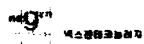
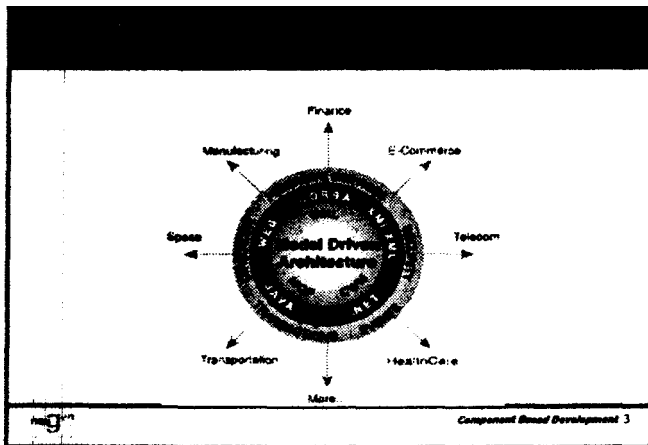


2002. 1. 31

류영규



1. MDA 소개 및 비전



How Systems Will Be Built

MDA™ provides an open, vendor-neutral approach to the challenge of interoperability, building upon and leveraging the value of OMG's established modeling standards: Unified Modeling Language™ (UML™), Meta-Object Facility™ (MOF™), and Common Warehouse Meta-model™ (CWM™). Platform-independent Application descriptions built using these modeling standards can be realized using any major open or proprietary platform, including CORBA®, Java, .NET, XML™, XML, and Web Base platforms.


Component Based Development 4

OMG's Mission


- Open, vendor-neutral interoperability Spec에 대한 제공
- 이를 통한 통합문제(Integration Problem) 해결

Next Step
in solving Interoperational Problem

Model Driven Architecture™



Component Based Development 5

- 1990년부터 Object Management Architecture(OMA) 활동
 - 초기 10년간 Roadmap 제공(CORBA)
 - 1997년 UML 표준 작성
 - MOF, XML, CWM 등의 기술 확장
 - OMG의 향후 방향 -> Model Driven Architecture
 - MDA를 통한 OMG 플랫폼 기술 완성(UML과 CORBA를 포괄한)
 - OMG의 proven, neutral, open standardization 프로세스 사용
 - OMG의 Vertical Market에 대한 적극적인 지원 (Healthcare, finance, telecommunications, life sciences research, manufacturing 등)
- 
- Component Based Development 6

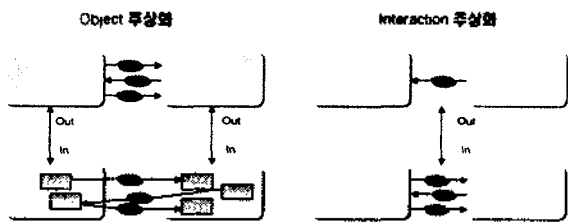
"Design once,
build it on any platform."

▪ MDA의 접근방식

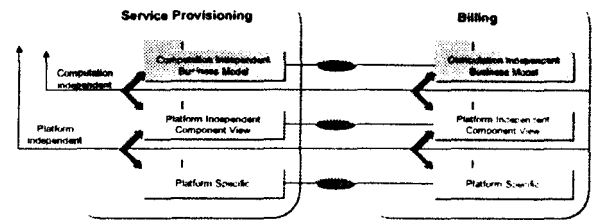
- 기능과 기술의 분리
 - 시스템의 기능과 구현 기술요소의 분리
- 모델 아키텍처 정의
 - 단계별 모델의 아키텍처를 정의
 - 하나의 기능 모델에서 여러 기술 플랫폼에 적용할 수 있는 표준 제공
- 시스템 통합 및 전환
 - 시스템 모델간의 통합을 통한 애플리케이션의 통합
 - 기술 플랫폼의 발전에 따른 시스템의 발전 제공

