

모델 중심 접근 방법과 아키텍처 모델

MDA 적용방안 탐색을 통한 길 찾기

One Step Ahead !!

발표자 : 송태국

2005.11.4

이비즈온(주)

전체 목차

- 개요
- MDA 뷰 – 소프트웨어 개발의 문제점
- MDA 솔루션
- MDA의 현실 적용
- 모델 중심 접근방법
- 아키텍처 모델 중심 접근방법
- 결론

개요

□ 소프트웨어 개발

- 하드웨어 성장에 비해 소프트웨어 성장은 느림
- 복잡한 대규모 시스템 구축 가능성 측면에서는 발전적임
- 소프트웨어 개발의 문제점
 - 노동 집약적임
 - 시스템은 항상 다른 시스템과 통신함으로써 동작
 - 요구사항은 지속적으로 변함

□ MDA

- 모델을 중심으로 이러한 소프트웨어 개발의 문제를 해결하고자 함
- 하지만, 현실적으로 많은 어려움이 존재함

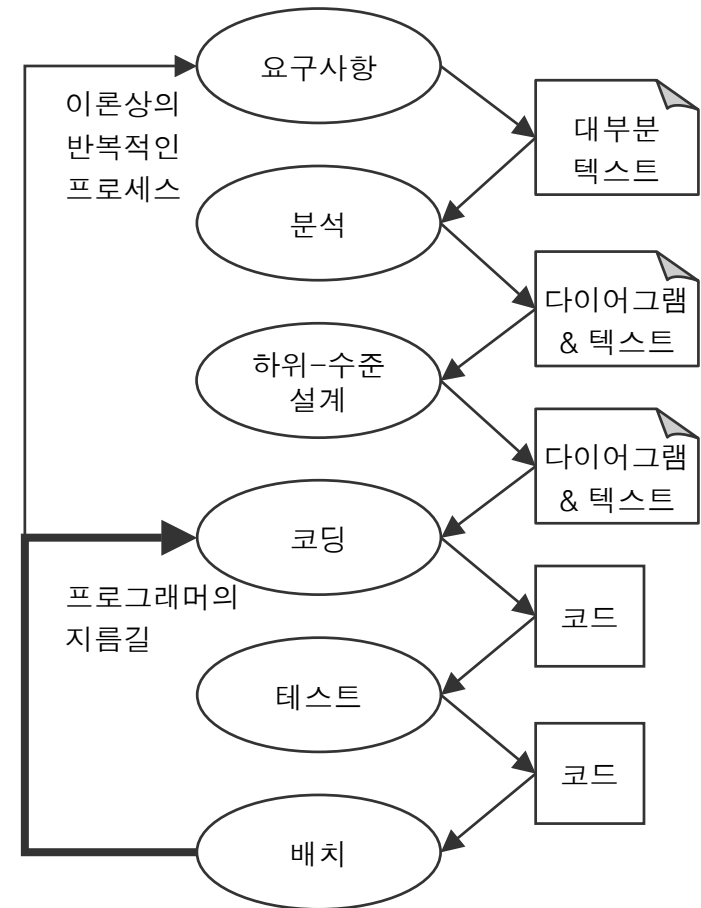
□ MDA 적용

- 사상, 기술, 아이디어 중 필요한 모든 것 활용
- 아키텍처 모델을 기준으로 수용할 수 있는 방안

MDA 뷰 - 생산성 문제

□ 생산성 문제

- 소프트웨어 개발 프로세스가 하위 수준 설계와 코딩으로부터 유도되는 경우가 존재함
- 전통적인 프로세스 단계
 - 1. 요구사항
 - 2. 분석
 - 3. 설계
 - 4. 코딩
 - 5. 테스트
 - 6. 배치
- 문서화 및 모델링 작업



MDA 뷰 - 이식성 문제

□ 이식성 문제

- 소프트웨어 산업의 특수한 특징
 - 매년 새로운 기술이 발견되고 대중화 됨
 - 예제) Java, Linux, XML, HTML, SOAP, UML, J2EE, .NET, JSP, ASP, Flash, Web Services 등
- 회사들이 신기술을 도입하는 이유
 - 고객의 요구(예; 웹 인터페이스)
 - 문제 해결(예; XML은 정보 상호 교환을, JAVA는 이식성을 해결)
 - 툴 벤더의 지원중단, 새로운 기술에 집중
- 새로운 기술
 - 회사의 이득 창출에 도움을 줌
 - 새로운 기술 자체도 지속적으로 변화함
 - 벤더들이 최근 2~3개의 버전만을 지원

