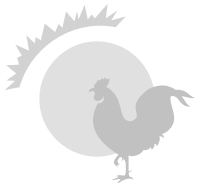


A solid orange vertical bar is positioned to the left of the title.

Metamodelling



Model and Metamodel

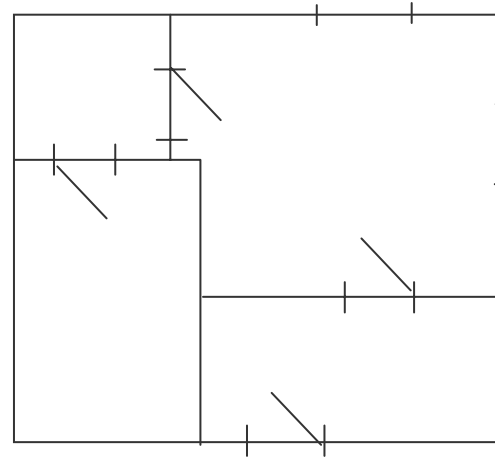
- Defining the modelling objects of a modelling language and their relations itself is a kind of modelling
- The model of a modeling language is called a metamodel
"A metamodel is a model used to model modeling itself."
- To model the modelling language we again need a language which is called a meta language

Model and Modelling Language in Architecture

real object



modell (plan)



modelling language
(metamodel)

object types:

_____ wall

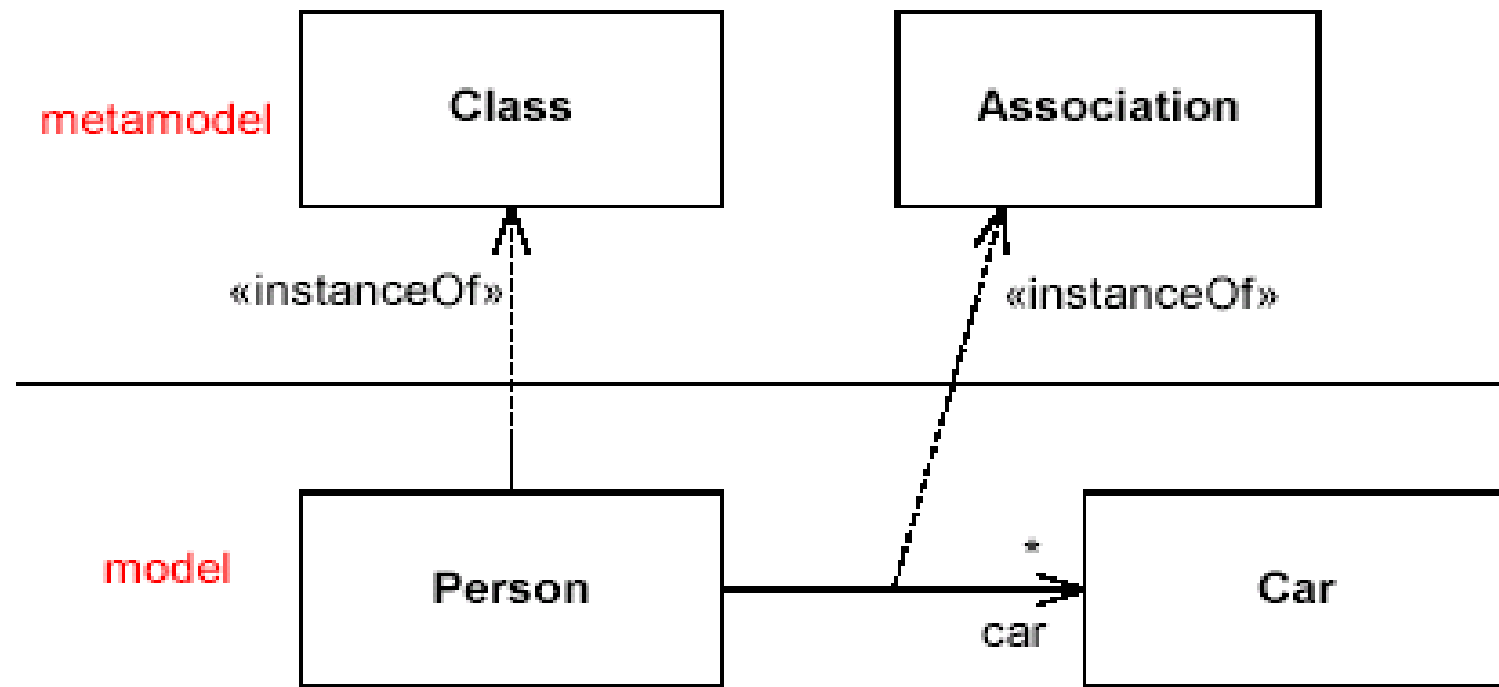
⌋ door

+—+ window

rules:

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

The relationship between Metamodel and Model



Meta Model Hierarchy

Language Level Models Language Name

Level 3

Meta-metamodel

described with → Meta²-Modelling Language

indirect
model of

direct model of

Level 2

Metamodel

described with → Metamodelling Language

indirect
model of

direct model of

Level 1

Model

described with → Modelling Language

indirect
model of

User data



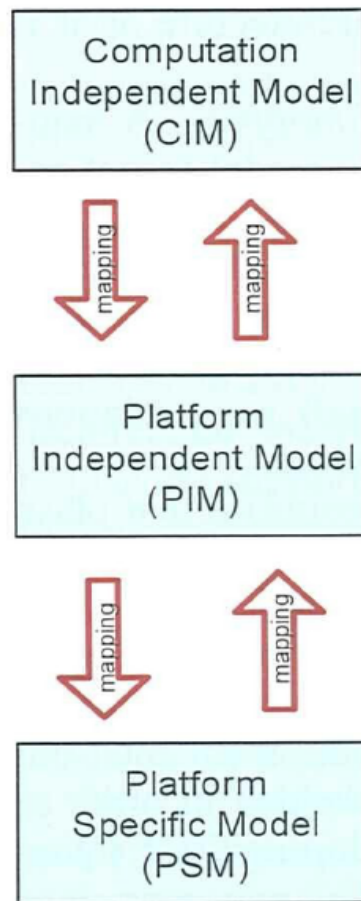
4 Layer Metamodel Architecture

Layer	Description	Examples
Metametamodel	Foundation for a Metamodeling Architecture. Defining the language to describe metamodels	MetaClass, MetaAttribute, MetaOperation
Metamodel	An Instance of a metamodel. 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

Use of explicit Metamodels

- A metamodel is a model used to model modeling itself.
- Metamodels 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

Model-Driven Architecture MDA



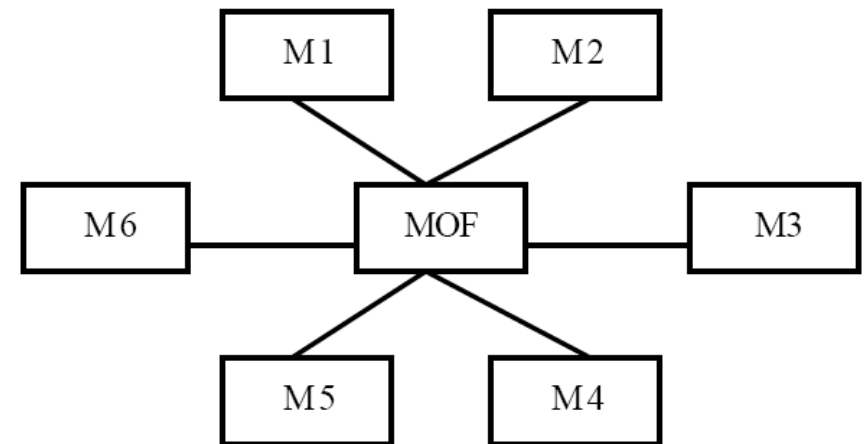
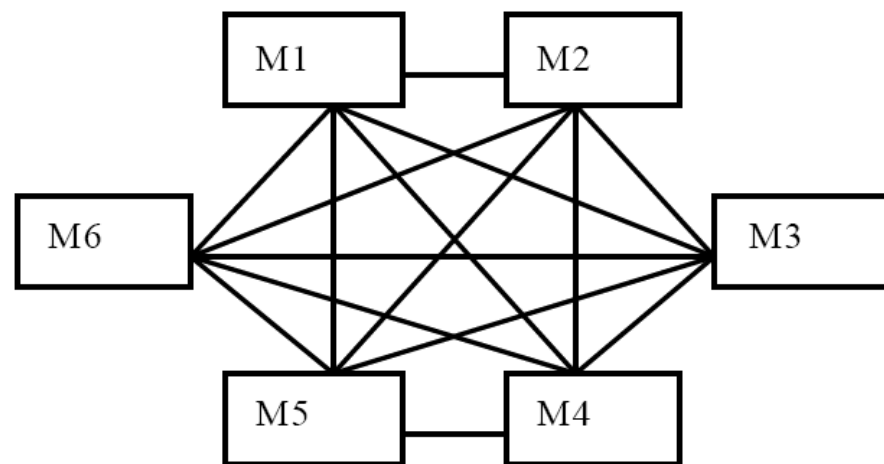
- MDA comprises three levels of abstraction
 - ◆ CIM – Computation Independent Model
 - ◆ PIM – Platform Independent Model
 - ◆ PSM – Platform Specific Model
- For the mapping OMG defined two standards:
 - ◆ **XMI - XML Metadata Interchange**
Standard Syntax for the Exchange of Model
 - ◆ **MOF – Meta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

