



MODEL DRIVEN SOFTWARE DEVELOPMENT

LECTURE : 6



FOUR PRINCIPLES OF MDA

- Models must be expressed in a well-defined notation, so as to enable effective communication and understanding
- Systems specifications must be organized around a set of models and associated transformations
- Implementing mappings and relations between the models.
 - Multi-layered and multi-perspective architectural framework.
 - Models must be compliant with meta-model.
- Increase acceptance, broad adoption and tool competition for Model Driven development

TERMS IN MDA

- **System:** The subject of any MDA specification (program, computer system, federation of systems)
- **Problem Space (or Domain):** The context or environment of the system
- **Solution Space:** The spectrum of possible solutions that satisfy the reqs.
- **Model:** Any representation of the system and/or its environment
- **Architecture:** The specification of the parts and connectors of the system and the rules for the interactions of the parts using the connectors
- **Platform:** Set of subsystems and technologies that provide a coherent set of functionalities for a specified goal
- **Viewpoint:** A description of a system that focuses on one or more particular concerns
- **View:** A model of a system seen under a specific viewpoint
- **Transformation:** The conversion of a model into another model

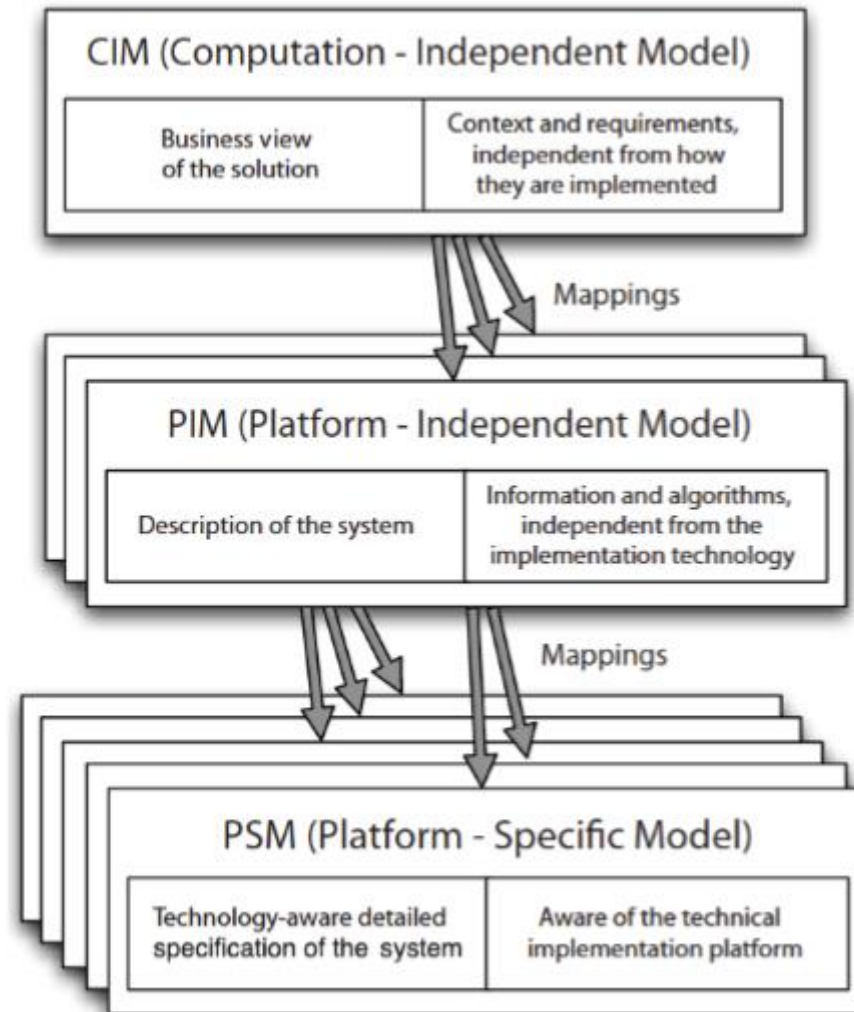
TYPES OF MODELS

- **Static models:** Focus on the static aspects of the system in terms of managed data and of structural shape and architecture of the system.
- **Dynamic models:** Emphasize the dynamic behavior of the system by showing the execution.