MDA 뷰 - 상호운영성 문제

□ 상호 운영성 문제

- 대부분의 시스템은 다른 시스템(종종 현존하는 시스템)과 통신함
 - 전형적인 예제)
 - 지난 몇 년 동안 많은 회사들이 새로운 웹-기반의 시스템을 구축해온 것을 볼 수 있었고, 새로운 최종-사용자 어플리케이션은 웹 브라우저(HTML, ASP, JSP와 같은 다양한 신기술을 사용) 상에서 실행되고, 이는 기존의 백-엔드 시스템으로부터 정보를 수집함
- 다양한 기술(때로는 이전 기술과 신기술의 조합)을 사용하여 시스템 구축
 - 예) EJB를 사용하는 시스템도, 저장소를 위한 관계형 데이터베이스를 사용함
- 지난 몇 년간, 대규모의 단일체 시스템은 존재하지 않았음
 - 대부분 서로 상호작용 함으로써 이와 동일한 역할을 수행하는 컴포넌트를 구축하려고 시도하고, 이는 시스템의 변경을 용이하게 만들었음
 - 서로 다른 컴포넌트 들은 가장 우수한 기술을 사용하여 만들어졌고, 서로 상호작용하며 동작하였음
 - 이와 같은 상황은 상호 운영성의 필요성을 제시하였음

MDA 뷰 - 유지보수 및 문서화 문제

□ 유지보수 및 문서화 문제

● 문서화

- 문서화는 소프트웨어 개발 프로세스와 연관성이 적음
- 개발자의 주요 작업이 코드 작성이라고 생각함
- 개발 시 문서작성은 프로세스를 지연시킴
- 프로젝트 참여자들 위한 문서가 아니라고 인식함
- 관리자를 위한 문서라고 인식함
- 따라서, 좋은 품질의 문서가 작성되지 않고, 문서가 최신의 상태를 유지하지 않음

● 개발자의 작업

- 변경이나 유지보수가 가능한 시스템 구축이 주된 목적임
- 즉, 문서를 작성하는 것이 개발자의 필수적인 작업 중의 하나임

● 해결 방법

- 소스 코드로부터 직접 문서를 생성
- Java와 같은 몇몇 프로그램 언어에서 문서화가 효과적으로 지원이 됨
- 하지만, 이런 해결 방법은 단지 하위-수준의 문서화 문제만을 해결하고, 보다 높은 상위-수준의 문서화(텍스트 & 다이어그램; 이는 반드시 수행해야 하는 작업) 문제는 해결하지 못한 채, 수작업을 수행하도록 남겨두고 있음

