
odeling
food for thought
etamodeling
egamodeling
tooling
with
and Jean-Marie

Objectives

After this course you should:

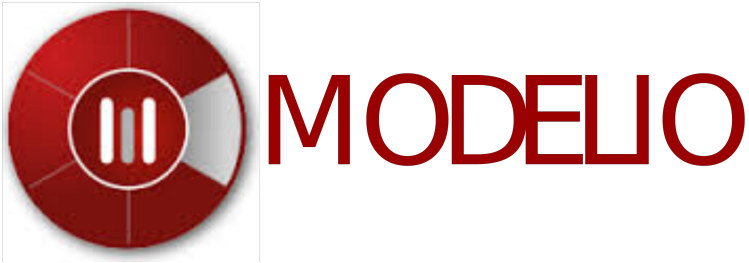
- know more about **(precise) modeling**
- understand what (meta/mega)modeling is about
- (be able to) create your own **metamodels**
- (be able to) use advanced features of UML **tools**
- (be able to) create extensions for UML

Part of the course "Advanced Engineering of Information Systems"

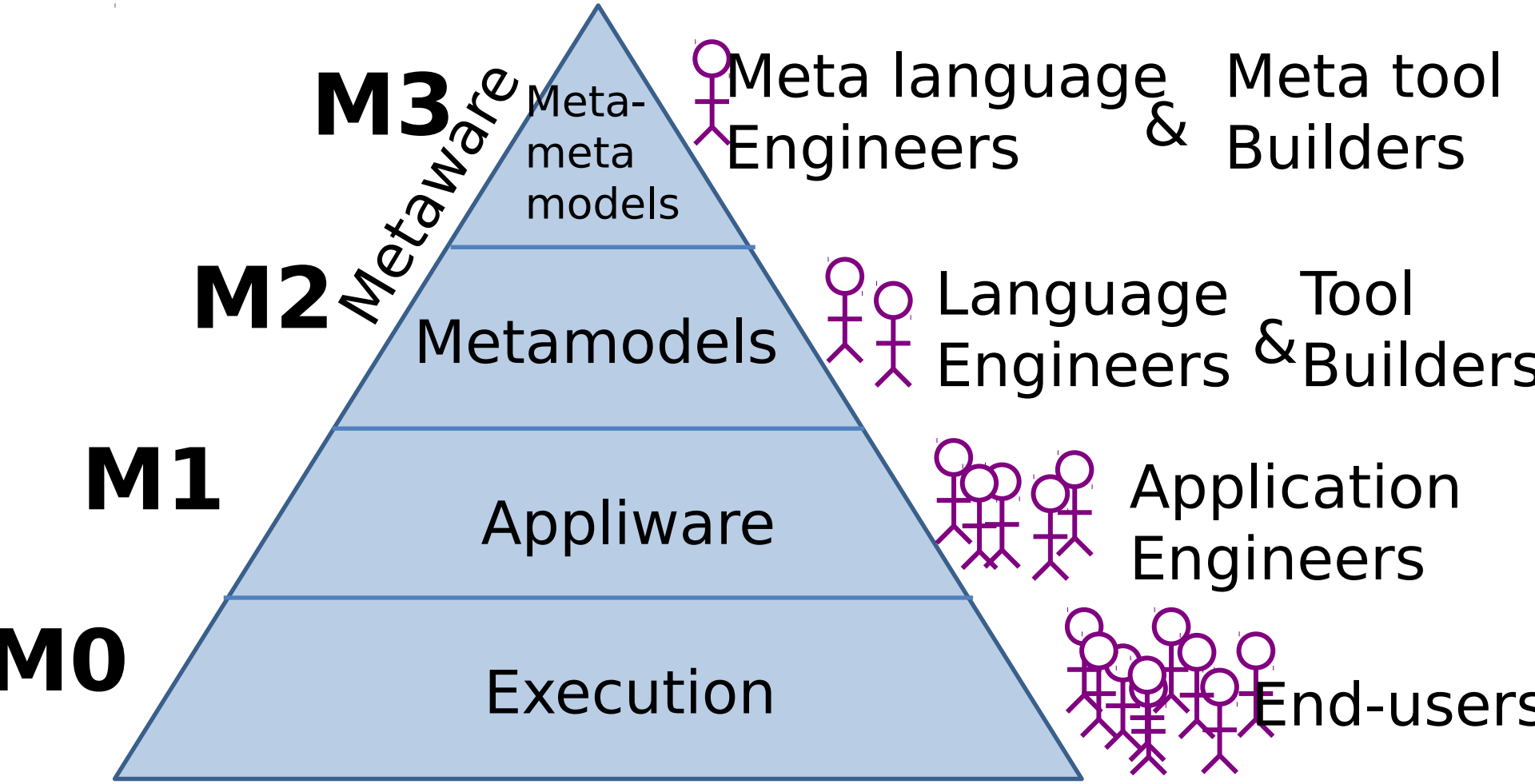
- UML takes its roots in IS modeling
- IS can be very complex and heterogeneous
- Open Enterprise
  - Many organization, many people, many languages
  - Many models, many metamodels, many tools
  - ==> Megamodels

