J. C. Brad, "Object-Oriented Programming", Reading Massachusetts : Addison-Wesley, 1986, pp. 12-18.

25) J. Rumbaugh et al, op. cit., pp. 9-15.

26) 시스템 기능을 확립하는 것을 목적으로 한 경우는 발주자가 원하는 시스템의 이상형을 말하며 실세계의 이해를 근거로 삼아 시스템 기능을 확립하는 것을 목적으로 한 경우에는 실세계의 모델화를 말하며 전자에 해당하는 것이 구조적 분석이고, 후자는 객체 지향 분석이다. 본고에서는 객체 지향 방법을 적용하였다.



〈그림 2〉 객체 지향분석의 구성³⁰⁾

성과 확장성 및 유지보수를 효율적으로 성취할 수 있다는 점이다.²⁷⁾ 이러한 특성들과 장점들이 설계 단계에서 올바르게 시스템에 적용되고 구성되기 위해서는 기본적으로 요구사항의 출현에서부터 객체 지향적 메커니즘이 고려되어야 한다. 결국 요구사항에 대해서 객체 지향적 분석 결과를 통한 설계 수립이 가장 이상적인 객체 지향적 개발의 과정이라고 할 수 있다.²⁸⁾

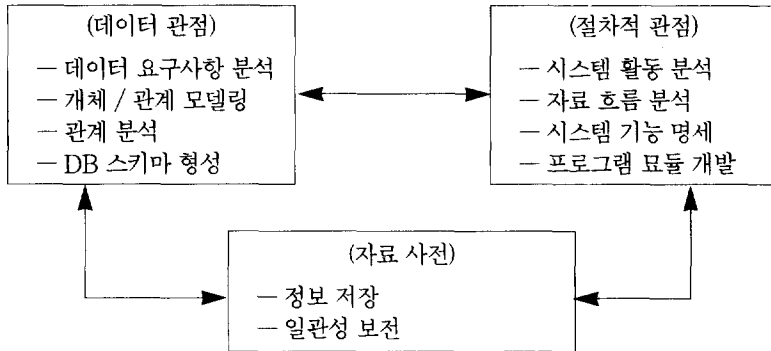
(1) 전통적인 분석방법과의 비교

객체 지향 분석은 정보 모델링 (information modeling), 다음 〈그림 2〉과 같이 객체 지향 프로그래밍 언어 (object-oriented languages), 지식기반 시스템 (Knowledge-Based System)으로부터 그 개념을 정립하고 있다. 즉, 정보 모델링으로부터는 Attribute,

Connections, Generalization, Specialization, Whole-Part 개념을 도입하고 객체 지향 프로그래밍어 및 지식기반 시스템으로부터는 속성의 캡슐화 (encapsulation of Attributes), exclusive service, communication with message, Generalization-Specialization, 및 상속 (Inheritance)을 도입하였다.²⁹⁾

전통적인 시스템 개발 방법은 정보시스템을 두가지 관점 즉, 데이터적 관점과 절차적 관점으로 분리하고 있다. 데이터 관점은 개체/관계 모델링 (Entity-Relationship Modeling), 관계성 분석 (Relational Analysis), 궁극적인 데이터베이스 스키마 생성 및 데이터베이스의 물리적인 구현에 관한 것이고, 절차적 관점 (Process Perspective)은 시스템의 기능적 요

27) B. Meyer, "Reusability : the case for object-oriented design", IEEE Software, Vol.4, No.2, 1987, pp. 50-64.
 28) W. E. David, D. K. Barry, S. N. Woodfield, "Object-Oriented Systems Analysis : A Model-Driven Approach", Yourdon Press, Prentice Hall, 1992, pp. 2-16.
 29) P. Coad, E. Yourdon, "Object-Oriented Analysis", Prentice-Hall, Inc. 1991, pp. 30-32.
 30) P. Coad, E. Yourdon, op. cit., p. 31.



〈그림 3〉 전통적인 구조화된 분석

구사항에 관련된 것으로 자료흐름도 (DFD : Data FLOW Diagram). 시스템 기능 및 프로그램 코드의 컴파일 단위 (Mod-ules)에 관한 것이다. 반면에 객체 지향적인 개발 방법에서는 집단화 (aggregation)의 원리를 데이터의 추상화에 중점을 두어 모든 기능 (function)들이 특정한 객체와 관련되어야 하고 그래서 그러한 기능들이 어떤 추상화된 데이터 상에서 작용하는 것이면 이를 모두 함께 묶는다. 이런 방식으로 객체는 상황 (state)과 행동 (behaviour)을 캡슐화 한다. 전통적인 구조화된 분석

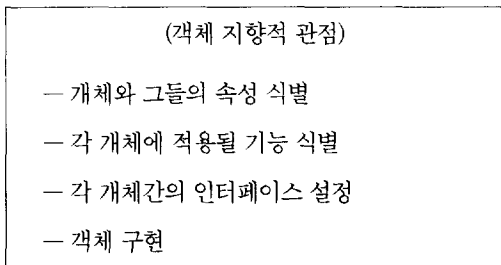
방법과 객체 지향적인 분석방법을 도표화 하면 다음 〈그림 3〉, 〈그림 4〉과 같다.

(2) 객체 지향 분석 모델

객체 지향 분석 모델링은 시스템의 정적 (static)이고 구조적 (structure)인 데이터 측면을 나타내는 객체모델 (object model)과 시스템의 일시적 (temporal)이고 행동적 (behavioral)인 통제 (control) 측면을 나타내는 동적모델 (dynamic model), 시스템의 변환적 (transformational) 기능 (function) 측면을 나타내는 기능 모델 (functional model)의 3가지 형태로 표현된다.³¹⁾ 분석 모델을 개략적으로 설명하면 다음과 같다.

