

객체지향기술 '소프트웨어 위기' 를 해결하기 위한 해결책(Ⅲ)

Object-Oriented Technology: A Silver Bullet for Software Crisis

REPORT



임성택
고려대학교
경상대학 경영정보학과 조교수
Rim, seong-taek, ph.d. Assitant Professor,
Dept of Management Information Systems
College of Economics & Commerce,
Korea University

연재순서

1. 서론
2. 객체지향의 기본개념
3. 객체지향 소프트웨어공학
- ▶ 4. 객체지향 데이터베이스
5. 결론

4. 객체지향 데이터베이스

앞에서 우리는 '소프트웨어 위기 (Software Crisis)'의 해결책으로서 객체지향 소프트웨어 공학에 대해서 살펴 보았다. 하지만 소프트웨어의 위기를 해결하기 위해서는 소프트웨어를 잘 만들 수 있는 방법을 개발하는 것만으로는 충분하지 않다. 그것 이외에 데이터를 효과적으로 관리할 수 있는 기술이 필요하다.

지난 20여년동안 조직이 갖고 있는 데이터의 양은 엄청나게 증가 했으며, 데이터의 내용 또한 훨씬 다양하고 복잡해졌다. 과거의 정보시스템은 단순히 텍스트나 숫자만을 저장할 수 있으면 충분하였다. 하지만 오늘날의 정보시스템은 스프레드시트 이미지, 다이어그램,

지도, 오디오, 비디오 등 여러가지 멀티미디어 데이터를 다룰 수 있어야만 한다.

객체지향기술은 이러한 문제를 해결하고, 데이터를 관리하기 위한 아주 새로운 방법을 제시할 수 있는 가능성을 지니고 있다. 사실 객체지향 데이터베이스는 더이상 가능성만이 논의되는 미래의 기술이 아니라 현실이다. 이미 20여개의 업체가 객체지향 데이터베이스 관리시스템(이하 ODBMS로 약칭)을 판매하고 있으며, 주요 관계형 데이터베이스 관리시스템(이하 RDBMS로 약칭)를 제공하는 업체들도 그들의 제품에 이러한 객체지향적인 특성을 삽입하려는 노력을 기울이고 있다.

이번 절에서는 객체지향 데이터베이스에 대해 알아보려 한다. 우선, 객체지향 데이터베이스의 필요성에 대해서 살펴본다. 그다음, ODBMS가 갖추어야할 기본적인 요소들에 대해서 살펴보고 마지막으로, 현재 시중에 나와있는 상업용 ODBMS를 조사해 본다.

4.1 객체지향 데이터베이스의 필요성

객체지향 데이터베이스의 필요성은 다음 두가지로 요약될 수 있다. 하나는 객체지향 프로그램의 실행에 의해 생성된 객체를 어떻게 저장하고 꺼낼 수 있느냐 하는 객체의 영속적인 저장(Persistent Storage)에 대한 필요성이고, 또다른 하나는 전통적인 데이터베이스 관리시스템(그중에서 특

REPORT



히 RDBMS)에 의해서는 적절하게 다루어지기가 힘든 복잡한 형태의 데이터를 어떻게 관리할 수 있으냐하는 복잡한 데이터구조를 구현할 필요성이다.

1) 객체의 영속적인 저장장소

객체지향 프로그래밍 언어는 일반적으로 주기억장치에 바탕을 두고 있으며, 단일 사용자를 위한 응용시스템에 맞도록 되어 있다. 따라서 대부분의 객체지향 프로그래밍 언어들은 프로그램 실행시에 만들어진 객체들을 프로그램이 끝난 후에 저장할 방법을 갖고 있지 않다. 하지만 많은 응용시스템들은 실행과 실행사이에 (Between Runs) 객체를 저장하고 추후에 다시 꺼내서 쓸 수 있어야 한다. 그렇다면 프로그램이 실행되지 않을 때 이 객체들을 어디에 보관하고 있어야 하는가?

