

UML 모델링과 Software Archeology를 품에 안은 Java 개발툴

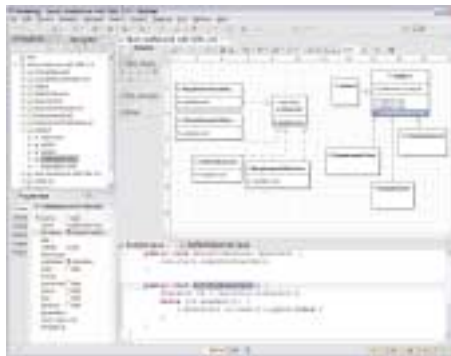
JBuilder 2008 R2



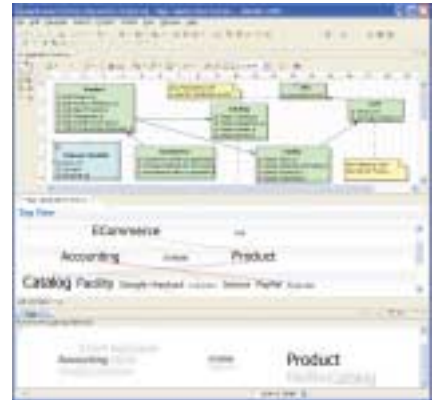
데브기어 (02-595-4288, www.devgear.co.kr)

- 종류 : UML 모델링 기능을 포함한 Java IDE
- 특징 : 풍부한 투게더 모델링 기능에 팀 단위 개발 생산성과 표준화를 위한 획기적 기능
- 평가 : ★★★★★
- 가격 : 별도 문의(240페이지 이벤트 참조)

Eclipse는 자바 개발에 있어 개발 플랫폼을 구축하고 광범위하게 확장 가능한 프레임워크를 제공한다. 또한 다양한 개발툴들(Building, 배포를 위한 런타임, 애플리케이션 라이프사이클 전반에 걸친 소프트웨어 관리)을 위해 한정된 오픈소스 커뮤니티를 대표한다. 한편으로, Eclipse는 어느덧 여러 상업적인 목적의 자바 IDE에 기반 플랫폼으로 적용되어 있다. 이번에 리뷰하는 JBuilder 2008 R2 역시 그러한



<화면 1> Live Source - UML 모델링



<화면 2> Application Factory 1

대표적인 IDE로서, Eclipse 3.4.1 버전과 WTP 2.0을 기반으로 하고 있다. 필자는 현재 금융권 SI 프로젝트를 주로 수행하는 조직에서 개발과 관련된 기술표준화 영역(소프트웨어 아키텍트)에 종사하고 있다. 본 리뷰는 필자가 업무상 반드시 고려해야 하는 개발 생산성의 향상, 개발표준화의 관점에서 진행하려고 함을 밝힌다.

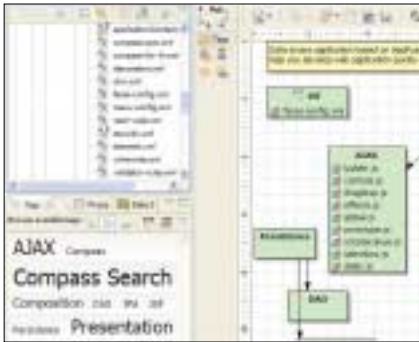
투게더 모델링 기능 - Live Source

JBuilder의 여러 기능들 중 가장 먼저 눈에 띄는 장점은 별도의 모델링 툴을 구매하지 않아도 내장된 투게더 기능(Live Source)으로 UML 2.0 기반 자바 모델링이 가능하다는 점이다. JBuilder 2008 R2는 UML의 9가지 다이어그램을 포함하고 있다. 필자가 주로 진행하는 SI 프로젝트에서는 요구사항, 분석, 설계, 구현 테스트, 배포 등 소프트웨어 라이프 사이클이 각각 필요

로 하는 산출물을 방법론과 표준에 따라 작성하는데, 이 중 요구사항, 분석, 설계 단계에서 작성하는 산출물은 대개 UML에서 정의하는 다이어그램이다.

이 작업을 위해 보통은 별도의 모델링 툴을 구매해 작업하는데, JBuilder를 이용하면 별도의 모델링 툴 구매 없이 모델링을 진행할 수 있다는 점이 매력적이다.

또한 전통적으로 투게더의 탁월한 실시간 소스 코드 생성 기능은 유명한데, 수년 전 수행했던 프로젝트에서 우리 팀은 아키텍트가 사전에 정의한 패턴에 따라 분석/설계모델을 생성하는 모듈을 투게더의 템플릿 API를 이용해 개발했다. 그 모듈은 설계자를 위한 기본 모델을 생성했고, 설계자는 이 기본 모델 패턴에서 출발해 실제 분석/설계모델을 작업하는 형태였다. 그리고 이 결과 자동으로 만들어진 소스 코드를 개발자가 완성하는 식으로 개발을 진행했다.