1) 객체모델 (object model)

객체모델은 시스템에 존재하는 객체와 그 객체들간의 관계 그리고 객체의 각 클래스를 특징 지우는 속성 (attribute)과 연산 (operation)을 나타내도록 시스템을 정적인 구조로 그리고자 하는 것이다. 객체 모델은 객체 클래스



〈그림 4〉 객체 지향 분석

31) I. Jacobson, M. Christerson, P. Jonsson, G. Overgaard, op. cit., pp. 132-138.

스를 포함하는 객체 다이어그램(object diagram)을 이용하여 도식적으로 나타낸다. 이때 클래스는 각 객체 실례(instance)에 의해 전달되는 속성값과 각 객체가 수행하는 연산을 정의한 것이다. 객체 다이어그램에서 클래스는 공통된 구조와 행동을 공유할 수 있도록 계층 구조로 정렬하고 다른 클래스와 연관지을 수 있도록 그려진다.³²⁾ 이러한 객체 모델은 동적 모델이나 기능 모델들이 형성될 수 있는 기본적인 형태(framework) 제공한다.

2) 동적모델(dynamic model)

어떤 특정 시점에서의 객체와 그들간의 관계를 정적으로 나타내는 객체모델에 대비하여 시간이 지남에 따라 객체와 그들 간의 관계가 변화하는 것을 고찰하고자 하는 것이 동적 모델이다. 동적 모델에서는 상황 다이어그램(state diagram)을 이용하는데 이 다이어그램은 외부의 자극에 의해서 일어나는 사건(event)과 이에 따른 객체의 값을 나타내는 상황을 시간이 지남에 따라 변화하는 모습을 동적으로 도식화하는 것이다.

3) 기능 모델(functional model)

기능모델은 값과 값, 또는 함수에 의한 기능 종속성을 데이터 흐름 다이어그램(DFD : Data-Flow Diagram)을 이용하여 프로세스

의 내에서 처리 순서나 객체의 구조에 무관하게 시스템내의 여러 값들이 어떻게 처리되고 계산되어지는가를 정형적으로 나타낸다.

(3) 객체 모델링(Object Modeling)의 평가

모델링의 주요 목적은 첫째, 구현하기 전에 물리적인 개체를 테스트하기 위해, 둘째, 사용자와의 효율적 의사소통을 하는데 있으며, 셋째, 문제의 가시화(可視化)와 넷째, 복잡성의 감소를 들 수 있다.

객체 지향 설계 방법은 현실 세계의 문제들을 객체로 규정지어 컴퓨터로 사상시켜 소프트웨어 객체들로부터의 문제에 대한 행동들을 반영하는데 있으며 시스템이 수행하는 기능보다는 취급하는 데이터를 중심으로 모듈을 분해하는 것이기 때문에 기존의 기능중심의 분해에 의한 설계방법보다는 데이터 중심의 시스템 분석방법인 객체 지향적 설계방법이 소프트웨어의 생산성 향상과 유지 보수를 위한 효율적인 방법이라고 할 수 있다.³³⁾ 이는 데이터와 데이터를 조작하는 연산을 하나로 묶는 캡슐화(encapsulation)와 모듈이 갖는 기능들을 명세한 인터페이스를 통해서만 접근이 되는 정보은닉의 개념이 있기 때문이다.³⁴⁾ 또한 상속(inheritance)과 정보은닉(information hidden)이란 특성을 통하여 소프트웨어 시스

32) J. Rumbaugh et al, op. cit., pp. 15-20.

33) B. Meyer, "Object-Oriented Software Construction", Prentice-Hall, Inc. 1988, pp.114.

34) 캡슐화(encapsulation)와 정보은닉(information hidden)의 개념을 적용함으로써 얻을 수 있는 가장 큰 장점은 크게 유지보수 용이성(maintainability)과 확장성(extensibility)이다. 즉, 소프트웨어가 사용환경의 변경 등으로 인해 생기는 변경 요구를 어느 정도 용이하게 받아들이는가, 또는 운용 중에 발견된 소프트웨어 결함의 수정이 어떻게 간단히 이루어지는가 등의 성질이라 할 수 있다. 정보은닉과 캡슐화는 각각의 모듈간의 인터페이스를 적게하고, 한 모듈은 다른 모듈의 시행에 무관하게 사용함으로써, 다른 모듈의 실행의 변화에 거의 영향을 받지 않게 한다. 따라서 유지보수가 쉽고, 새로운 프로그램의 확장이 용이하다.

템의 장기적인 제반 사항에 대한 많은 변화 요인에 용이하게 적응할 수 있는 능력을 제공함으로써 소프트웨어의 재사용성과 확장성 및 유지보수를 효율적으로 성취할 수 있다는 강점이 있다.³⁵⁾ 이러한 특성들과 장점들이 설계 단계에서 올바르게 시스템에 적용되고 구성되기 위해서는 기본적으로 요구사항의 출현에서부터 객체 지향적 메커니즘이 고려 되어야 한다. 결국 요구사항에 대해서 객체 지향적 분석(OOA : Object Oriented Analysis)이 수반된 분석 결과를 통하여 설계가 수립이 될 때 가장 이상적인 객체 지향적 개발의 과정이라고 할 수 있다.³⁶⁾

4. 이용자 시스템 객체 지향 모델링 방법론

도서관의 전산시스템 구축과 도서관에 소장되어 있는 문헌정보의 디지털화는 최근 논의되고 있는 전자도서관 실현의 절대 조치이다.³⁷⁾

그러나 도서관에 소장된 문헌정보가 데이터베이스로 구축이 되고 전산시스템이 잘 갖추어져 있다 하더라도 이용자의 다양한 요구를 충족시키지 못하면, 도서관의 본질적 기능인 이용자 중심 서비스를 만족하게 실현시키기가 어렵다.

