



Methods for the Development Of Dependable and Adaptive Information Systems

Carolina Gomez Hernandez

Index of Contents

- History of Modeling
- Methods for the Development of DAIS:
 - Model Driven Architecture (MDA)
 - An MDA specific Application
 - Enterprise Knowledge Development Change Management Method (EKD-CMM)
- MDA vs. EKD-CMM
- Conclusions

History of Modeling

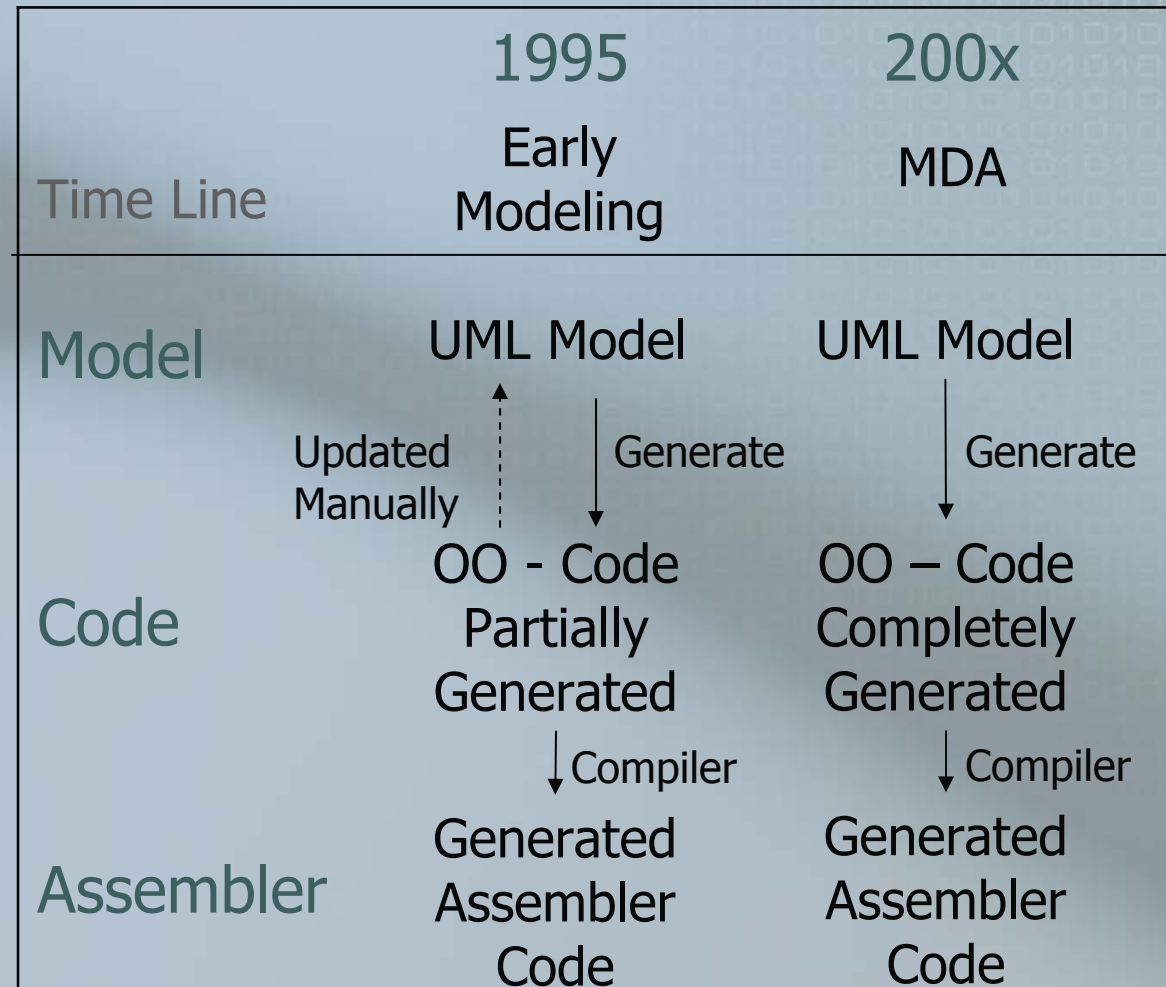
■ Modeling Evolution

Time Line	1950	1960	1980
Model	∅	∅	∅
Code	∅	High Level Programming (FORTRAN) ↓ Compiler	O-O Programming (JAVA) ↓ Compiler
Assembler	Assembler Program	Generated Assembler Code	Generated Assembler Code



History of Modeling

■ Modeling Evolution



History of Modeling

Modeling has evolved in order to:

- Handle the complexity of today's software systems
- Achieve a higher level of abstraction that allows a human to better understand a problem.

History of Modeling

- Complexity of today's software system:

FORMERLY

- Applications were developed by thinking of technical specifications of a specific technology.

TODAY

- The development of Information Systems involves a wide range of concepts including the implementation technology. This rising complexity requires a higher level of abstraction.