MOF – Meta Object Facility

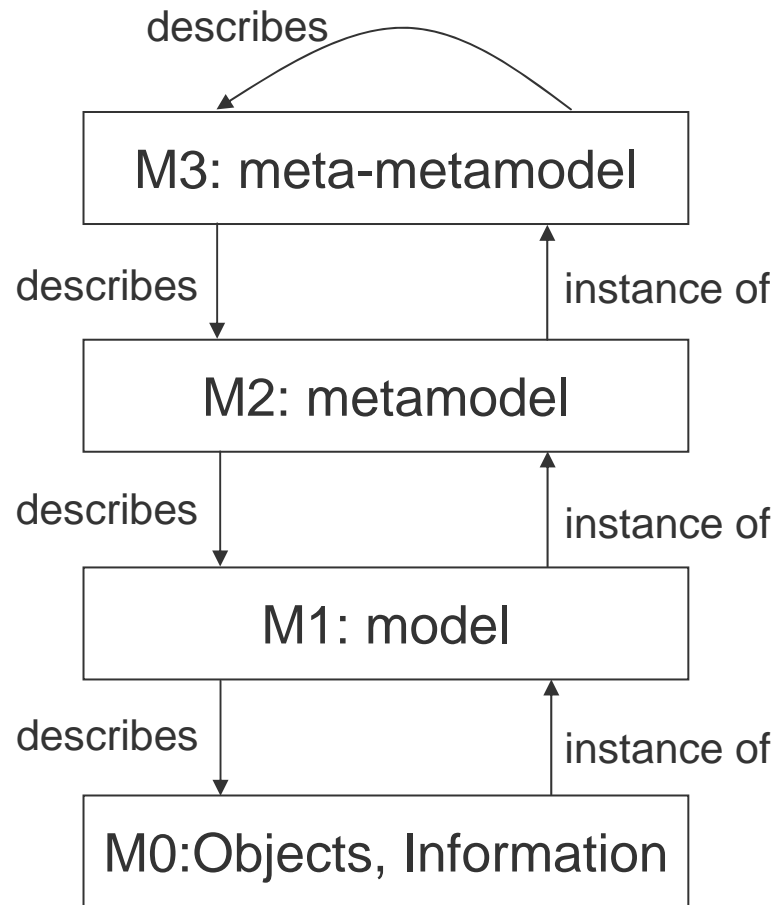
- The theoretical underpinnings of XMI are defined by the Meta Object Facility (MOF), an OMG metamodeling standard closely based on Unified Modeling Language (UML).
- MOF is itself a *meta-metamodel*, a specification describing how one may build metamodels.
- Thus, for example, MOF provides a specification for how to model the fact that a 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.