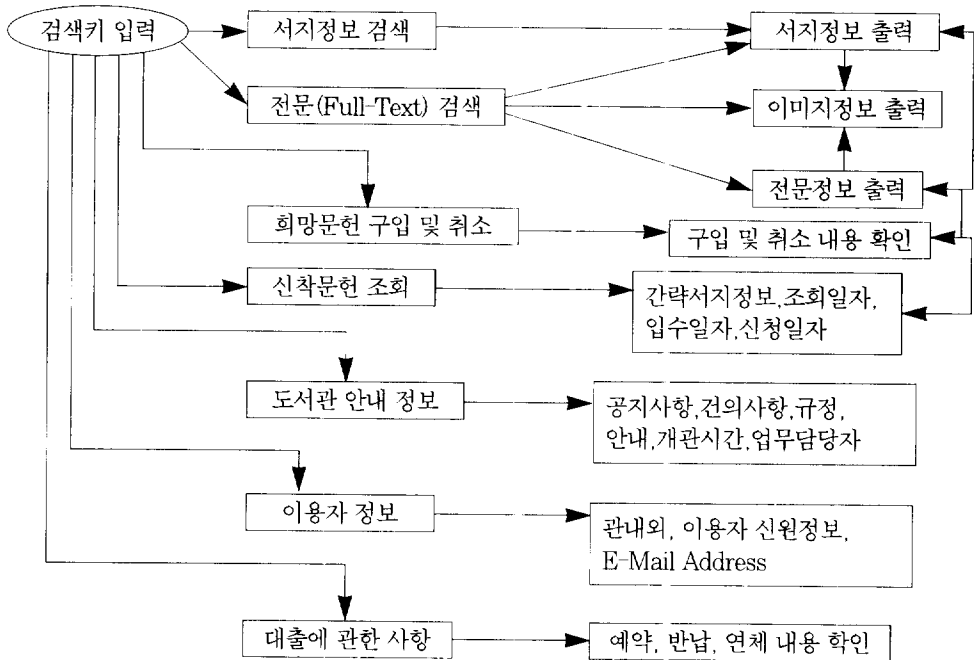
이용자 시스템은 도서관의 이용자 중심 서비스를 실천하는데 중요한 과제이다. 전자도서관의 이용자 시스템은 문헌정보 검색에 관련된 정보의 신속한 수집과 데이터베이스의 적정한 분석 처리뿐만 아니라, 도서관의 상호협력 체제를 유기적으로 실현하여 이용자의 만족도에 결정적인 영향을 주는 시스템이다.³⁸⁾ 본 시스템은 객체 지향 데이터베이스 구축이 도서관에도 적용되어야 한다는 논의가 최근 활발하게 이루어지고 있는 것에 대비하여 객체 지향 데이터베이스 전 과정에 필수적으로 요구되는 객체 지향 모델링을 함으로서 개발자와 사용자간의 의사소통의 원활을 기하고자 하는데 있다. 기존에 제안된 OMT를 적용하되, 문헌정보시스템과 도서관의 특징상 개선된 방법 즉, 정형화된 객체 지향 모델링 방법을 이용자 시스템을 모델로 하여 설계하였다. OMT에서 제안하고 있는 문제기술은 이용자 시스템을 분석하는 분석팀(사서, 관리자)이 이해하기 쉽도록 기능 구성도로 개선하였으며, 기능 구성도는 총괄 분석도와 서브시스템 분석도로 나누어 작성하였다. 이에 따라 객체 지향 모듈 분해 기준을 정형화하고, 객체 지향 설계 절차와 명세화를 위한 객체모델링을 제시하였다.

35) B. Meyer, "Reusability : the case for object-oriented design", IEEE Software, Vol. 4, No. 2, 1987, pp. 50-64.

36) P. Coad, E. Yourdon, op. cit., pp. 13-15.

37) 유양근, "대학도서관의 학술정보 전산망 구성 방안에 관한 연구", 강남대학교, 논문집 26집, 1995, PP. 504-511.

38) 유양근, "대학의 정보화를 위한 전자도서관 구성 방안", 디지털 도서관, 1996(여름호), pp. 21-30.



〈그림 5〉 전자도서관의 이용자 시스템 검색 과정

4.1 이용자 시스템의 개념

전자도서관의 이용자 시스템은 〈그림 5〉와 같이 문헌정보뿐만 아니라, 도서관 이용자 요구에 관련된 정보의 신속한 수집과 데이터베이스의 적절한 분석처리와 도서관의 상호 협력 체제를 유기적으로 실현하여 이용자의 만족도에 결정적인 영향을 주는 시스템이다.

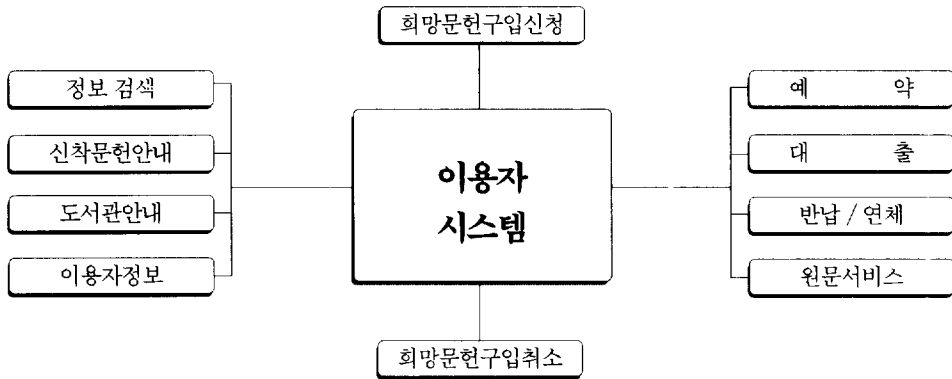
이용자가 원하는 방법으로 문헌정보를 검색하고, 도서관 이용자나 도서관 사서들이 대출에 관한 사항, 예약에 관한 사항, 또는 연체에 관한 사항 등의 정보를 검색하여 확인할 수 있고, 구입을 요구하는 문헌정보를 신청하거나, 도서관 상호협동에 의한 타도서관 데이터베이스 정보 이용, 원문서비스(전문 검색) 및 신청

등을 온라인 네트워크를 통하여 처리할 수 있는 사용자 인터페이스를 지원해주는 시스템을 의미한다. 이용자 요구분석은 이러한 문제를 해결하기 위하여 좀더 발전하고 정형화 시킬 수 있도록 분석되어야 하며 개발자가 이해하도록 모델링되어야 한다.