$2 + 3 = 5$

User Requirements
Business Processes
Domains

1950

1960

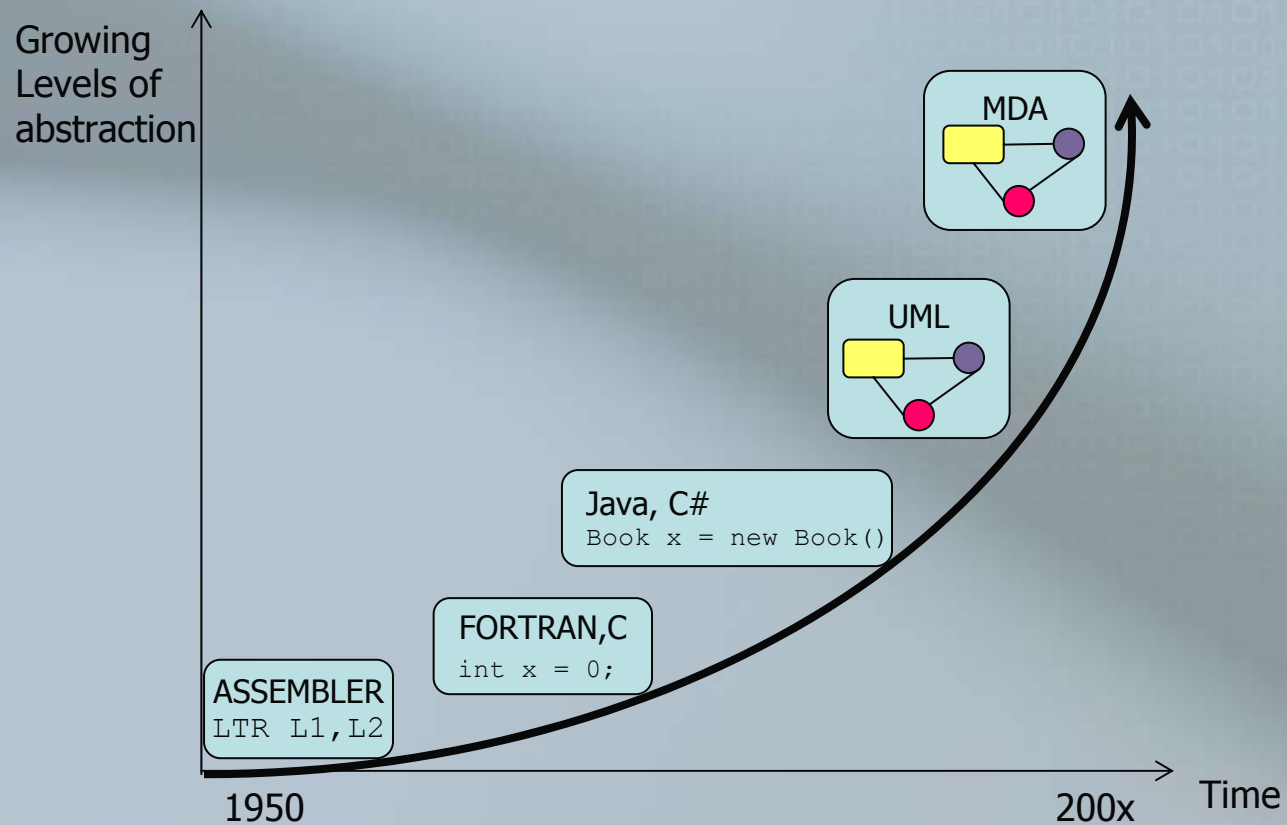
1980

1995

2000

History of Modeling

- A higher level of abstraction allows a human to better understand a problem.



Methods for the Development of DAIS

- Model Driven Architecture (MDA): For the development of an information system
- Enterprise Knowledge Development Change Management Method (EDK-CMM): Allow to choose an information system suitable for an enterprise

Model Driven Architecture (MDA)

DEFINITION

- Method used for developing software that uses models to specify the behavior and requirements of a system, independently from the platform in which the system is finally implemented.

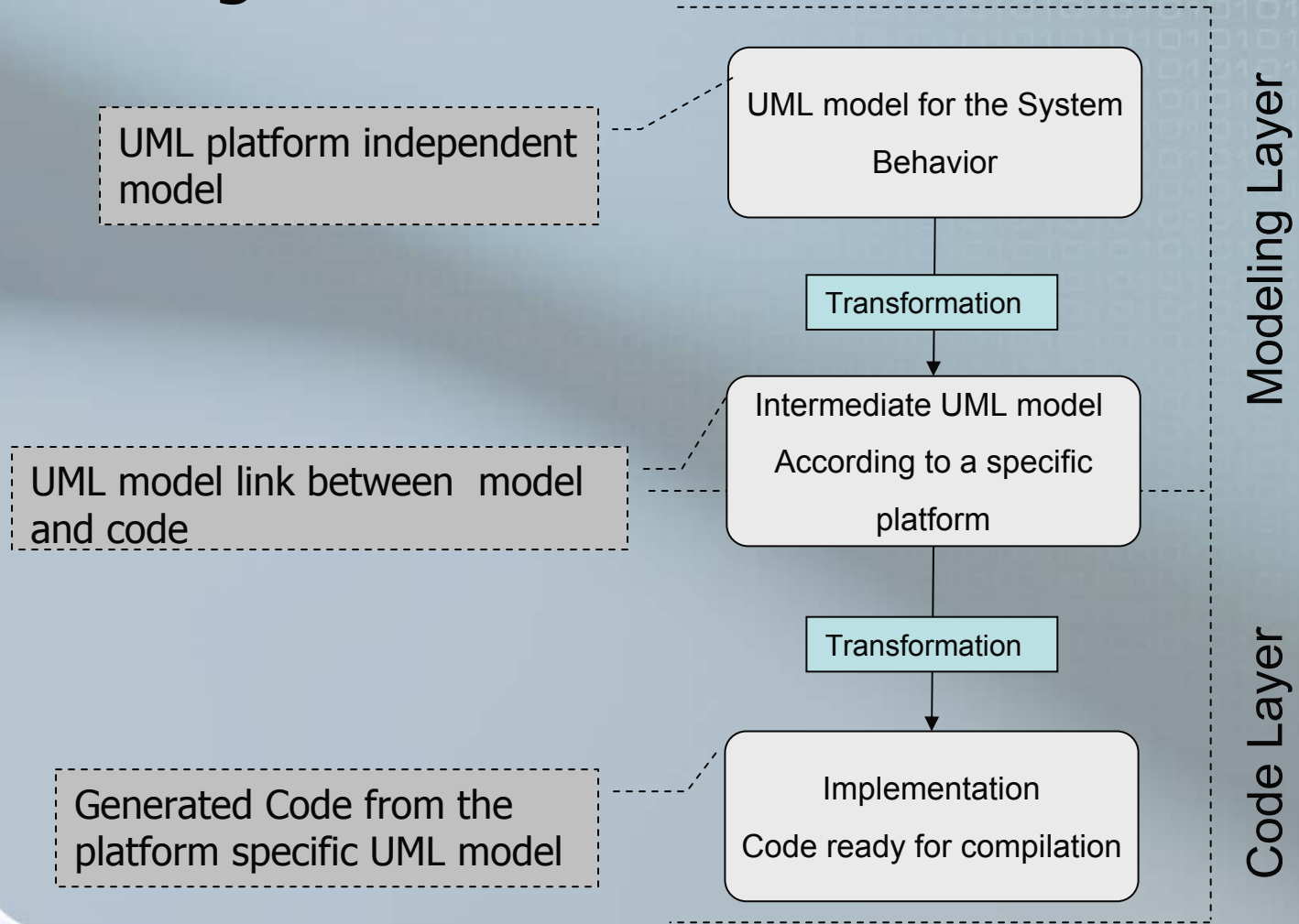
Model Driven Architecture (MDA)

GOALS

1. To make a distinction between the logic of business and the platform implementation.
2. To offer a standard to develop and create models.
3. To create code implementation directly from models.