▪ MDA의 기본 개념

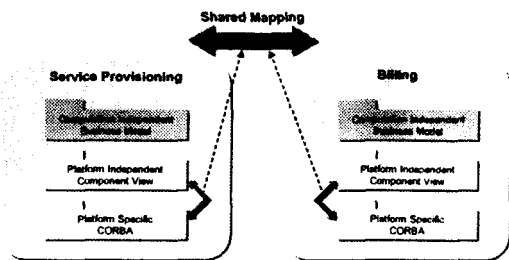
- 추상화 와 Zooming in & out



▪ MDA의 기본 모델 Architecture



▪ Shared Pattern of PIM <-> PSM Mappings

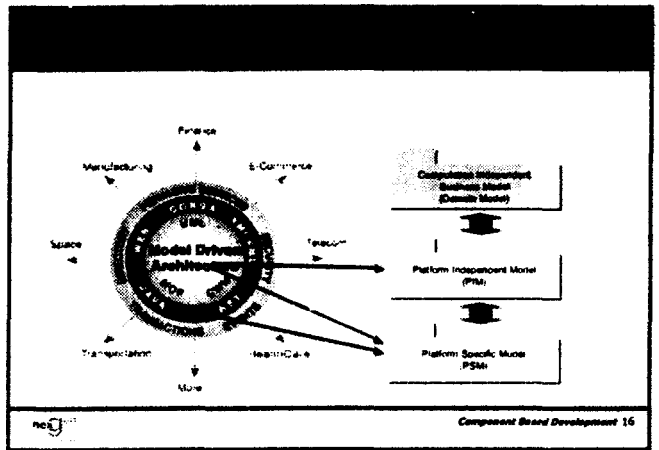
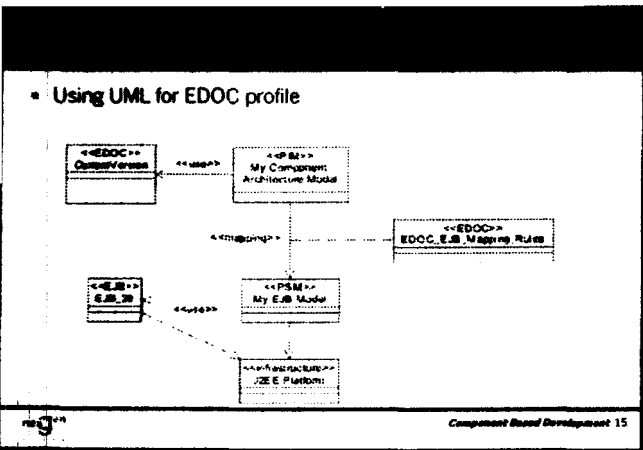
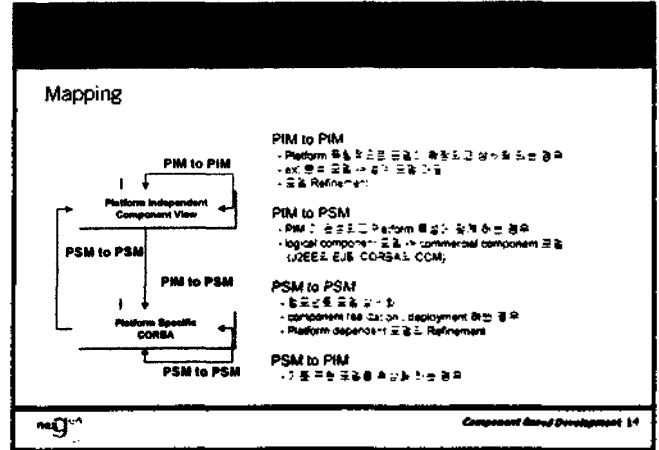
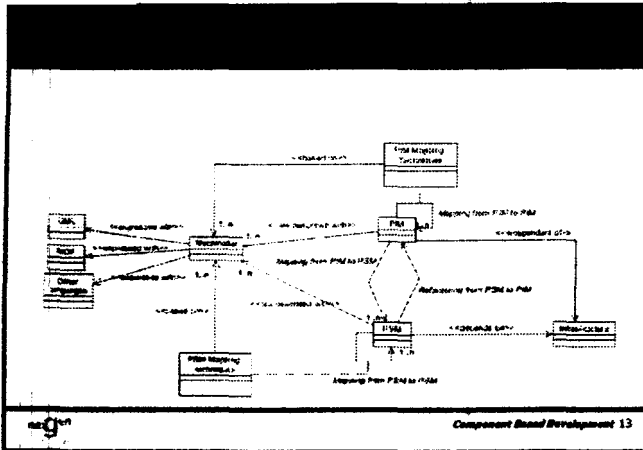


▪ Platform Independent Models in UML

- UML의 장점
 - UML을 기반으로 한 설계 방식이 발전되어 왔다
 - Textual/graphical 모델링 가능
 - 다른 모델링 언어 보다 풍부한 표현이 가능
- UML을 기반으로 PIM을 표현한다.

