MODEL DRIVEN SOFTWARE DEVELOPMENT

LECTURE : 2

MDSD

- Models can be at different levels of abstractions
- Model at one level of abstraction can be automatically transformed into models or implementations at different levels of abstractions

MODEL DRIVEN ARCHITECTURE

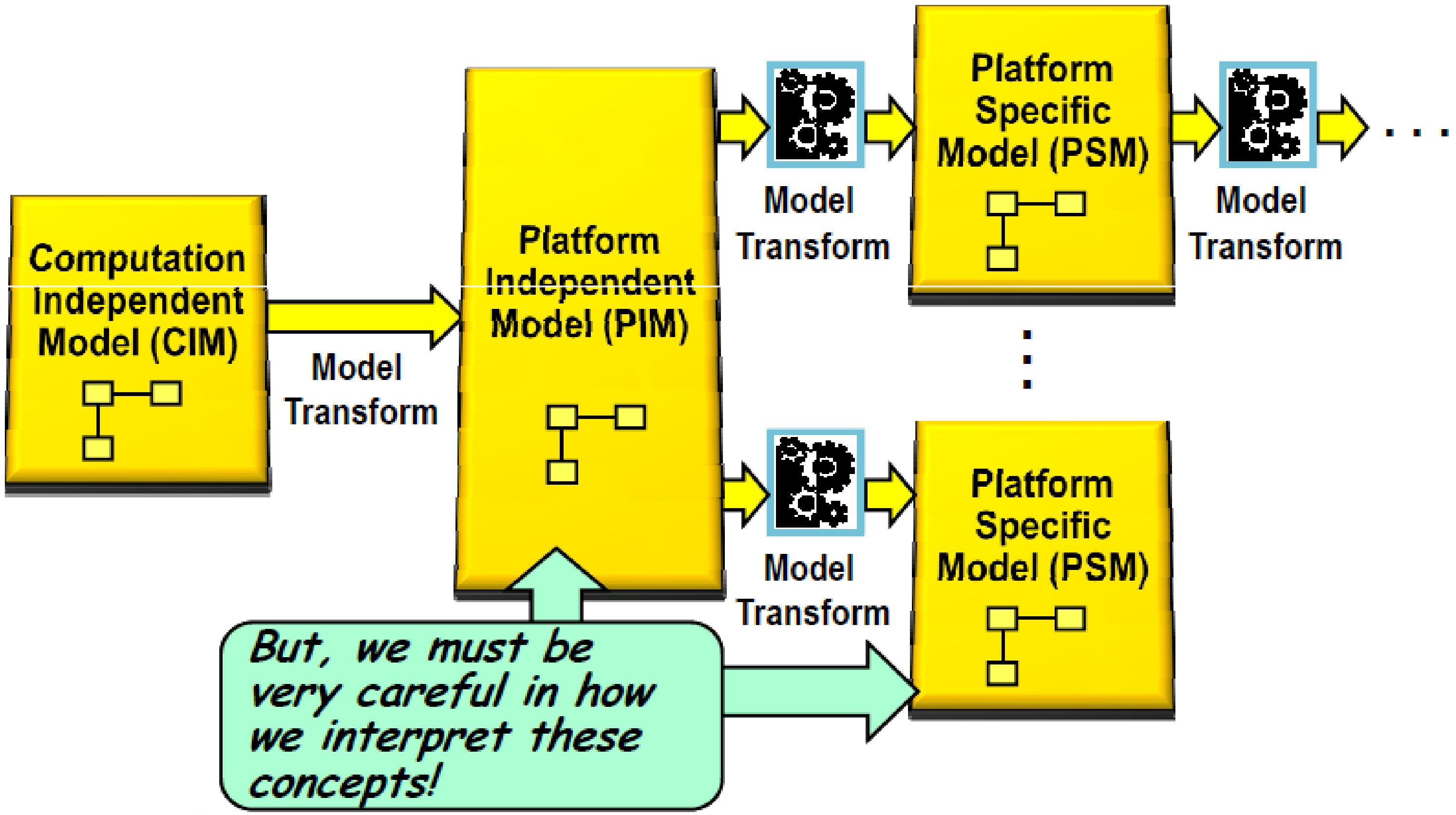
- The MDA is a specification that provides a set of guidelines for structuring specifications expressed as models