4.2 이용자 시스템 분석

(1) 이용자 시스템 구성

이용자 시스템은 도서관의 문헌정보 데이터베이스 시스템에서 이용자가 원하는 다양한 방법으로 문헌정보를 검색하고, 도서관 이용자나 도서관 사서들이 대출에 관한 사항, 예약



〈그림 6〉 이용자 시스템의 구성

에 관한 사항, 또는 연체에 관한 사항 등의 정보를 검색할 수 있고, 구입을 요구하는 문헌정보를 신청하거나, 도서관 상호협동에 의한 타 도서관 데이터베이스 정보 이용, 원문복사 및 신청 등을 온라인 네트워크를 통하여 처리할 수 있는 사용자 인터페이스를 지원해 주는 시스템을 의미한다.

이용자 시스템은 〈그림 6〉과 같이 10개의 하위시스템으로 구성된다.³⁹⁾

(2) 이용자 시스템 주요기능

이용자 시스템의 각 10개의 서브시스템의 기능을 살펴보면 〈표 2〉과 같으며 이를 다시 분류하면 정보검색, 대출, 원문제공, 희망문헌 시스템, 신착문헌 안내, 도서관 안내 및 이용자 정보의 6가지로 구분할 수 있다. 구체적인

기능은 다음과 같다.

1) 정보검색 기능

정보검색시스템은 데이터베이스에 구축된 여러 가지 형태의 정보자료를 검색대상으로 하여 이용자가 다양한 검색키⁴⁰⁾를 사용하여 원하는 자료를 검색하는 기능을 가진 시스템이다.

2) 대출 기능

이용자가 대출, 연체, 예약상황을 조회할 수 기능을 가진 시스템으로서 이용자번호와 비밀번호를 입력하면 조회를 시작할 수 있어야 하며 상황을 디스플레이할 수 있어야 한다.

3) 원문서비스 기능

원문서비스는 국내외 학위논문의 이미지정보를⁴¹⁾ 포함하여 전문(Fulltext)⁴²⁾, 목차, 초록

39) 시스템 분석은 시스템 운영자나 이용자의 의견을 충분히 고려하여야 한다. 분석 내용은 기능을 정확히 파악하는데 절대적 요소이다. 명세서의 양식은 간략하고 작성하는 사람이 이해할 수 있도록 만들어져야 한다. 양식의 복잡성, 지나친 전문성에 의해 오히려 내용이 허술할 수 있기 때문에 기재할 내용이 무엇인지를 정확히 알 수 있도록 기재방법 설명을 첨부하면 더욱 좋다.

40) 검색키는 서명,저자명, 주제명, 출판사명, 분류기호, ISBN, ISSN, CODEN, I.CCN, 보고서 번호 등을 말한다.

〈표 2〉 시스템별 기능

이용자 시스템	총괄 기능	구 성
정보 검색	이용자가 도서관의 소장자료를 다양한 검색기를 사용하여 원하는 자료를 검색하는 기능	검색대상자료, 검색키종류, 검색방식, 검색기법, 출력표시, 저장/정렬방법, 인쇄
예약	대출을 원하는 자료가 대출중이거나 예약중인 자료를 예약하거나, 이용자가 예약한 내용을 취소하는 기능	예약신청(예약입력항목, 예약만료일 입력) 예약취소(예약취소입력항목, 출력항목)
대출	이용자ID, 비밀번호를 입력하면 대출, 연체, 예약 현황을 볼수 있는 기능	서지사항, 대출반납일, 대출 총권수, 가능권수
반납/연체	이용자가 반납일자 및 연체현황을 볼 수 있는 기능	신청번호, 신청일자, 신청자사항, 서지정보, 자동복본조회
신착 문헌 안내	이용자가 신청한 문헌이나 도서관에서 구입한 신착문헌을 볼 수 있는기능	서지사항, 대출반납일, 연체사항정보
희망문헌구입신청	이용자가 필요한 자료의 구입을 희망하는 경우에 온라인으로 신청하는 기능	간략서지정보, 조회일자, 입수일자, 신청일자
희망문헌구입취소	신청한 자료를 이용자가 취소할 수 있는 기능	취소입력정보, 출력정보 취소일자
원문 서비스	소장자료는 물론 타기관 소장원문자료 제공기능	원문복사신청 및 취소,처리현황조회
도서관 안내	도서관 이용안내, 도서관정보, 공지사항 정보제공기능	공지사항, 건의사항, 업무 및 담당자, E-Mail Address, 규정, 안내, 개관시간
이용자 정보	도서관이용자를 등록하여 이용자가 변경사항 등을 수정할 수 있는 기능	이용자등록, 변경사항

및 색인을⁴³⁾ 이용할 수 있도록 하며 국가별, 연도별, 학위종별, 분야별, 수여대학별 정보 등은 물론 다양한 통계 현황도 제공하도록 한다. 또한 내부 이용자가 원문복사를 신청할 수 있는 기능과 외부 이용자에 대한 자관 소장자료

의 원문제공서비스를 할 수 있도록 한다.

4) 구입 희망문헌 신청 및 취소 기능

온라인으로 자료를 신규 구입하도록 신청하는 기능, 신청한 자료를 취소하는 기능, 구입 신청한 문헌에 대한 처리 현황을 이용자에게

41) 이미지 정보 관리 모듈 : 이미지정보 관리 모듈은 이미지를 스캔하여 저장하며, 저장된 이미지를 검색하고, 이미지의 구조를 편집하고, 초록 및 목차를 저장하여 이미지와 연동하여 검색할 수 있도록 구현한다. 이미지 저장시 이미지의 특성에 따라 해당 이미지의 서지정보, 목차정보 혹은 초록을 참조하지 않아도 이미지의 내용을 쉽게 파악할 수 있도록 이미지 설명문을 이미지와 함께 저장할 수 있도록 한다.