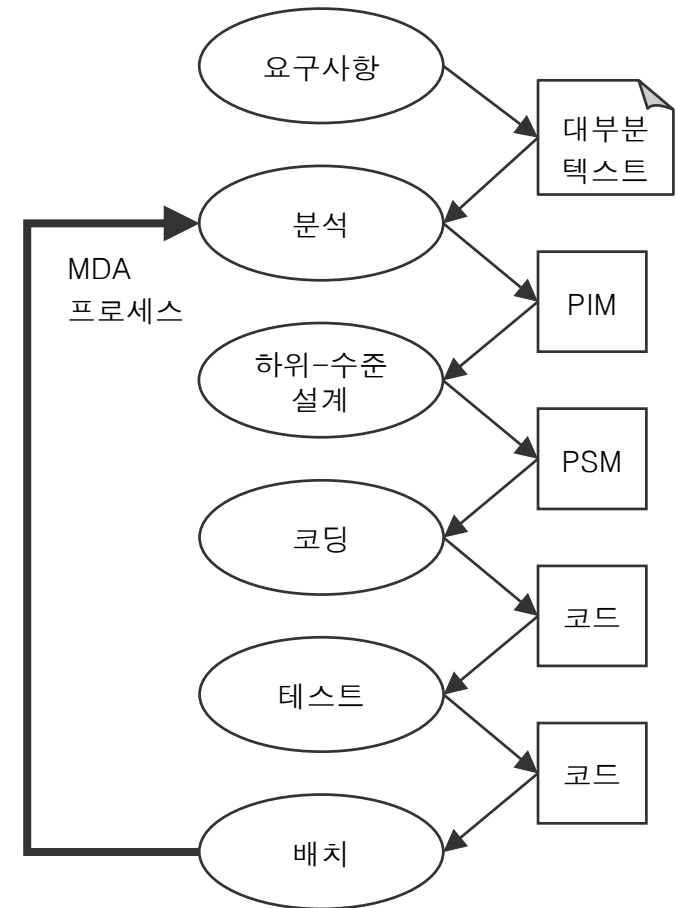
MDA 솔루션 - 개요

□ MDA 개요

- MDA는 소프트웨어 개발을 위한 프레임워크 임
- MDA의 핵심은 소프트웨어 개발 프로세스에 있어 **모델의 중요성**을 강조함
- 소프트웨어 개발 프로세스는 소프트웨어 시스템을 모델링 하는 활동으로부터 나옴

□ MDA 개발 라이프 사이클

- MDA 소프트웨어 개발 라이프 사이클
 - 전통적인 프로세스와 의 근본적인 차이점은 개발 프로세스의 산출물
 - 산출물은 컴퓨터에서 이해 될 수 있는 정형화된 모델임
- MDA 핵심 모델
 - PIM (Platform Independent Model)
 - PSM (Platform Specific Model)
 - 코드 모델
 - 서로 다른 추상화 수준을 표현함



MDA 솔루션 - 개발 프로세스

□ MDA 개발 프로세스

● PIM

- 높은-수준의 추상화를 표현하는 구현 기술과 독립적인 모델임
- 플랫폼 독립적인 모델이라고 불림
- 특정 비즈니스를 지원하는 소프트웨어 시스템을 설명함
- 비즈니스를 지원하는 방법에 대한 관점에서 시스템을 모델링 함

● PSM

- PIM은 하나 혹은 그 이상의 특정한 플랫폼 모델로 변환됨
- PSM은 특정 구현 기술에 사용되는 구현자적인 관점에서 시스템을 명세하기 위하여 조정됨
 - 예제) EJB PSM은 EJB의 “홈 인터페이스”, “엔티티 빈”, “세션 빈”과 같은 EJB 용어를 사용함
- 특정 플랫폼에 해박한 개발자에게만 명백한 정보를 제공함
- 특정 기술 플랫폼에 따라 서로 다른 PSM이 생성됨
- 하나의 PIM 으로부터 다양한 PSM이 생성되는 것이 일반적임

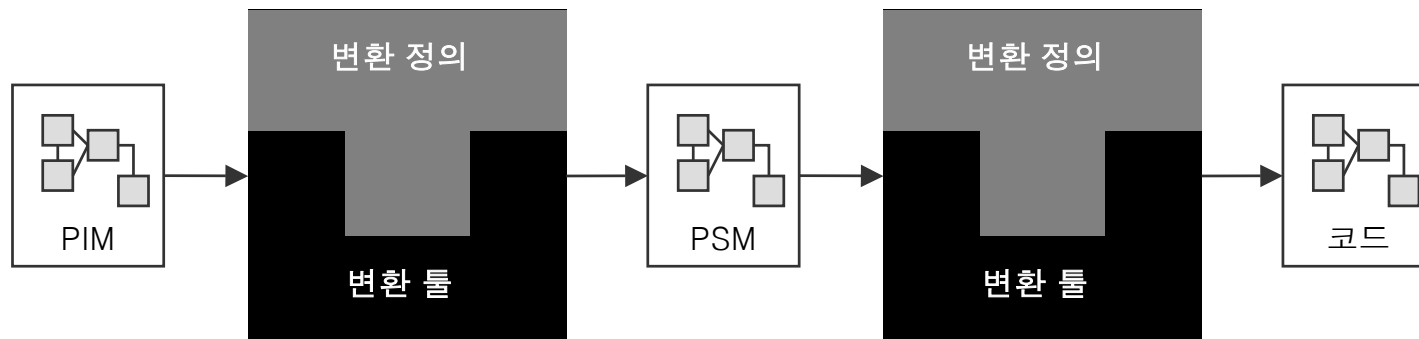
● 코드

- 마지막 단계로 각 PSM은 코드로 변환되어짐
- PSM은 직접 코드 변환이 가능함
- MDA는 PIM, PSM, 코드를 정의함과 동시에, 이들 간의 연관성에 대해서도 정의함
- 가장 복잡한 단계는 PIM이 하나 이상의 PSM으로 변환되는 것임

