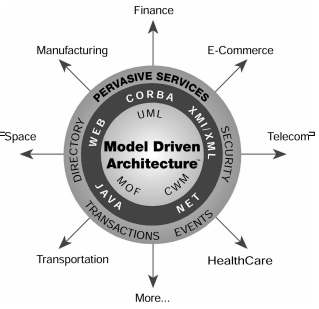
MDA Model driven architecture



No Universal Component Middleware

Distributed Software Components

Major Challenge!

CORBA	EJB	.NET	. . .

Challenge and solution

- Preserve investment
  - as enterprise borders shift requiring software, applications and platforms to change
  - as platforms change and new technology is adopted
    - EJB 1.1 -> EJB 2.0; CORBA 2 -> 3; webservices
- Isolate information and processing logic from technology specifics
  - Build platform-independent models
  - Map these models to specific platforms
  - Maintain the separation at the implementation level

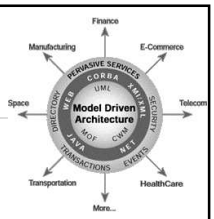
Enterprise system integration trend

- "middleware" addresses integration within the enterprise with a distributed computer
- today aim for similar integration between enterprises
  - Avoid typing in from computer-printouts
- ebXML, .NET, RosettaNet and other Web Services initiatives aim to be "the" middleware to link enterprises
  - How to cope with them all?

Program generation from models

- MDA is about using modeling languages as programming languages rather than merely as design languages
- can improve productivity, quality, and longetivity outlook
- new term, old invention
  - CASE-tools in 1980's, integrated object-oriented development environments in 1990's, UML tools, etc.

Model Driven Architecture



- New orientation for OMG activities
  - New step beyond the Object Management Architecture (OMA)
- Models are centric!
  - Target middleware is not important!
- Focus on Platform Independent Models (PIM)
  - Without middleware details
- Abstract Platform Specific Models (PSM)
  - Including all middleware details
- Define PIM to PSM transformations
- Preserving PIM when new middleware appears!

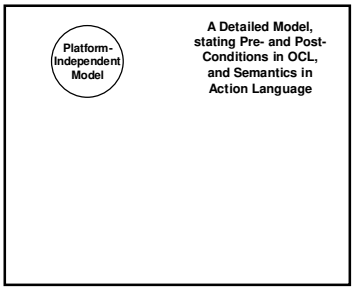
Some Key Terms

- Model
  - A formal specification of the function, structure and/or behavior of a system. (Model .ne. UML)
- Platform
  - Technological and engineering details that are irrelevant to the fundamental functionality of a software component.
- Platform Independent Model (PIM)
  - A formal specification of the structure and function of a system that abstracts away technical details (e.g., funds transfer)
- Platform Specific Model (PSM)
  - The technical details (CORBA, SOAP, EJB)

Basic modeling technology

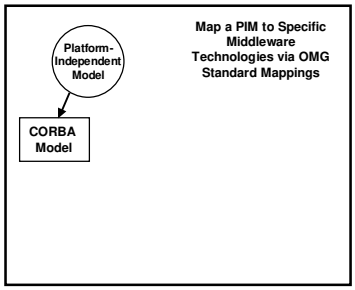
- UML – unified modeling language
- MOF – meta object facility
- CWM – common warehouse metamodel
- XMI – XML metadata interchange

Building an MDA Application



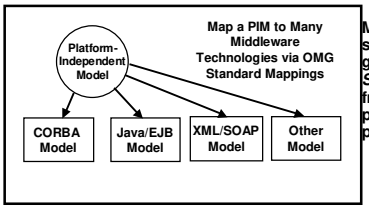
Start with a ***Platform-Independent Model*** (PIM) representing business functionality and behavior, undistorted by technology details.

Generating Platform-Specific Model



MDA tool applies a standard mapping to generate ***Platform-Specific Model*** (PSM) from the PIM. Code is partially automatic, partially hand-written.

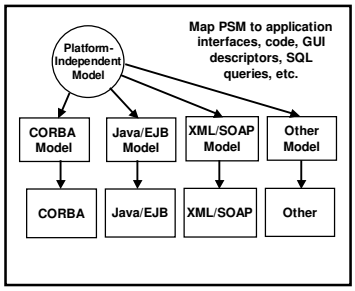
Mapping to Multiple Deployment Technologies



MDA tool applies an standard mapping to generate ***Platform-Specific Model*** (PSM) from the PIM. Code is partially automatic, partially hand-written.

