

CBD(Component-Based Development) 방법론 및 적용사례 소개

전 동 수*

* 삼성SDS

Abstract

하드웨어와 네트워크의 성능은 지속적으로 성장하는 반면 소프트웨어의 성능과 품질은 비즈니스 활동을 적시에 지원하지 못하고 있어 소프트웨어의 현실은 심각하다. 소프트웨어 개발은 아직도 수작업에 의지하고 있어 납기지연, 부적절한 품질, 긴 개발주기, 높은 원가, 유지보수 어려움 등의 문제에서 벗어나지 못하고 있다. 이의 대안으로 소프트웨어 패키지가 등장했으나 구현기간도 길고 비용도 많이 소요되며 구현자체도 어려울뿐 아니라 통합과 기능 개선도 어려워 적절히 해결책이 되지 못하고 있다.

이러한 배경에서 등장한 것이 바로 CBD 방법론과 Component 산출물이다.

CBD의 주 이점인 재사용성에 대한 연구는 1980년대말 객체지향방법론을 기반으로 계속 발전해 왔다. 그 이후 OMG(Object Management Group)는 분산환경에서 객체의 재사용 표준인 CORBA를, Microsoft사는 OLE와 ActiveX를 기반으로 하는 COM/DCOM을, Sun사는 Enterprise JavaBeans 등의 Component Model을 발표하였다. 그 이후 여러 미들웨어 벤더들이 표준을 지원하므로 Component는 객체와 분리된 독립적인 개념으로 등장하게 된다.

Component의 장점인 Interoperability(Language, Environment, Platform)와 재사용에 의한 생산성 효과가 산업계에 인식되면서 분석단계부터 Component 개념을 도입한 CBD 방법론이 본격적으로 등장하게 된다.

CBD 방법론은 다음과 같이 소프트웨어의 근본적인 문제를 해결할 수 있는 해답을 제시한다.

첫째는 비즈니스와 기술의 변화에 적응이 용이하고 최고의 품질과 높은 생산성을 보장한다.

둘째는 기존 시스템과 기술을 새로 개발하는 애플리케이션에 쉽게 결합할 수 있는 실용적인 Transition Framework을 제공한다. 이는 기존 투자를 보호할 뿐 아니라 점진적인 기능 개선이 가능하고 비즈니스 요구 변화에 신속히 대응할 수 있다.

CBD 방법론의 급속한 확산으로 하드웨어의 플러그 앤 플레이 방식처럼 소프트웨어도 검증된 소프트웨어 Component를 구입하여 기존 운영환경의 혼란 없이 기개발 애플리케이션에 붙여 기능을 개선하게 될 것이다.

결과적으로 애플리케이션의 성능 개선 방법으로 비용이 많이 드는 재개발 대신에 비용과 시간이 훨씬 적게드는 Component를 이용한 점진적 개선이 가능하게 되었다.

본 논문은 CBD 방법론을 적용하여 Component 분석/설계/개발 및 기존의 애플리케이션에 적용한 사례를 소개한다. ERP의 주요 모듈인 SD(Sales & Distribution)의 주문관리기능을 구현하고 서브기능 중에 재사용이 빈번하게 발생하는 Availability Check 기능을 Component 개발대상으로 선정하여 설계는 Sterling S/W사의 Model TOOL인 COOL:SPEX로 개발은 I-CASE TOOL인 COOL:GEN을 사용하였다. 개발후에 주문관리시스템에 플러그인하여 재사용하였다.