University of Applied Sciences Northwestern Switzerland
School of Business

Master of Science
Business Information Systems

Modeling and Meta-Modeling

Models and Modeling

Modeling

Describing and Representing all relevant aspects of a domain in a defined language.

Result of modelling is a model - an exemplary reproduction of reality.

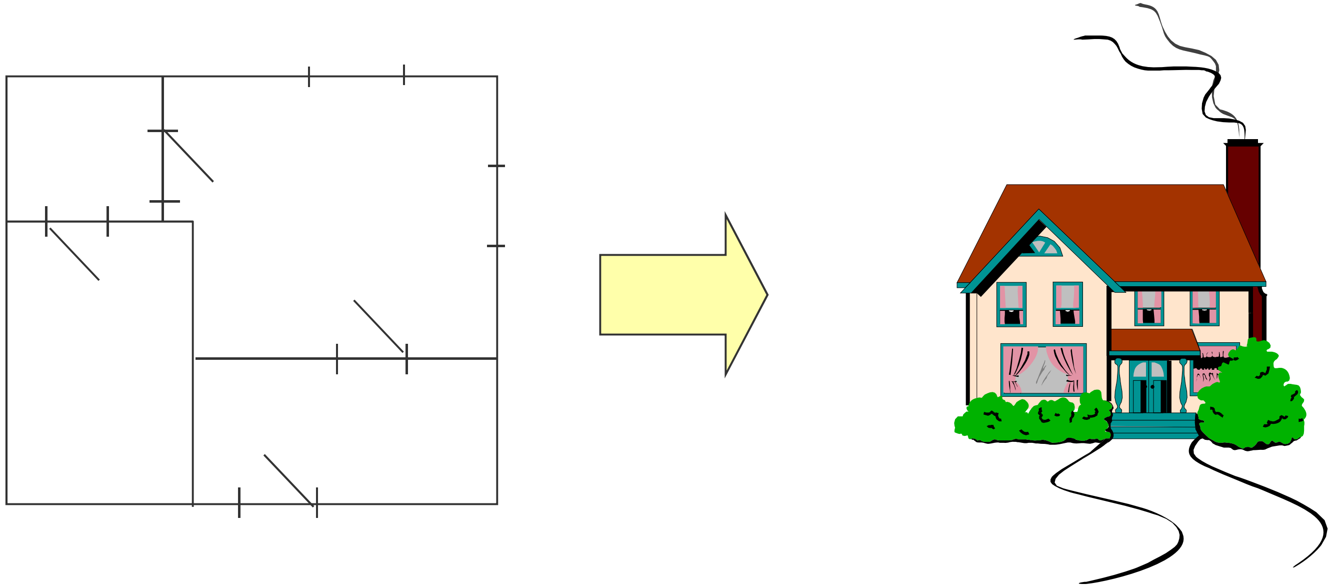
Model

A reproduction of the part of reality which contains the essential aspects to be investigated.

Model and Real Object in Architecture

modell (plan)

real object

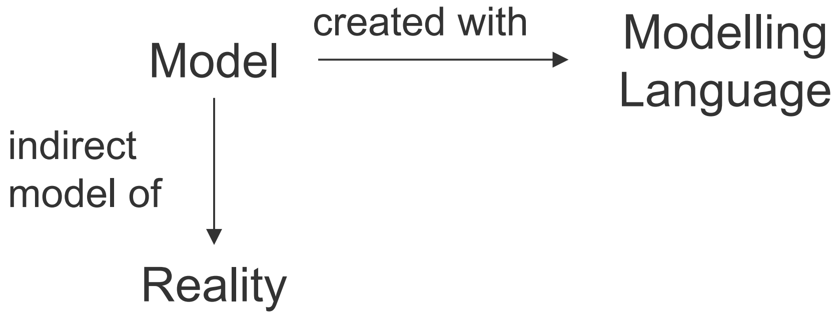


Rationale for Modelling

- Models provide abstractions of a physical system that allow engineers to reason about that system by ignoring extraneous details while focusing on relevant ones.
- All forms of engineering rely on models to understand complex, real-world systems.
- Models are used in many ways:
  - predict system qualities
  - reason about specific properties when aspects of the system are changed
  - communicate key system characteristics to various stakeholders