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

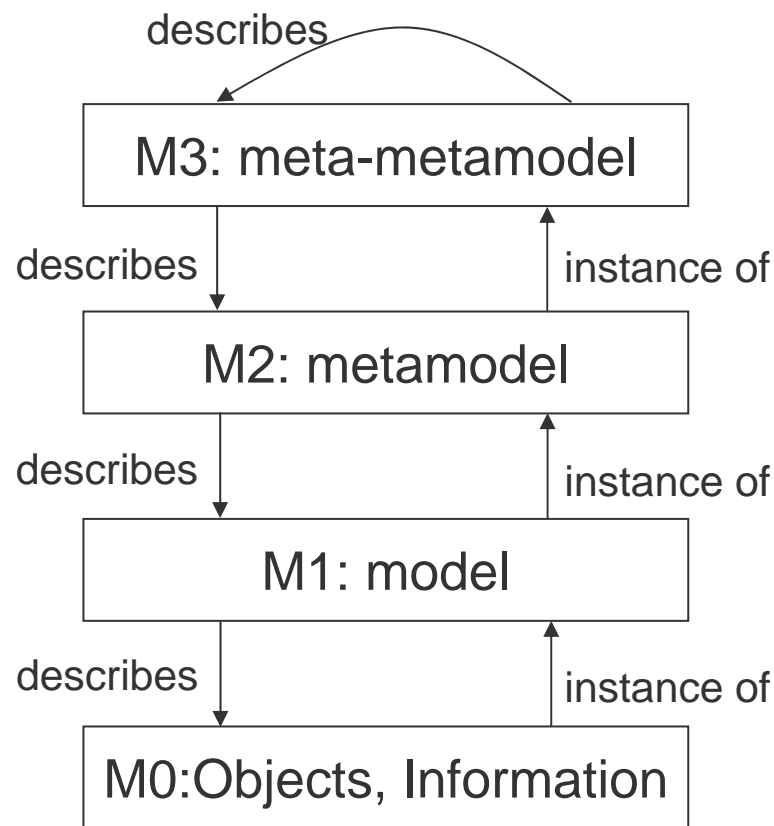


The Meta Object Facility (MOF) distinguishes four levels:

- 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"