42) Fulltext 관리 모듈 : 논문의 전문을 문자인식도구(OCR)등을 이용 DB에 저장하고 검색하며 구조를 색인화한다.

43) 초록, 색인, 목차는 이미지 관리 모듈과 동일한 방법으로 구현한다.

〈표 3〉 객체 및 클래스 분석 테이블

main class No		sub class No																							
기호	클래스	1				2				3				4				5							
A	이용자	내부(A-1)				외부(A-2)				(A-3)				(A-4)											
B	자료	단행본				정간물				비도서자료															
		a	b	c	d					a	b	c	d												
C	예약																								
D	도서신청																								
E	복사	원문				초록				목차															
		a	b	c	d																				
F	외부기관																								
G	원문복사	단행본				정간물																			
H	목록																								
I	초록																								
J	도서관안내																								

제공하는 기능을 말한다.

5) 신착문헌 안내 기능

신착문헌 안내 기능은 이용자들이 신청한 문헌정보나 도서관에 신규 입수된 자료를 이용자에게 제공하는 기능으로서 목록, 목차, 색인 및 초록 등을 조회할 수 있어야 한다.

6) 도서관 안내 및 이용자 정보 제공

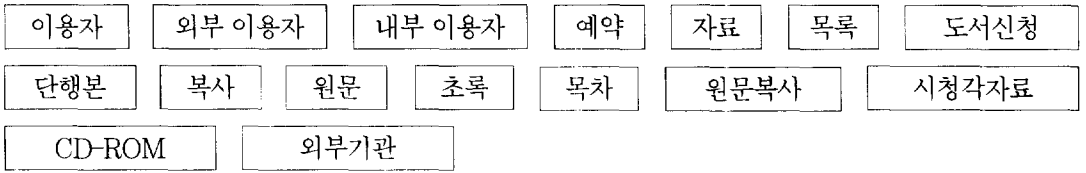
도서관 이용 안내 정보와 이용자 정보⁴⁴⁾ 및 도서관에서 이용자에게 공지하고자 하는 정보

등을 이용자에게 제공하는 기능이다.

4.3 객체 모델링

객체모델링(object modeling)은 전체 모델링 과정의 기초가 되며 객체 지향 데이터 모델 설계 및 프로그래밍의 안정된 바탕을 제공할 수 있도록 한다. 이 단계에서는 객체 및 클래스 정의와 이들간의 관계, 그리고 클래스의 인

44) 이용자 정보 : 이용자 정보는 내부 이용자 및 외부 이용자 정보로서 도서관을 이용하는 사람들의 인력 정보를 말한다. 인력 정보는 이용자 신상정보로서 전공 분야, 소속, 학과, 직위, 주소, 학번, 주민등록 번호 등이 구축되어야 한다.



스탄스를 나타내는 속성과 연산을 나타내며, 또한 클래스를 계층적으로 상속 관계를 정의하여 시스템의 정적인 표현을 한다.

상향식(bottom-up)방법에 의해서 다음과 같은 단계에 따라 객체 모델링을 구축하였다.

제 1 단계 : 요구명세서나 시스템별 구성도에서 식별성을 가진 객체를 추출한다.

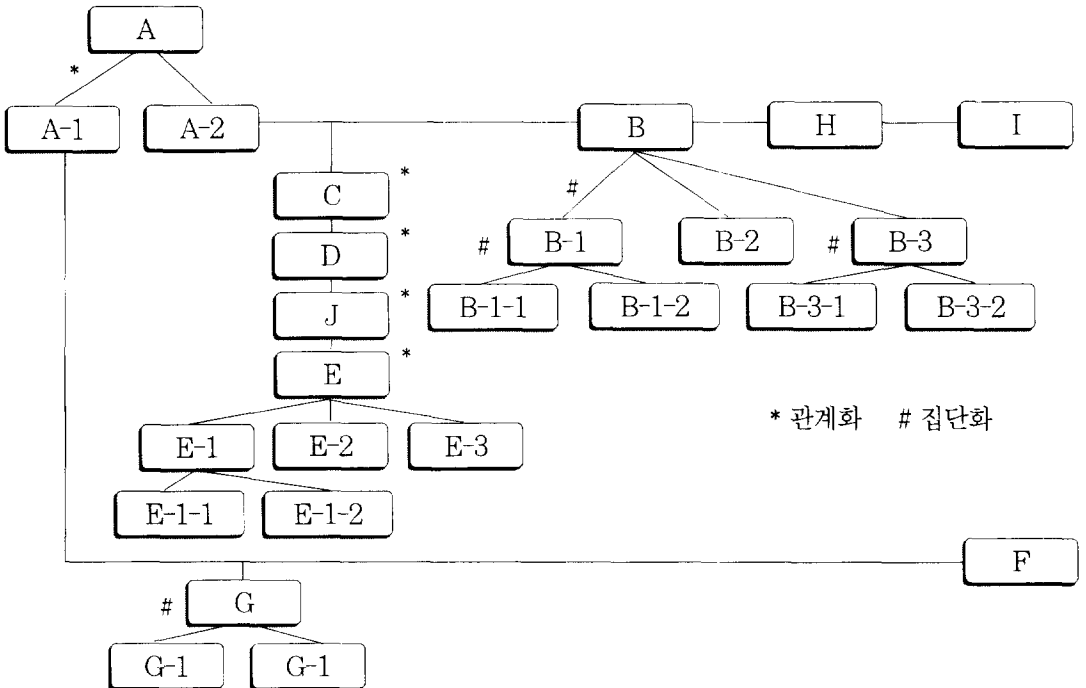
제 2 단계 : 객체를 기호화하여 공동 의미, 행동을 가진 객체를 그룹화하여

클래스를 정한다.