MDA 솔루션 - 변환

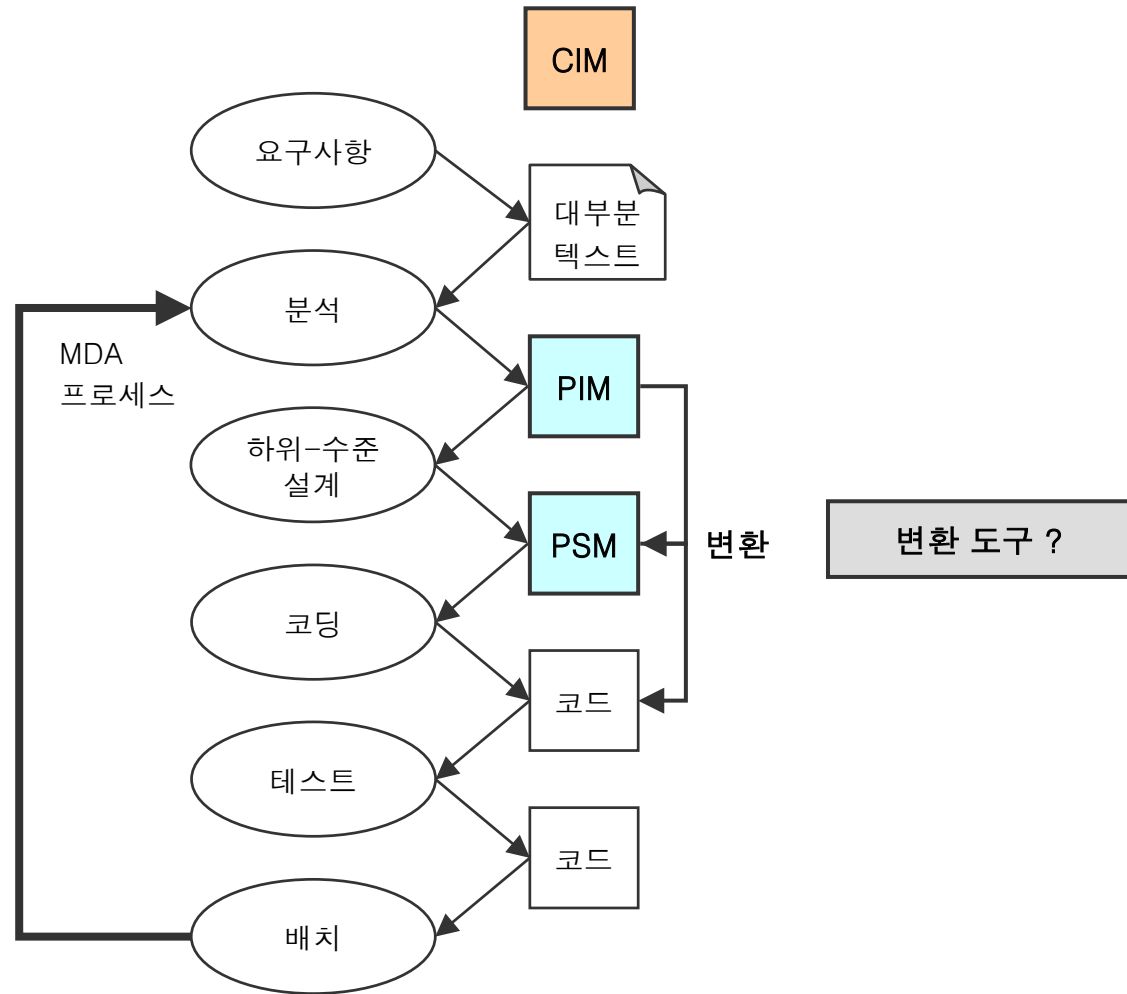
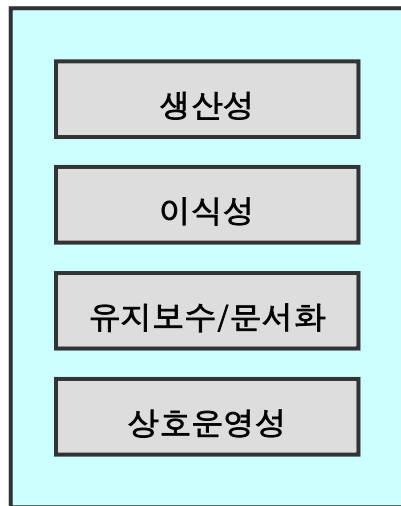
□ 변환(transformation)의 자동화

