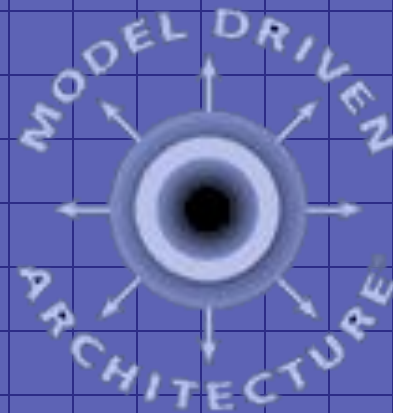
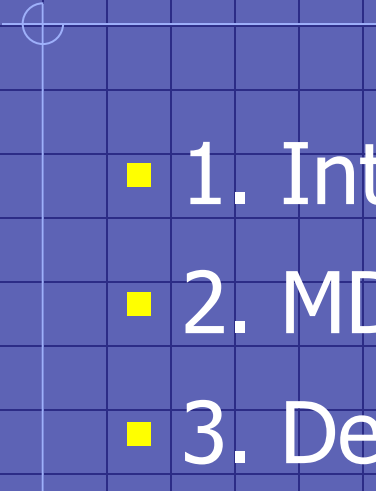


# OMG's Model Driven Architecture



- 
- 1. Introduction
  - 2. MDA Overview
  - 3. Developing in MDA
  - 4. MDA in the Practice



# 1. Introduction

# A heterogeneous world...

- Programming languages: *(see also: the DoD survey)*
  - ~3 million COBOL Programmers
  - ~1.6 million VB Programmers
  - ~1.1 million C/C++ Programmers
- Operating systems:
  - Unix, MacOS, Windows(3.1->XP), PalmOS
  - Embedded devices
- Networks:
  - Ethernet, IP, USB, FireWire
  - Bluetooth, 802.11b, HomeRF

# Where can we agree?

- Heterogeneity hinders the development of enterprise distributed systems
- There will not be consensus on
  - Hardware
  - Operating systems
  - Network protocols
  - Programming languages
- We can agree at a higher level
  - Middleware

# Middleware



- A software layer that masks heterogeneity
- Placed between operating systems and application components

# Middleware proliferation

- Middleware itself has proliferated:
  - CORBA
  - COM / DCOM / MTS(Microsoft Transaction Server)
  - Java / EJB
  - XML / SOAP
  - C# / .NET
- None of them prevails over the others
- The problem remains

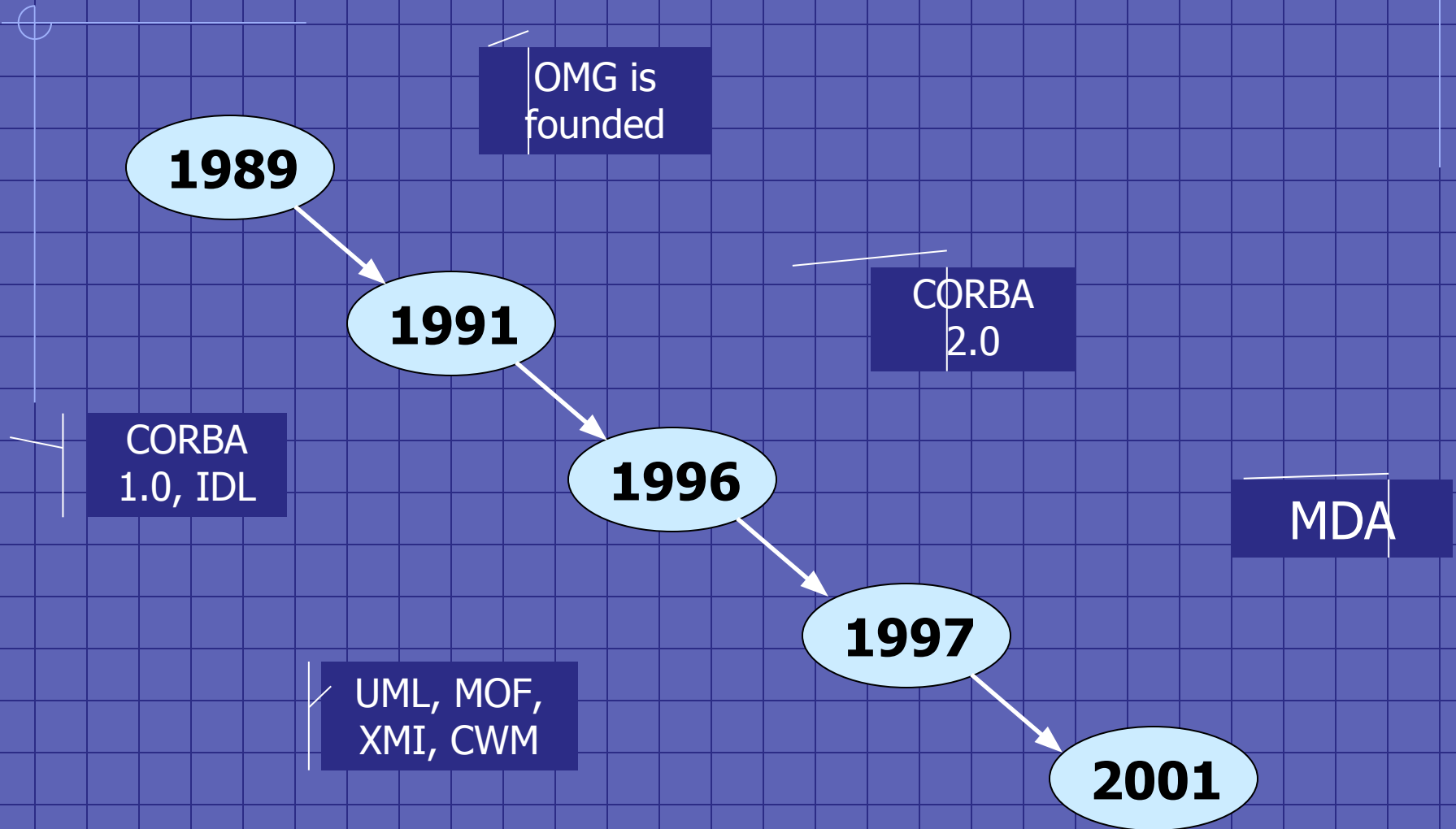
# The Object Management Group (OMG)

- An open membership and no-profit consortium
- Produces and maintains computer industry specifications for interoperable enterprise applications

Who are OMG?