▪ Platform Specific Models in UML

- UML Profile을 이용한 확장을 통해 정의할 수 있다 (stereotypes.tagged values 등 사용)
- UML Profile for CORBA
- UML Profile for EDOC
- UML Profile for EJB 등



- 애플리케이션 라이프사이클 전반에 걸쳐 비용 감소 (Reduced cost)
 - 새로운 애플리케이션에 대한 개발 시간 단축 (Reduced development time)
 - 애플리케이션의 품질 향상 (Improved application quality)
 - 기술투자에 대한 수익 증대 (Increased return)
 - 기존 시스템에 새로운 기술의 빠른 적용 (Rapid inclusion of emerging technology)
- Component Based Development 17

- 메타 데이터 공유의 중요성 (The Importance of Shared Metadata)
 - 일반적인 서비스 및 프로그래밍 모델 (Common Services and Programming Models)
 - 플랫폼 정의 (Platform Specification)
- Component Based Development 18

- 지식 기반의 적용(Knowledge-Based Orientation)**
 - 지식에 기반한 모델 정의(추상화: 공통 모델 정의)를 통해 시스템에 효과적인 정보를 추출하고 활용
 - 구현, 저장방식, 미들웨어 서비스, 메타 모델 관리 등을 효과적으로 반영할 수 있는 방안
- 동적 아키텍처(Dynamic Architecture)**
 - 현재의 Mapping 방식(Mapping with shared Metamodel)을 모델이 Software에 의하여 직접적으로 해석될 수 있도록 대체
- 적응력을 갖춘 시스템(Adaptive Systems)**
 - 시스템이 변경이 되는 경우 시스템 모델의 변경을 통한 적응
 - 소프트웨어 전문가 없어도 수행 가능

neC Component Based Development 19

* Relationships Between Standards

neC Component Based Development 20

- Metadata Integration**
 - CWM(Common Warehouse Metamodel)**
 - 비즈니스와 기술적인 빌드레이아웃을 표현하는 메타 모델용 용어
 - 데이터 클래스, 분석, warehouse 관리, D/W비즈니스 환경등을 표현하는 메타 모델로 구성되어 있음(OMG의 CWM 참조)
 - UML(Unified Modeling Language)**
 - 이러한 객체, 객체간의 상호작용, 데이터 모델링, 개발과 배포(환경)등의 모델을 지원하는 표준 모델링 언어
 - MOF(Meta Object Facility)**
 - MDA에서 사용되는 표준 모델링과 변환구조를 제공한다
 - XMI(XML Metadata Interchange)**
 - 다양한 플랫폼, 저장장치, 미들웨어 사이에 사용되는 표준 변환 키텍서
 - UML, MOF 모델로부터 자동으로 XML DTO를 생성할 수 있다.

neC Component Based Development 21

- Common Services and Programming Models**
 - Java and J2EE(Java 2 Platform, Enterprise Edition)**
 - 분산환경의 애플리케이션 기반 아키텍처를 정의하는 Spec의 뒤를
 - Java 기술과 OMG MDA가 Mapping이 되도록 개발 중에 있다
 - JMI(Java Metadata Interface)**
 - 차별언어와 OMG의 MOF의 정형화된 특성을 제공
 - Unsys 주도하여 개발 진행중
 - JOLAP(Java OLAP Interface)**
 - OLAP Server의 자바 API
 - Hyperion Solutions Corporation 주도하여 개발 진행중
 - JDM(Java Data Mining API)**
 - 데이터 마이닝 기술을 제공하는 비즈니스 인텔리전스 어플리케이션의 자바 API
 - Oracle 주도하여 개발 진행중