제 3 단계 : 클래스를 그룹하기 위해 분석 테이블을 작성한다.<표 3 참조>

제 4 단계 : 분석테이블을 이용하여 개략적으로 모델링한다.<그림 7 참조>

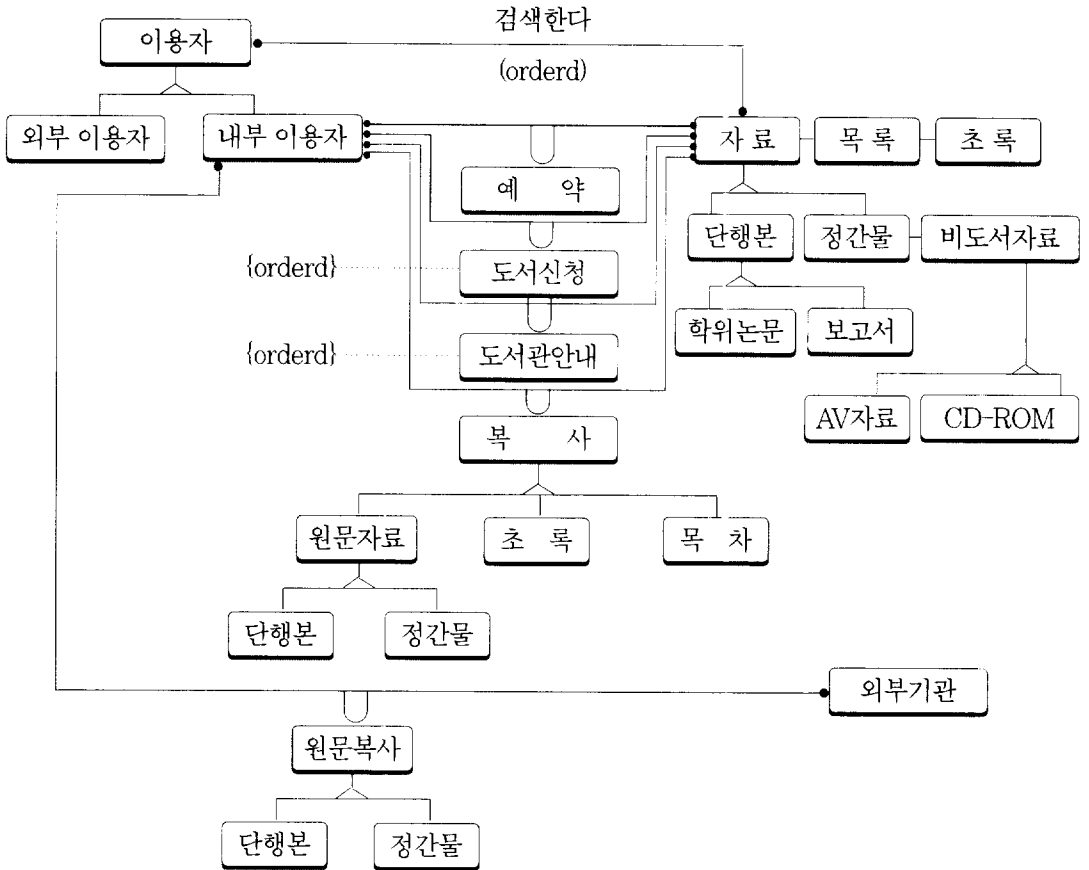
제 5 단계 : 개략적 모델링을 분석하여 통합된 객체모델을 작성하고 속성과 연산을 나타내는 표를 작성한다.<그림 8 참조>



* 관계화 # 집단화

<그림 7> 개략 객체 모델링

(1) 이용자 시스템 객체 모델링



〈그림 8〉 이용자 시스템 통합 객체 모델링

이용자	예 약	서지정보	초 록
ID, 직분 성명	예약 일시, 대출 예정일 예약 민료일, 예약 수위	서지 정보(서명, 저자명, 출판사항 등)	서지 정보, 형태 코드, 신청 일 자, 초록 작성자
	예약 현황 리스트 예약 통보 및 취소	문헌 정보 리스트 (주제별, 연도별 등)	잡지별 현황 신청 리스트
도서신청		목 차	원문복사
신청 번호, 신청 일자, 가격 정보, 용도, 신청 권수, 서적상명, 희망 기간, 출처, 복본 구입 유무, 구독 기간		잡지명, 제목, 출판년	신청 일자, 시작과 끝, 페이지, 부수, 처리 방법, 자료 유형
구입 희망 도서 신청 취소 구입 희망 도서 처리 현황 조회			입력/수정/조회, 신청 리스트 신청 처리 현황
외부기관	단행본	정간물	A/V 자료
조회기관 및 조회 도서관명	등록번호, 서명, 저자 명, 출판사항, 출판 년, 청구기호	자료명, 권호, 출 판년, 기사명, 저 자명, 청구기호	자료명, 제작자, 요약, 청구기호
협력기관 및 협력도서관 현황	형태코드	형태코드	형태코드
CD-ROM			
자료명, 제작자, 제작처, 요약, 청구기호			
형태코드			