# OMG's Milestones



The background is a solid blue color with a white grid pattern. There are several white lines and corner crop marks. A vertical line is on the left side, and a horizontal line is near the top. Another vertical line is on the right side, and a horizontal line is near the bottom. There are small white circles at the corners where these lines meet, resembling crop marks.

## 2. MDA Overview

# What is the MDA?

- An approach to IT system specification that separates the specification of system functionality from the specification of the implementation of *that* functionality on a particular technology platform
- “Design once, build it on any platform”

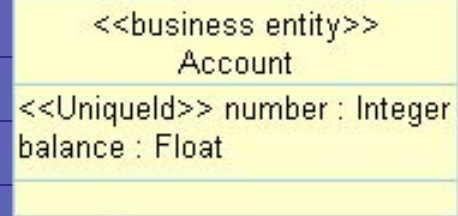
# Basic concepts of MDA

- A *model* is a formal specification of the function, structure and/or behaviour of a system
  - Examples:
    - Source code is a model
    - An UML-based specification is a model
- Models of different systems are structured explicitly into:
  - Platform Independent Models (PIM)
  - Platform Specific Models (PSM)

# Platform Independent Model (PIM)

- A “formal” specification of the structure and function of a system that abstracts away technical detail
- Expressed using UML

# PIM: an example



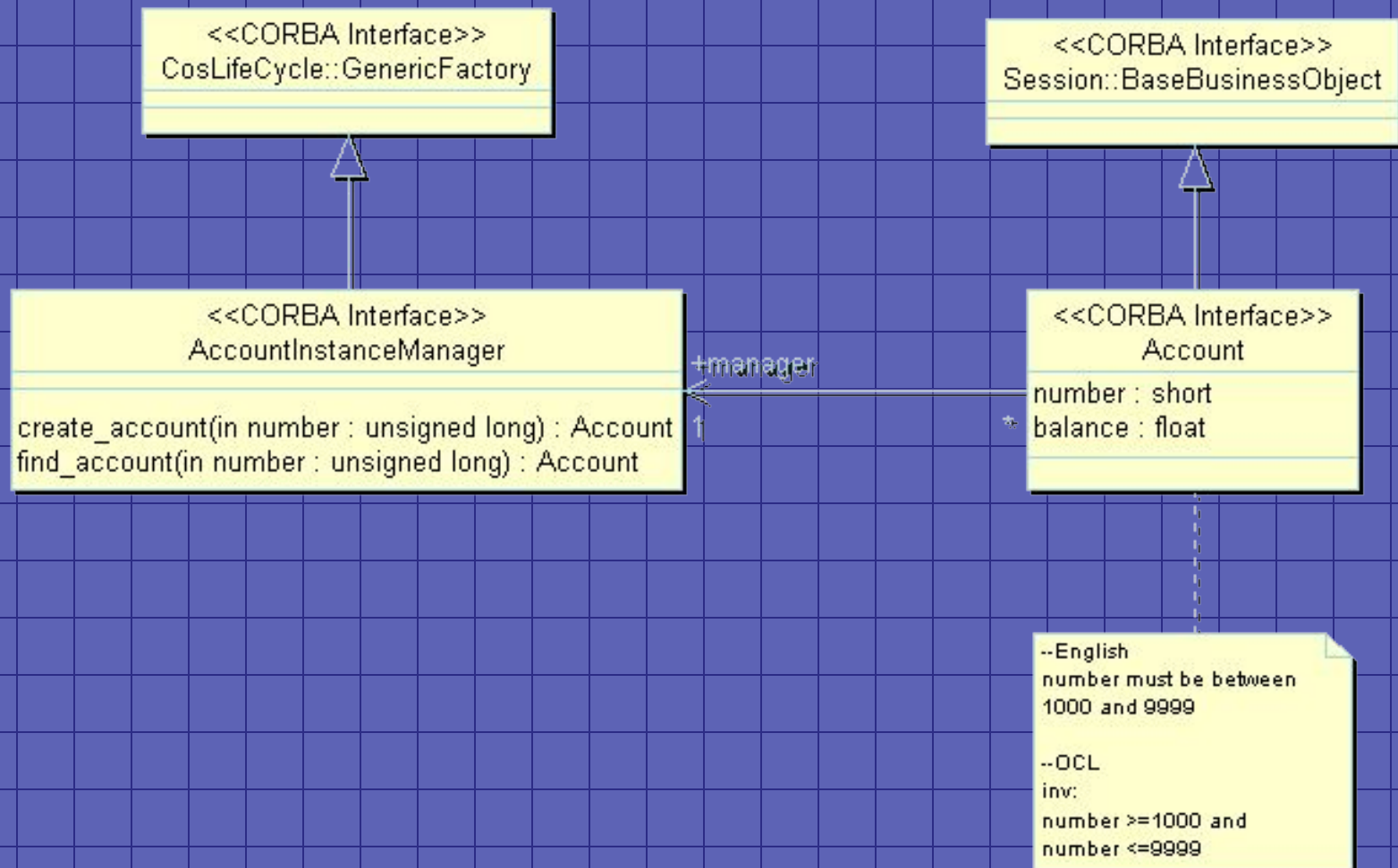
--English  
number must be between  
1000 and 9999