neC Component Based Development 22

2. MDA 기반 J2EE 시스템 구축 방안

Convergent Architecture Metamodel and Stylistic Guidelines

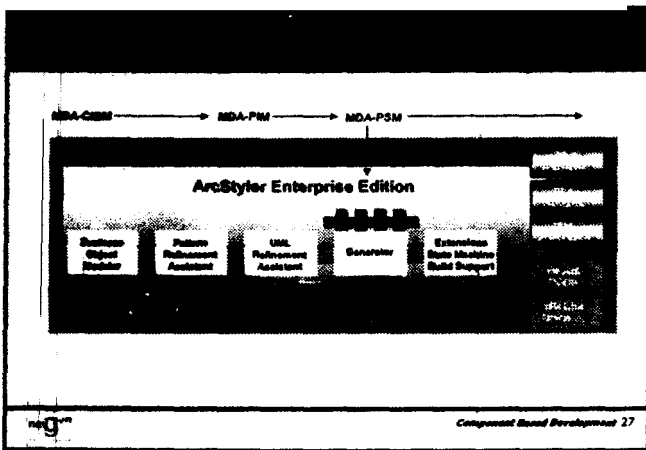
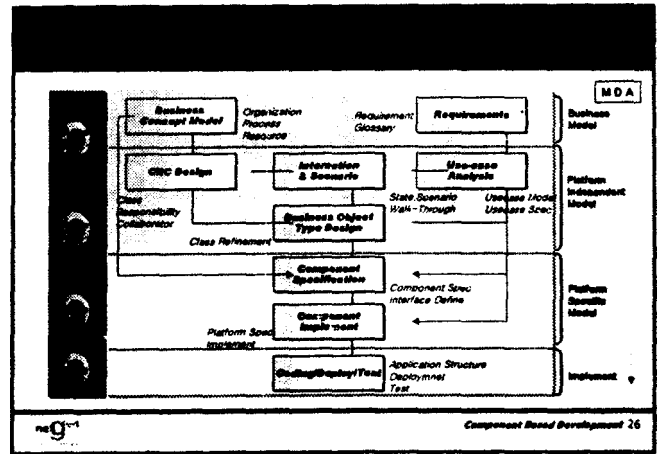
IT-Org. I RUP Workflow | Component Models in UML/XML/J2EE

neC Component Based Development 24

The patterns, component metamodels and fully-cycle technology analysis provided by Architectural Style enables complete coverage development assistance via, for example, rule based verifiers

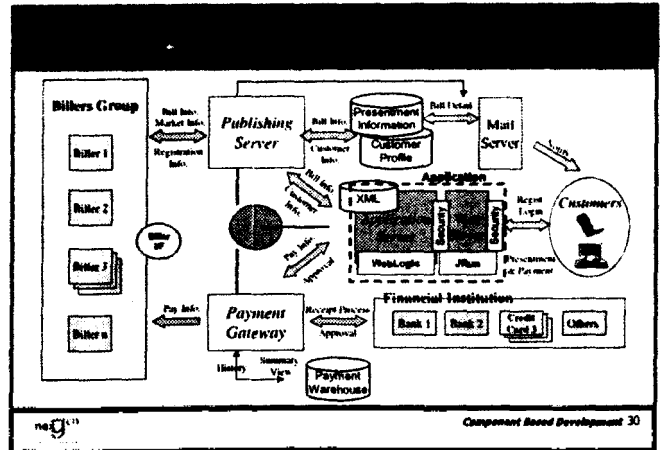
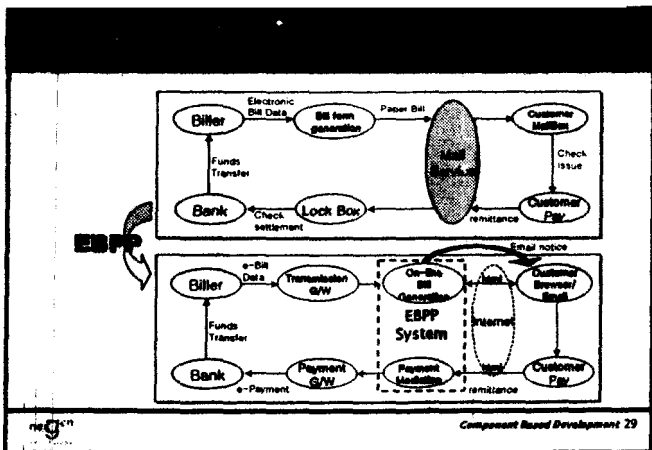
- Structural, completeness and plausibility verifiers
- Pattern integrity and completeness verifiers
- Modal verifiers for Pattern,UML,EJB and Java integrity
- Technological feasibility verifiers : EJB container sensitive
- Structural completeness and optimized build integrity.
- Complete visual coverage(reflection) and dynamic testing

