University of Applied Sciences Northwestern Switzerland
School of Business

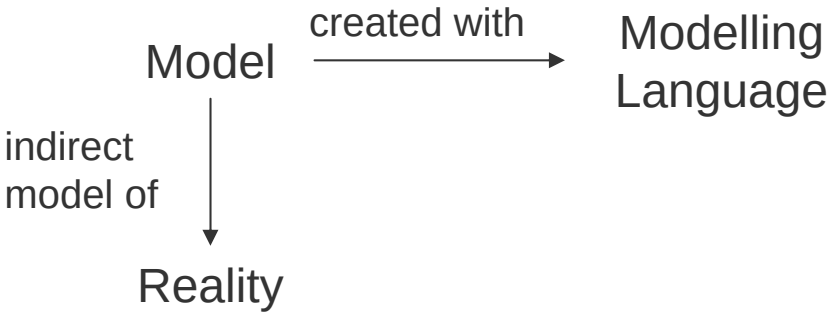
Metamodelling

Model and Metamodel

- Defining the modelling objects of a modelling language and their relations itself is a kind of modelling
- The model of a modeling language is called a metamodel

  "A metamodel is a model used to model modeling itself."
- To model the modelling language we again need a language which is called a meta language

Modelling Language

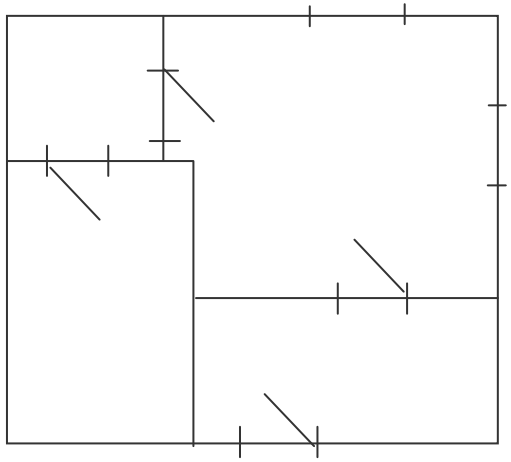


- A modelling "language" specifies the building blocks from which a model can be made.
- There can be different types of modelling languages, depending on the kind of model
  - graphical model
  - textual description
  - mathematical model
  - conceptual model
  - physical model

Model and Modelling Language in Architecture

real object

modell (plan)



modelling language (metamodel)

object types:

wall

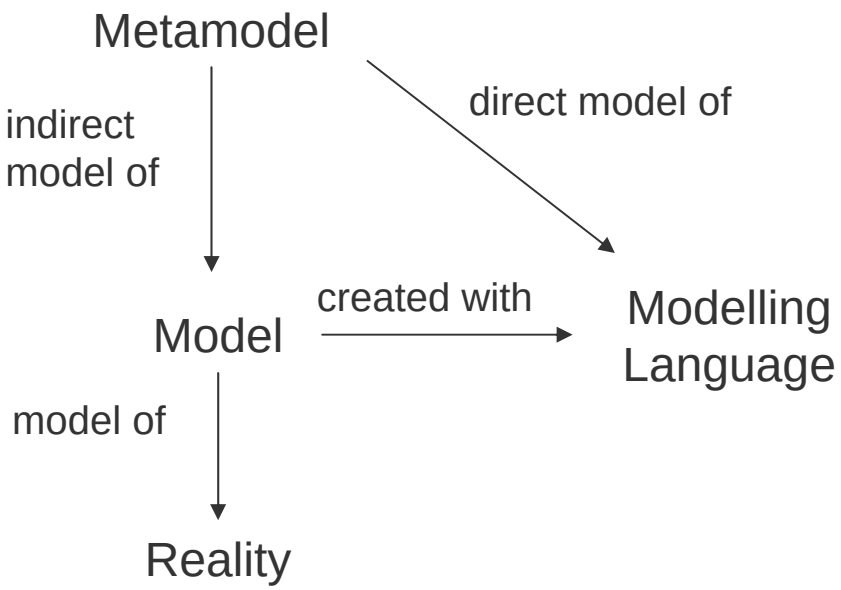
door

window

rules:

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

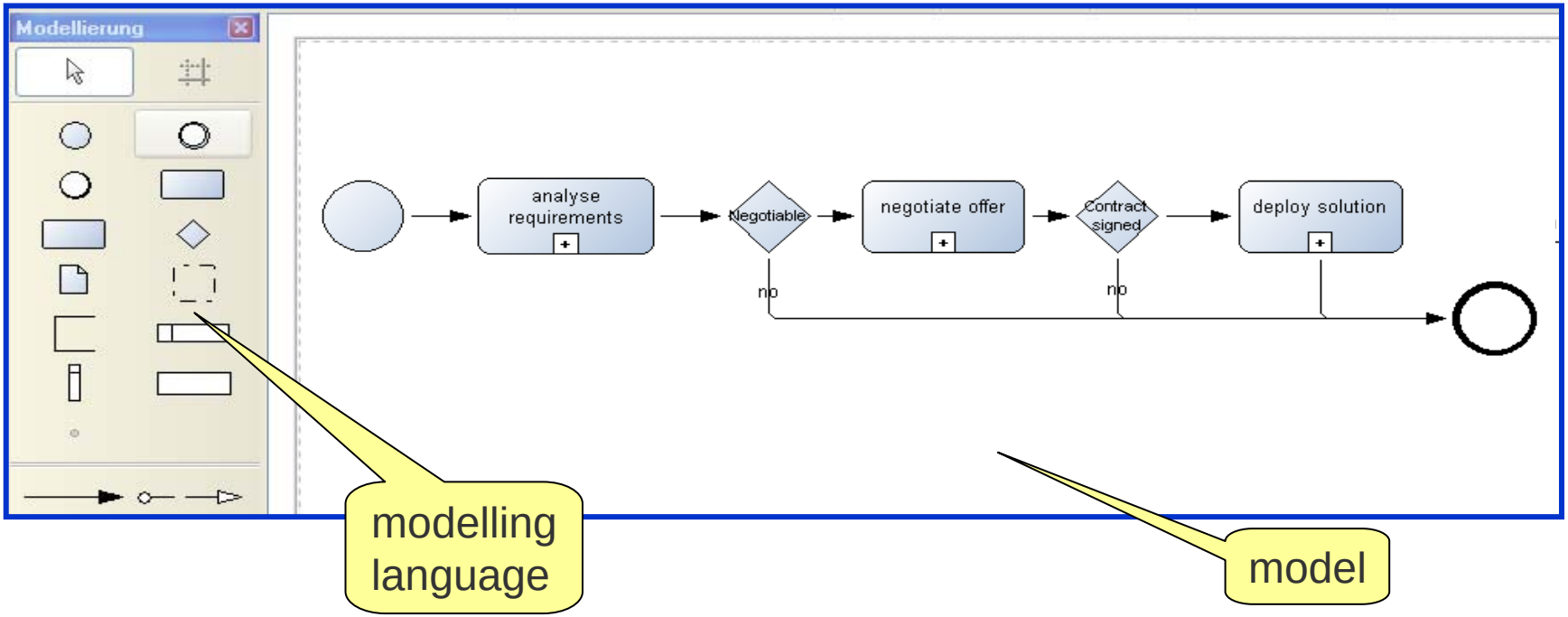
Meta Model



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

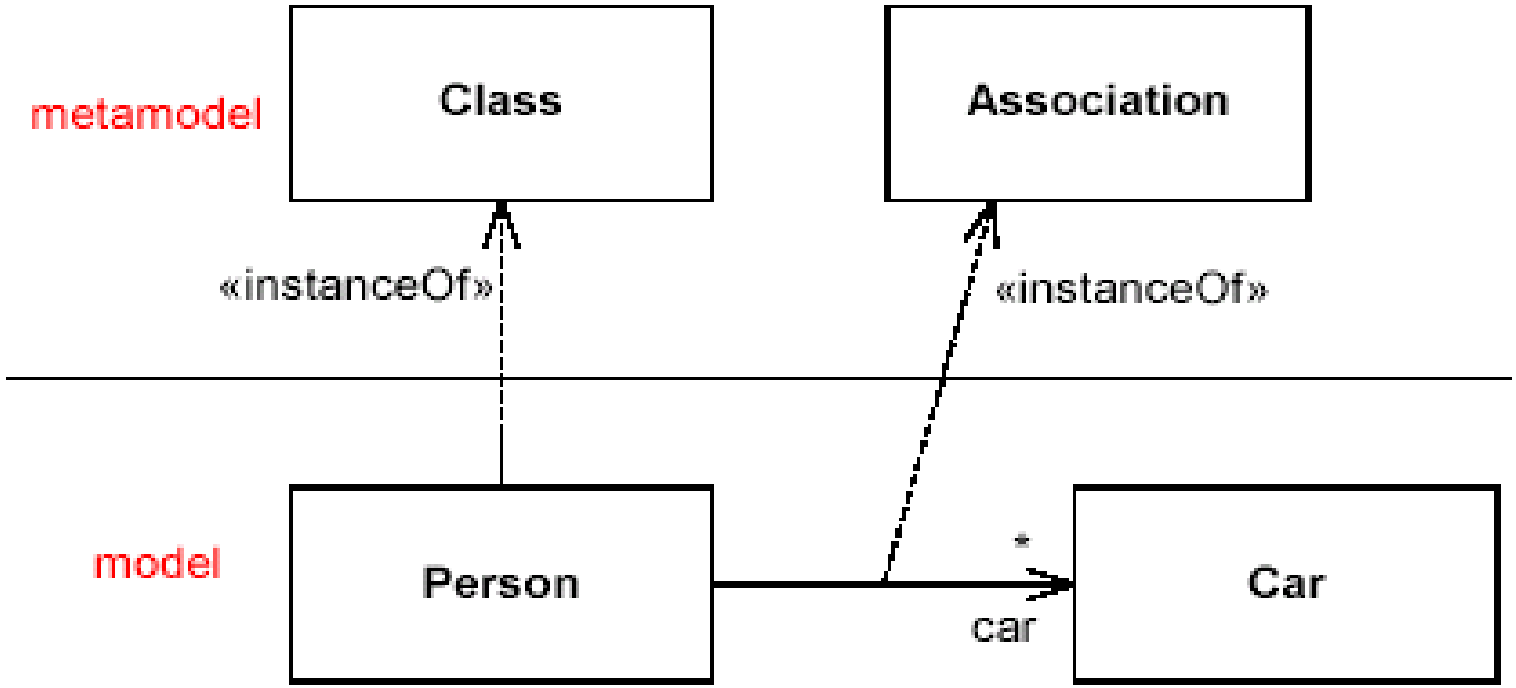
- the object types that can be used to represent a model
- relations between object types
- attribtues of the object types
- the meaning of the object types
- rules to combine object types and relations