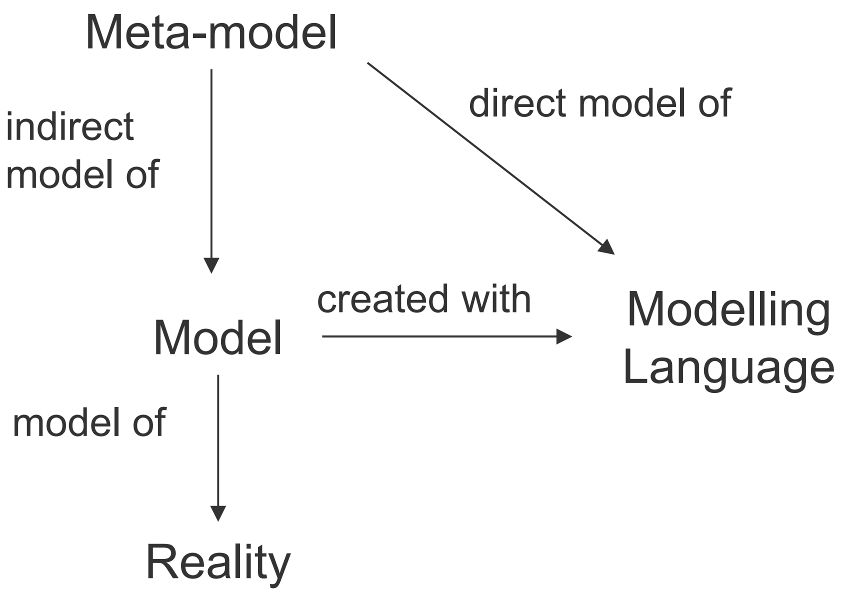
(Brown 2004)

Modelling Language



- A modelling "language" specifies the building blocks from which a model can be made.
- There can be different types of modelling languages, depending on the kind of model
  - ♦ graphical model
  - ♦ textual description
  - ♦ mathematical model
  - ♦ conceptual model
  - ♦ physical model

Meta-model



A meta-model defines the modelling language, i.e. the building blocks that can be used to make a model. It defines the

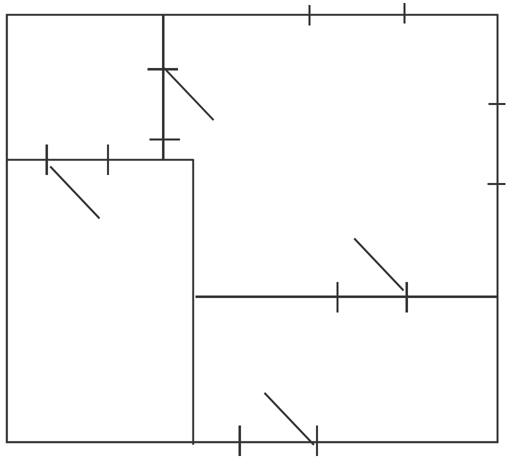
- object types that can be used to represent a model
- relations between object types
- attribtues of the object types
- meaning of the object types
- rules to combine object types and relations

Model and Meta-Model in Architecture

real object

house

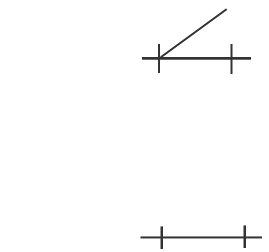
model



architect's drawing (plan)

meta-model
(modelling language)

object types:



wall

door

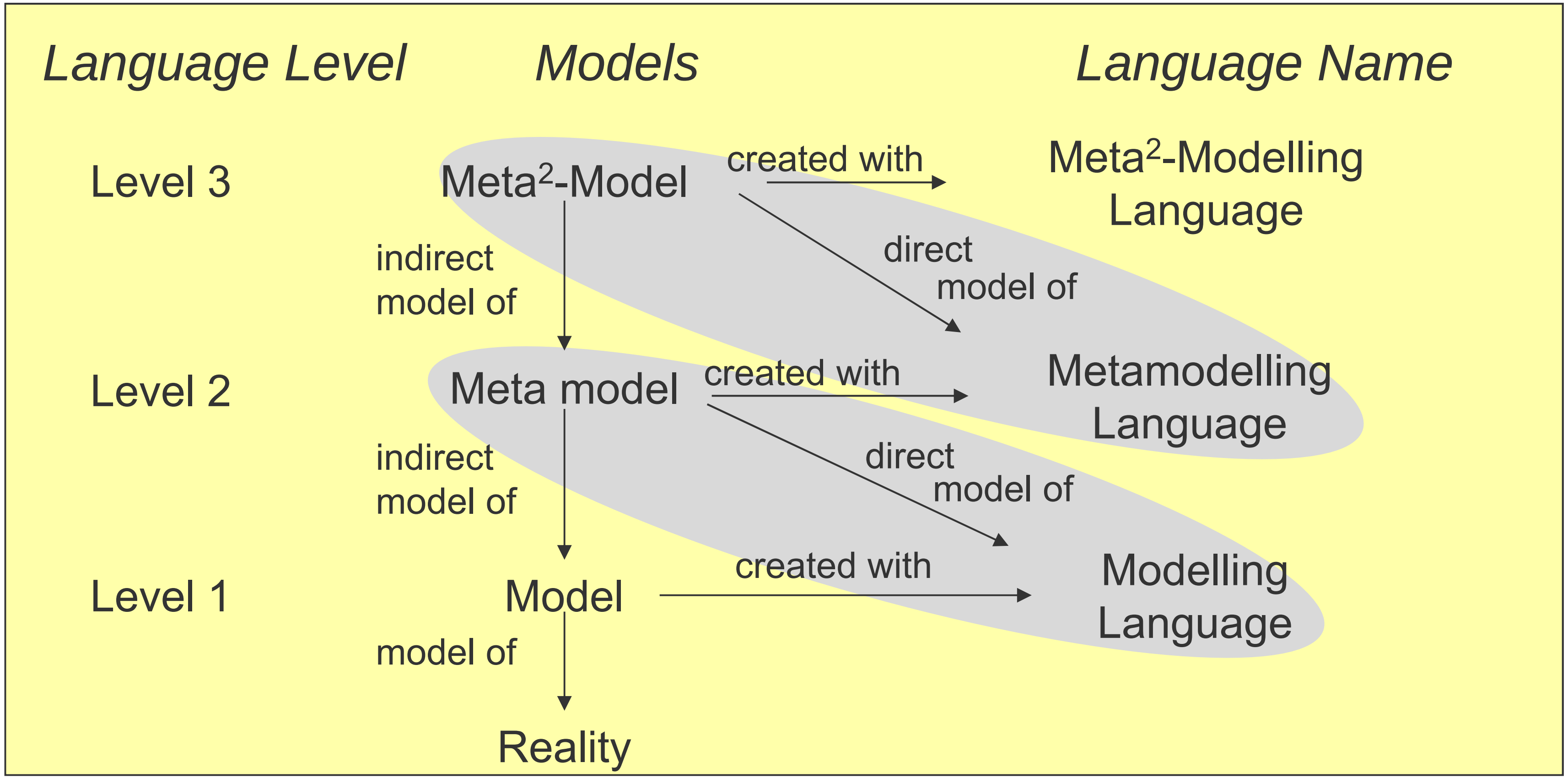
window

rules:

- a door is adjacent to a wall on both sides
- Windows are on outer walls.

Meta Model Hierarchy

The meta-model must again be described in some language, which has to be specified in a meta-model



Often the meta-model and the modeling language are unified and not distinguished.

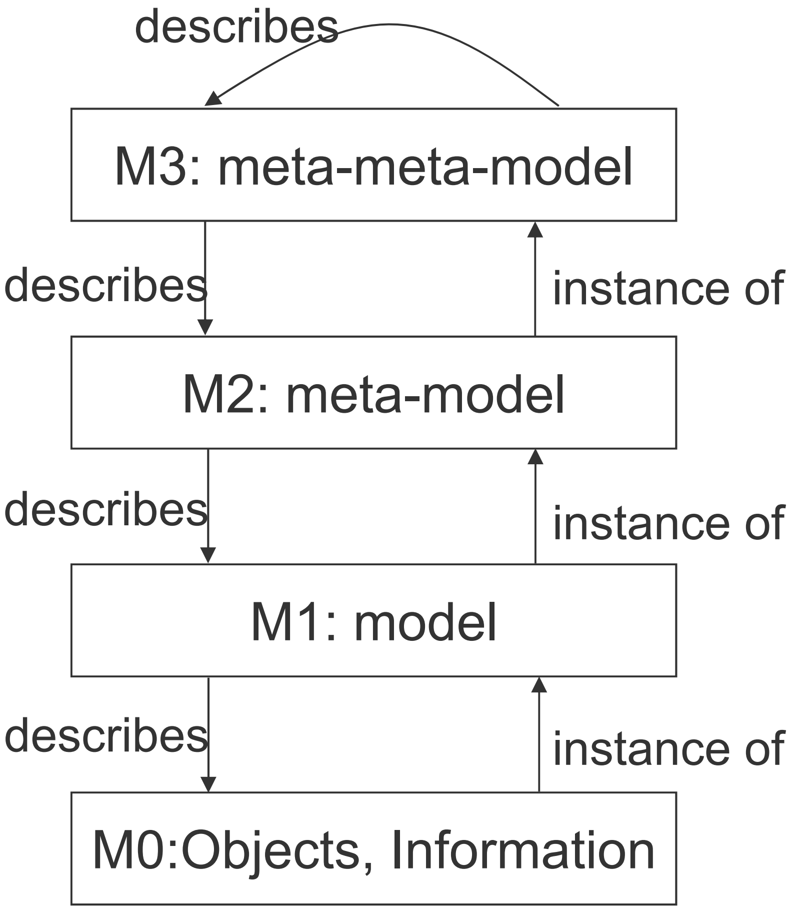
4 Layer Meta-model Architecture

Layer	Description	Examples
Metametamodel	Foundation for a Meta-modeling Architecture. Defining the language to describe meta-models	MetaClass, MetaAttribute, MetaOperation
Metamodel	An Instance of a meta-meta-model. Defining the language to describe models.	Class, Attribute, Operation, Component
Model	An Instance of Meta-model. Defining a language to describe the information object domain.	Customer, Product, Unit Price, Sale, Detail
User Objects **(User Data)**	An Instance of a Model. Defines specific information Domain	<Knut>, <Peter>, <Knut’s phone>, $600