In Practice

- Precise modeling
  - with OCL
- Metamodels and Meta-metamodels
  - exercices
  - close look at UML metamodel
- UML environment
  - usage
  - extension



Pyramid of Actors




M3
food for thought
odeling

Motto #1

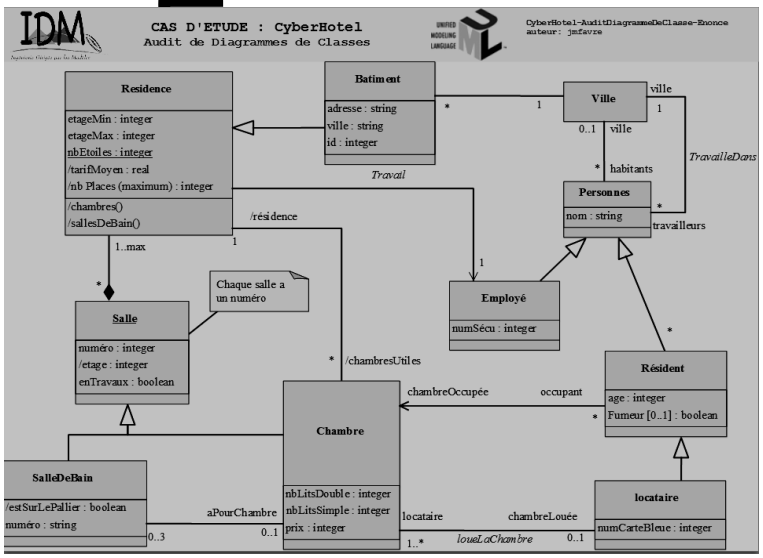
Model everything!