Model Driven Architecture-MDA

1. Logic of Business vs. Platform

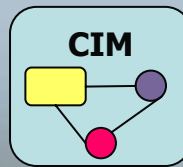


Model Driven Architecture (MDA)

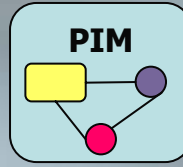
■ Models

Modeling Layer

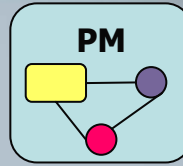
Computation Independent Model



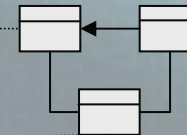
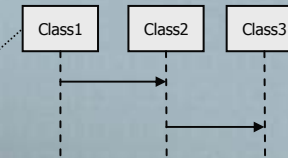
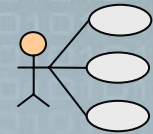
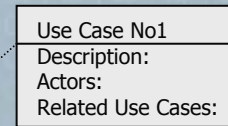
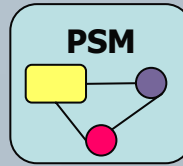
Platform Independent Model



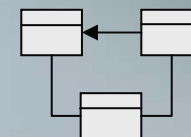
Platform Model



Platform Specific Model



UML Profile for .NET

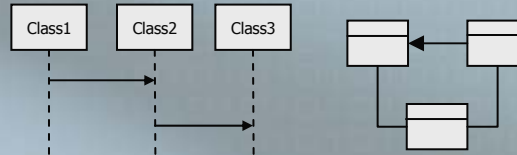
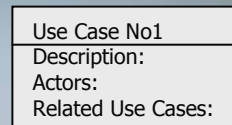
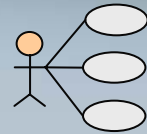


Model Driven Architecture (MDA)

2. Modeling Standards and Languages

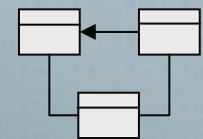
- Unified Modeling Language (UML) : Used to create graphical representation of systems requirements and behavior.

PIM



PSM

UML Profile for .NET



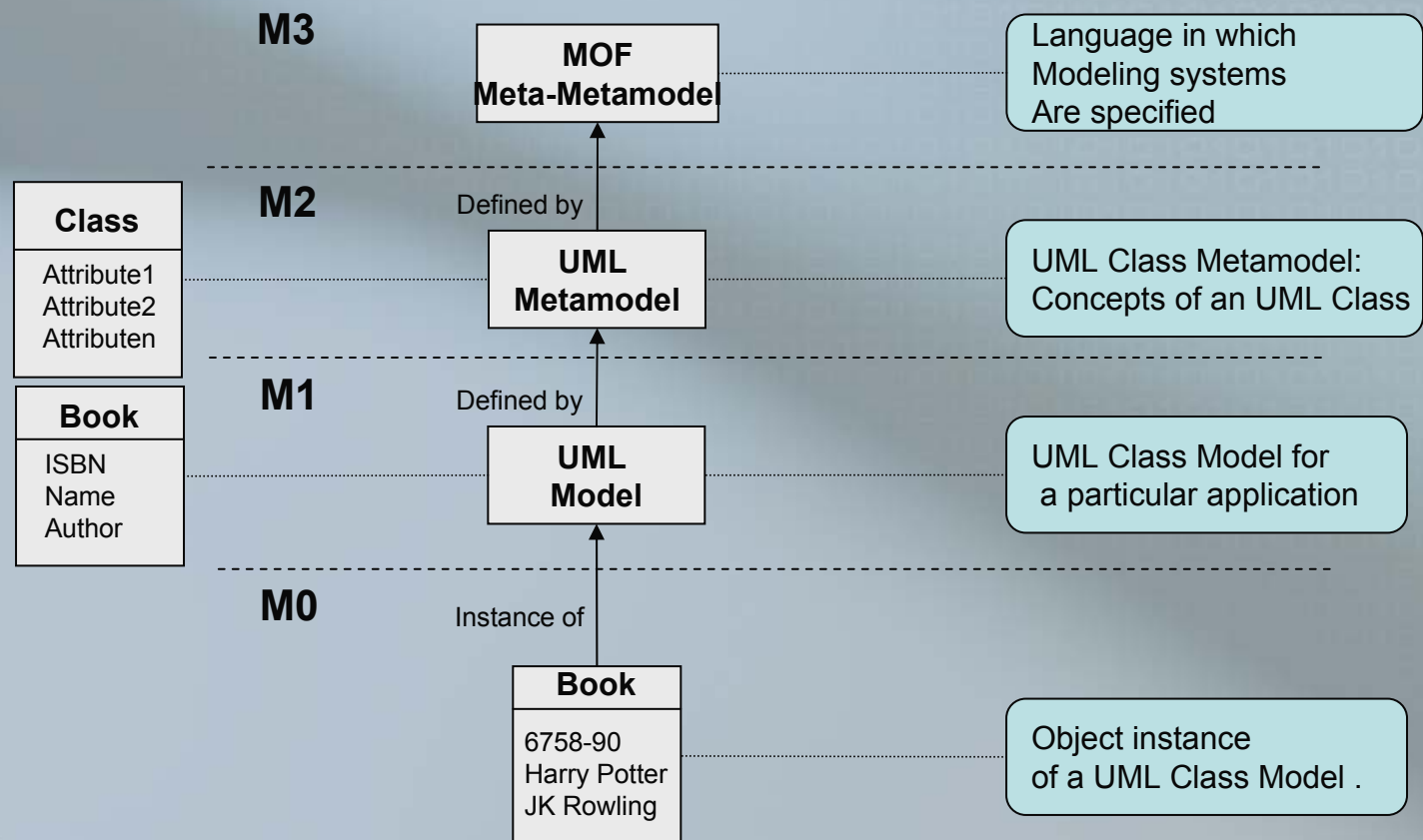
- XML Metadata Interchange (XMI): Used to integrate XML data and model objects



Model Driven Architecture (MDA)

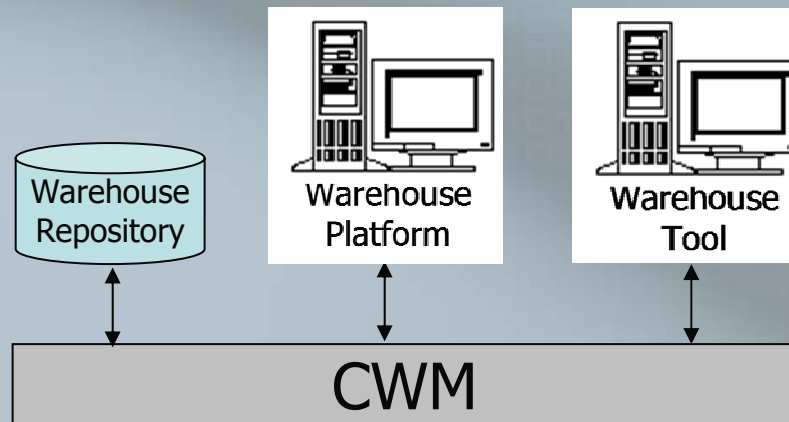
2. Modeling Standards and Languages

■ Meta Object Facility (MOF)



Model Driven Architecture (MDA)