Model and Modelling Language for BPMN



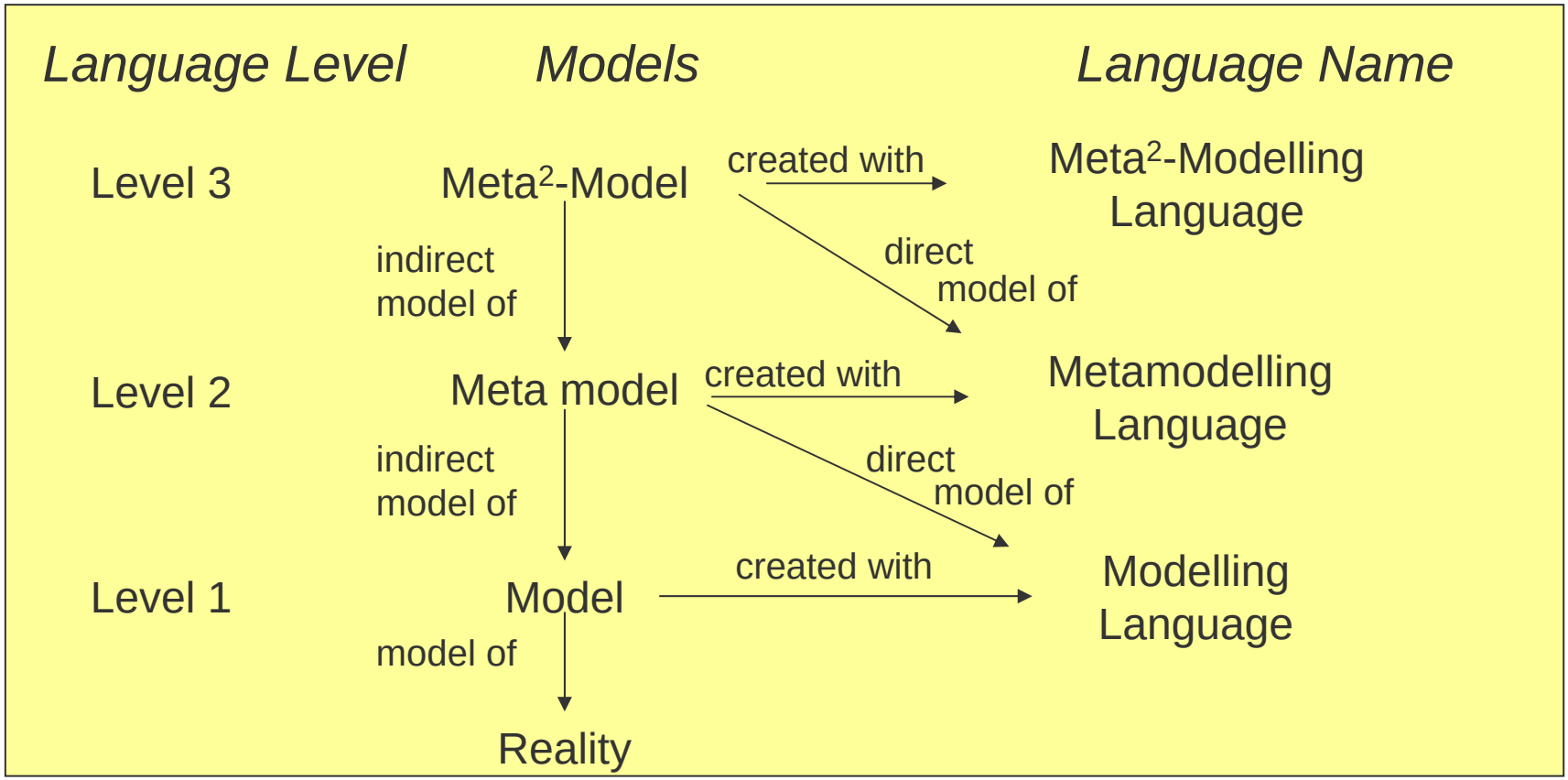
- The metamodel is the definition of the modelling language (or the model of the modelling language)

The relationship between Metamodel and Model for the UML class diagram



Meta Model Hierarchy

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

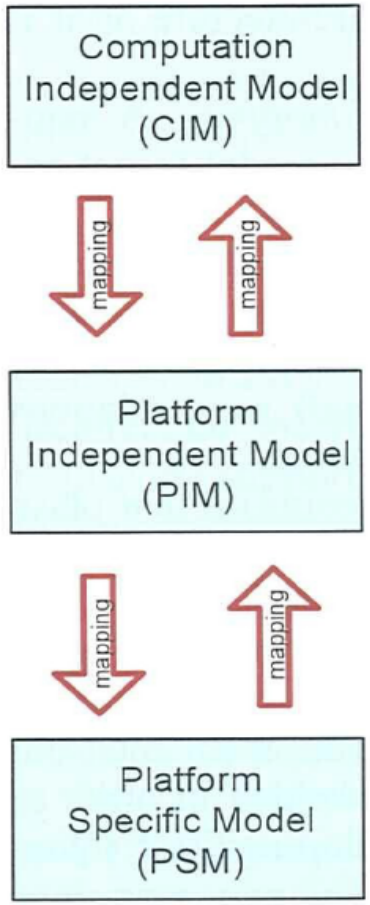


Often the metamodelling language is a kind of natural language, represented in a manual.

4 Layer Metamodel Architecture

Layer	Description	Examples (UML)
Metametamodel	Foundation for a Metamodeling Architecture. Defining the language to describe metamodels	MetaClass, MetaAttribute, MetaOperation
Metamodel	An Instance of a metametamodel. Defining the language to describe models.	Class, Attribute, Operation, Component
Model	An Instance of Metamodel. Defining a language to describe the information object domain.	Product, Unit Price, Customer, Sale, Detail
User Objects **(User Data)**	An Instance of a Model. Defines specific information Domain	<phone>, <Desk>, $100, $200