- One of the main aims of the MDA is to separate design from architecture and realization technologies facilitating that design and architecture can alter independently

- The design addresses the functional (use case) requirements while architecture provides the infrastructure through which non-functional requirements like scalability, reliability and performance are realized

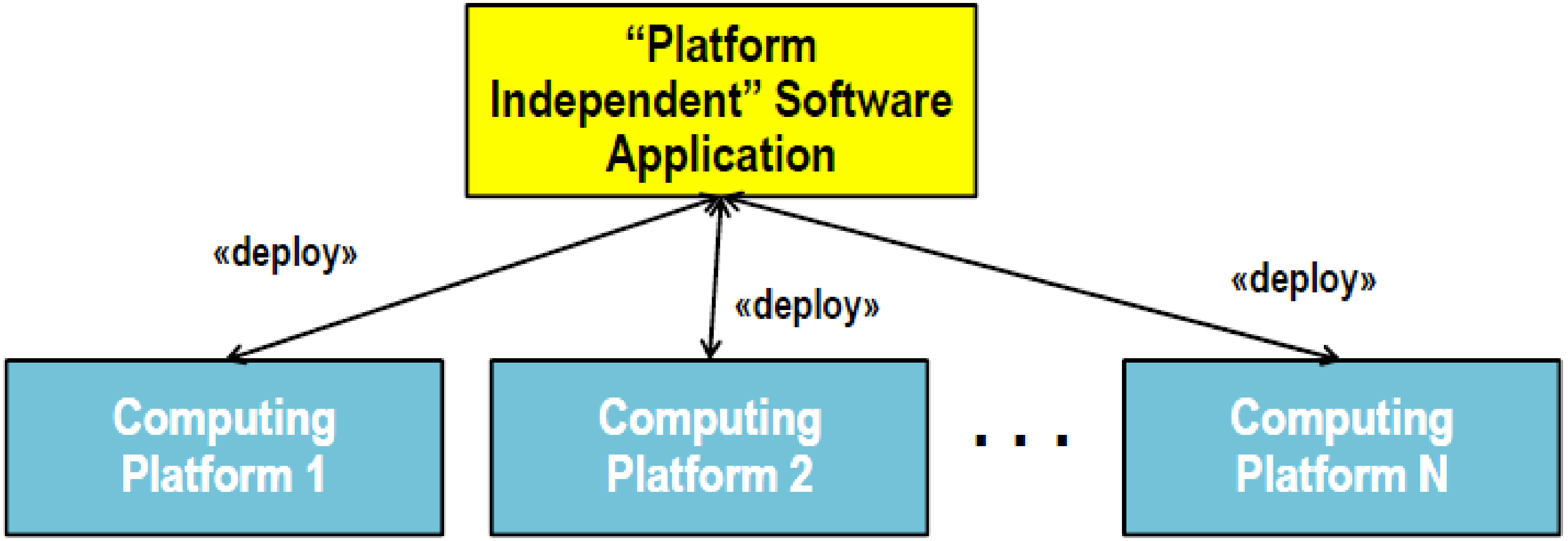
MDA INTERPRETATION OF MDSD

- A cascade of successively refined models leading to one or more implementations


Computation Independent Model (CIM)
Model Transform
Platform Independent Model (PIM)
Model Transform
Platform Specific Model (PSM)
Model Transform
. . .
⋮
Model Transform
Platform Specific Model (PSM)
But, we must be very careful in how we interpret these concepts!

THE CONCEPT OF "PLATFORM INDEPENDENCE"?