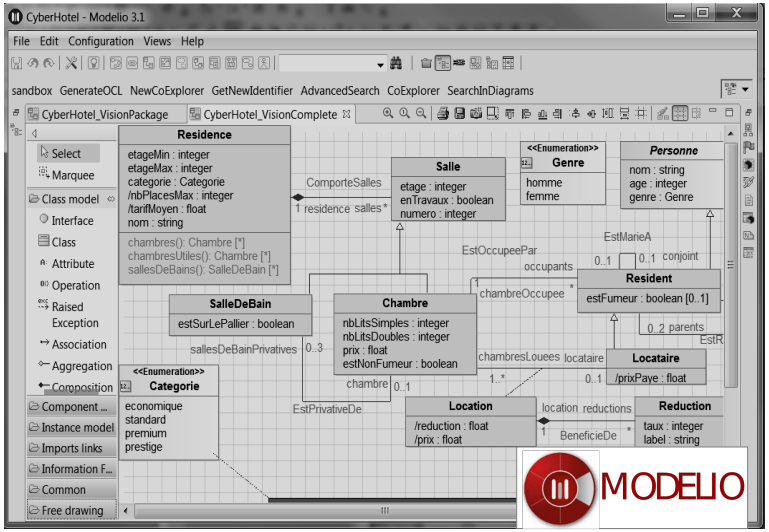
Models

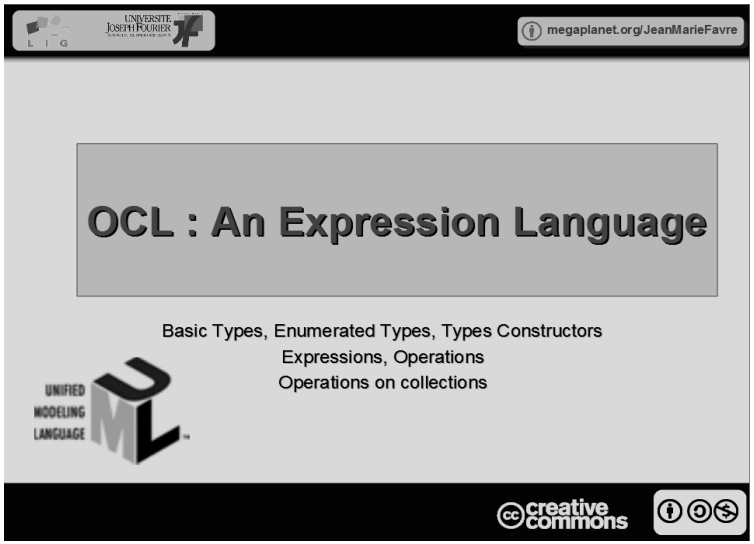
A **model** is a simplified representation of a system elaborated for a given purpose.