2. Modeling Standards and Languages
 - Common Warehouse Metamodel (CWM):
Used to integrate warehouse metadata.

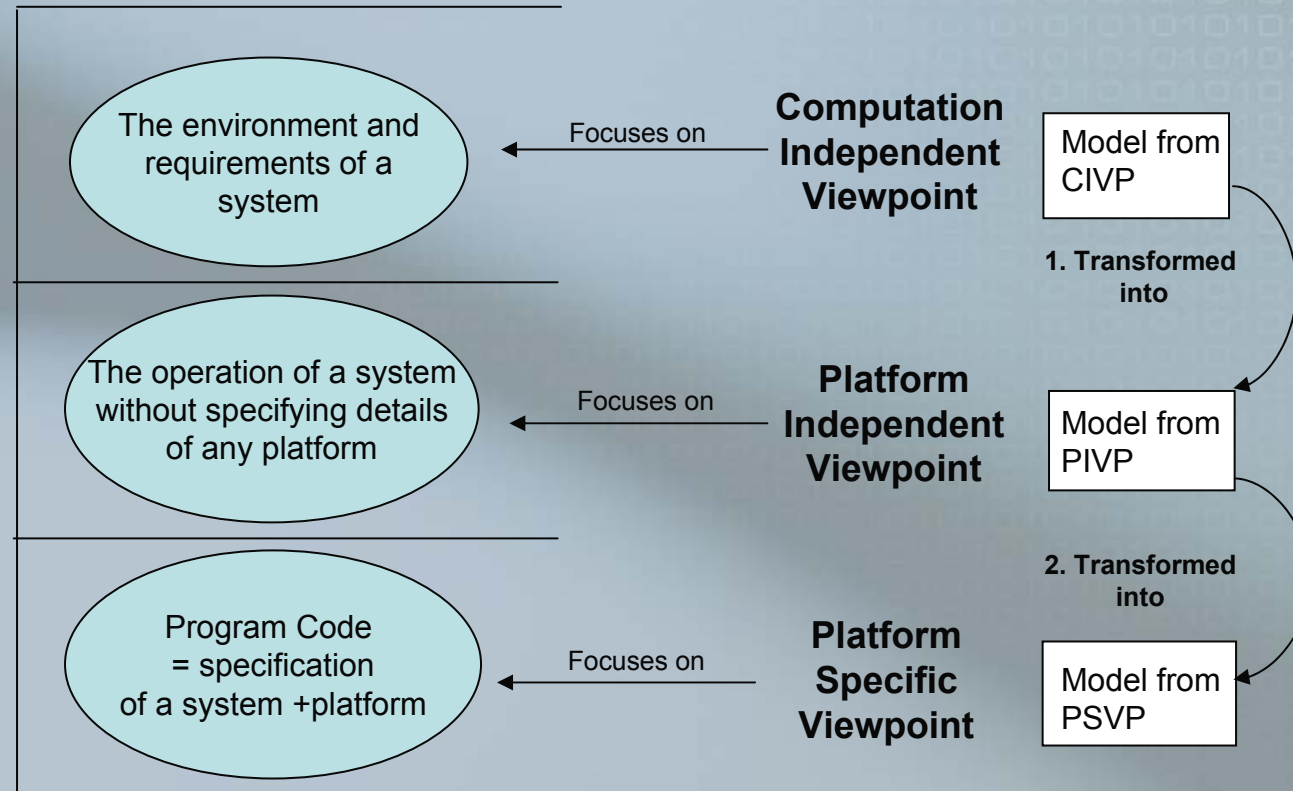


Model Driven Architecture (MDA)

3. Implementation code derived from models: MDA Transformation
 - Transformation in MDA is the process of converting one model into another of the same system.
 - Within a system models are classified in three different layers, called viewpoints.

Model Driven Architecture-MDA

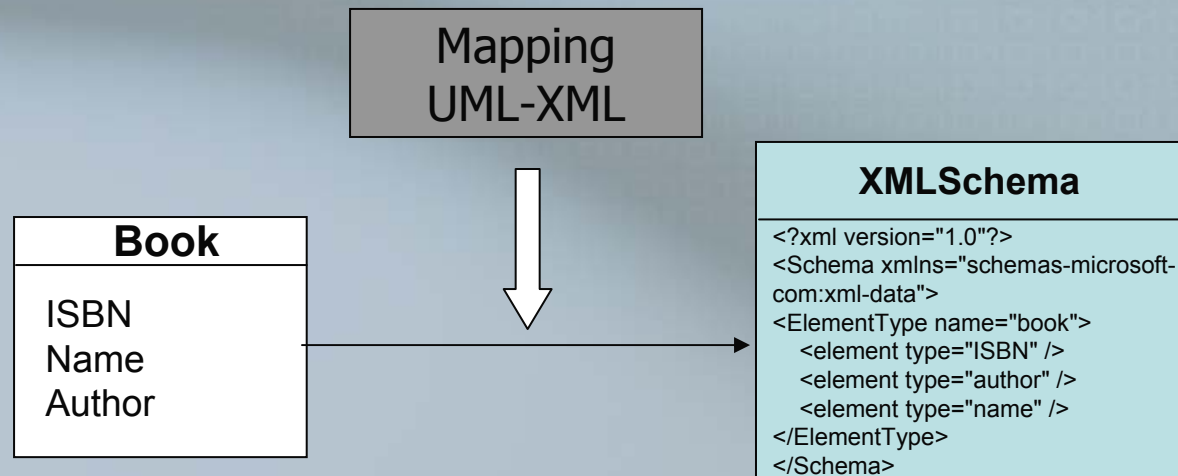
■ MDA Transformation: Viewpoints



Models are transformed from one to another viewpoint.

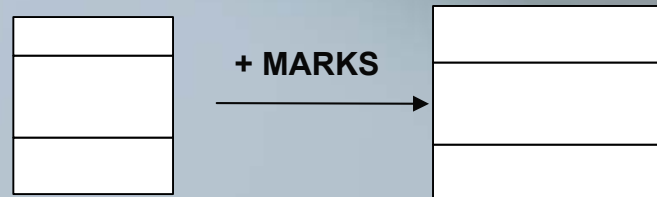
Model Driven Architecture (MDA)

- MDA Transformation Guides: Mappings
- Mapping gives rules and specifications to transform one model into another of a specific platform.