--OCL  
inv:  
number >= 1000  
and  
number <= 9999

# Platform Specific Model (PSM)

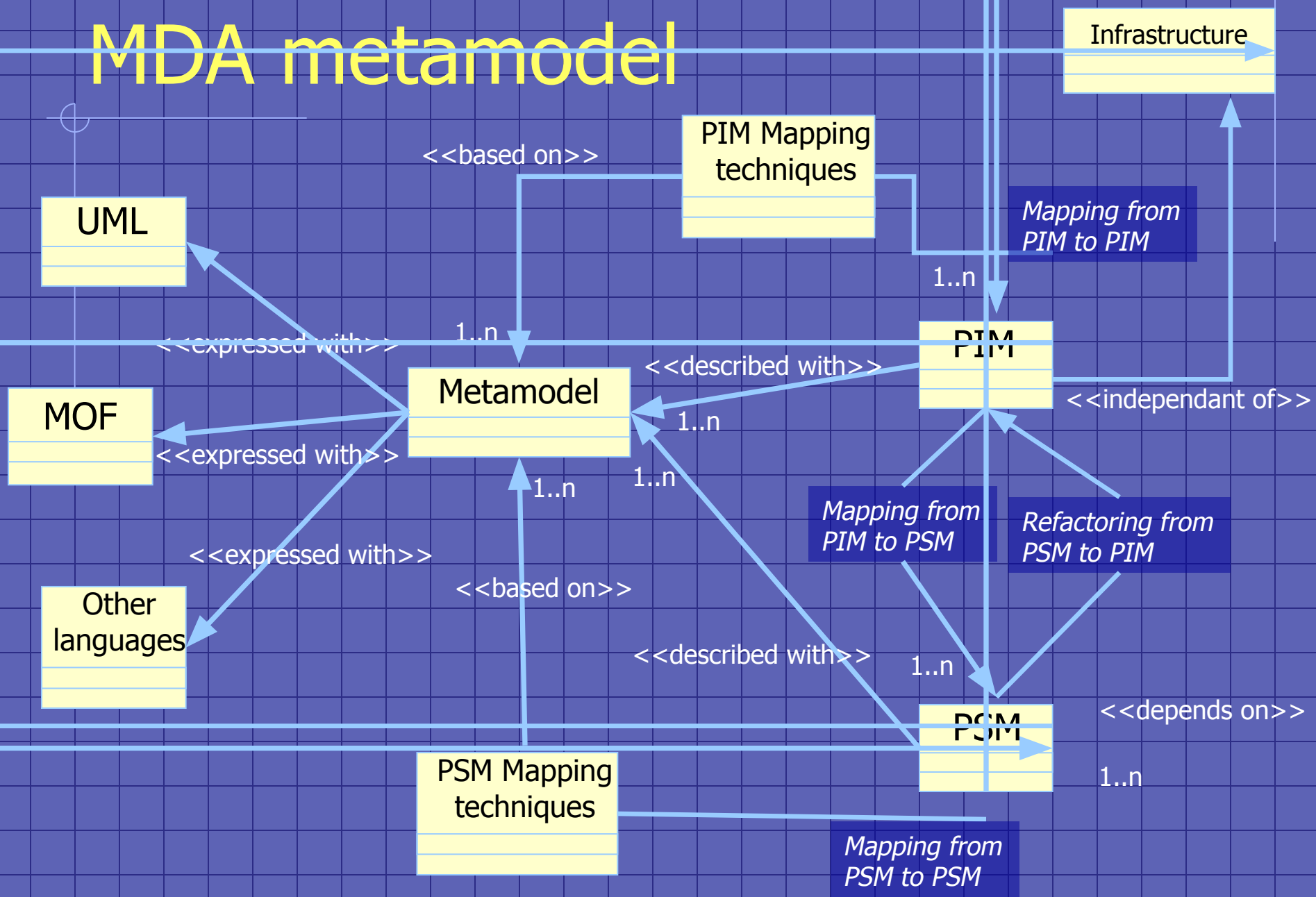
- Specifies how the functionality specified in a PIM is realized on a particular platform
- Expressed using UML extended with platform specific UML profiles

