

SW 생산성·품질향상 위한 최상의 개발기법

오브젝소프트·데이텍 등 전문회사 활발한 활동 보여

김수동/ 숭실대학교 컴퓨터학부 교수

90년대에 들어서 기업내 소프트웨어(SW) 개발 및 구매 비용이 급증하고 있다. 이는 사용자의 SW에 대한 요구사항이 점점 복잡해지고 네트워킹, 그래픽 사용자 인터페이스(GUI), 데이터베이스(DB), 클라이언트 서버 등의 기술 채택이 보편화되면서 목표한 SW의 개발 난이도와 개발기간이 크게 늘어나기 때문이다.

따라서 많은 기업들은 SW비용을 절감하기 위한 효과적인 방법과 기술을 모색해왔다. 객체지향 프로그래밍(OOP : Object Oriented Programming) 기술은 이처럼 복잡한 SW의 개발기간과 비용을 크게 단축시키고 품질을 크게 향상시키는 혁신적인 SW기술로 정착되고 있다. 현재로서는 객체기술이 SW개발 생산성과 품질향상을 위한 가장 효과적인 개발기법으로 알려져 있다.

객체관리그룹(OMG)은 객체지향과 관련된 기술에 대해 표준화를 하기 위하여 구성된 국제적인 표준화 기구로서 현재 800개 이상의 컴퓨터 관련 회원사를 두고 있다. OMG는 객체모델(Object Model)이란 OOP개념의 표준을 정의하고 있으며 현재 이 표준이 가장 널리 수용되고 있다.

객체지향 프로그램의 핵심개념은 객체, 클래스, 상속의 세가지로 구성되며 추상 클래스와 제너릭(Generic) 클래스의 고급개념도 가지고 있다. 객체는 SW 문제영역 즉 목표 도메인내의 사람, 개체, 개념 등 일체의 정보를 모델링하며 클래스는 유사한 객체들의 속성과 메소드(Method)들을 정의하는 OOP의 단위다. 상속은 하위 클래스가 상위 클래스의 특성을 상속받아 마치 하위 클래스에서 정의된 특성처럼 사용하게 해주는 재사용 장치다.

이러한 개념으로 구성된 객체지향 프로그램은 COBOL이나 C

등의 3세대 언어로 작성된 프로그램에 비하여 그 크기가 약 3분의 1 정도로 줄어들어 SW 개발비용과 기간을 크게 단축시킨다. 또한 OOP는 SW 유지보수를 할 때 일부 클래스의 수정과 추가만으로도 가능하므로 프로그램 수정기간과 유지보수비를 크게 줄여준다.

객체기술이 SW개발에 본격적으로 적용되기 시작한 80년대 중반에는 주로 프로그램 코딩방식으로 활용되어 왔으나 현재는 객체기술이 전산의 거의 모든 분야에 매우 폭넓게 적용되고 있다. 객체지향 개발기술은 대학의 전산학과 교과과정, 산업계의 SW 개발 방법론, 시스템 아키텍처, 프로그래밍 언어, DB, 케이스(CASE) 도구 및 기타 여러 분야의 기본적인 모델로 정착되었다.

또한 객체기술은 전산관련 여러 분야의 표준화제정에 기본 모델로 채택되었는데 분산컴퓨팅의 표준으로 코바(CORBA), DCOM, 통신망 서비스 및 관리 분야에서 TMN, TINA, 멀티미디어 표준, SW재사용 컴포넌트 등의 예를 들 수 있다. 그 결과 OOP는 OOP의 의미를 넘어서 객체지향 패러다임의 의미로 사용된다.

고품질의 SW를 개발하기 위해서는 요구사항을 정확히 분석하고 목표 SW를 설계하는 분석, 설계작업이 매우 중요하다. 모든 개발방법론에서 분석모델을 바탕으로 설계가 진행되고 설계모델을 기반으로 구현되지만, 구조적 기법 등 전통적 개발방법론은 분석, 설계, 설계, 구현 단계간의 효과적인 전이방법이 부족했다.