〈화면 3〉 Application Factory 2



〈화면 4〉 Team Insight



〈화면 5〉 Swing Designer

Application Factory

‘개발자가 가진 지식을 다른 사람들과 공유하도록 만들기’는 모든 프로젝트에서 애플리케이션 아키텍트의 역할이지만, 쉽지 않은 일이며 또 여러 툴들이 필요하다. 우리는 개발 프로젝트를 시작할 때와 끝난 후 항상 이전 프로젝트에서 쌓았던 지식(Asset)을 어떻게 효과적으로 재활용할 수 있을까 하는 고민을 하게 된다. 그저 프로젝트마다 꾸준히 추가되어 그래서 원래의 의도를 알 수 없는 코드들의 조합을 써야 하는 상황에 직면한다면 이는 심각한 상황이 아닐 수 없다. 프로젝트를 수행하면서 만 들었던 환경과 각종 개발 표준, 그리고 표준 모듈들을 재활용하기 위해 가이드라인을 만들고 팀을 정리하고 기존 소스를 인스펙션하는 등의 작업을 해서 프로젝트 리파지토리에 보관하는 등의 노력을 끊임없이 기울임으로써 이후 유사 도메인에서의 프로젝트를 대비한다. 하지만 이것은 매번 작업을 하는 입장에서는 녹록치 않은 작업일 수밖에 없다. JBuilder가 제공하는 Application Factory에서 그에 대한 힌트를 찾아보자. JBuilder 2008 R2에서는 Struts 2, Spring MVC, JSF용 데이터베이스 웹 애플리케이션, e-commerce와 기타 템플릿 애플리케이션이 기존에 생성한 모듈로 기본 제공되고 있다. 하지만 그 뿐이면 재미 없다. 우리가 작성한 모듈을 팩토리에 저장한다. 그리고 프로젝트/파일 각각에 이후 활용을 위한 우리의 의도를 글과 그림(아키텍처 다이어그램)으로 기록할 수 있다. 심지어 초기 활용을 위한 스타트 프로그램도 작성할 수 있다.

Application Factory는 개발조직에서 애플리케이션의 진화(버전업)를 추적해 나갈 수 있도록 애플리케이션 고유의 메타데이터를 저장하고 관리하는 기능을 제공한다. 그리고 태그를 활용해 개발자의 의도를 기록해서 이를 통해 애플리케이션을 구조화하고 관련 코드를 생성한다. Application Factory는 애플리케이션 모듈과 베스트 프랙티스를 통한 빠른 시작을 도와줄 수 있을 것으로 보인다. 관리 대상으로 선정된 애플리케이션 모듈은 각자의 생명 주기를 갖고 리파지토리에 보관한다. 애플리케이션 모듈뿐

만 아니라 개발자의 이해와 개발 의도, 진화 과정(노하우)을 모두 공유해 바로 재사용할 수 있게 구조화한다. 이렇게 되면 프레임워크를 선택할 때 애플리케이션 주도 개발 모델 방식으로 접근하게 한다. 계속되는 개발 과정 동안에 애플리케이션 모듈은 좀 더 구조화되고 진화하게 될 것이고 그 내용은 모두 메타데이터화해서 공유한다. 멋진 아이디어다. 개발조직에서는 신규 멤버가 투입되었을 때 교육시간과 프로젝트 적응 시간을 단축할 수 있을 것이다.

Software Archeology

대체로 개발자들은 다른 사람이 작성한 소스는 인수하기를 꺼려한다. 명확하고 적용하기 쉬운 개발표준이 부재하고 또 있는 개발표준도 지키기가 쉽지 않아 분석과 수정이 어렵기 때문이다. 사실 개발 프로젝트에서 표준화하는 중요한 부분 중 하나가 소스 표준을 수립하고 이를 개발자들이 준수하도록 하는 것인데, 이것이 현실적으로는 대단히 어렵다. 이 때문에 꼭 필요한 것이 소스의 인스펙션이고 이를 위해 필자의 회사를 비롯한 대부분의 SI 프로젝트 개발조직은 많은 비용을 지불하는 것이 현실이다. JBuilder 2008 R2에서는 별도 추가 모듈 없이도 툴 자체에서 코드 감사/매트릭스를 제공한다. 풍부하면서도 커스터마이징 가능한 규칙을 활용한 감사/매트릭스, 코드 커버리지 측정이 가능하다. 그리고 프로젝트 유형을 사전에 구성할 수 있으며, 산출문서를 생성하는 기능을 제공한다. ProjectAssist와 TeamInsight는 Borland StarTeam을 포함한 ALM 툴 선택 확장이 가능한 오픈 API를 제공한다. 대개 IDE가 포함하고 있는 Performance/Memory Profiler의 동작도 흥미롭다. 정적 코드 커버리지, 프로파일링, 힙 분석, 쓰레드 디버깅을 통한 분석, Java EE 프로파일을 위한 리퀘스트 분석기 기능이 포함되어 있다. 그 외에 성능 툴인 옵티마이저 기능이 번들되어 있고 시각적 EJB, JPA, 웹서비스 개발이 가능하다. ⊕

김영리 barnett@hanmail.net