# PSM: an example

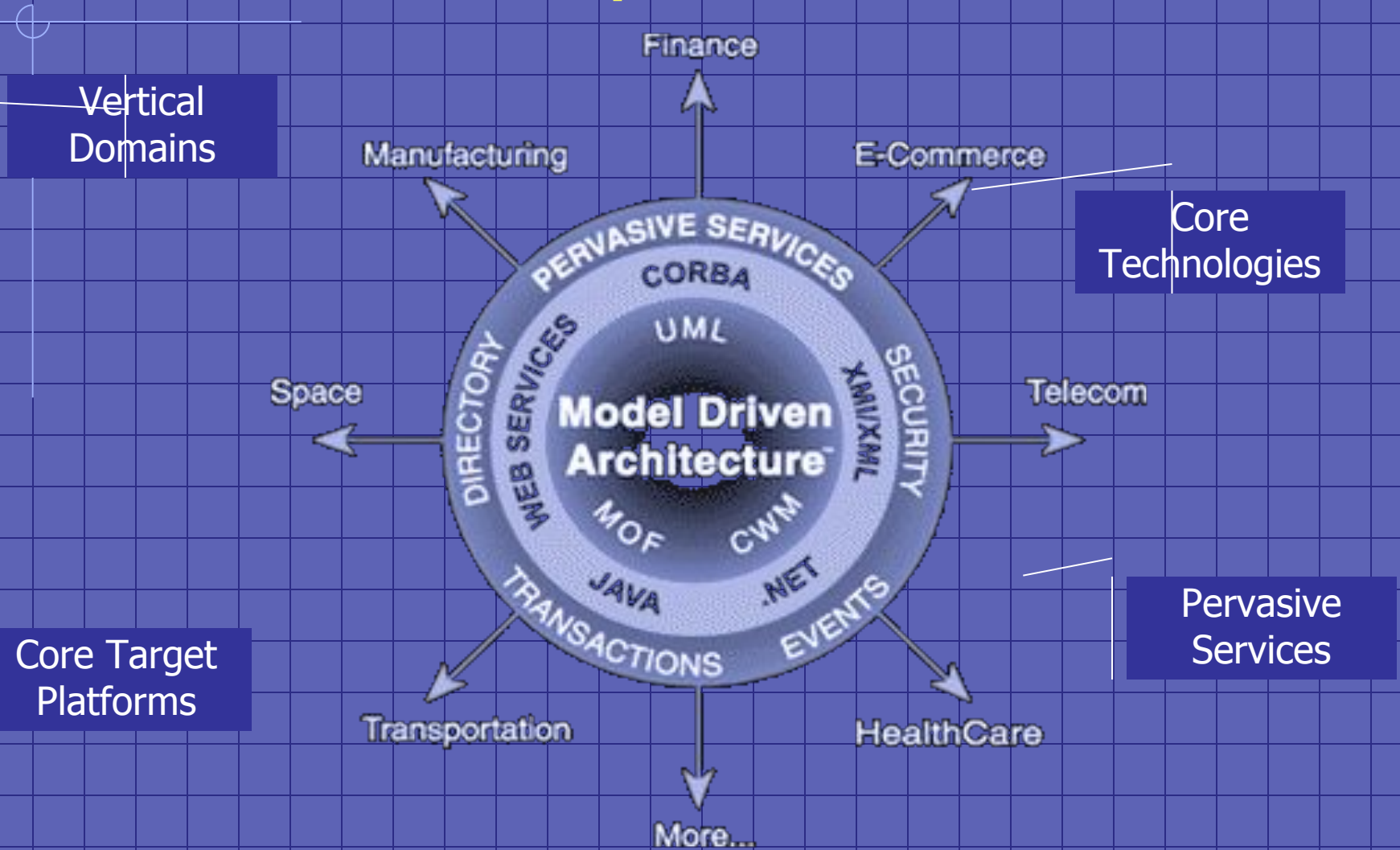




# MDA metamodel

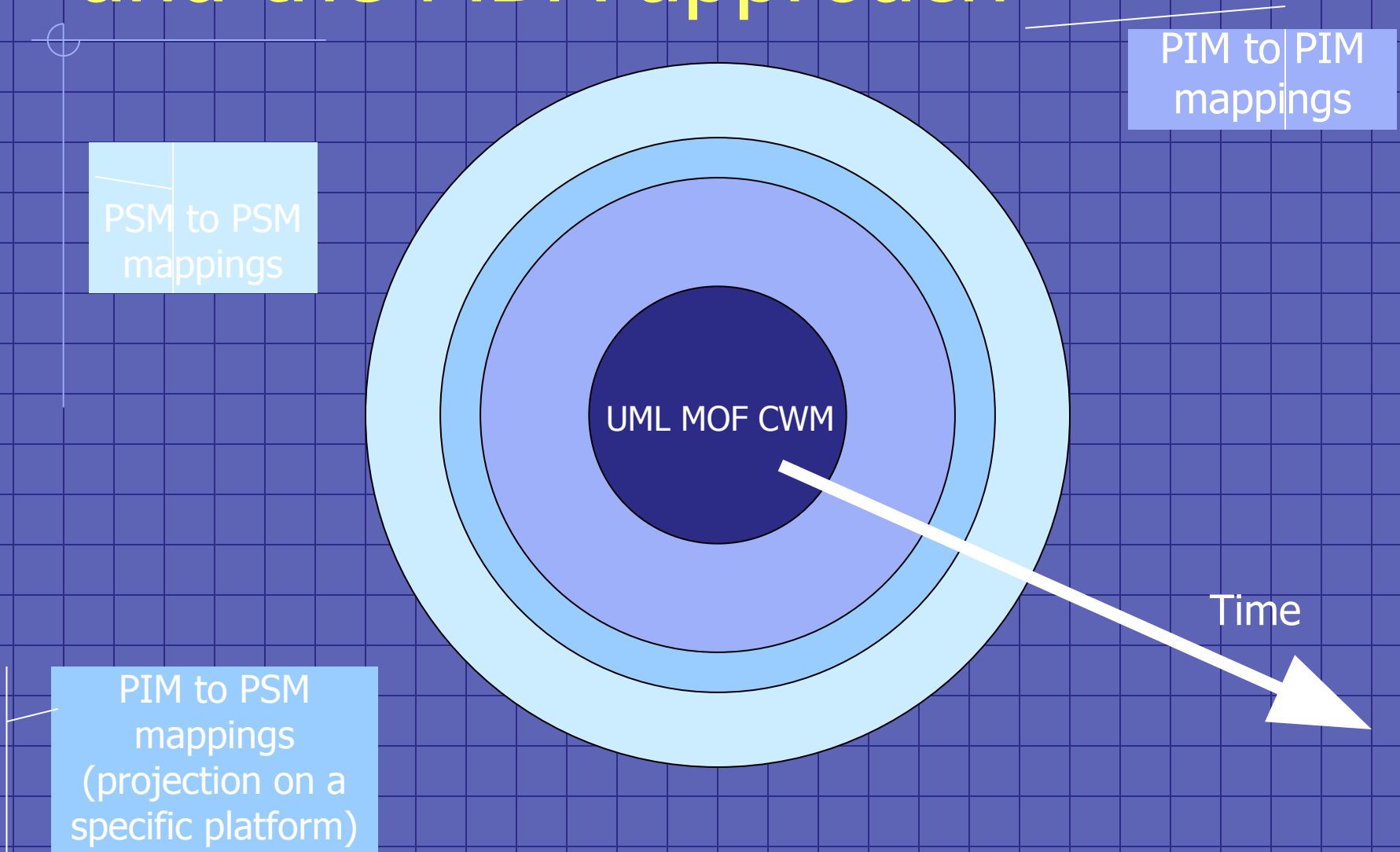


# MDA in a Snapshot



# 3. Developing in MDA

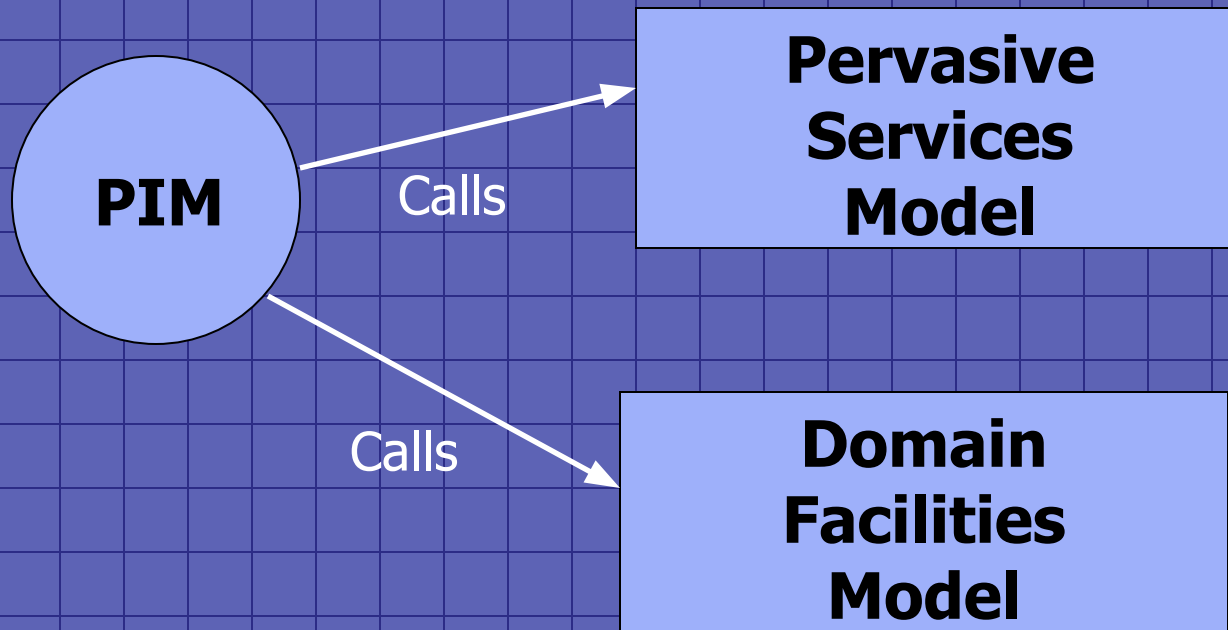
# System Development Lifecycle and the MDA approach



# Developing in MDA – Step 1: the PIM (1/2)

- All MDA development projects start with the creation of a PIM