객체지향 SW를 개발하는 경우, 분석과 설계 단계에서 사용되는 다이어그램 및 모델링 장치들이 동일하며 객체지향 분석, 설계(OOA/D)모델은 선정된 객체지향 언어로 체계적으로 번역되어

코드가 생성될 수 있기 때문에 충분한 분석과 정확한 설계가 과제의 성공여부에 더욱 중요한 요소가 된다. 국내에서는 아직 객체지향 설계모델의 중요성 및 유용성이 덜 인식되어 개발기간을 단축시키는 객체기술의 장점을 잘 활용하지 못하고 있는 실정이다.

80년대 후반과 90년대 초반에 방법론 전쟁이라 할 만큼 다수의 분석, 설계 기법이 소개되었는데 그중 럼보(Rumbaugh)의 객체모델링기술(OMT)기법이 산업계에서 가장 널리 사용되어 왔다. 그러나 객체관련 표준화기구인 OMG에서 97년 11월 부처, 럼보, 야콥슨의 방법론들을 통합한 통합모델링언어(UML : Unified Modeling Language)를 표준으로 제정하였고 그후 UML이 산업계의 표준 OOA/D 방법론으로 널리 활용되고 있다.

UML은 사용예(Use Case) 다이어그램, 클래스 다이어그램 등 8개의 다이어그램을 기반으로 객체지향 SW를 개발하기 위한 풍부한 분석 및 설계 장치를 제공하고 있어 향후 상당기간 동안 산업계의 표준으로 활용될 것으로 예상된다.

UML을 가장 잘 적용할 수 있는 SW개발 프로세스로 올 11월에 미국 래셔널(Rational)사는 통합프로세스(Unified Process) 5.0을 발표하였다. 이 프로세스는 웹 애플리케이션 개발에 효율적이고 개발팀의 생산성을 극대화하며 UML의 장점을 최대한 살릴 수 있도록 고안된 실무형 개발 프로세스다. 비즈니스 모델링, 요구사항 관리기법, 컴포넌트 기반 개발기법, 자료공학, 아키텍처 및 형상관리 기법, 테스트를 포함한 몇몇 SW개발 요소부터 일관성 있는 프레임워크까지 진보된 개발기법과 개념을 가지고 있는 새로운 개발 프로세스로 산업계에서의 넓은 수용이 예상된다.

SW개발 공정 및 모델링 과정을 효율적으로 지원하는 CASE도 구 분야에서도 지난 수년간 많은 변화가 있었다. 대다수의 도구가 객체지향 다이어그램과 기법을 지원하기 위한 기능을 추가했으며 특히 지난 2~3년간에 소개된 객체지향 케이스(OCASE)도구는 대부분 UML을 지원하고 있는데 대표적으로 Paradigm Plus, ROSE, SELECT, Stp/UML 등을 들 수 있다.

객체지향 언어는 놀랍게도 이미 67년의 SIMULA 언어에서 시작하여 순수 객체언어로 객체지향 개념을 정립시키는 데 큰 공헌을 한 스몰토크(Smalltalk) 언어, 80년대 중반부터 산업계 SW개발에 본격적으로 활용되어온 C++, 90년대 중반에 소개되어 인터넷기반 객체지향 SW개발에 가장 널리 사용되고 있는 자바(Java)에 이르기까지 그 역사가 전산학의 역사에 버금가는 30년

간에 걸쳐 발전되어 왔다.

C++와 같은 객체지향 언어는 OOP뿐 아니라 절차적 프로그래밍 방식도 지원하는 혼성언어이며 Smalltalk, 자바 등은 OOP만 가능케 하는 순수 객체지향 언어다. 산업계에서는 90년에 접어들면서 순수 객체지향 언어들을 더 선호하는 경향이 국제학술대회 등을 통해 나타나고 있는데 향후에도 이런 추세는 계속될 것으로 예상된다. 이는 단일 패러다임의 프로그래밍이 개발 생산성을 높여주며 혼성 패러다임방식의 프로그램 설계시 중복된 모델로 인한 복잡성을 해결하기 때문이다.