- MDA 프로세스는 전통적인 프로세스와 변환에서 차이가 있음
- 전통적인 변환
 - 모델과 모델 사이의 변환, 모델과 코드 사이의 변환은 주로 수작업으로 수행됨
 - 다양한 툴이 모델로부터 코드를 생성하긴 하였지만, 이는 단순한 템플릿 코드에 불과하였고, 세부적인 코드 작업은 역시 수작업에 의하여 이루어짐
- MDA 변환
 - 툴을 사용하여 변환을 자동으로 수행
 - 현재의 MDA 툴로는 PIM에서 PSM으로 PSM에서 코드로의 100% 전환이 어려움
 - 개발자들의 수작업을 통하여 변환된 PSM과 코드를 다듬어 줄 필요가 있음
 - 그러나, 현재의 툴로도 PIM으로부터 실행 어플리케이션의 생성이 가능하고, 이는 기본적인 기능을 제공함



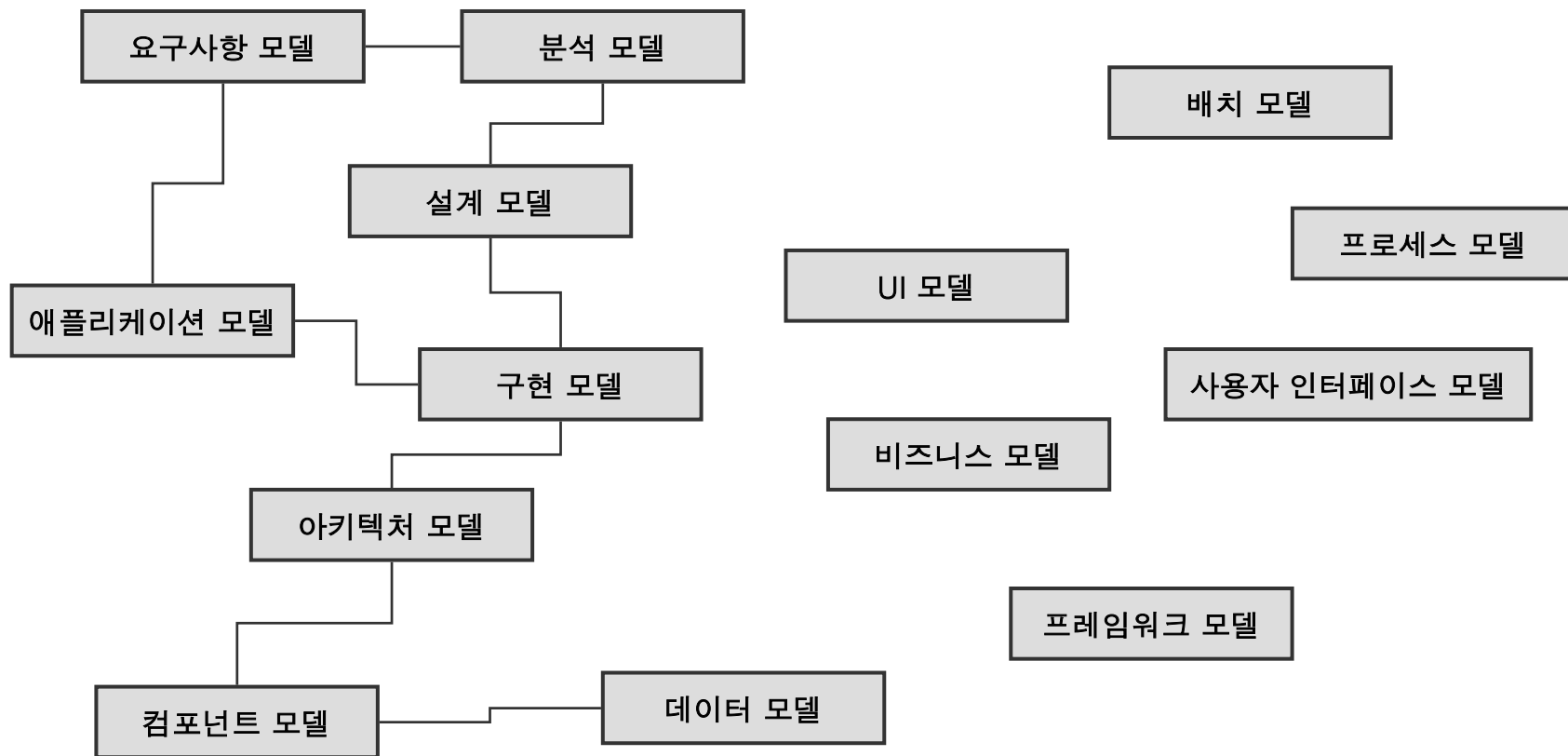
MDA의 현실 적용

- 모든 개발 노력을 모델에 집중
- 하지만, 현실은...