- PIM at this level represents business functionality and behaviour, undistorted by technology details
- MDA application-modeling tools contain representations of Pervasive Services and Domain Facilities allowing them to be used and/or incorporated in the application via a menu selection

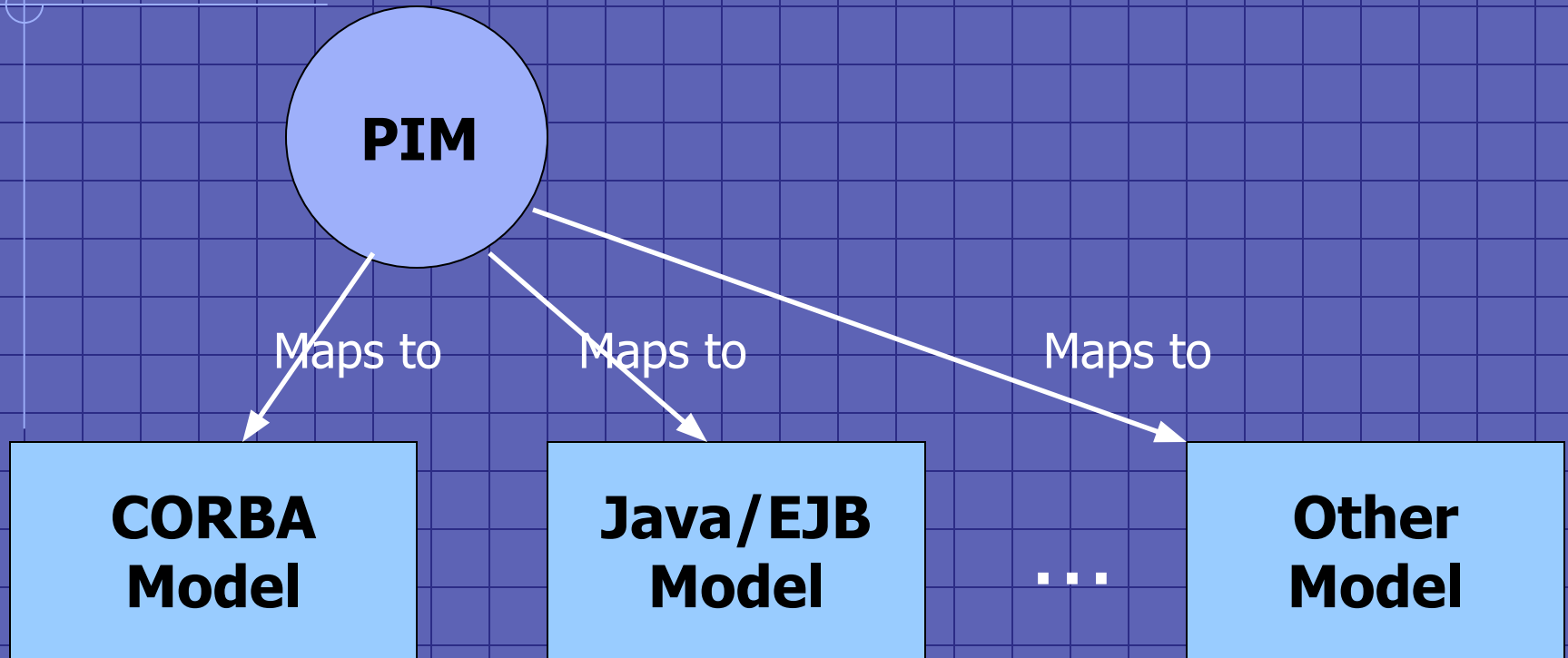
# Developing in MDA – Step 1: the PIM (2/2)