MOF – Meta Object Facility

- The Meta Object Facility (MOF) is an OMG meta-modeling standard.
- MOF is itself a *meta-meta-model*, a specification describing how one may build meta-models.
- MOF is closely based on Unified Modeling Language (UML):
  - Meta-models are represented with class diagrams of UML (with some minor constraints necessitated by the nature of metamodeling).
- MOF defines the theoretical underpinnings of the XML Metadata Interchange (XMI)
  - XMI is a standard syntax for the Exchange of Models

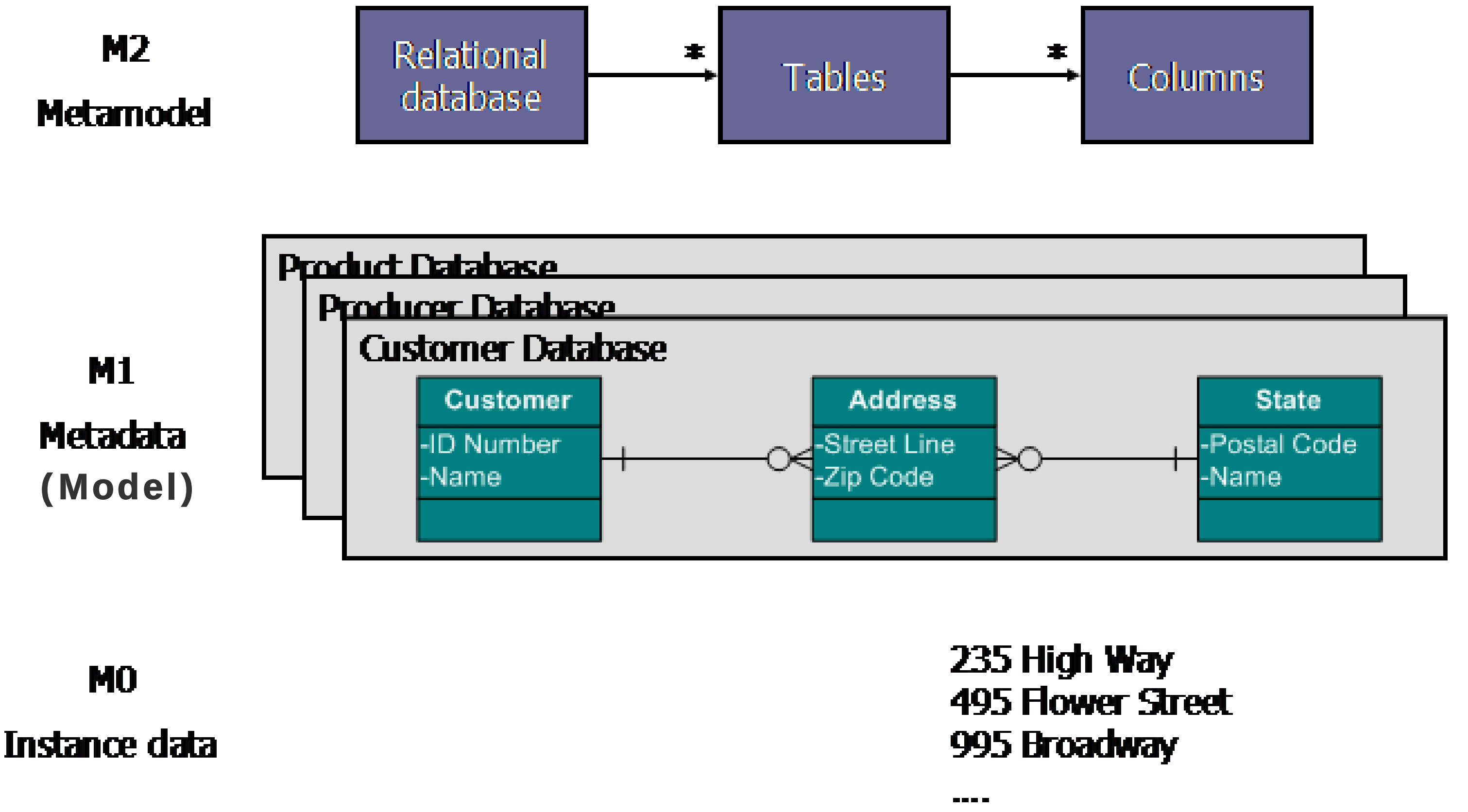
The OMG Model Stack



The Meta Object Facility (MOF) distinguishes four levels:

- M0 is the basic data, the lifeblood of the business
  - the customer name "Peter Miller", the price "$291.70".
- M1 is the metadata: schemas and interfaces describing the structure of the data.
  - a table customer with a name column
- M2 is the meta-model, or the "IT language" - specifying the concepts of the modelling language
  - "A relational database has tables, each table has zero or more columns".
  - "UML has classes, associations, attributes etc."
- M3 is the MOF specification itself, which allows us to draw the boxes-and-arrows of UML

Meta-model and Model for Relational Databases



M2

Metamodel

Relational database → * Tables → * Columns

M1

Metadata

(Model)

Product Database

Producer Database

Customer Database

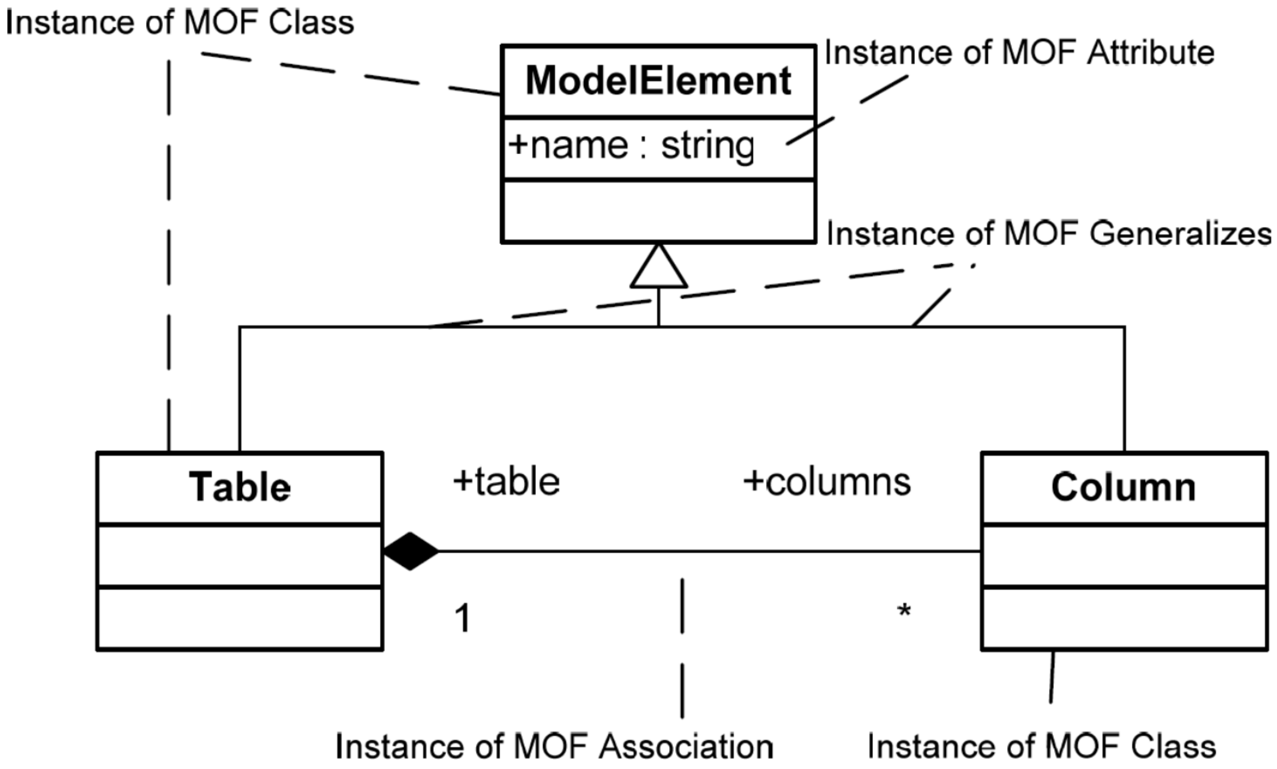
Customer	Address	State
-ID Number -Name	-Street Line -Zip Code	-Postal Code -Name