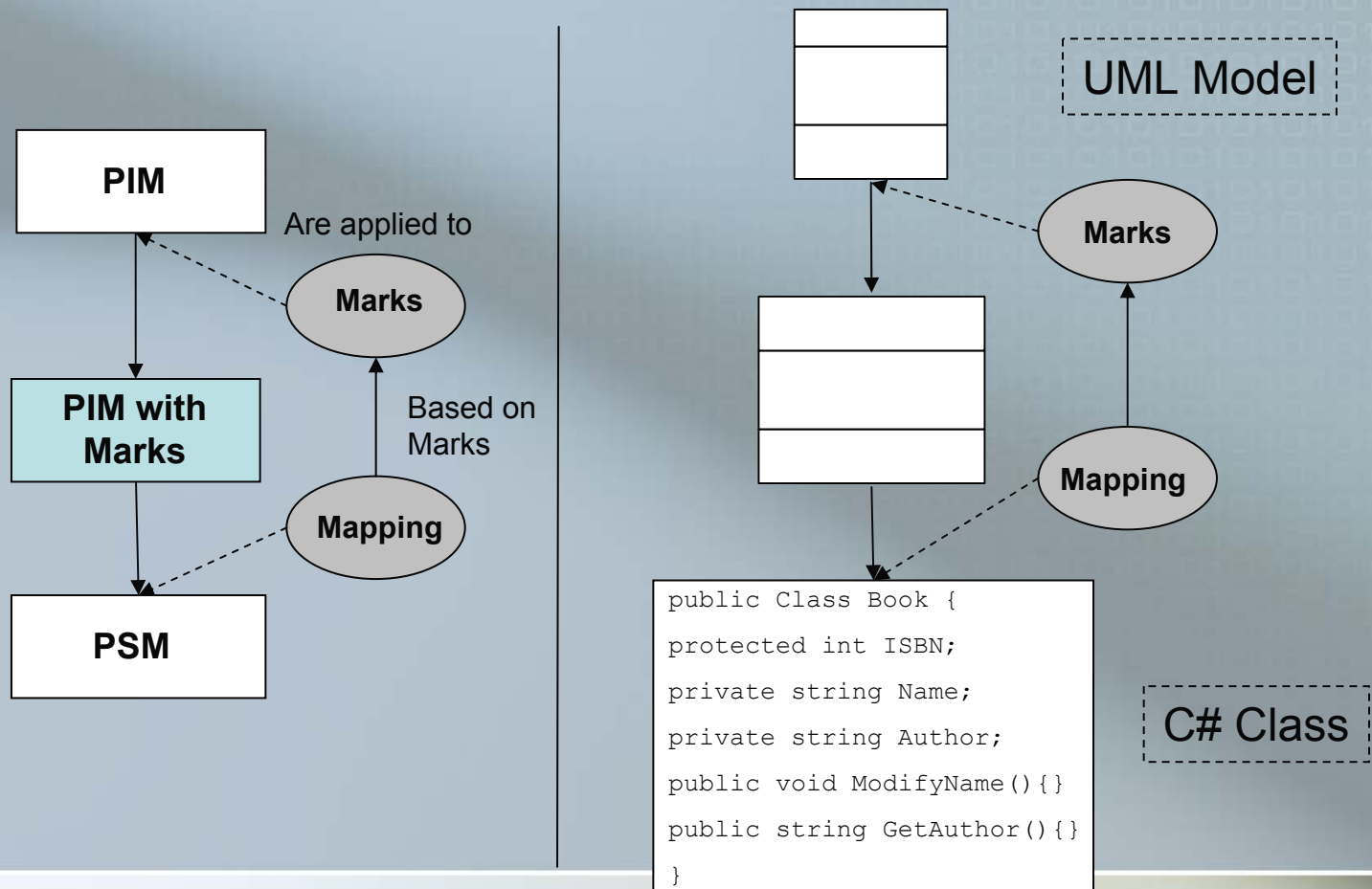
Model Driven Architecture (MDA)

- MDA Transformation Guides: Mappings
- Model Type Mappings: Mapping Rules are defined by using a Platform Model.
- Model Instance Mappings: Mapping Rules are defined by using Marks.



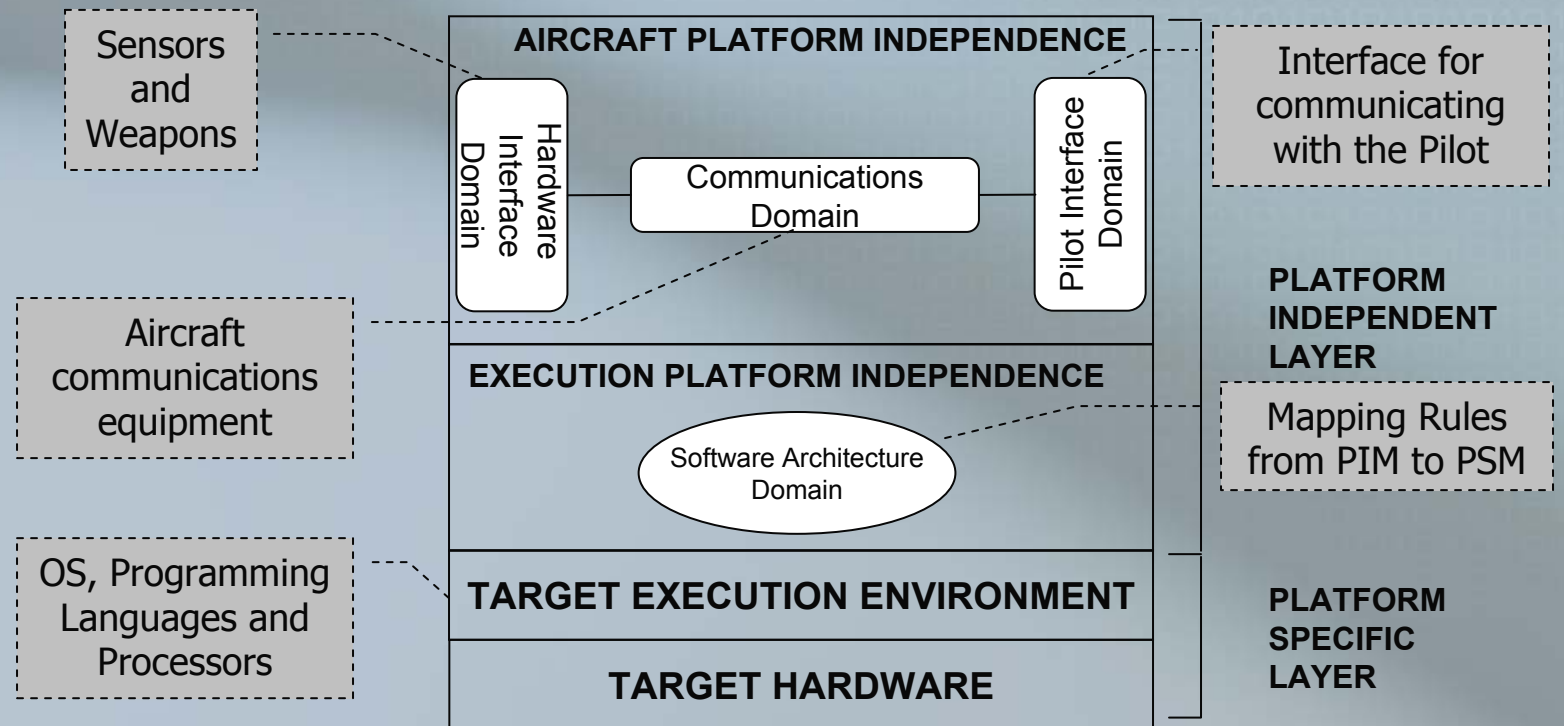
Model Driven Architecture (MDA)