〈그림 9〉 클래스의 속성과 연산

(2) 이용자 시스템 데이터 사전
데이터 사전(data dictionary)은 요구정의

분석단계에서 저의된 기능 구성도로부터 식별
된 클래스나 데이터 구조에 대한 정의 명세서

〈표 4〉 데이터 사전

CLASS	Description
예 약	이용자예약기록 속성 : 예약일자, 예약일시, 이용자번호, 예약만료일, 예약순위 연산 : 입력/조회/출력/삭제, 예약통보, 예약현황리스트
도서신청	일반이용자, 학과단위, 부서단위 구입을 희망하는 도서를 온라인으로 신청한다 속성 : 자료명, 저자명, 출판사, 출판년, 판차, 추정가격, 화폐단위
도서관 안내	이용자가 도서관 안내를 온라인으로 신청한다 속성 : 신청번호, 신청자, 신청일자, 처리방법, 안내내용(공지사항, 대출규정, 개관시간, 담당자 성명, 협력기관, 협력도서관 등) 연산 : 입력/조회/출력/삭제
원문복사	이용자가 자료의 원문복사를 온라인으로 신청, 취소, 처리상황을 조회한다. 속성 : 신청번호, 신청자, 신청일자, 처리방법, 자료구분(연간물, 단행본), 시작과 끝페이지 연산 : 입력/조회/출력/삭제
단행본원문복사	원문복사의 서브 클래스로 단행본의 원문복사를 온라인으로 신청, 취소, 처리상황을 조회한다. 속성 : 입력/조회/출력/삭제, 잡지명, 출판사, 출판년, 권호 등
정간물원문	원문복사의 서브클래스로 연속간행물의 원문복사를 온라인으로 신청, 취소, 처리상황을 조회 복사한다. 속성 : 입력/조회/삭제, 잡지명, 출판사, 출판년, 권호 등
초 록	단행본 정간물 각종 보고서 학위논문의 초록을 온라인으로 신청, 취소, 처리현황을 조회한다 속성 : 입력/조회/삭제, 잡지명, 출판사, 출판년 등
목 차	기사의 내용을 알기위하여 목차를 온라인으로 신청, 취소, 처리현황을 조회한다. 속성 : 입력/조회/삭제, 잡지명, 기사명, 출판년, 출판사항 등
시청각자료	시청각 자료의 목록을 온라인으로 신청, 취소, 처리현황을 조회한다. 속성 : 입력/조회/삭제 자료명, 자료 주제명, 자료형태, 제작처 등
CD-ROM	소장된 CD-ROM 자료의 목록 조회 및 온라인으로 CD-ROM 내용 검색 속성 : 자료명, 주제명, 자료형태 등
외부기관	협력기관 및 협력도서관 현황 조회 속성 : 기관명, 기관코드, 도서관명, 도서관코드

이다.

이러한 데이터 사전은 정보를 저장하여 데이터베이스를 설계 운용하고, 확장하는 과정에서 데이터에 관한 정보를 제어하고 관리하기 위해 필요하다. 또한 데이터 사전은 데이터에 관한 문서로서 데이터에 관한 기록을 유지하여 사용자 및 개발자들이 원하는 항목을 선택할 수 있도록 하여, 데이터를 집중화하고 데이터를 제어하고 관리할 수 있는 실질적인 데이터 사전이 되어야 한다.

데이터 사전은 다음과 같은 내용이 포함된

[클래스] = { 문제에 대한 정의 및 관계성, 속성, 연산기술 }

형식을 갖추어야 한다.

데이터 사전 항목의 영역에는 이외에, 제약 조건, 연관 관계 등이 포함될 수 있다. <표 4>의 데이터 사전에서는 도서관 이용자 시스템의 클래스에 대한 문제 정의 및 관계, 속성, 연산 등을 기술한다.

5. 결론 및 제언

국내에서 전자도서관을 터키시스템으로 추진하면서 도서관 요구사항이 제대로 분석되지 않아 개발후에 많은 문제점이 나타나는 것을 종종 볼 수 있다.

전자도서관 구축은 용역회사가 제안한 제안서에만 의존할 것이 아니라 도서관에서 개발

에 필요한 요구사항을 체계적으로 제시하여 상호 협동관계로 개발하여야 한다. 또한 이용자 중심의 전자도서관을 구축한다면서 이용자가 무엇을 원하고 있는지를 정확하게 파악지 못하고 사서의 사무적 지원시스템 중심으로 개발한다면 더욱 문제가 된다고 본다. 객체 지향 시스템 개발에 필요한 객체 지향 모델링은 기존의 구조적시스템분석 설계에 비하여 개발기간이 단축되고, 이식성이 뛰어나기 때문에 최근 들어 확산 보급되고 있는 개발 방법이다.

본 논문에서 밝혀진 객체 지향 모델링은 다음과 같은 특징으로 나타나 있다.

첫째, 요구정의 방법 변경에 유연하게 대응한다.

둘째, 그룹개발(분산개발)을 원활 신속하게 한다.

셋째, 지적 지원을 가능케 한다.

객체 지향 방법론의 큰 장점은 분석 설계 및 구현 단계들 사이에 의미적 차이가 거의 없다는 것이며, 분석 단계의 클래스 단위 작업은 각 단계간의 의미적 차이를 줄이는 데 효과적이다. 그러나 도서관 이용자 시스템의 문제 영역을 추상화를 통하여 모델링하고, 속성별로 객체들을 묶어서 하나의 클래스를 만들어 내는 과정에서 객체 지향 개념과 특성을 적용시킨다는 것은 간단치 않은 문제이다

본 논문에서 제안된 이용자 시스템 모델링은 다음과 같은 장점이 있다.

첫째, 문헌정보시스템과 도서관의 특성에 맞춰 작성된 기능 구성도는 이해하기가 쉽다.

둘째, 기능구성도를 기초로한 객체 및 클래스 추출은 서브시스템별로 상세하게 작성하였

기 때문에 정확성이 높다.

여러 가지 객체 지향 분석 방법과 구조적 분석은 제각기 일장 일단이 있고, 개발 대상에 따라 적용기법의 효율이 다르게 나타날 것으

로 보아 업무 영역의 특성에 따라 선택하여 사용할 필요가 있다고 본다. 경우에 따라서는 서로의 결점을 보충하는 상호 보충적인 이용 방법도 생각할 수 있다.