C++ 언어는 C 언어 구문들과의 호환성을 최대한 유지하면서 OOP를 지원하는 장치를 지닌 언어다. 이 언어의 장점으로는 C 언어와의 높은 호환성으로 기존에 개발된 C 프로그램과 연동 및 재사용이 가능하다. 또한 C 언어처럼 운용체계 레벨의 시스템 프로그래밍을 지원하며 실행시 효율성도 Smalltalk, 자바에 비해 매우 높다. 반면에 이 언어는 복잡한 가시성 정의 기능, 레퍼런스 타입, 프렌드 장치, 다양한 데이터 타입 및 사용 규칙에다 C 언어의 복잡성을 그대로 가지고 있어 배우기가 상대적으로 어렵고 많은 언어장치들로 인한 에러발생 가능성의 단점도 가지고 있다.

이러한 단점에도 불구하고 이 언어는 애플리케이션 및 시스템 SW개발에 상당히 오랜 기간 사용될 것으로 예상된다.

객체지향 기술이 무르익으며 C++ 언어의 복잡성이 널리 인식되면서 선(Sun)사가 96년에 발표한 자바 언어는 산업계에 큰 변화를 가져왔다. 90년 초반부터 시작된 인터넷의 급속한 보급에 힘입어 인터넷 기반의 OOP 언어로 빠른 속도로 부상하고 있다.

자바 언어는 웹 애플리케이션 개발에 유용한 애플릿, 소켓 등의 장치를 제공하며 자바 원격메소드호출(RMI)을 사용할 경우 인터넷 기반의 효율적인 클라이언트 서버시스템 개발이 가능하다. 현재 상용 웹 브라우저가 자바 가상머신(JVM)을 기본으로 포함하고 있어 웹 애플리케이션에서 클라이언트 프로그램을 자바로 구현하는 것이 보편화되고 있다.

자바 언어의 다른 큰 장점으로는 호환성, 이식성이 높은 SW개발이 가능하다는 것이다. 자바 컴파일러는 하드웨어, 운용체계에 독립적인 바이트 코드를 생성하고 이를 자바 해석기가 실행하므로 JVM만 있으면 어떤 환경에서도 동일한 GUI를 통하여 애플리케이션이 실행될 수 있다. 또한 이 언어는 C++에 비하여 구문이 훨씬 간단해져서 배우기가 쉬운 장점이 있다. 전산학과의 입문

프로그래밍 언어로 자바를 가르치는 대학이 늘어가고 있다.

자바 개발환경인 JDK 2.0은 분산객체컴퓨팅 아키텍처인 CORBA의 IIOP 프로토콜을 기본으로 지원할 예정이어서 향후 자바를 기반으로 한 클라이언트 서버 및 분산구조의 웹 애플리케이션 개발이 더욱 보편화될 전망이다. 자바 언어의 가장 큰 한계점은 바이트 코드의 해석방식으로 인한 실행속도인데 최근에 적시(Just In Time)컴파일 기술의 발전으로 상당 부분 해소될 것으로 예상된다.

객체지향 애플리케이션은 객체로 구성되어 있으며 이중 일부 객체만이 영구(Persistent)객체가 되어 DB에 저장된다. 객체지향 DB는 관계형 DB가 데이터를 엔티티 테이블과 관계 테이블 형태로 저장하는 대신 객체 단위로 저장되기 때문에 멀티미디어, 캐드(CAD), 전자상거래의 상품 정보 등 비정형 데이터를 저장하기에 매우 효과적이다. 객체 DB에 저장된 객체가 자신의 데이터를 처리할 수 있는 메소드를 함께 가지고 있으므로 애플리케이션과 DB간의 데이터 전송을 줄일 수 있으며 DB 트랜잭션처리 성능도 크게 향상된다.

DB는 1970년대의 계층적 DB에서 1980년대의 관계형 DB를 거쳐 객체지향 DB로 발전하고 있다. 각 DB기술은 초기상태에서 기술 개발이 완성된 수준까지 10여년의 기간이 소요된다. 객체지향 DB의 경우 2000년 초반에 이르러 오늘날의 관계형 DB와 같은 수준의 기술적인 성숙을 이룰 것으로 예상되지만 현재의 기술 및 ODBMS의 제품 수준이 상용 애플리케이션 개발에 충분할 정도이므로 미국 등에서는 객체지향 애플리케이션 개발에 그 적용이 빠른 속도로 증가하고 있다.