# Developing in MDA – Step 2: the PSM (1/2)

- Once the first iteration is complete, the PIM is input to the mapping step which will produce a PSM
- Code is partially automatic and partially hand-written
- PIM can be mapped either to a single platform or to multiple platforms

# Developing in MDA – Step 2: the PSM (2/2)

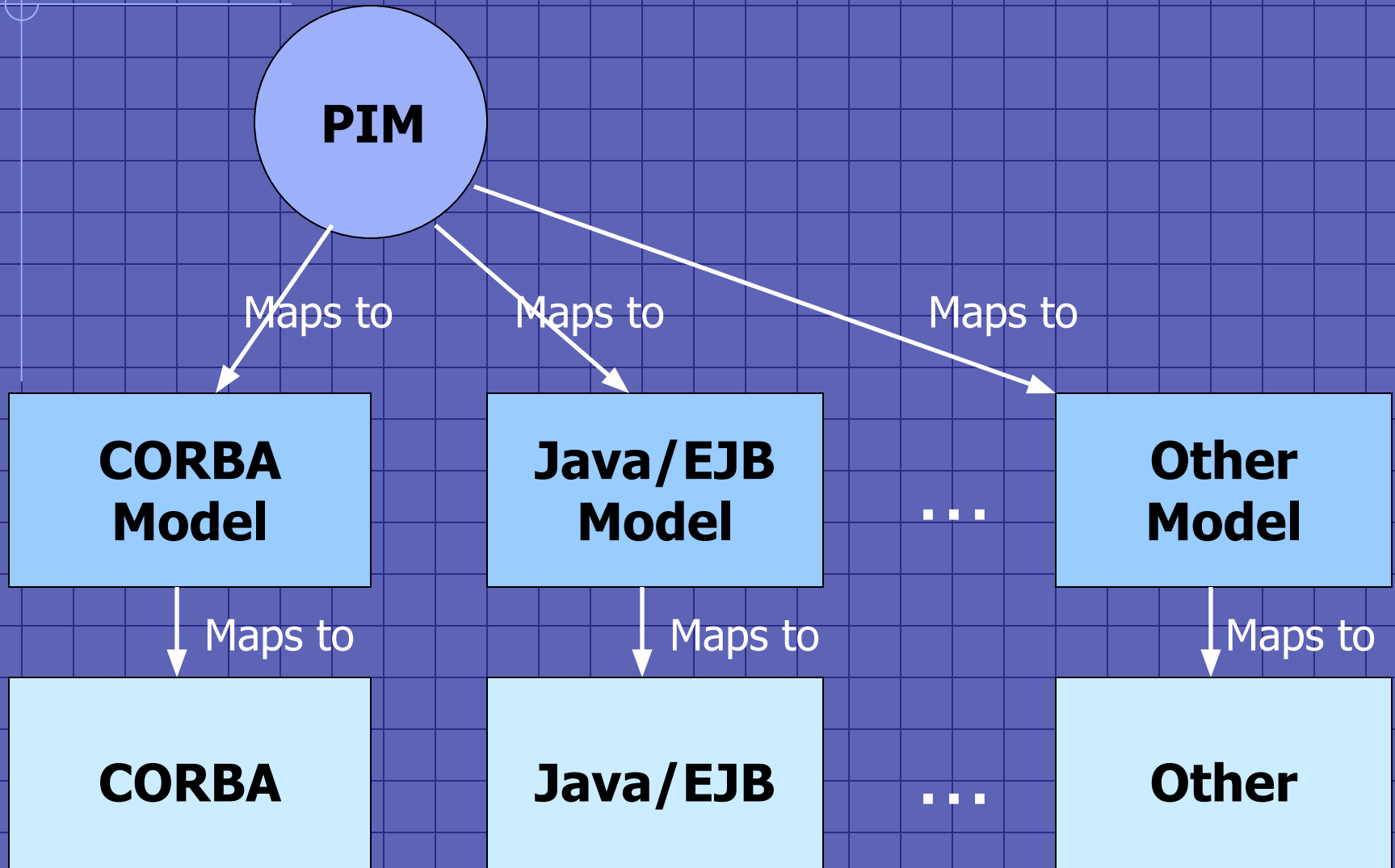




# Developing in MDA – Step 3: Generating Application (1/2)

- An MDA tool generates all or most of the implementation code for the deployment technology selected by the developer
- Re-integration on new platforms can be done by reverse engineering the existing application into a model and redeploy

# Developing in MDA – Step 3: Generating Application (2/2)



The background is a solid blue color with a fine white grid pattern. There are several white lines and corner crop marks. A vertical line is on the left side, and a horizontal line is near the top. Another vertical line is on the right side, and another horizontal line is near the bottom. Small white quarter-circle marks are at the intersections of these lines, indicating where the image was cropped.