M0

Instance data

235 High Way
495 Flower Street
995 Broadway
....

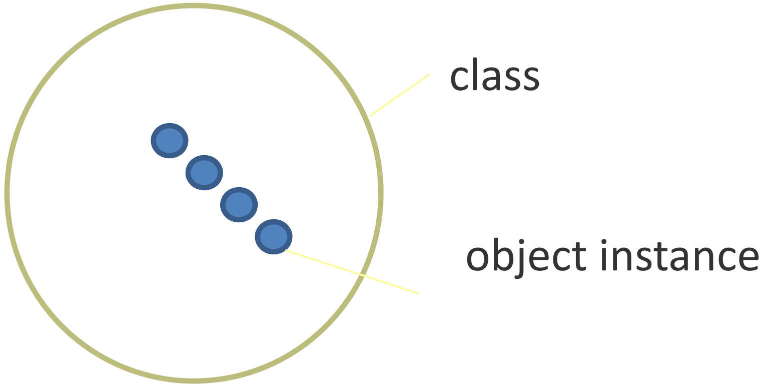
Example of a M2 Metamodel



Basic Idea: How to define an Object

Reality

Set: Employees of company A



class

object instance

M0 Layer

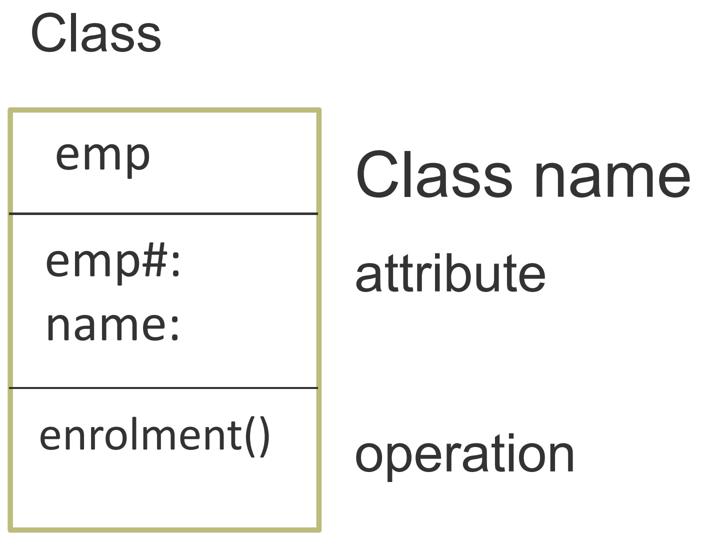
emp #	Name		
0800101	Adam Smith		
0800102	Jon Due		
0800103	Hajime Hori		

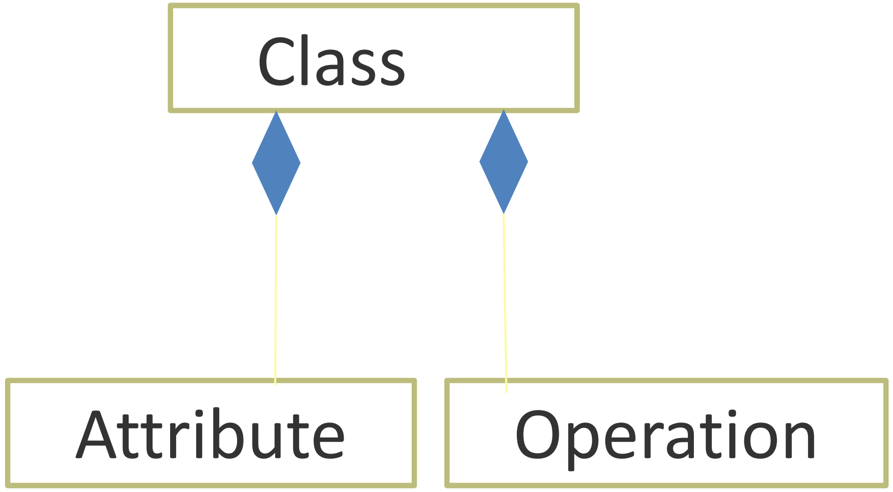
Class Emp={ people | people working for company A}