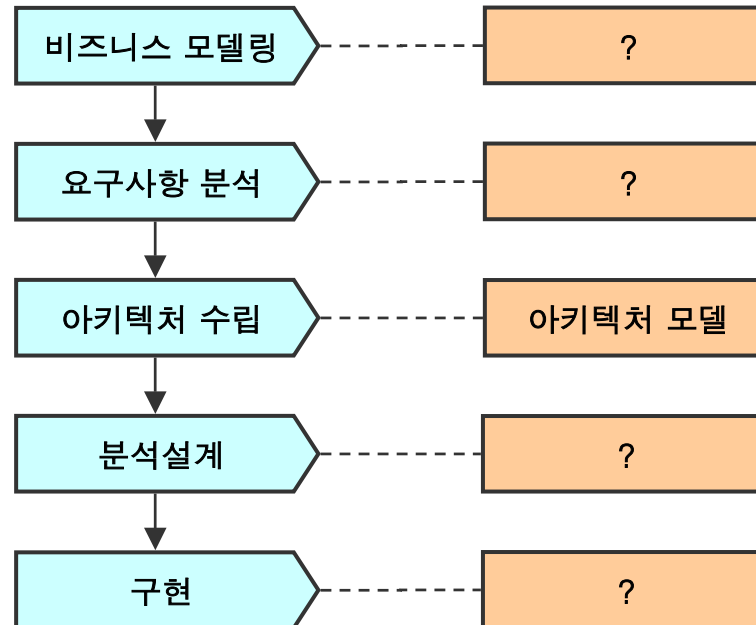
모델 중심 접근방법 - 다양한 모델

- 다양한 모델
- 어떤 모델을 중심으로....