## 4. MDA in the Practice

# MDA Adoption Status

- Adoption of MDA is at an early stage
- UML profiles underway:
  - CORBA (adopted)
  - EJB (adopted)
  - SOAP/XML (in process)
  - .NET (to be started)
- UML 2.0 in process

# Future Trends

- “A market for MDA will be created as OO modeling and development tool vendors incorporate MDA into their offerings”
- “It will be two or three years before mature MDA tools emerge”
- “This is the most exciting enterprise software initiative I’ve seen since UML”

(Paul Harmon, Senior consultant and Market Analyst, CUTTER consortium, 2001)

# Early Adopters

- Interactive Objects Software GmbH
  - [ArcStyler](#)
- Kennedy Carter
  - [iUML, iCCG](#)
- Kabira
  - Adaptive Realtime Infrastructure
- Secant technologies
  - Model-Driven Infrastructure
- Sun's NetBeans was added to this list on May 7th 2002
  - ...stay tuned!

# Conclusions (1/2)

- Abstracting out the structure and behaviour of a system in the PIM makes easier:
  - To validate the correctness of the model
  - To produce implementations on different platforms
  - The application of tool supported solutions

# Conclusions (2/2)

- The major drawback is that MDA does not provide a standard for the specification of mappings
  - Different implementation of mappings can generate very different code or models
  - This can create dependencies between the software and the mapping solution used



# Platform



- Technological and engineering details that are irrelevant to the fundamental functionality of a software component



# UML Profile

- A standardized set of extensions (stereotypes and tagged values)
- Defines an UML environment tailored to a particular use, such as modeling for a specific platform
  - UML profile for CORBA was standardized in 2000



# MOF (Meta Object Facility)

- OMG's standard for defining metamodels
- Provides standard modeling and interchange constructs used in MDA
- UML are defined in terms of MOF constructs
- The three main metadata modeling constructs provided by the MOF are:
  - Class
  - Association
  - Package



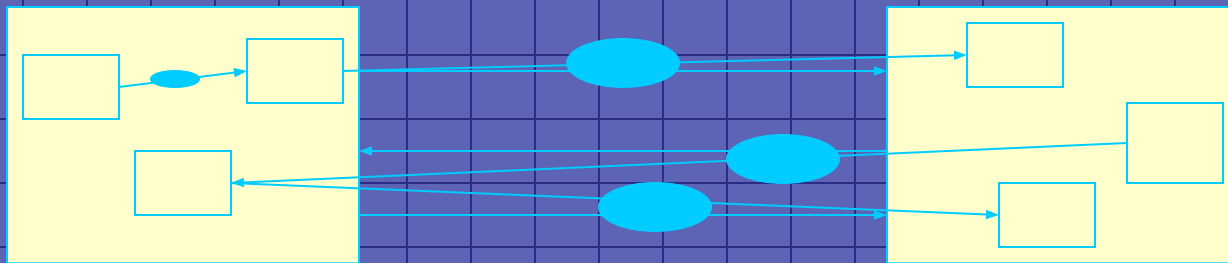
# Mappings

- *Mapping* is defined as a set of rules and techniques used to modify one model in order to get another model
- Mappings are described using UML



# Refining

- Makes a model less abstract
- Permits “zooming” in (and out of) a model:



# PIM to PIM mapping

- Used when models are enhanced, refined or filtered during the development lifecycle without needing any platform dependant information
- Example: analysis to design models transformation



# PIM to PSM mapping (1/3)

- This transformation is used when the PIM is projected to the execution infrastructure
- Projection is based on platform characteristics
- These characteristics should be described using a UML description (and eventually a profile for describing common platform concepts)
- There are multiple ways to transform a PIM into a corresponding PSM

*More on mapping*

# PIM to PSM mapping: techniques (2/3)

- A human could study the PIM and:
  - Manually construct a platform-specific model
  - Utilize models of known refinement patterns to reduce the burden in constructing the PSM



# PIM to PSM mapping: techniques (3/3)

- An algorithm could:
  - Be applied to the PIM and create a skeleton of the PSM to be manually enhanced by hand
  - Create a complete platform specific model from a complete platform independent model, explicitly or implicitly recording the relation for use by other automated tools



# PSM to PSM mapping

- This transformation is needed for component realization and deployment
- Generally related to platform dependent model refinement



# PSM to PIM refactoring

- This transformation is required for abstracting models of existing implementations in a particular technology into a platform independent model
- Represents a “mining” process hard to be fully automated
- Result of this transformation will match the corresponding PIM to PSM mapping



# Infrastructure

- A set of software or hardware pieces assumed to be already present by some stakeholders when he/she develops a software artifact



# Core models

- The core target platforms of the MDA:
  - CORBA
  - Java/EJB
  - C#/.NET
  - XML/SOAP
- MDA provides UML profiles for these target platforms



# Pervasive Services

- Essential services for many applications
  - transactions
  - directory services
  - security
  - persistence
  - others...
- OMG will define them at PIM level