Object Concept and Metamodel

M1 Layer

M2 Layer

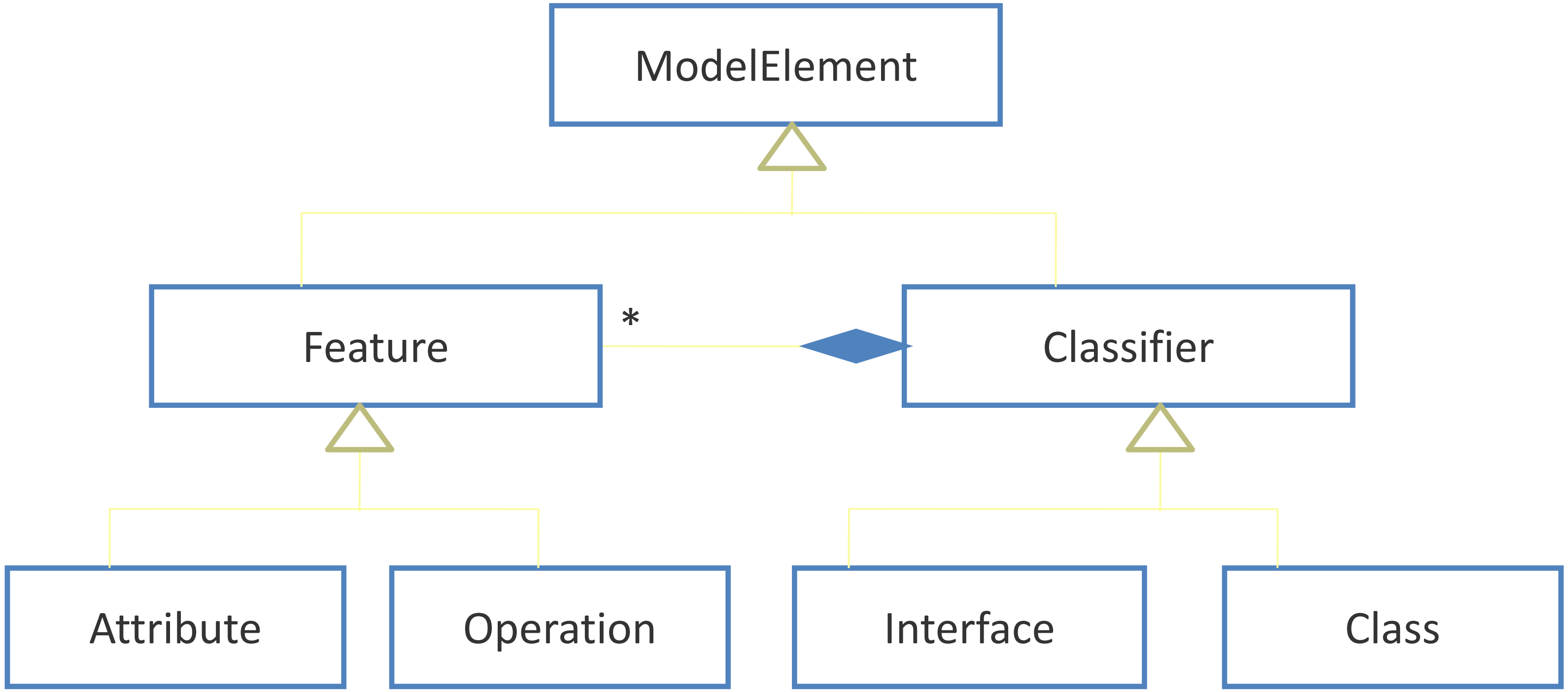




Class has Attributes and Operations

Overview of M3 Layer

The M3 Layer of MOF is represented as a UML Class Diagram



Use of explicit Meta-models

- A meta-model is a model used to model modeling itself.
- Meta-models provide a platform-independent mechanism to specify the following:
  - The shared structure, syntax, and semantics of technology and tool frameworks
  - A shared interchange format (using XML).
  - A shared programming model for transformation and querying of models

Use Cases for the Meta Levels

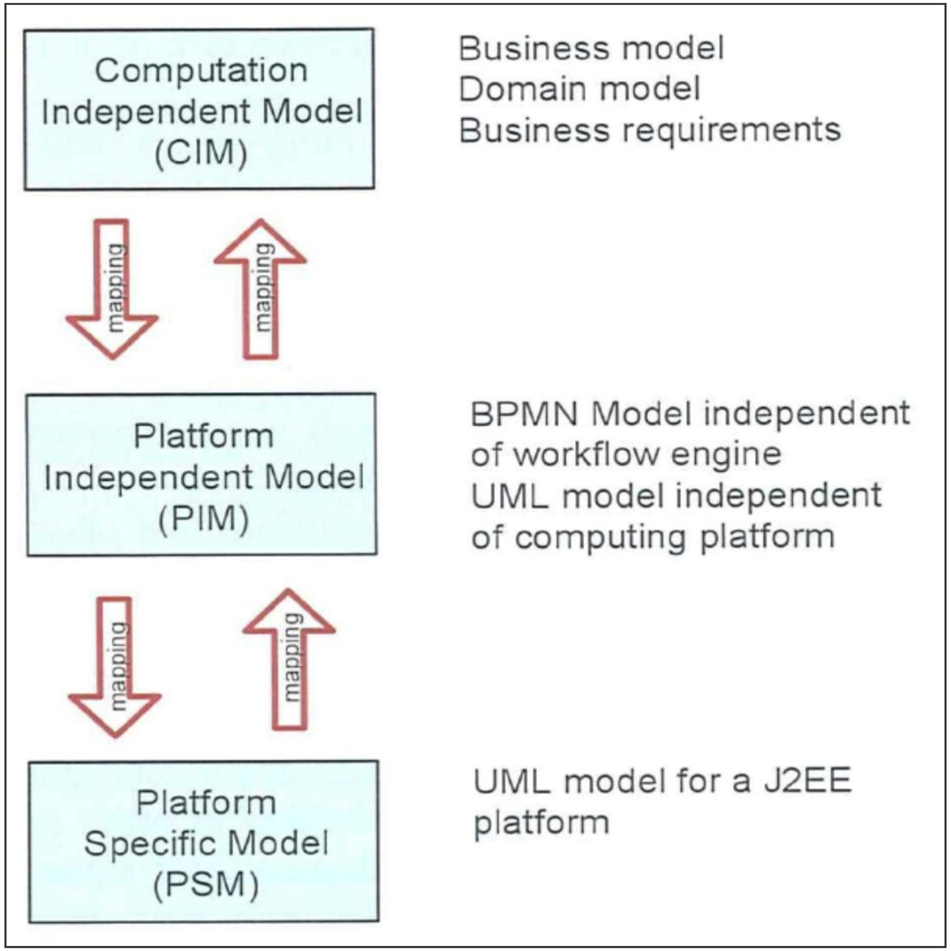
- The different meta-levels have quite different use cases:
  - data is used by the business,
  - metadata is used by IT, and
  - meta-models are used by metadata repositories (allowing metamodels to be configured rather than hard-coded).
- There is generally less metadata than data, and much less variety in metadata languages (metamodels) than in metadata.
  - A given enterprise, for example, may have millions of database rows, hundreds of schemas, but only a few different varieties of data bases are installed.