MODELING LEVELS

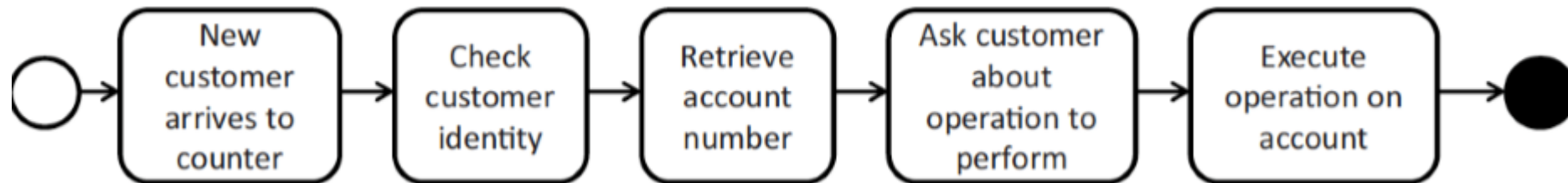
- **Computation independent (CIM)**
- **Platform independent (PIM)**
- **Platform-specific (PSM):** define all the technological aspects in detail.

MODELING LEVELS



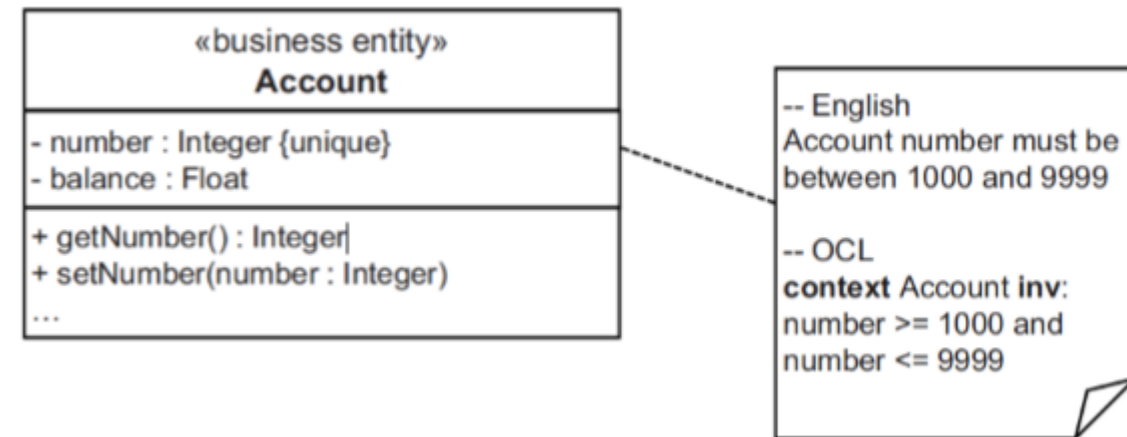
COMPUTATION INDEPENDENT MODEL (CIM)

- At the CIM level, the process is defined as a business process model listing the set of tasks to be performed and their dependencies.
- Describe requirements and needs at a very abstract level, without any reference to implementation aspects (e.g., description of user requirements or business objectives)
- E.g., business process



PLATFORM INDEPENDENT MODEL (PIM)

- Specification of the **behavior of the systems** in terms of stored data and performed algorithms, **without any technical or technological details**
- It describe that the software application will do some parts of the job, in terms of information and behavior models i.e. **Constraints**
- Abstraction of structure and behaviour of a system with the PIM simplifies the following:
 - Validation for correctness of the model
 - Create implementations on different platforms
 - Tool support during implementation



PLATFORM SPECIFIC MODEL (PSM)

- Even if it is not executed itself, this model must contain all **required information** regarding the behavior and structure of an application on a **specific platform** that developers may use **to implement the executable code**.
- Specifies how the functionality described in the PIM is realized on a certain platform
- Using a UML-Profile for the selected platform, e.g., EJB
- PSM level, the implementation platform (whose description should be defined separately) is precise descriptions of the **technical details** associated with that **platform**
- Including all the details regarding the invocations of the **platform specific APIs** and so on, will be provided within the models.