객체지향 DB의 표준화를 위해 91년에 ODBMS 회사들의 비영리 컨소시엄으로 객체데이터 관리그룹(ODMG)이 출범했고 다양한 ODBMS 제품들간의 상호 연동을 지원하기 위한 최근의 표준은 97년에 발표된 ODMG 2.0이다.

이 표준은 독자적인 표준을 지양하고 기존의 OMG 표준, SQL92 표준, ANSI에서 제정한 언어 표준을 수용, 반영하여 객체지향 DB개발에 활용될 수 있도록 만들어졌다. ODMG 2.0은 객체정의언어(ODL), 객체질의언어(OQL) 및 언어 바인딩 등 3가지 주요 표준으로 구성되어 있다. 특히 최근의 자바 언어 바인딩으로 인해 웹 애플리케이션에서 객체지향 DB의 사용이 본격화될 것으로 예상된다.

관계형 DBMS에 비하면 아직 ODBMS의 시장점유율은 낮은 편이나 객체기술이 보편화되어가고 있고 인터넷 기반의 전자상거래 등이 활성화될 것을 고려하면 ODBMS의 활용은 점차 증가할 것으로 기대된다. 현재 대표적인 상용 ODBMS로는 Object Store, O2 등이 있다.

클래스의 작은 단위에서의 재사용을 보완하고 객체간의 전형적인 메시지 흐름도 재사용할 수 있는 기술로 프레임워크(Framework)기술이 부상되고 있다. 프레임워크는 재사용을 목적으로 한 준(Semi Completed) 애플리케이션이다. 특정 도메인에서 여러 애플리케이션에 공통으로 적용될 수 있는 클래스와 메시지 흐름을 이용하여 주요 기능을 구현해 놓은 대단위 재사용부품이다. 하나의 프레임워크는 특정 기능을 수행하기 위한 클래스들, 이들간의 메시지 교환, 프레임워크 인터페이스, 각 애플리케이션에서 구체화해야 하는 훅(Hook) 메소드 등으로 구성되어 있다.

프레임워크는 프레임워크 인터페이스를 통하여 해당 객체의 메소드를 실행시켜줌으로써 재사용하는 개발자가 프레임워크의 내부 구조와 내역을 모르더라도 재사용할 수 있다. 프레임워크 모듈성은 설계와 구현의 변화에 대한 영향을 지역화함으로써 존재하는 SW를 유지하고 이해하는 데 요구되는 노력을 최소화할 뿐만 아니라 SW의 품질을 향상시키는 데 도움을 준다. 애플리케이션별로 특정 기능을 다른 방식으로 수행할 경우를 대비하여 훅 메소드와 접속용 객체(Pluggable Object)장치도 제공하고 있다.

현재 IBM의 샌프란시스코라는 상업용 프레임워크가 개발되어 평가단계에 와있다. 프레임워크 기술부문에서는 국내기술 수준도 상당히 높아 한국전자통신연구원(ETRI)의 소프트웨어공학 연구부에서는 프레임워크 개발방법론과 인터넷용 프레임워크를 개발했고 숭실대학교 객체기술 연구실에서는 전자유통 및 인터넷 상거래용 프레임워크를 개발하고 있다.

객체지향 재사용 컴포넌트를 개발하기 위한 모델로 선사의 기업형 자바빈스(EJB : Enterprise JavaBeans)가 크게 각광을 받고 있다. EJB는 분산객체 구조를 기반으로 하는 자바 애플리케이션의 개발과 배포를 위한 컴포넌트 모델인데 자바의 컴포넌트 모델을 기반으로 한 서버용 컴포넌트 개발에 유용하게 사용된다. 프레임워크가 특정 언어에 독립적인 컴포넌트 개념을 정의한 것이라면 EJB는 블랙박스 프레임워크를 자바를 사용하여 구현할 수 있는 상용 컴포넌트 개발기술이다. 현재 EJB를 지원하는 도

구가 다수 개발되어 있어 향후 본격적인 컴포넌트 기반 SW개발이 가능해졌다.

최근들어 컴포넌트를 이용한 SW 개발방법에 관한 학문인 컴포넌트 기반 소프트웨어공학(CBSE : Component Based Software Engineering)이 국제학술대회나 국제저널 등에서 심도있게 다루지고 있고 산업계에서의 기술 및 제품개발이 활발하여 향후 SW 개발방식에 큰 변화를 가져올 것으로 예상된다.