OMG's Model-Driven Architecture

- MDA is provided by Object Management Group OMG
- Aims to provide an open, vendor-neutral approach to interoperability
- Builds upon OMG's modelling standards
  - UML: Unified Modelling Language
  - MOF: Meta Object Facility
  - XMI: XML Metadata Interchange
- MDA wants to raise the level of abstraction at which software solutions are specified
  - generate code from models and views
  - Example: specify software in UML instead of programming it in Java
- Recently, OMG has extended the focus of MDA to cover business aspects of a company, e.g.
  - Business process modelling notation BPMN
  - Business motivation model BMM
  - Semantics for Business Vocabulary and Rules SBVR

(Lankhorst et al. 2005, p. 25f)

Model-Driven Architecture MDA



Computation Independent Model (CIM)

Business model
Domain model
Business requirements

Platform Independent Model (PIM)

BPMN Model independent of workflow engine
UML model independent of computing platform

Platform Specific Model (PSM)

UML model for a J2EE platform

MDA comprises three levels of abstraction with mappings between them

CIM Computation-Independent Model

- ♦ modelling the requirements for the system describing the situation in which the system will be used
- ♦ hiding much or all information about the use of IT systems

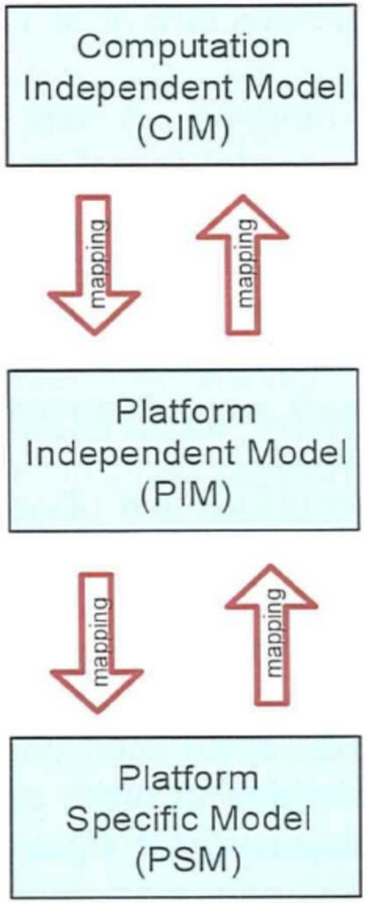
PIM Platform-Independent Model

- ♦ describing operations of the system while hiding details for a particular platform
- ♦ describing those parts of the system specification that do not change from one platform to another

PSM Platform-Specific Model

- ♦ Combines specifications of PIM with details about a particular type of platform

Model-Driven Architecture MDA


Computation Independent Model (CIM)
mapping
mapping
Platform Independent Model (PIM)
mapping
mapping
Platform Specific Model (PSM)

- MDA comprises three levels of abstraction
  - CIM – Computation Independent Model
  - PIM – Platform Independent Model
  - PSM – Platform Specific Model
- For the mapping OMG defined two standards:
  - **XMI - XML Metadata Interchange**
    Standard Syntax for the Exchange of Models
  - **MOF – Meta Object Facility**
    Well-defined Semantics of the Modeling Constructs