Model-Driven Architecture MDA

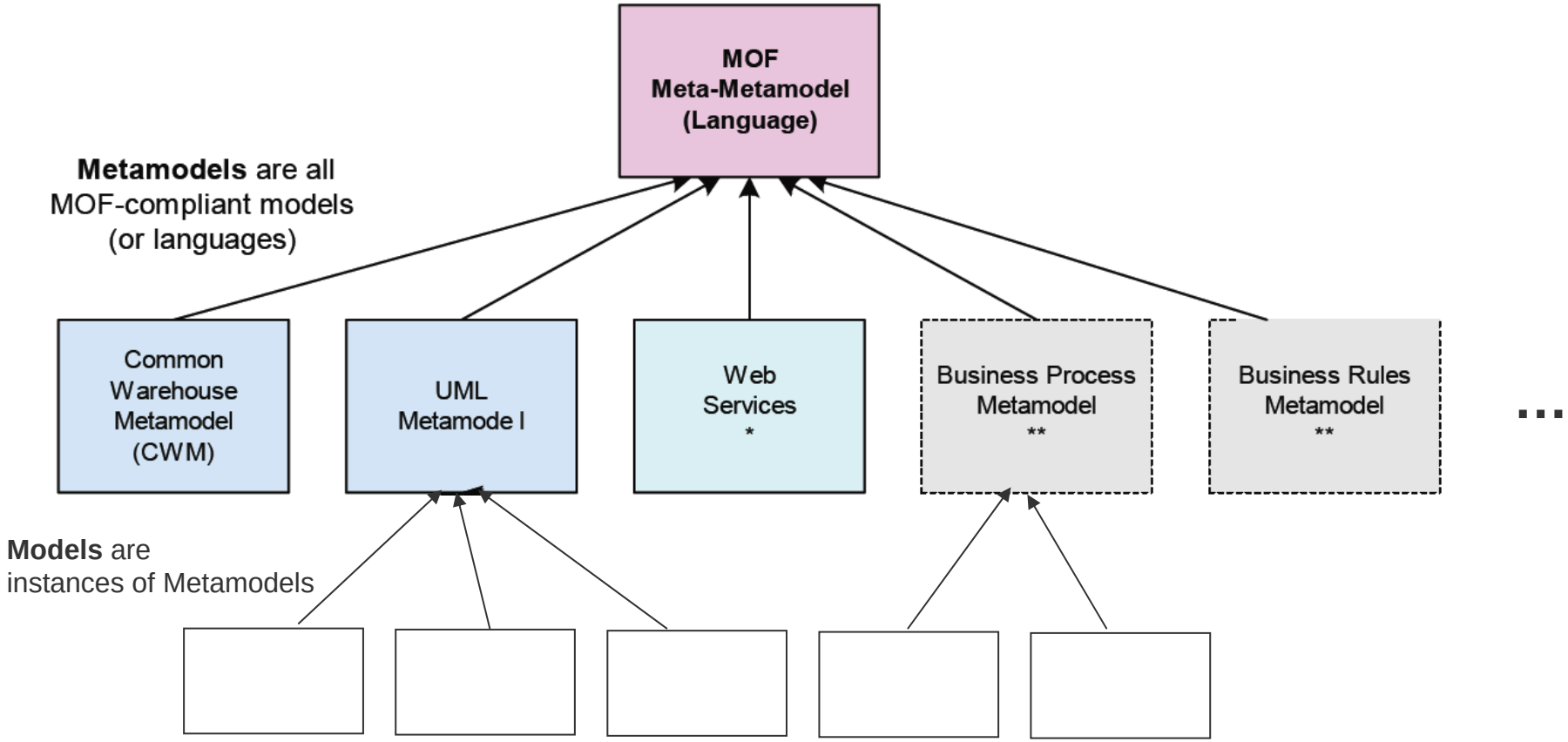


- MDA comprises three levels of abstraction
  - CIM – Computation Independent Model
  - PIM – Platform Independent Model
  - PSM – Platform Specific Model
- For the mapping OMG defined two standards:
  - **XMI - XML Metadata Interchange**
    Standard Syntax for the Exchange of Model
  - **MOF – Meta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

MOF – Meta Object Facility

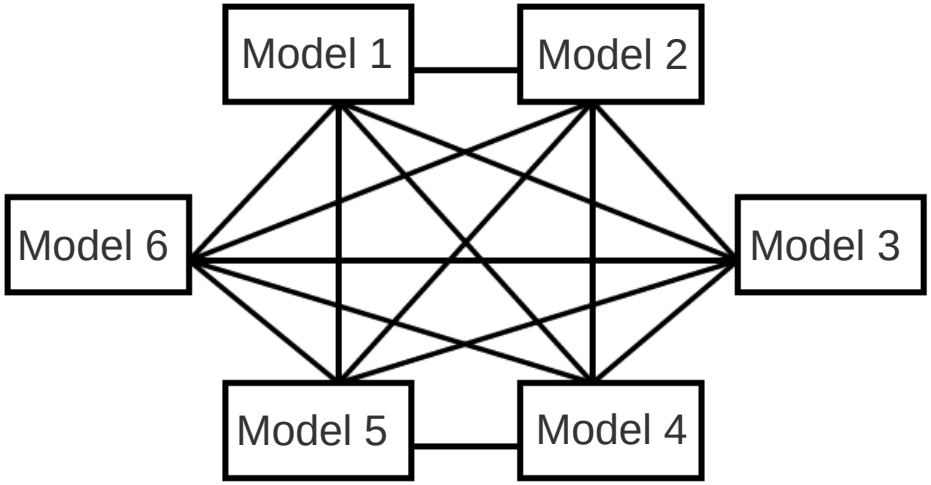
- The theoretical underpinnings of XMI are defined by the Meta Object Facility (MOF).
- MOF is itself a *meta-metamodel*, a specification describing how one may build metamodels.
- Thus, for example, MOF provides a specification for how to model the fact
  - that in a class diagramm class has attributes and operations, or
  - that a Web service has service endpoints.
- This is done using the familiar "boxes and arrows" notation of UML, with some minor constraints necessitated by the nature of metamodeling.