# Some of the OMG members:

- AT&T
- BEA
- Borland
- Boeing
- CA
- Citigroup
- Compaq
- Ericsson
- Ford
- Fujitsu
- Glaxo
- HP
- Hitachi
- Hyperion
- IBM
- IONA
- io Software
- Kabira
- KC
- John Deere
- Microsoft
- MITRE
- MSC Soft
- NASA
- NEC
- NetGenics
- NTT
- OASIS
- Oracle
- Pfizer
- Rational
- SAGA
- SAP
- SAS
- Secant
- Siemens
- Sprint
- Sun
- Unisys
- Vertel



# IoSoftware's ArcStyler

Rational Unified Process

## Arc Styler Core Modules

Business  
Object  
Modeler

Pattern  
Refinement  
Assistant

UML  
Refinement  
Assistant

MDA Cartridges

Generator  
Engine  
with Meta IDE

Build, Deploy  
and Test  
Support

### Optional Integrated Tools

IDS ARIS

Rational Rose

Java IDE

UML/XML (XMI) Repository

MDD™  
Technology Projection

J2EE/EJB,.NET

BEA WebLogic

IBM WAS NI, z/OS

Borland BES

IONA E2A

Custom Infrastr.

System Definition (Analysis, Design)

System Development

Verification / Test / Deployment



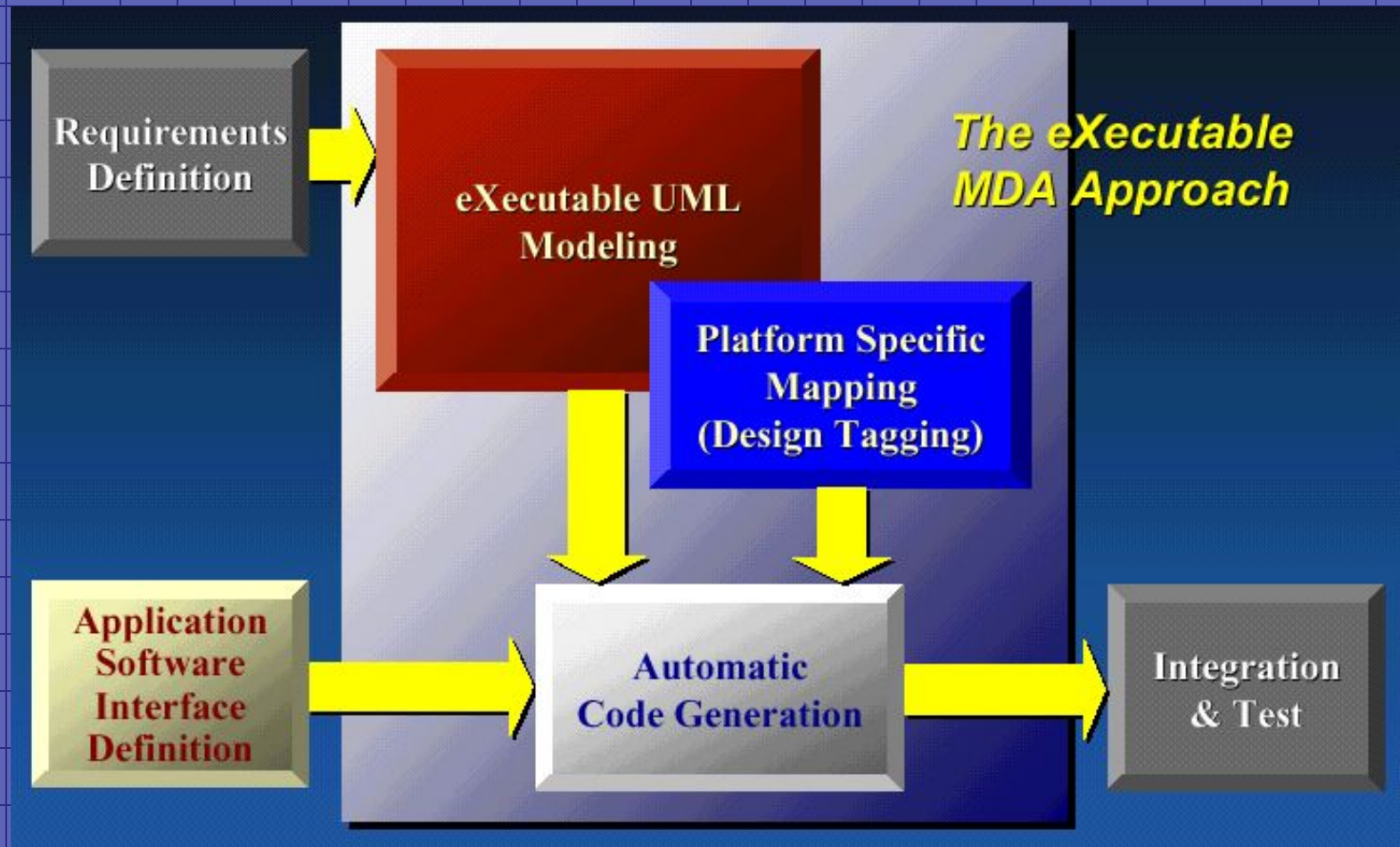
# ArcStyler Success Story

- Deutsche Bank Bauspar AG used ArcStyler to embed existing COBOL mainframe application into modern web-based system
- A customized Cartridge was created



# The KC's xMDA approach

xMDA: eXecutable UML + MDA



# xMDA Success story

- Lockheed Martin used MDA to develop the F-16 Modular Mission Computing software
- They used the KC's iUML tool and more recently iCCG (intelligent Configurable Code Generator) to specify an Ada code generator which can automatically generate 100% of the implementation