- Case of transformation
 - Using Model Instance Mappings



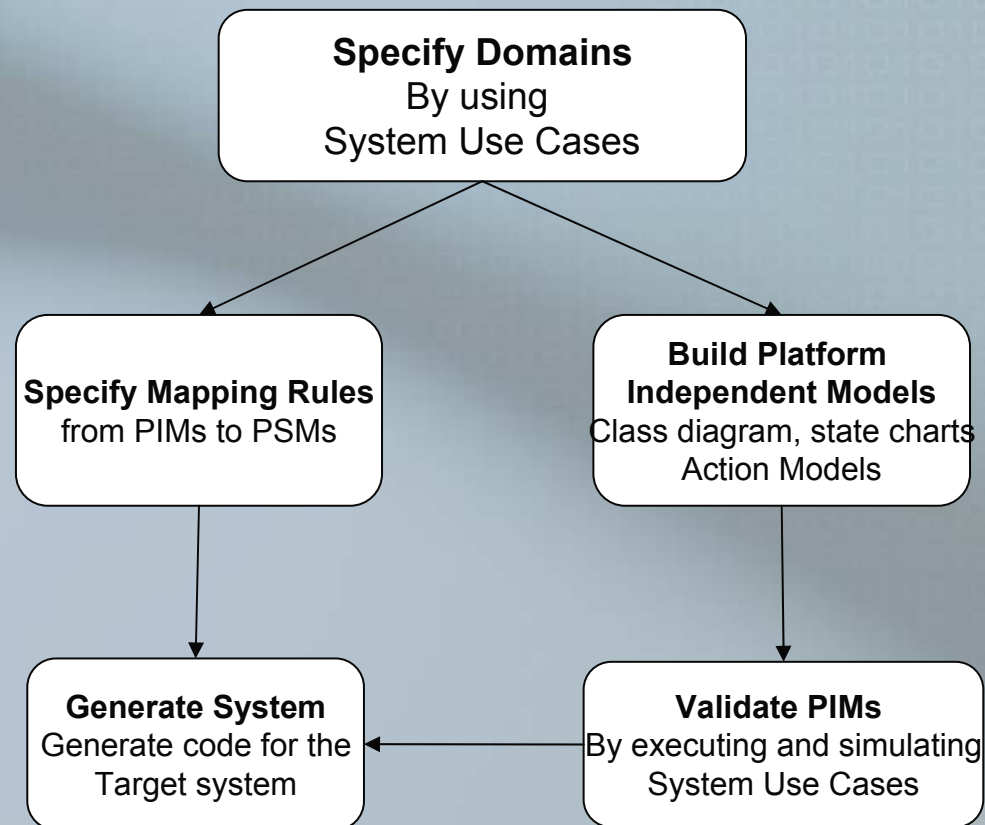
MDA & Avionics Industry

- Why MDA in the avionics?
- To improve their existing method of developing software: "waterfall method"
- To integrate all functionality blocks in a standardized way



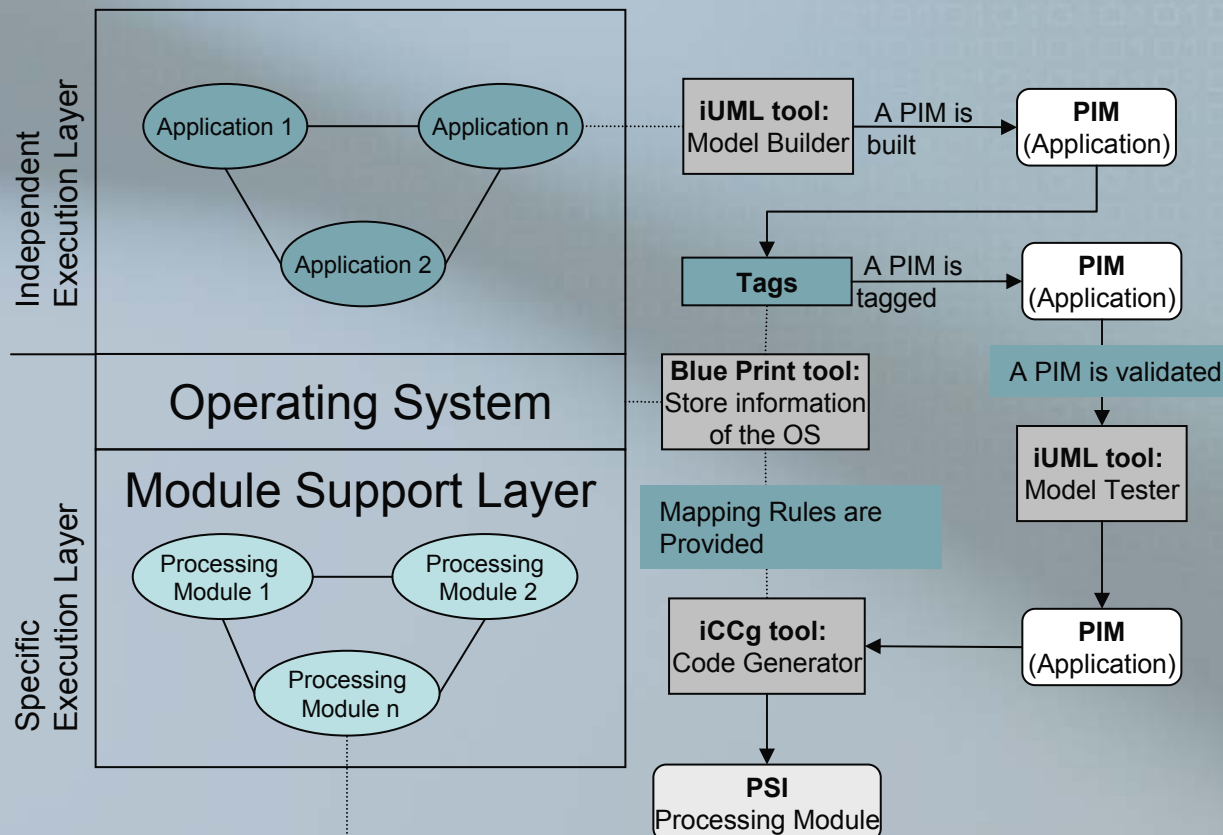
MDA & Avionics Industry

- MDA approach used by the Military Aircraft System



MDA & Avionics Industry

- Example of the MDA approach:
- Transform an application into a set of processing modules



MDA & Avionics Industry

- Why does MDA improve dependability?
 - It verifies platform independent models against the system requirements before a transformation.
 - It performs an automated transformation process.