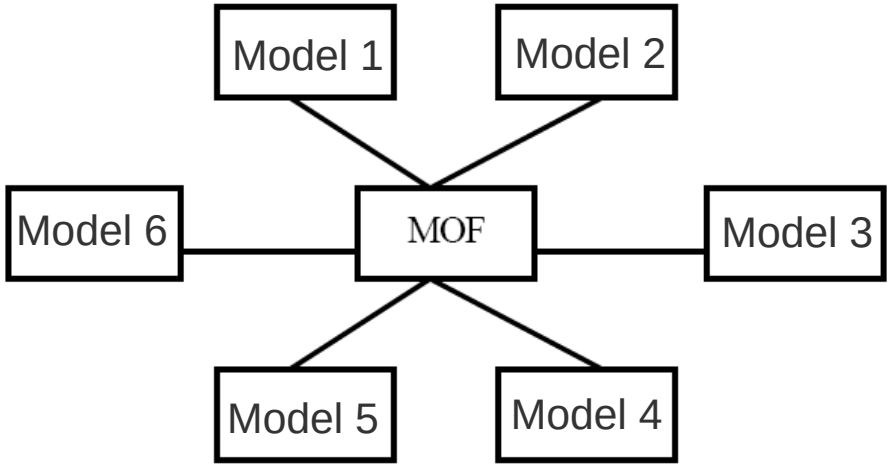
Model Levels of MDA



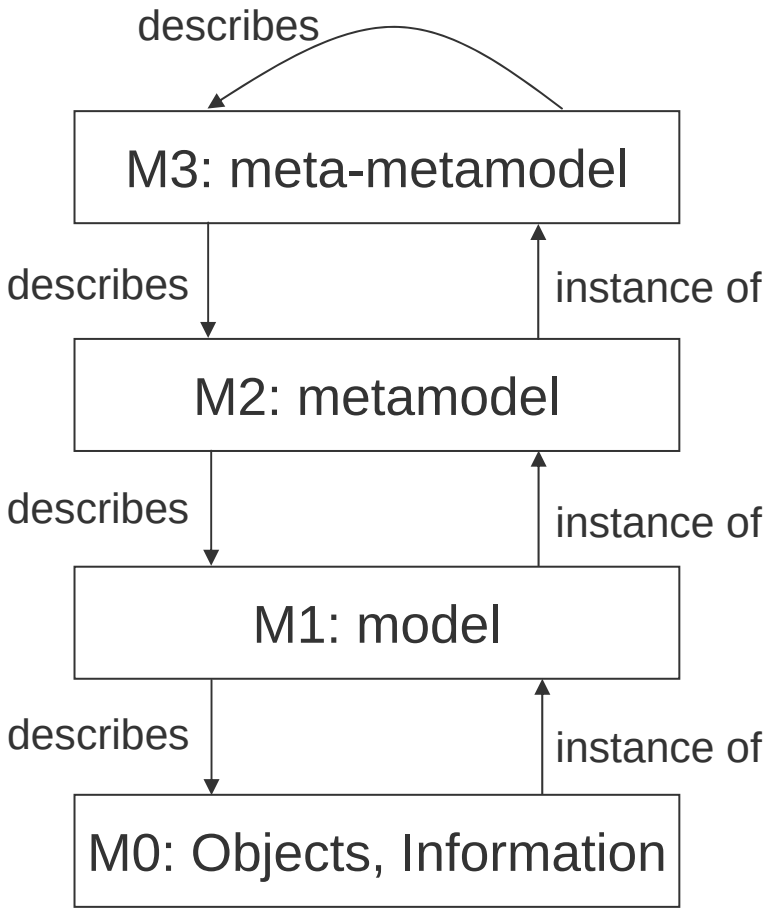
MOF as Intermediate Language for Mappings

- With MOF the number of interfaces between different models and tools can be reduced
- Instead of mappings between each two models, for each model only a mapping to MOF has to be defined





The OMG Model Stack

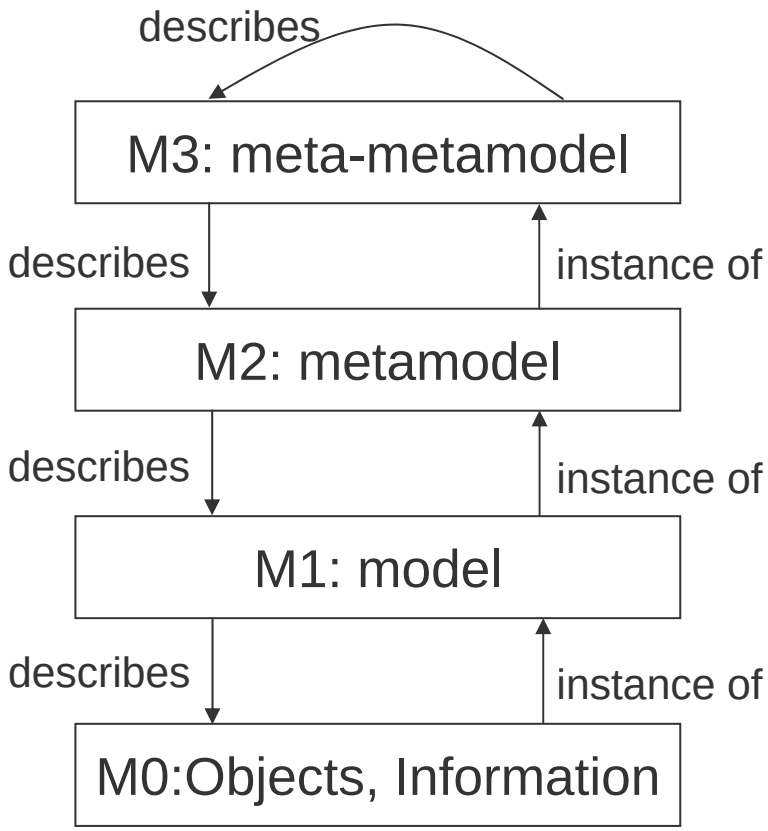


The Meta Object Facility (MOF) distinguishes four levels:

- 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. It is the information describing how the enterprise shares and stores its data.
  - a table customer with a name column
- M2 is the metamodel, or the "IT language". This class is central for 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