- A highly desirable objective
  - Separation of concerns – reduces apparent problem complexity
  - Enables portability



PLATFORM INDEPENDENT MODELS

- A PIM aims to capture implementation-independent information about the system
- Within a PIM, the system is modeled from the viewpoint of how it best supports the business.
- The implementation of the system plays no role in PIM
- Whether a system will be implemented as
  - Web-services with XML as storage mechanism
  - a relational database, or
  - on an EJB application server

PLATFORM SPECIFIC MODELS

- A PSM aims to provide the implementation detail tuned to a specific deployment environment.
- A PSM is tailored to specify the system in terms of the implementation constructs that are available in one specific implementation technology

- For example, an EJB PSM is a model of the system in terms of EJB structures
  - It typically contains EJB-specific terms like "home interface," "entity bean," "session bean," and so on
- A relational database PSM includes terms like "table," "column," "foreign key," and so on

BENEFITS OF MDSD -
PORTABILITY & PRODUCTIVITY

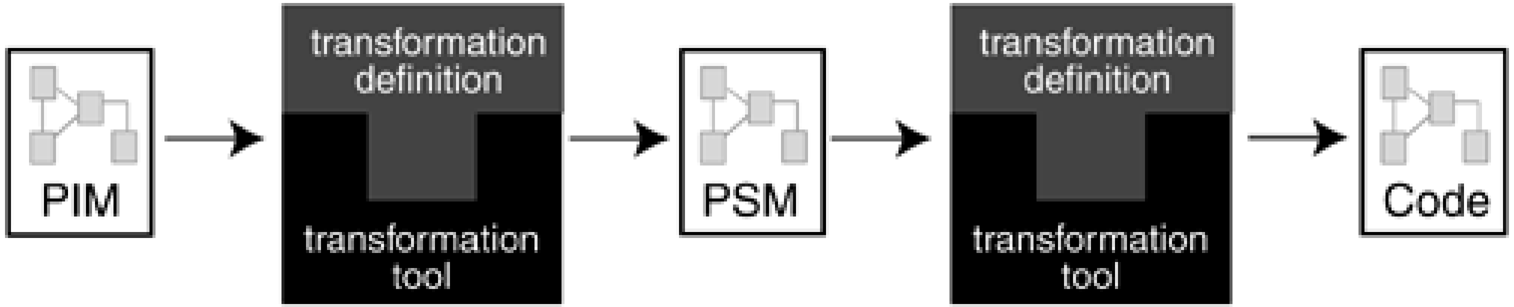
- The focus of a developer shifts to the development of models containing actual business logic
- The required implementations are generated from this model
- The transformation is to be defined once for each platform
- One model can be automatically transformed into multiple implementations for different platforms

BENEFITS OF MDSD – MAINTENANCE & DOCUMENTATION

- The models are not abandoned after writing
- Models are first-class artifacts of software development
- Changes made to the system will eventually be made by changing the models and regenerating the implementations

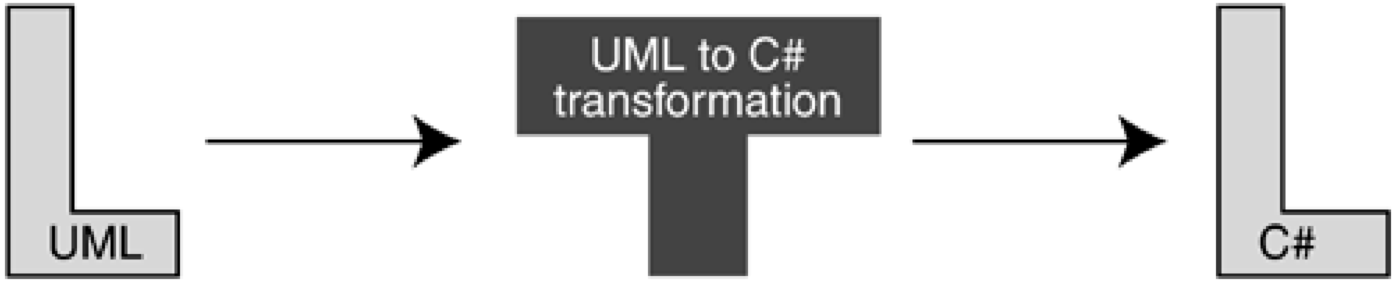
TRANSFORMATIONS

- A transformation tool takes a model and transforms it into another model or implementation
- Transformations are essential in the MD development process



TRANSFORMATIONS

- A transformation definition consists of **a set of transformation rules**
- These rules describe how a model in the source language can be transformed into a model in the target language
- To apply transformation specification independent of the source model the transformation specification relates constructs from the source language to constructs in the target language.
- We can, for example, define a transformation definition from UML to C#, which describes which C# should be generated for a (or any!) UML model


UML
UML to C# transformation
C#