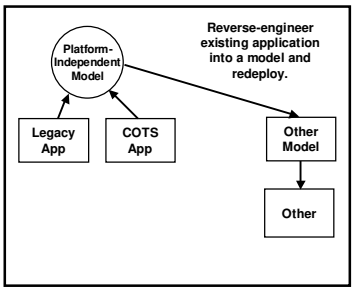
PSM can be viewed as
- presentation model (Web): data schemas, web comps, etc
- Business logic model (EJB): data schemas, key classes, etc
- Data models (DBMS): tables, columns, keys, etc

Generating Implementations



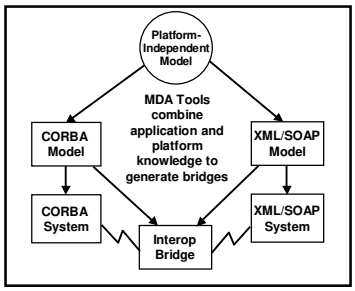
MDA Tool generates all or most of the implementation code for deployment technology selected by the developer.

Integrating Legacy & COTS



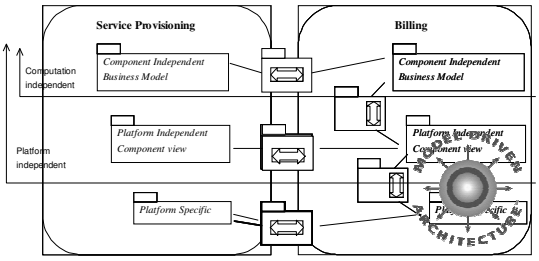
MDA Tools for reverse engineering automate discovery of models for re-integration on new platforms.

Automating Bridges

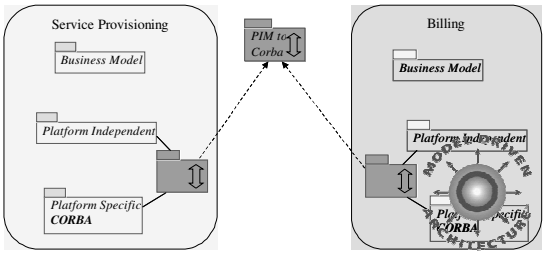


Bridge generation is simplified by common application models, simplifying creation of integrated applications both within and across enterprises.

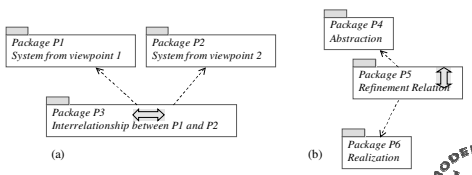
Concepts: Refinement



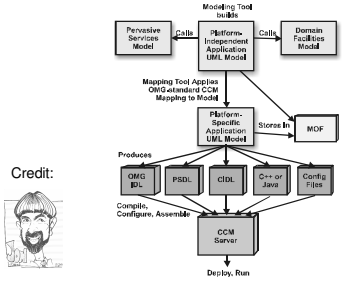
Concepts: Patterns



Concepts: Packages

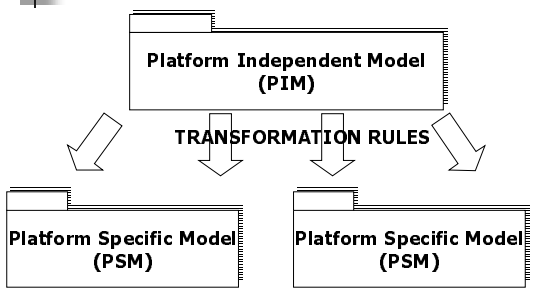


MDA Example

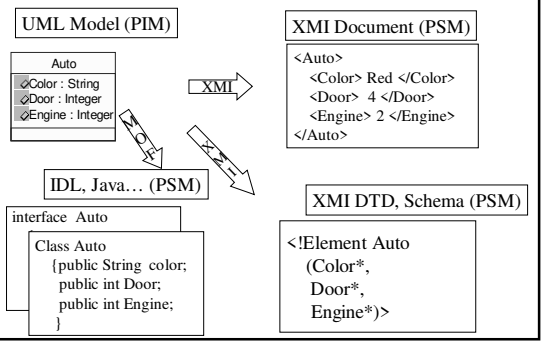


Credit:

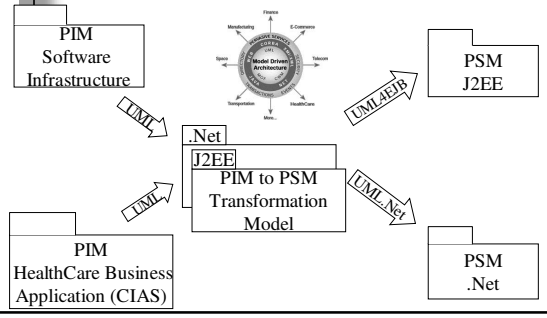
PIM, PSM, and Transformations in MDA



PIM to PSM Transformation Examples with MOF/XMI



Transformation Models in MDA



Why New Component Models?

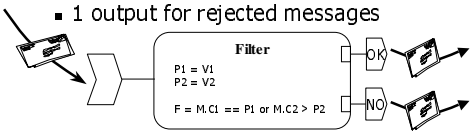
- Often application designers want to use concepts not present in PSM
  - e.g. CCM has no active JavaBeans properties

1) Creating a new PSM but also requires to develop
   the associated middleware
   - Very expensive and time consuming!!!

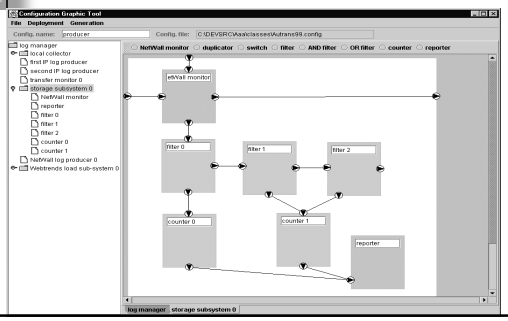
2) Creating a new PIM and defining PIM to PSM transformation rules

The Message Filtering Example

- A filter has
  - 1 input for receiving messages to filter
  - Configurable filtering properties
  - 1 filtering function
  - 1 output for accepted messages
  - 1 output for rejected messages



Application = Assembling Filters



Problems

- All must be done manually!
  - Defining OMG IDL for message eventtypes, filter component and home types
  - Implementing CCM eventtypes, components and homes
  - Writing CCM XML descriptors
- Extremely verbose, time consuming and error prone!!!
- How improving productivity and quality?
- How dealing with thousand messages and filters?
- How dealing with other Message Oriented Middleware (MOM)?
  - Like EJB 2.0 message beans

Next Step on the Message Filtering Meta-Model

- Only allow definition of message and filter types
- Only generate OMG IDL types, implementations, and CCM XML packaging descriptors
- Complete the meta model to describe filter instances and connections between them
- Then allow generation of CCM XML assembling descriptors for automatic deployment

Drawback around MDA-oriented Tools

- Few MOF compliant tools to experiment the creation of new component meta models!
- Developing UML profiles are time and money consuming
  - UML Profile Builders are still expensive and proprietary
- No portability of developed UML profiles between UML Tools!!!
- Few formalisms to express MDA transformation
  - OMG RFP under work
- Few PSM meta models for
  - Programming languages like Java, C++, …
  - Component standard middleware

Kirjallisuutta

- Siegel, Jon, Developing in OMG's Model-Driven Architecture. Object Management Group White Paper, Nov 2001.ftp://ftp.omg.org/pub/docs/omg/01-12-01.pdf (pakkollinen)
- OMG Architecture Board, Model Driven Architecture (MDA), Jul 2001, ORMSC/2001-07-01 http://cs.ua.edu/630/Resources/MDA%20White%20Papers/MDA%20-%20Technical%20Perspective%20-%20OMG%20Board%20-%20ormsc-01-07-01.pdf (täydentävä)

Varhaisia välineitä

- iUML, ICCG www.kc.com
- Adaptive framework www.adaptive.com
- www.kabira.com
- ArchStyler (in Borland's Enterprise studio 2) www.io-software.com