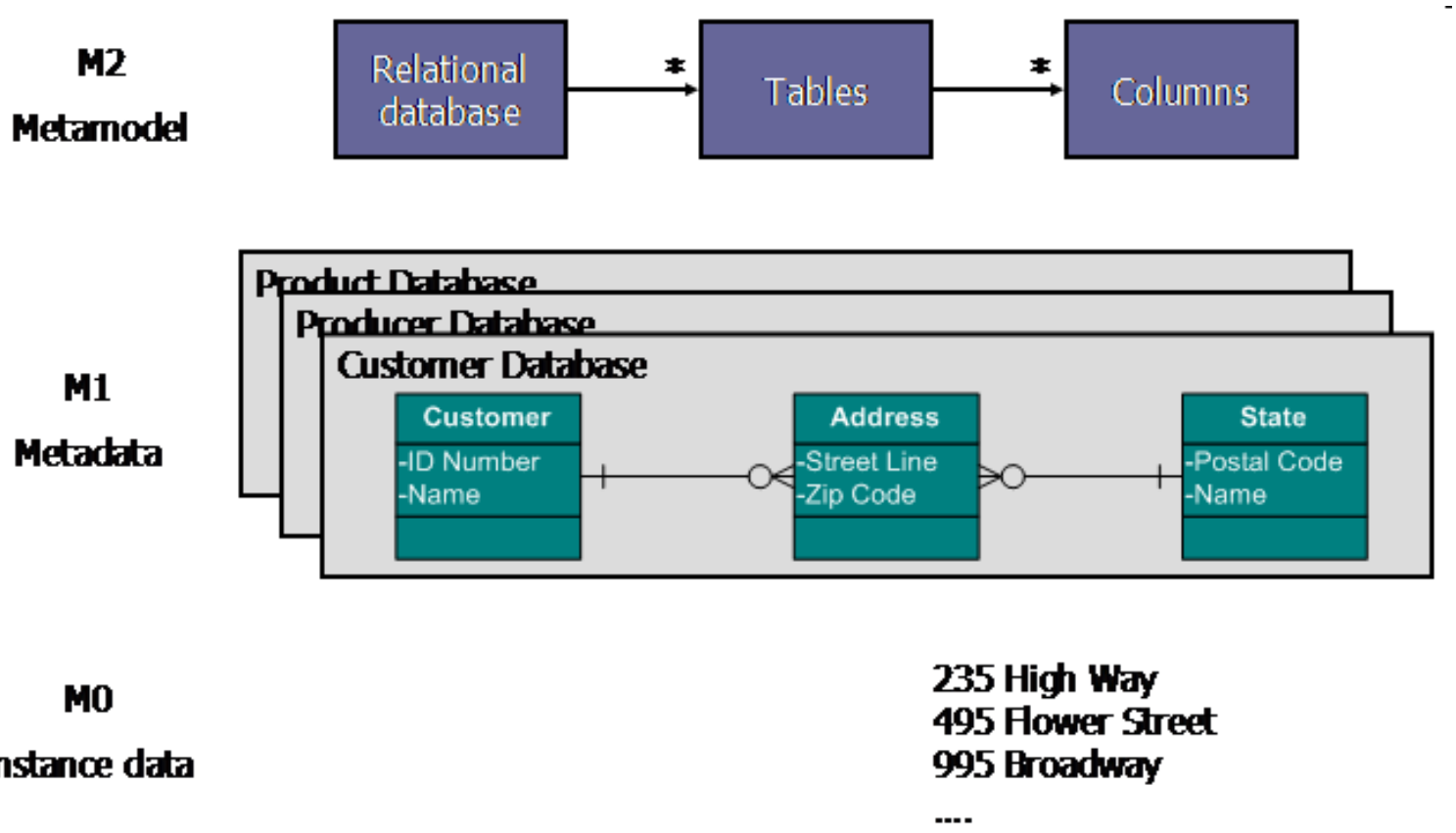
Metamodels in MOF

MOF can represent any meta model, e.g. object-oriented models and relational database

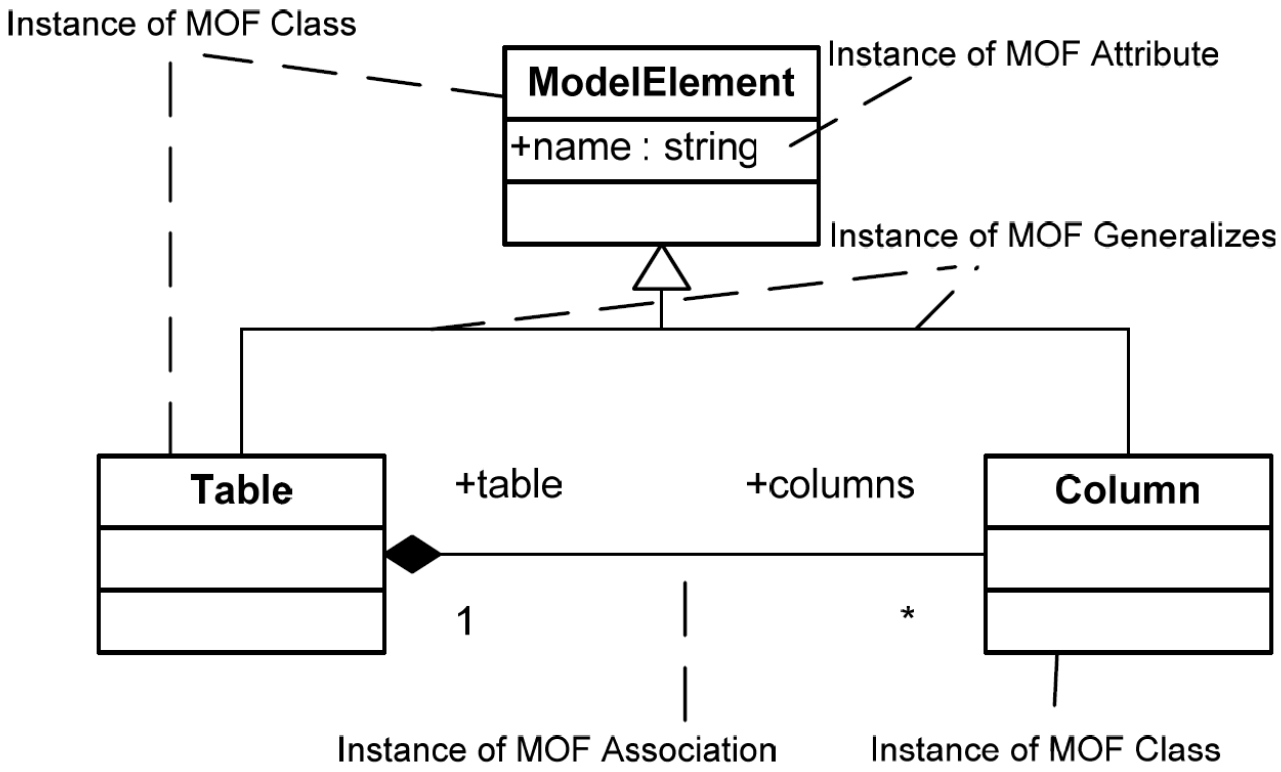


Object-oriented modelling in UML	***relational databases***
MOF Class, MOF Attribute, MOF Association	
UML Class UML Association	RDB tables and columns
class: product class: producer	table product with columns manufacturer and price
product: myphone producer: Apple price: 650.00	product myphone with price "650.00"