첫번째 해결방법은 파일 속에 저장하는

것이다. 예를 들어 Smalltalk/V는 응용 프로그램에 있는 모든 변수의 값을 포함하여, 현재 상태의 이미지를 저장한다. 그리하여 시스템을 재실행시켰을 때, 이 이미지 화일로부터 각 객체들의 내부상태를 원상복귀할 수 있도록 한다. 하지만 이러한 방법들은 결국 데이터베이스로 귀결되는 전통적인 화일처리방식의 여러가지 문제점을 그대로 지니게 된다.

두번째 해결방법은 전통적인 데이터베이스 저장방식을 사용하는 것이다. 하지만 전통적인 데이터베이스 관리시스템은 텍스트나 숫자와 같이 아주 단순한 형태의 데이터만을 저장할 수 있도록 설계되었기 때문에 객체지향형 프로그래밍 언어가 허용하는 아주 다양한 형태의 데이터를 취급하기가 어려울 뿐만 아니라 객체가 갖고 있는 메소드(행위)를 저장할 방법이 없다. 이러한 문제를 해결하기 위해 복잡한 객체를 전통적인 데이터베이스 관리시스템이 쉽게 취급할 수 있는 단순한 요소로 전환하는 장치(Object Converter)가 필요하다.

세번째 해결방법은 새로운 종류의 데이터베이스인 객체지향 데이터베이스 관리시스템(ODBMS: Object Database Management Systems)를 사용하여 저장하는 것이다. 객체지향 데이터베이스가 어떻게 구성되어 있는 지는 다음 절에서 좀 더 구체적으로 살펴볼겠지만, 객체데이터베이스는 객체를 저장하고 꺼내올 때 분해하고 조립하는 비능률적인 일을 하지않고도 데이터베이스 서비스를 모두 제공할 수 있다는

점에서 객체를 저장하는 가장 좋은 방법이라 할 수 있다. 예를들면 복잡한 객체를 관계형 데이터베이스로 저장하는 것은 차고에 차를 주차하는 대신에 차를 분해해서 차고에 넣는 것과 마찬가지로 할 수 있다.

2) 복잡한 데이터

앞에서는 객체지향 데이터베이스의 필요성을 객체지향 프로그래밍 언어와의 관계에 의해서 살펴보았다. 객체지향 데이터베이스가 필요하게 된 또다른 하나의 이유는 정보시스템의 발달로 인하여 종래의 데이터베이스 관리시스템으로는 관리할 수 없는 복잡한 형태의 데이터가 출현하게 되었다는 것이다. 이들 새로운 데이터를 살펴보면 다음과 같다.

첫째, 설계 데이터를 들 수 있다.

엔지니어링 설계 데이터베이스는 CAD/Manufacturing/소프트웨어 공학 시스템에 아주 유용하다. 그러한 시스템에서는 Complex Objects는 여러 개의 작은 객체로 재귀적으로 분할될 수 있다. 객체지향 데이터베이스는 이러한 복잡한 구조를 표현하는 데 아주 적절하다. 또한 객체지향 데이터베이스는 구조자체가 객체들간에 직접 Reference로 저장되어 있기 때문에 Retrieval시간이 훨씬 짧다.

둘째, 멀티미디어 데이터를 들 수 있다.

오늘날의 정보시스템에서 데이터는 단순히 텍스트나 숫자만이 아니라 이미지, 그래픽, 오디오, 비디오 등을 포함한다. 그러한 멀티미디어 데이터는 가변

REPORT

길이를 가진 일련의 바이트로 저장되며, 데이터의 세그먼트들이 서로 쉽게 찾기 위해 연결되어 있다. 가변길이의 자료구조는 관계형 데이터베이스와는 잘 맞지 않는다. 객체지향 데이터베이스는 새로운 데이터 타입에 쉽게 조화를 이룰 수 있기 때문에 멀티미디어 데이터의 저장에 아주 적절하다.

세제, 인공지능과 전문가 시스템에 쓰이는 데이터를 들 수 있다.

인공지능과 전문가 시스템은 하나의 지식베이스를 형성하는 사실(Facts)와 규칙(Rules)로 정보를 나타내고 있다. 이러한 응용시스템에서는 관계형 모델의 단순한 구조를 넘어서는 아주 복잡한 구조를 요구하고 있다.