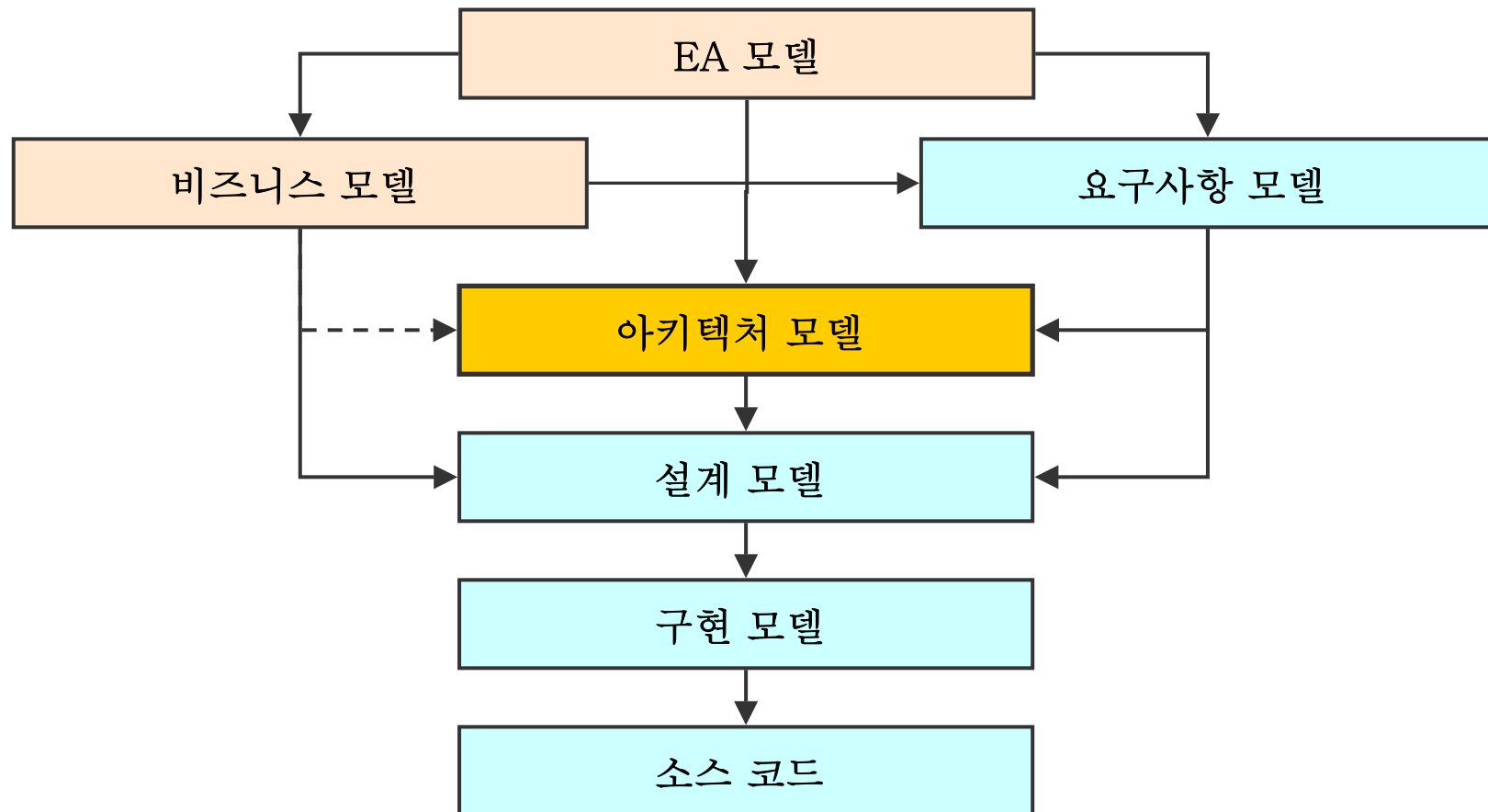
모델 중심 접근방법 - 아키텍처 모델 중심

- ❑ Architecture-Centric 시스템 구축
- ❑ 모든 시스템은 하나의 아키텍처를 가진다(IEEE-1471)