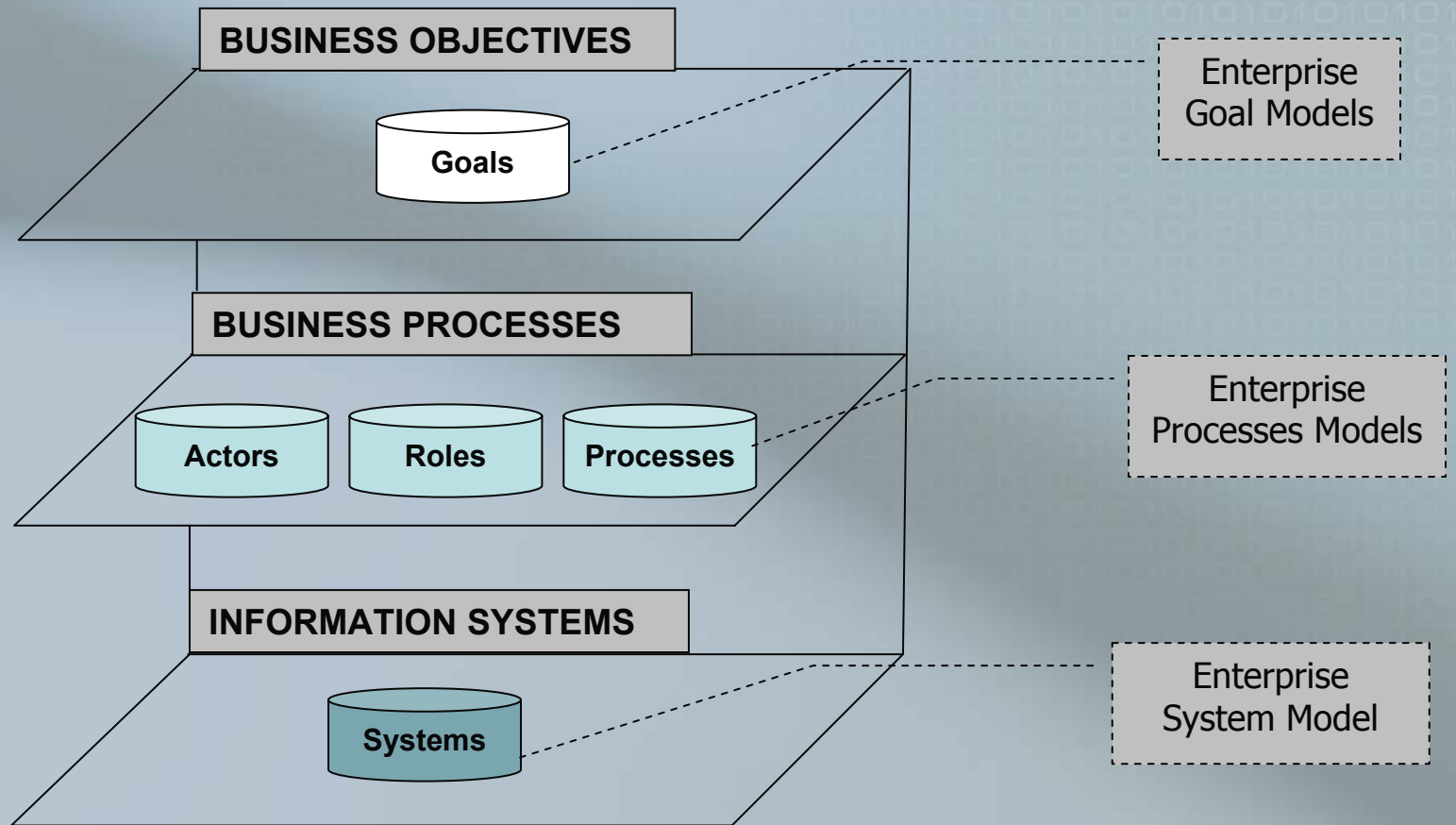
EKD-CMM

Enterprise Knowledge Development
Change Management Method

- DEFINITION
- Choose an information system suitable for an enterprise:
 - First define the objectives and processes of an enterprise.
 - Second, define the requirements of the information system suitable for the enterprise.

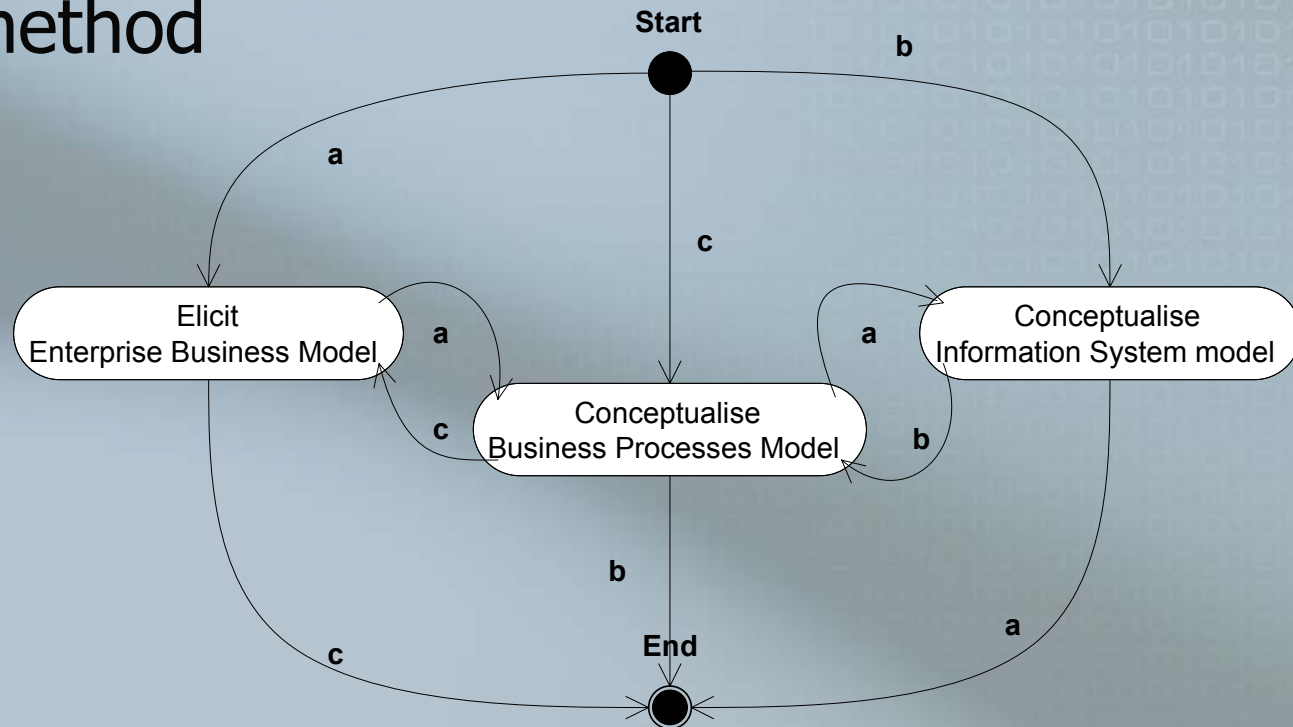
EKD-CMM Enterprise Knowledge Development Change Management Method