4.2 객체지향 데이터베이스 시스템의 기본요소

하나의 데이터모델은 실세계 객체들, 그들위에 부과되어 있는 여러가지 제약들, 그들간의 관계에 대한 논리적인 조직이다.

데이터베이스 언어는 하나의 데이터모델을 위한 구체적인 문법이며, 데이터베이스 시스템은 데이터 모델을 구현시킨다.

RDBMS가 관계형 데이터모델을 구현시키는

데이터베이스 시스템인 것처럼 ODBMS는 객체지향 데이터모델을 직접 지원할 수 있는 데이터베이스 시스템이다. 하나의 표준화된 객체지향 데이터모델이 없기 때문에 무엇이 ODBMS의 기본요소인지 한마디로 잘라 말하기는 힘들다.

따라서 여기서는 어떤 데이터베이스 시스템과 마찬가지로 ODBMS가 반드시 갖추어야할 기본적인 요소만을 살펴보기로 하자.

무엇보다도 ODBMS는 객체와 그것의 스키마를 위한 영속적인 저장소를 제공하여야 한다. 또한 스키마 정의와 수정, 객체의 생성과 접근 등을 위한 사용자 인터페이스를 제공하여야만 한다.

이러한 기본적인 기능 이외에도 질의언어가 정의되어 있어야 하며, 트랜잭션 관리와 트리거링과 같은 데이터의 무결성(Integrity)을 유지시켜줄 수 있어야한다.

4.3 여러가지 상용 객체지향 데이터베이스 시스템

ODBMS가 맨처음 상용화된 것은 1987

〈표 1〉 상용용 객체지향 데이터베이스 관리시스템

제 품	제 공 업 체	주요 응용분야
Gemstone	Servio Corporations	사무자동화, CAD/CAM, CASE, 제조업 등
ITASCA	Itasca Systems	지리정보시스템, CAD/CAM
Level 5 Object	Information Builders, Inc.	전문가 시스템
Open ODB	Hewlett-Packard	제조업, CAD/CAM
Object/ DB	Digital Equipment Corp.	Case, 제조업, CAD/CAM
Object Store	Object Design, Inc	Imaging, Case 전문가시스템
Objectivity/DB	Objectivity, Inc	CAD/CAM, 멀티미디어
Stalice	Symbolics	사무자동화, 하이퍼팩트 등
Versand Object-Base	Versand Object Technology	사무자동화, 전문가시스템, CAD/CAM
Zeitgeist	Texas Instrumens	CAD/CAM Manufacturing, CASE

년 Servio회사의 Gemstone 과 Ontos 社の Vbase(Ontos의 전신)이다. 현재 까지 약 20여개의 ODBMS가 시중에 나와있으며, 여러개의 ODBMS 관련 연구프로젝트가 진행중이다. <표 1>은 시중에 나와있는 ODBMS와 그것들이 지원하는 응용분야를 나열하고 있다. ODBMS를 판매하고 있는 업자의 대부분이 주요 컴퓨터 업체임을 알 수 있다.

5. 결론

객체지향기술은 이미 소프트웨어를

개발하는 방법을 변화시키고 있으며, 머지않아 소프트웨어의 성격 그 자체를 변하게 할 것임에 틀림없다. 하지만 이러한 변화는 관리자들의 이해와 지원없이 이루어질 수 없다. 따라서 이 글에서는 객체지향기술에 대한 관리자들의 이해를 돕기 위해 이를 기본 개념, 객체지향 소프트웨어공학, 객체지향 데이터베이스 세부분으로 나누어 살펴보았다. 여기서는 주로 기본적인 개념의 소개에 초점을 맞추었기 때문에 세세한 내용에 대한 설명은 가급적 삼가하였다. **DIC**

REPORT

Abstract

The Purpose of this Section is to Provide an Understanding for Object-Oriented Databases. First, We will Describe the Needs for Object-Oriented Databases in Terms of 1) the Persistent Storage in an Object-Oriented Programming Language and 2) Complex Data. Then, We will Define What Object-Oriented Database Management Systems Should be. Finally, We will Survey Current Object-Oriented Database Management Systems in the Marketplace.