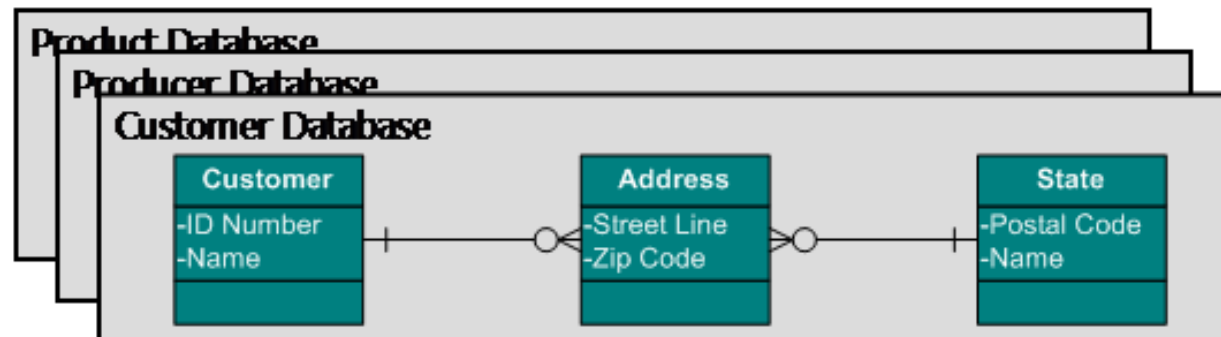


Metamodel and Model for Relational Databases

M2
Metamodel



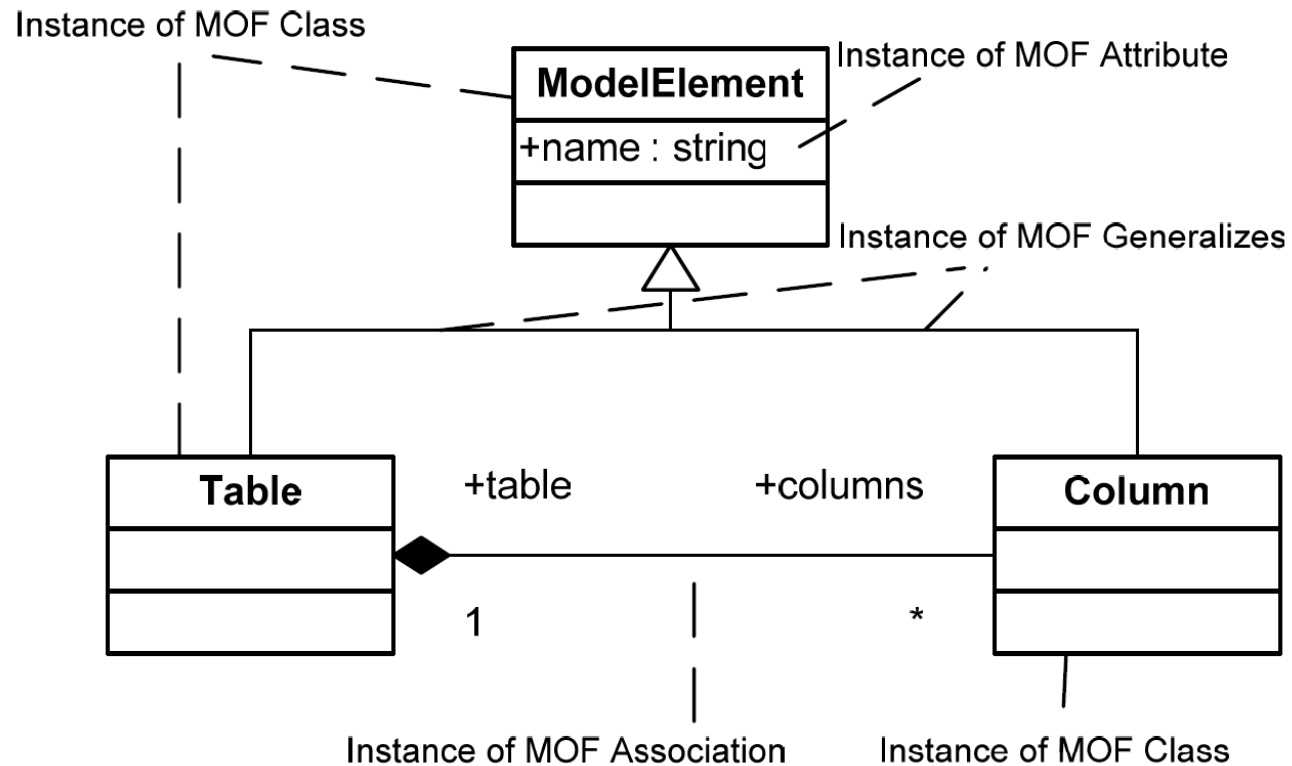
M1
Metadata



M0
Instance data

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

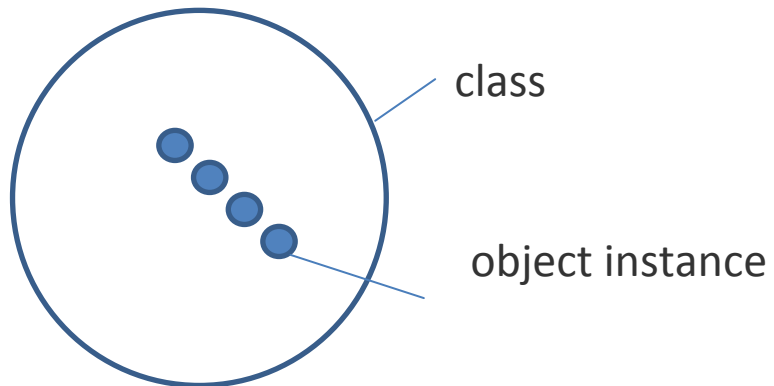
Example of a M2 Metamodel



Basic Idea: How to define an Object

Reality

Set: Employees of company A



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

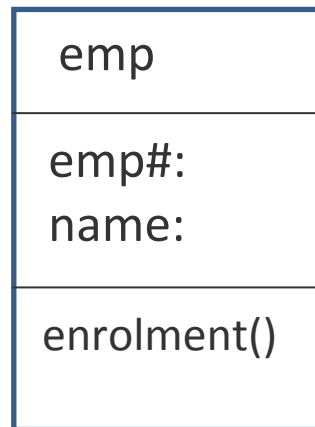
M0 Layer

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

Object Concept and Metamodel

M1 Layer

Class



Class name

attribute

operation

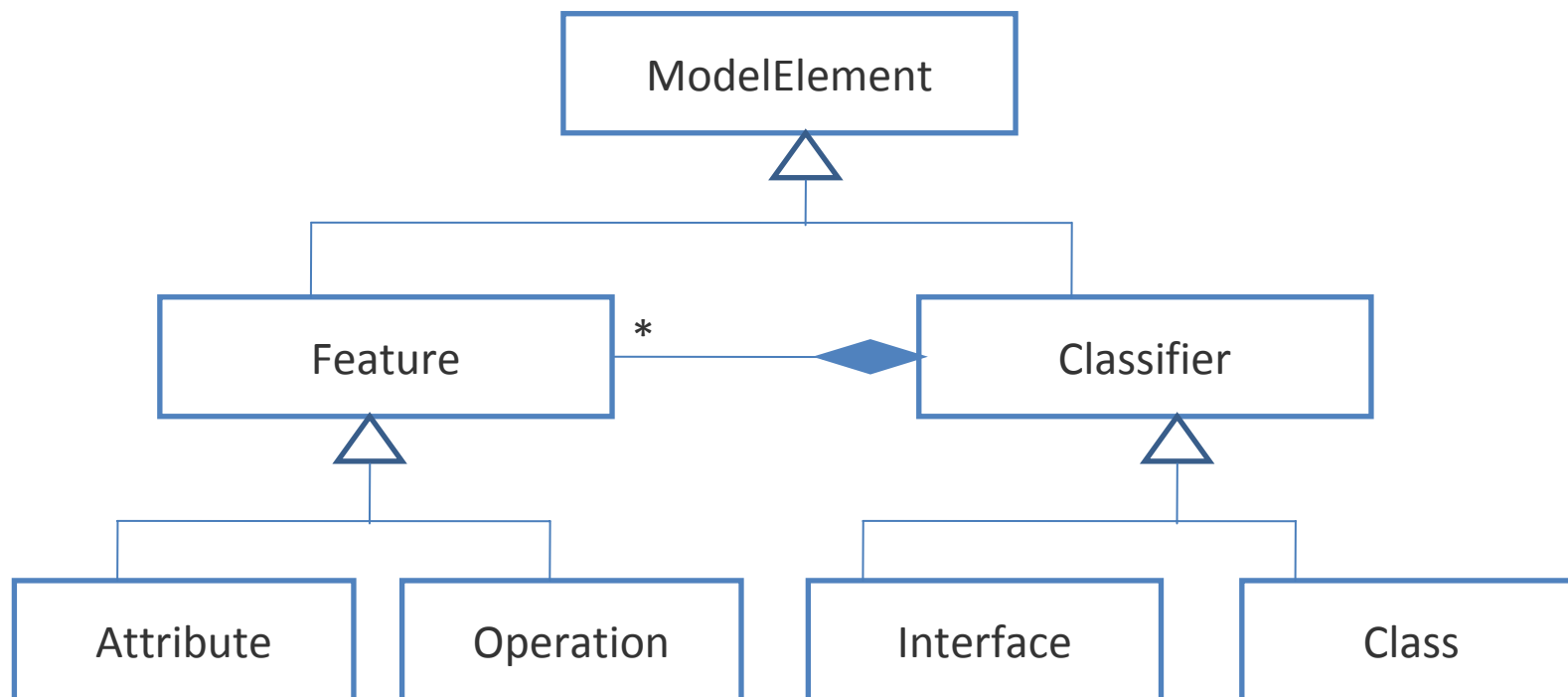
M2 Layer



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 by the business,
 - ◆ metadata is used 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.

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 (such as UML 2.0 and CWM etc.).
- Thus, MOF
 - ◆ defines a metalanguage kernel that can define (bootstrap) UML and also be reused to define other OMG MDA metamodels (e.g., MOF, BPMN, Common Warehouse Model CWM, Organisation Description Model ODM)
 - ◆ provides more powerful mechanisms to customize UML
 - ◆ 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
 - ◆ 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 or XSD.