■ Layers of Modeling



EKD-CMM Enterprise Knowledge Development Change Management Method

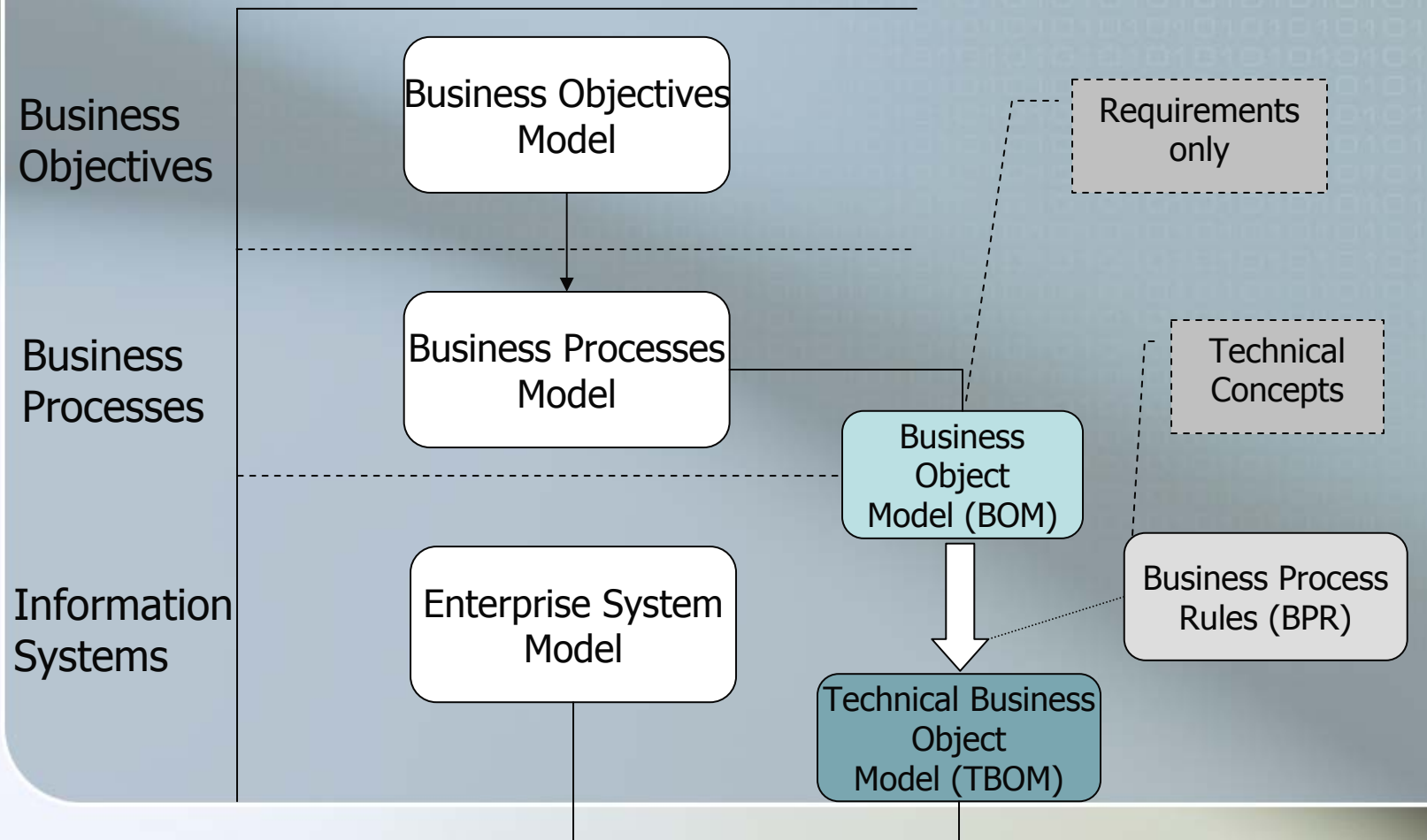
■ Road Map: Approaches to follow this method



- a. Forward Engineering
- b. Reverse Engineering
- c. Business Process Reengineering or Improvement

EKD-CMM Enterprise Knowledge Development Change Management Method

EDK-CMM approach example : Forward Engineering



MDA & EKD-CMM

- Similarities
- Use of Model Driven Development (MDD)
- Use of Transformations
- Use of the concept of layers
- Refinement of models to keep integrity



MDA & EKD-CMM

- Differences
- The final product of MDA is a platform specific model (PSM) : Implementation Code
- The final product of EDK-CMM is an Enterprise System Model: List of requirements of an Information System suitable for an Enterprise.

Conclusions

■ Dependability

■ MDA

- Reliability: MDA performs an automated transformation from models to code.
- Validation: The PIMs are validated against the requirements of a system before being transformed.
- Integrity: MDA provides an alternative to coding.

■ EKD-CMM

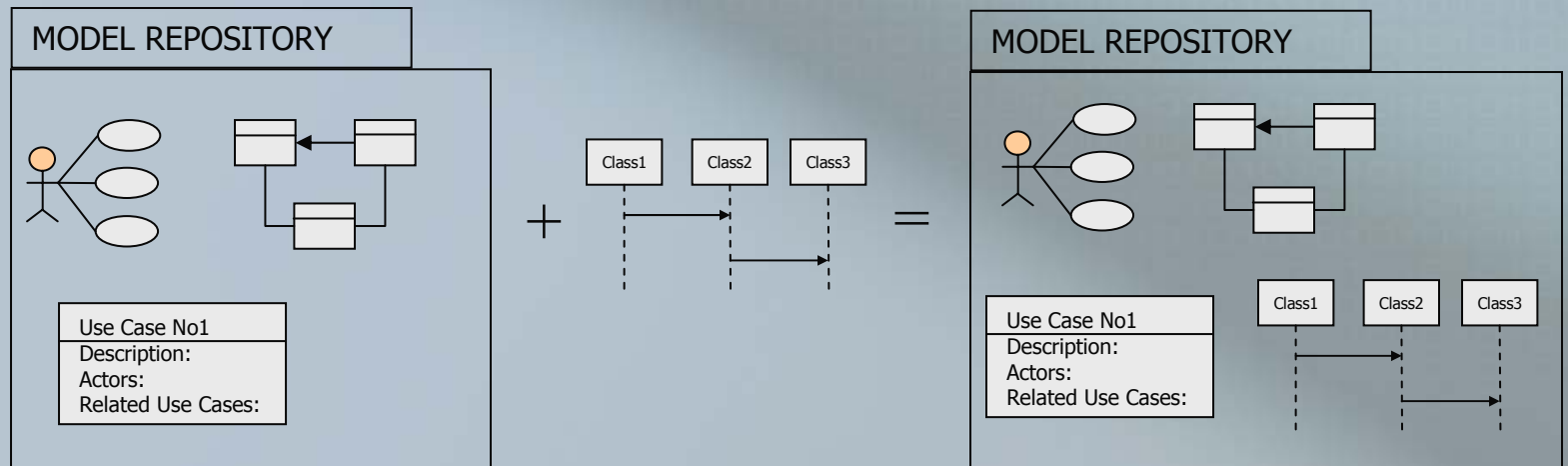
- EKD-CMM defines an enterprise system model, that describes how to support business processes at operational levels.

Conclusions

- Adaptability

- MDA

- In MDA, changes in the system are updated dynamically: MDA allow to make an update of the system directly in the models.





THE END