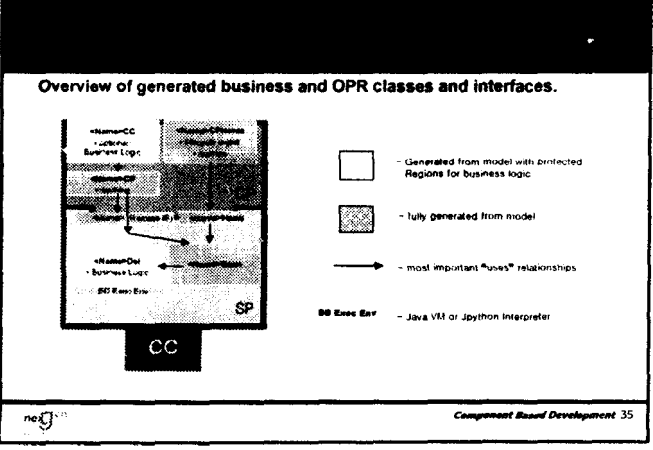
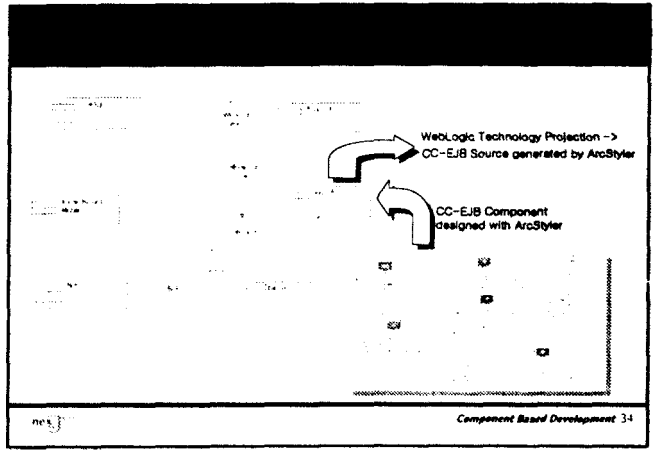
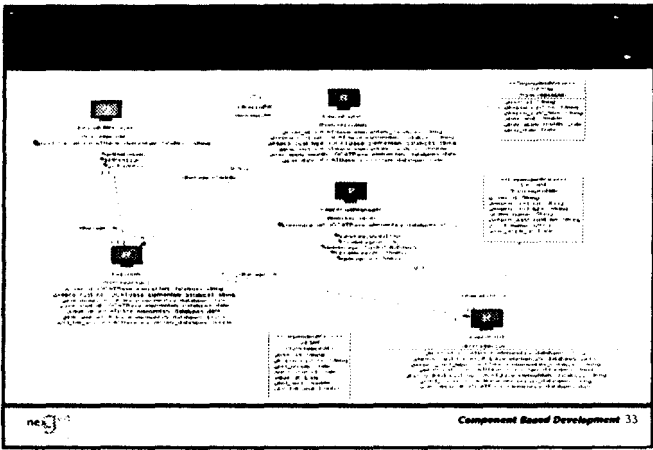
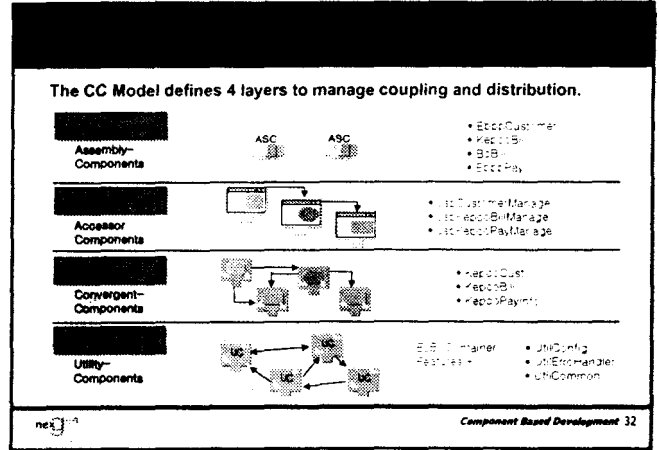
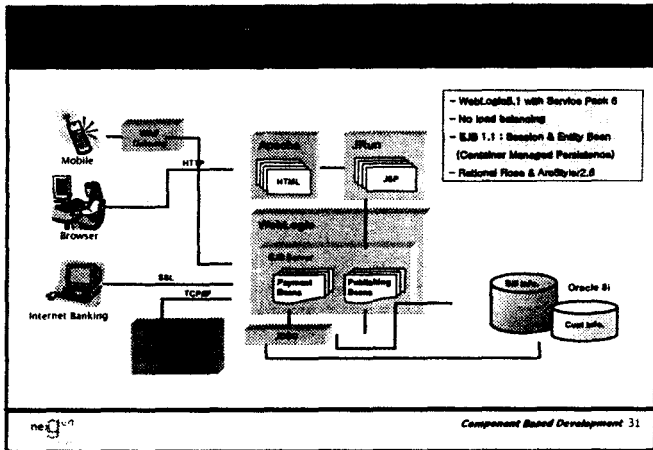
Component Based Development 25



Electronic Bill Presentment & Payment(EBPP) System

Component Based Development 28





Q&A

www.nexgentech.co.kr
hknv@nexgentech.co.kr