Metamodel and Model for Relational Databases



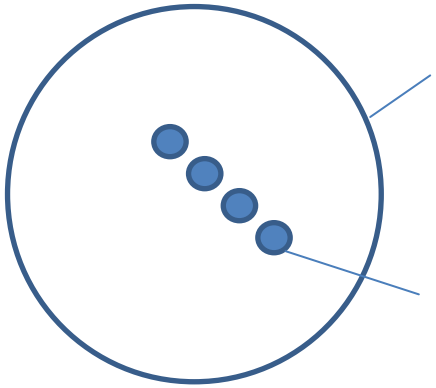
Example of a M2 Metamodel



Basic Idea: How to define an Object

Reality

Set: Employees of company A



M0 Layer

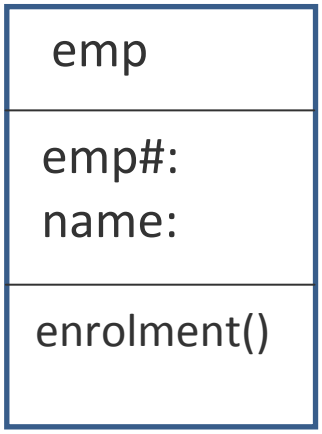
emp #	Name		
0800101	Adam Smith		
0800102	Jon Due		
0800103	Hajime Hori		