아키텍처 모델 중심 접근방법

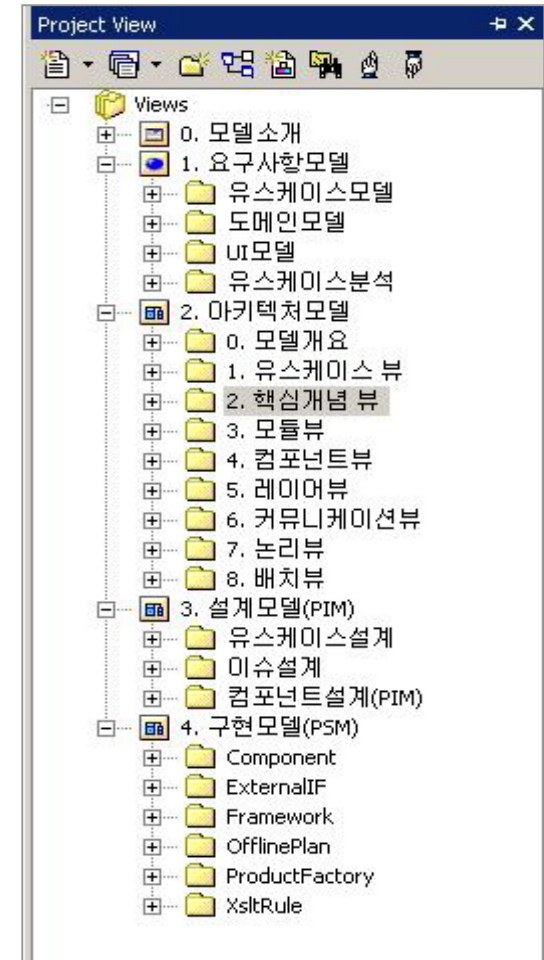
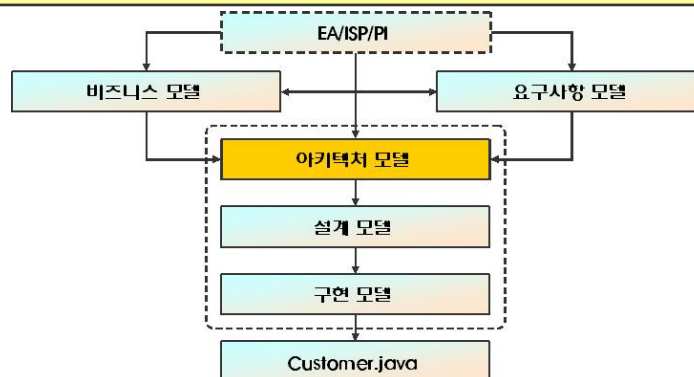
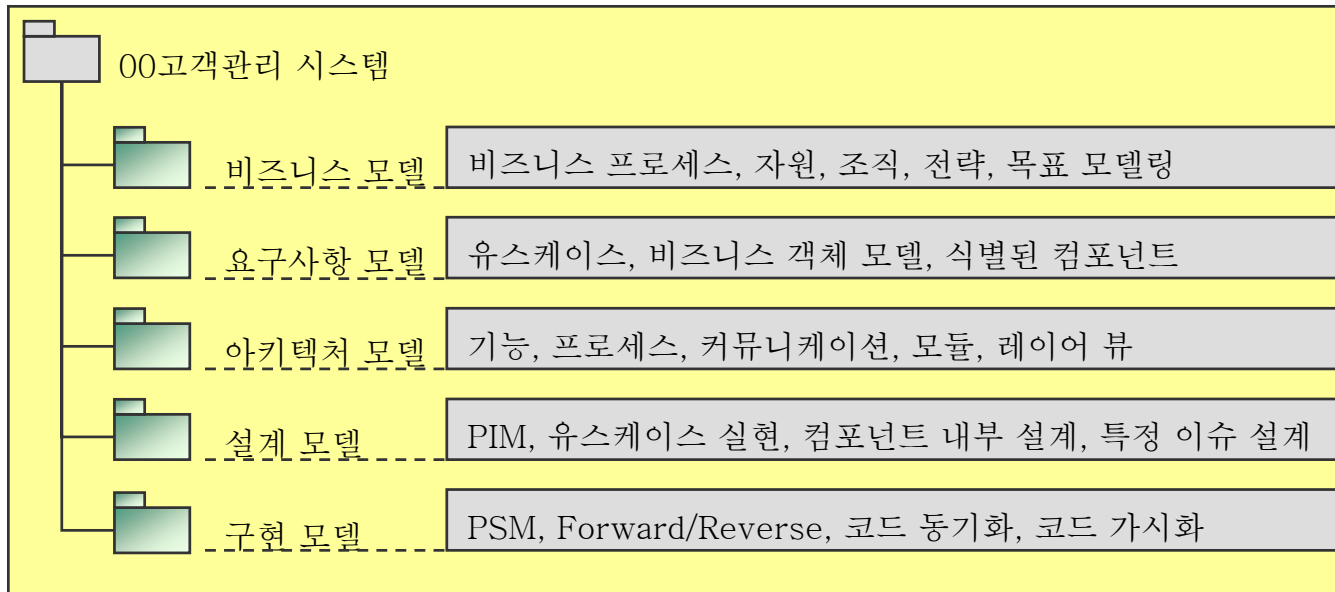
- 모델이 산출물의 중심
- 직접적인 참조에 의한 In/Out 관계 형성



아키텍처 모델 중심 접근방법 - 사례

- ❑ “형식이 내용을 규정한다.” = “모델 구조가 모델의 성격을 규정한다.”
- ❑ 5 모델 기반 설계 모델의 구조

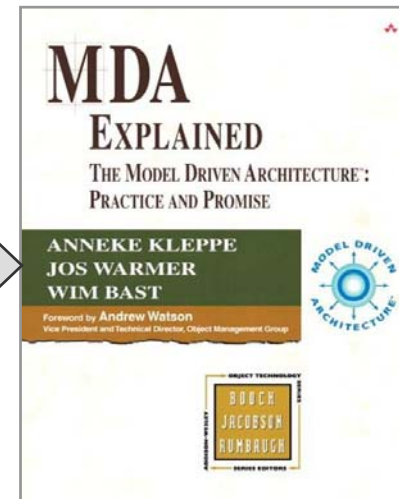
[모델의 예]



결론

- ❑ 아키텍처 모델은 전체 모델의 중심이 되어야 함
- ❑ 아키텍처 모델은 비즈니스와 시스템, 요구사항과 설계를 연결하는 다리
- ❑ OMG의 CIM→PIM부분에 대한 해결책 ?

MDA 참조 자료



연락처

❑ 감사합니다 !!

❑ 이비즈온

- 대표이사 : 손문일(neilson@e-bizon.com) : 02-430-2633
- 대표컨설턴트 : 송태국(tsong@e-bizon.com) : 02-430-2633
- 회사 URL : <http://www.e-bizon.com>
- 회사 주소 : 서울시 송파구 가락동 112-9 석종빌딩 4층

마지막 페이지 입니다 – 감사합니다 !