인터넷 기반의 웹 애플리케이션은 기본적으로 클라이언트 서버구조를 가지고 있다. 최근들어 객체지향기술과 클라이언트 서버기술이 접목된 분산객체 컴퓨팅이 인터넷 애플리케이션 개발에 필수 기술로 정착되어 가고 있다. OMG에서 제안한 CORBA와 마이크로소프트(MS)가 제안한 분산컴포넌트모델(DCOM) 등이 대표적인 아키텍처다.

CORBA는 서로 다른 프로그래밍 언어로 작성된 객체간에도 IDL 컴파일러를 거쳐서 구현되었으면 이들간에 메시지 교환을 가능하게 해주는 언어 독립성과 지역 투명성을 제공한다. 따라서 인터넷 기반의 자바 객체와 서버상의 C++ 객체간의 연동이 가능해지므로 보다 고성능의 웹 애플리케이션을 개발할 수 있다. CORBA는 네이밍(Naming), 이벤트, 트랜잭션 등 여러개의 서비스기능을 제공하고 있으며 CORBA 기반의 재사용 객체를 정의하는 재사용 퍼실리티(Facilities)를 가지고 있다.

CORBA의 중심부인 객체요청브로커(ORB)는 클라이언트 객체가 서비스를 요청하고 응답을 받을 수 있는 객체 버스(BUS)를 제공한다. 즉 객체간의 통신을 담당하는 미들웨어로서 수신 객체가 어디에 있는지 발견하여 바인딩을 해준다. 또한 CORBA는 서로 다른 ORB제품간의 상호 연동도 지원하여 개방형 분산객체시스템 개발에 매우 유용하다.

MS의 DCOM은 네트워크상에서 분산된 객체간의 통신을 지원하는 컴포넌트모델(COM)의 확장모델인데 현재는 윈도환경에서 데스크톱 애플리케이션 개발시 매우 효과적으로 사용될 수 있다. OMG와 MS는 각각 별도의 CORBA to DCOM 연동규격 및 제품을 개발하고 있어 향후 이 두 아키텍처간의 호환은 문제되지 않을 것으로 예상된다.

자바 언어, CORBA 표준, UML 등의 등장으로 국내의 SW산업계에서도 객체기술에 관심이 매우 높고 상당수 과제들이 객체지향 방식으로 개발되고 있다. 그러나 객체지향 과제를 수행하는

인터넷 기반의 웹 애플리케이션은 기본적으로 클라이언트 서버구조를 가지고 있다. 최근들어 객체지향기술과 클라이언트 서버기술이 접목된 분산객체 컴퓨팅이 인터넷 애플리케이션 개발에 필수 기술로 정착되어 가고 있다. OMG에서 제안한 CORBA와 마이크로소프트(MS)가 제안한 분산컴포넌트모델(DCOM) 등이 대표적인 아키텍처다.

과정에서 예상치 못한 어려움을 겪게 되는데 대부분의 경우는 객체지향 분석과 설계기법의 표기법은 잘 이해하고 있지만 UML과 같은 모델링기법의 여러 효과적인 장치를 개발시 어떻게 적용해야 하는지의 깊이 있는 이해와 실무의 경험이 부족하기 때문이다. 따라서 객체지향 과제수행에 앞서 이론적 강의방식이 아닌 파일럿 과제를 통한 실무 트레이닝을 통하여 객체지향 모델링의 실무기술을 익히는 것이 매우 중요하다.

미국 등 전산 선진국에서는 객체기술을 실제로 적용해보는 실무교육을 통해 OOA/D 및 구현기술을 습득하는 것이 보편화되어 있다. 이러한 객체기술의 발전과 산업계의 넓은 수용으로 인해 국내에서도 객체관련 고급기술을 전문으로 교육, 컨설팅하는 오브젝소프트, 넥스텍, 데이텍 등과 같은 전문회사가 생겨났다. 객체지향기술은 어느 SW기업도 피할 수 없는 큰 발전 방향이므로 조기에 기술을 확보하여 SW경쟁력을 향상시키는 것이 매우 중요하다. 