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

Object Concept and Metamodel

M1 Layer

Class

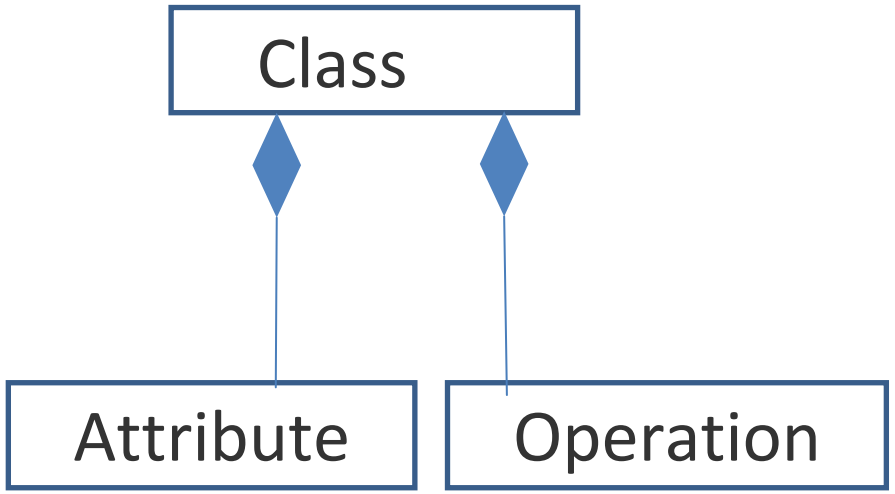


Class name

attribute

operation

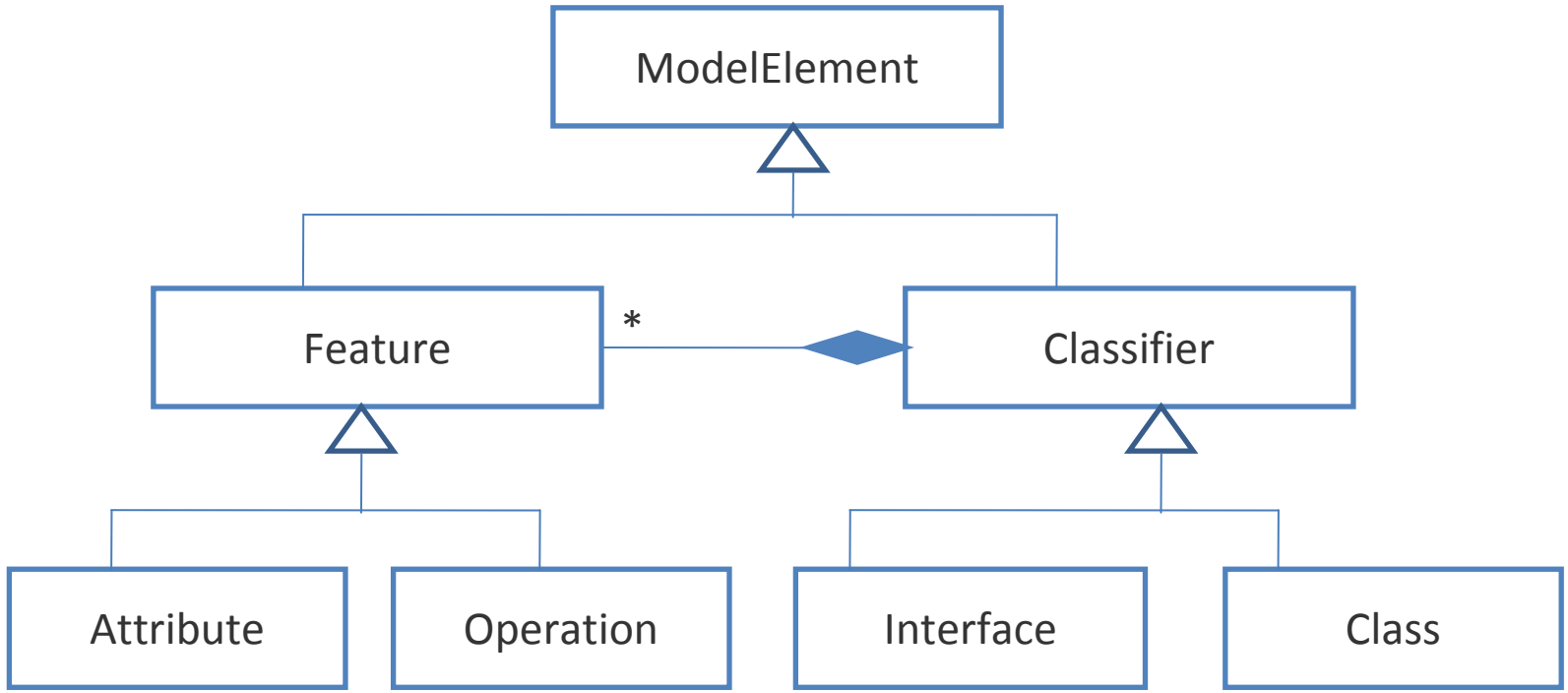
M2 Layer



Class has Attributes and Operations

Overview of M3 Layer

The M3 Layer of MOF is represented in UML

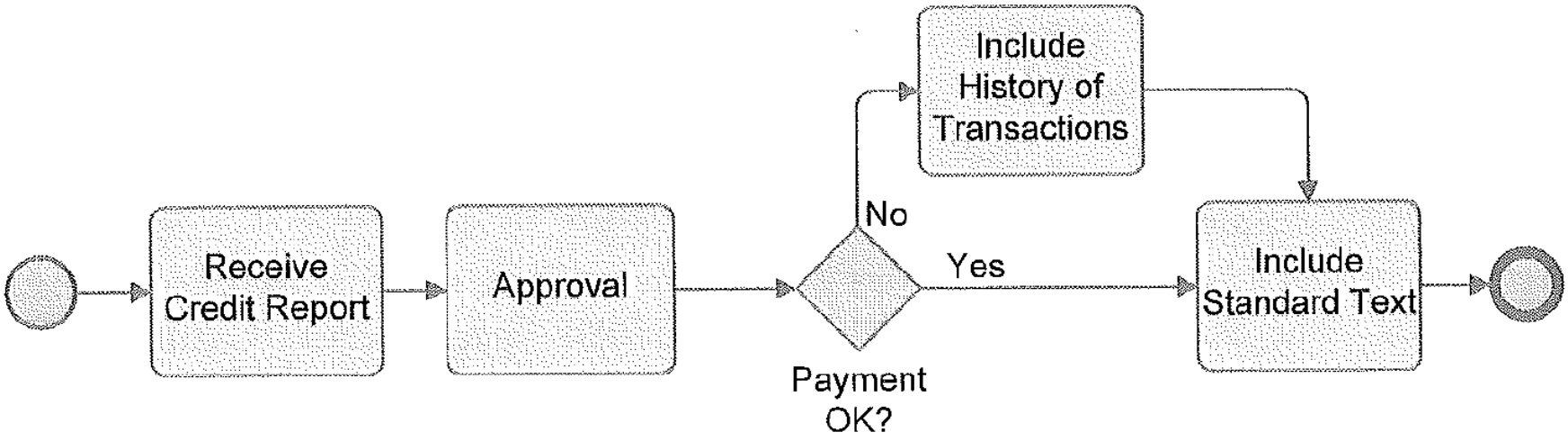


Use Cases for the Meta Levels

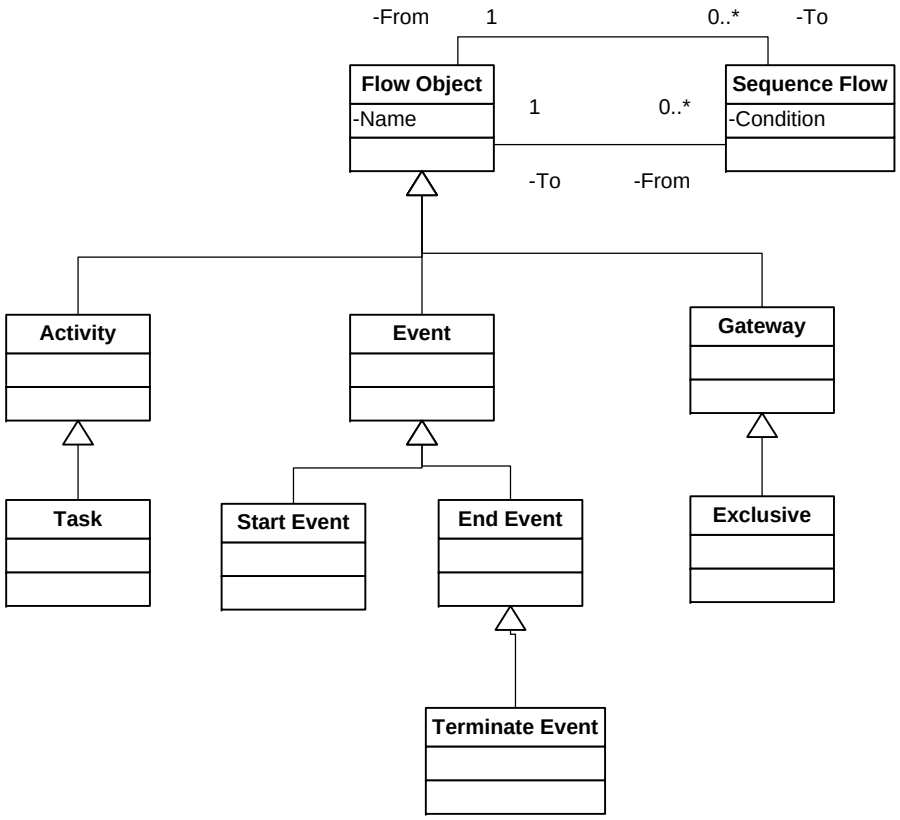
- The different meta-levels have quite different use cases:
  - data is used/provided by the business,
  - metadata is used/provided by IT, and
  - metamodels are used by metadata repositories (allowing metamodels to be configured rather than hard-coded).
- There is generally less metadata than data, 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.

BPMN Metamodel

- Which elements on levels M1 and M2?



M2 Model


-From
1
0..*
-To
Flow Object
-Name
1
0..*
Sequence Flow
-Condition
-To
-From
Activity
Event
Gateway
Task
Start Event
End Event
Exclusive
Terminate Event

MOF2.0 in UML2.0

- MOF 2.0 uses UML Infrastructure Library
- The MOF 2.0 Model is used to model itself as well as other models and other metamodels
- Thus, MOF
  - defines a metalanguage kernel that can define (bootstrap) UML and also be reused to define other OMG metamodels (e.g., MOF, BPMN, Common Warehouse Model CWM, Organisation Description Model ODM)
  - can be used to model arbitrary metadata or language dialects for
    - platforms (e.g., J2EE, .NET),
    - applications (e.g. software configuration or requirements metadata) and
    - domains (e.g., ebusiness, finance, etc.)

XMI – XML Metadata Interchange Standard

- Objective: answer the need for exchanging and storing metadata in a variety of different languages
  - Transforming between MDA model CIM, PIM, PSM
  - Mapping between perspectives of the same aspects in an EA framework (e.g. business process modell, workflow-model)
  - Exchange models between different tools
- In XMI, the metamodel of the given language is described according to the MOF
  - First, XMI gives the same general structure to metadata, regardless of the "language" used (e.g. relational databases, business processes, and class diagrams receive a consistent format).
  - Second, the metamodel of the given metadata language (in other words, the model which constrains the metadata) is also described in a standardized format such as either DTD ord XSD.

Illustration of a class using an instance specification