참고문헌

- 유양근, 류해영, "객체 지향 데이터베이스 시스템의 필요요건과 설계에 관한 연구", 한국정보처리학회, 제4권, 제2호, 1997.
- 유양근, "대학도서관의 학술정보 전산망 구성 방안에 관한 연구", 강남대학교, 논문집 26집, 1995, pp.504-511.
- , "대학의 정보화를 위한 전자도서관 구성 방안", 디지털 도서관, 1996(여름호), pp.21-30.
- C. Batini, S. Ceri, S. B. Navathe, "Conceptual Database Design", Benjamin/Cumming Publishing, 1992.
- C. Batini, S. Ceri, S. Navathe, "Conceptual Database Design : An Entity-Relationship Approach", The Benjamin/Cumming Publishing, 1992, pp.15-54.
- G. Booch., "Object-Oriented Development", IEEE Transactions on Software Engineering Vol. SE-12, No.2, 1986, pp.211-221.
- G. Booch, "Object-Oriented Design", Redwood City, Calif, Benjamin/Cum-mings, 1991.
- J. C. Brad, "Object-Oriented Programming", Rrading Massachusetts : Addison-Wesley, 1986, pp.12-18.
- P. Coad, E. Yourdon, "Object-Oriented Analysis", Prentice-Hall, 1990, pp.1-7.
- P. Chen, "The Entity-Relationship Model : Toward a Unified View of Data", ACM Trans. Database System, Vol. 1, No.1, 1976, pp.9-36.
- R. Elmasri, B. Navathe, "Fundamentals of Database System(2nd)", Benjamin/Cummings Publishing, 1994, pp.27-30.
- R. Elmasri, B. Navathe, " Object Integration in Logical Database Design", Proc. of the First International Conf. on Data and Knowledge Engineering 1, No. 1, May 1985.
- S. E. Hudson, R. King, "A Self-Adaptive Concurrent Implementation of an Object-Oriented Database Management System", ACM Transactions on Database System, Vol. 14, No. 3, 1989, pp.291-321.

- I. Jacobson, M. Christerson, P. Jonsson, G. Overgaard, "Object-Oriented Software Engineering : A Use Case Driven Approach", ACM, Inc. 1992, pp.43-62.
- T. F. Kraemer, "Product Development using object-oriented software technology", Hewlett-Packard Journal, 1989.
- B. Meng, "Object-Oriented Programing", MacWorld, Vol. 7, No. 1, January, 1990. pp.174-183.
- B. Meyer, "Object-Oriented Software Construction", Hertfordshire, England, Prentice Hall Inter, 1988.
- B. Maher, D. H. Sleeman, "Automatic Program Improvement : Variable Usage Transformations," ACM-TOPS, Vol .5, No. 2, 1983.
- J. Martin, "Information Engineering, Book I Instruction", Prentice-Hall, Inc. 1989, pp.57-58.
- J. Martin, "Information Engineering, Book III Desige and Construction", Prentice-Hall, Inc. 1989, pp.20-34.
- J. Martin, J. Odell, "Object-Oriented Analysis & Design", Prentice-Hall, 1992, pp.29-42.
- B. Meyer, "Object-oriented Software construction", Prentice-Hall Inc., 1988.
- B. Meyer, "Object-Oriented Software Construction", Prentice-Hall, 1988, pp.114.
- B. Meyer, "Lessons from the design of the Eiffel Libraries," CACM, Vol. 33, No. 9, 1990.
- J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen, "Object-Oriented Modeling and Design", Prentice-Hall International, 1991.
- J. M. Smith, C. P. Smith, "Database Abstractions : Aggregation and generalization", ACM TODS, Vol. 2, 1977, pp.105-133.
- J. Toby, T. Yang, D. James, P. Fry, "A Logical Design Methodology for Relational Database Using the Extend Entity-Relationship Model", Comuter Survey, Vol. 18, No. 2, 1986. pp.197-222.
- E. Yourdon, "Structured Design Workshop", Third Edition, Yourdon press, 1982, pp.25-30.
- B. Stroustrup, "What is Object-Oriented Programming?", IEEE Software, Vol. 5, No. 3, 1988.
- D. Tsichritzis, F. Lochovsky, "Data Models", Prentice-Hall, Englewood Cliffs, N. Y. 1982.
- W. E. David, D. K. Barry, S. N. Woodfield, "Object-Oriented System Analysis : A Model-Driven Approach", Yourdon Press, Prentice Hall, 1992.
- B. Stroustrup, "What is Object-Oriented Programming?", IEEE Software, Vol.

- 5, No.3, 1988. pp.10-20.
- Mannila, H., Raiha, K., The Design of Relational Database, Addison-Wesley, 1992.
- Elmasri, R., and Navathe, S. B., "Object Integration in Logical Database Design", Proc. of the First International Conf. on Data and Knowledge Engineering 1, No. 1, May 1985.
- Fleming, C., Halle, B., Handbook of Relational Database Design, Addison-Wesley, 1989. pp.89-95.
- Sheth, A.P., Larson, J.A., Cornelio, A., and Navathe, S.B., "A Tool for Integrating Conceptual Schemas and User Views", Proc. of International Conf. on Data Engineering, 1988, pp.20-72.
- Fleming, C., Halle, B., Handbook of Relational Database Design, Addison-Wesley, 1989, pp.45-70.
- Thalheim, B., "A General Framework for Database Design Strategies", East/West Database Workshop, 1994, pp.263-274.
- Batini, C., Ceri, S., and Navathe, S.B. Conceptual Database Design, Benjamin/Cumming Publishing, 1992, pp.271-303
- Wiederhold, G., File Organization for Database Design, Megraw-Hill, 1987, pp.210-235.
- Smith, J.M., and Smith, C.P., "Database Abstractions : Aggregation and generalization", ACM TODS, vol.2, 1977, pp.105-133.
- D. Tsichritzis and F. Lochovsky, "Data Models", Prentice-HALL, Englewood Cliffs, N. Y. 1982. pp.105.
- Elmasri, R. and Navathe, S. B., Fundamentals of Database Systems, The Benjamin/Cumming Publishing, 1989. pp.120-132.
- Batini, C., Ceri, S. and Navathe, S., Conceptual Database Design : An Entity-Relationship Approach, The Benjamin/Cumming Publishing, 1992. pp.15-54.
- Chen, P., "The Entity-Relationship Model : Toward a Unified View of Data", ACM Trans. Database System, Vol. 1, No.1, 1976. pp.9-36.
- Toby, J. T., Yang, D., James and P. Fry, "A Logical Design Methodology for Relational Database Using the Extended Entity-Relationship Model", Computer Survey, Vol. 18, No. 2, 1986. pp.197-222.