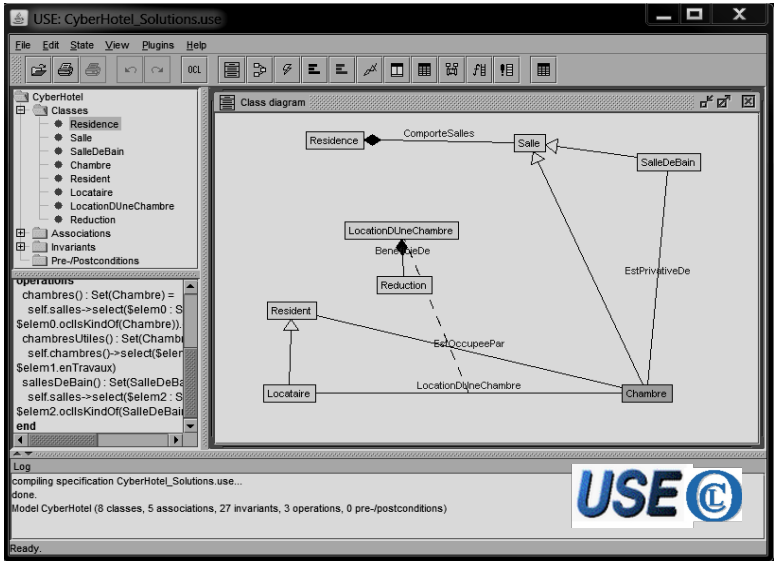
3Modeling

Precis Modeling with e

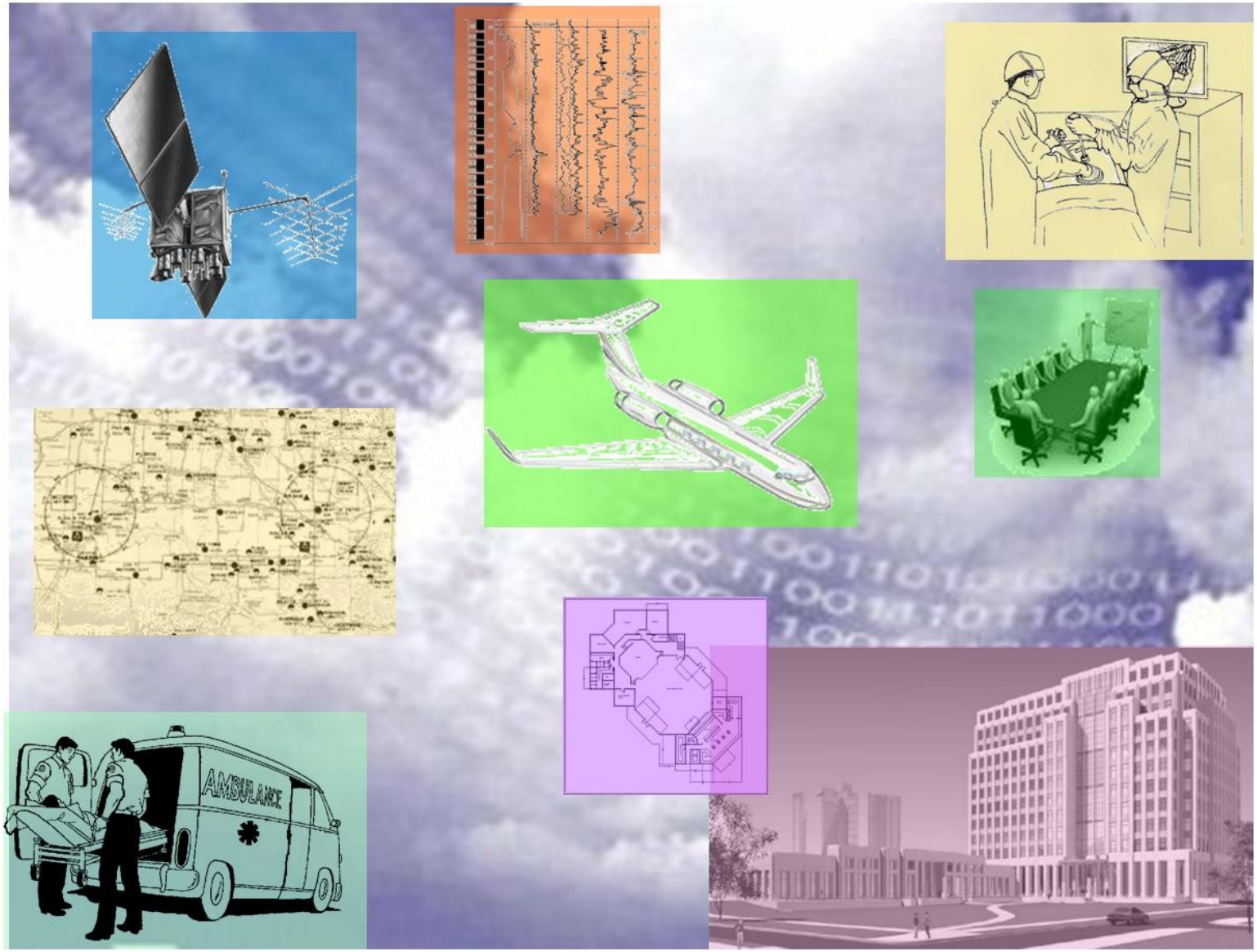








Ultra-large Scale Systems

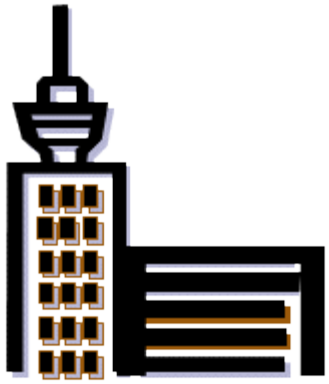


Complex Systems & Stakeholders

Model for firefigthters

Models for architects

Model for cadastre



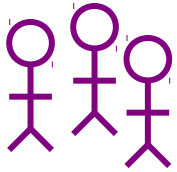
system

Models for plumbers

Model for banks

Models for electricians

Model for insurance



Models for software engineers

Model for notaries

Models for tenants

Modeling -in-the-large

Dealing with complexity

Many models

Many languages

Many modelers

Many and stakeholders

Many modeling tools

Many standardization bodies

...

Metamodeling
Megamodeling

Modeling Language Ecosystems

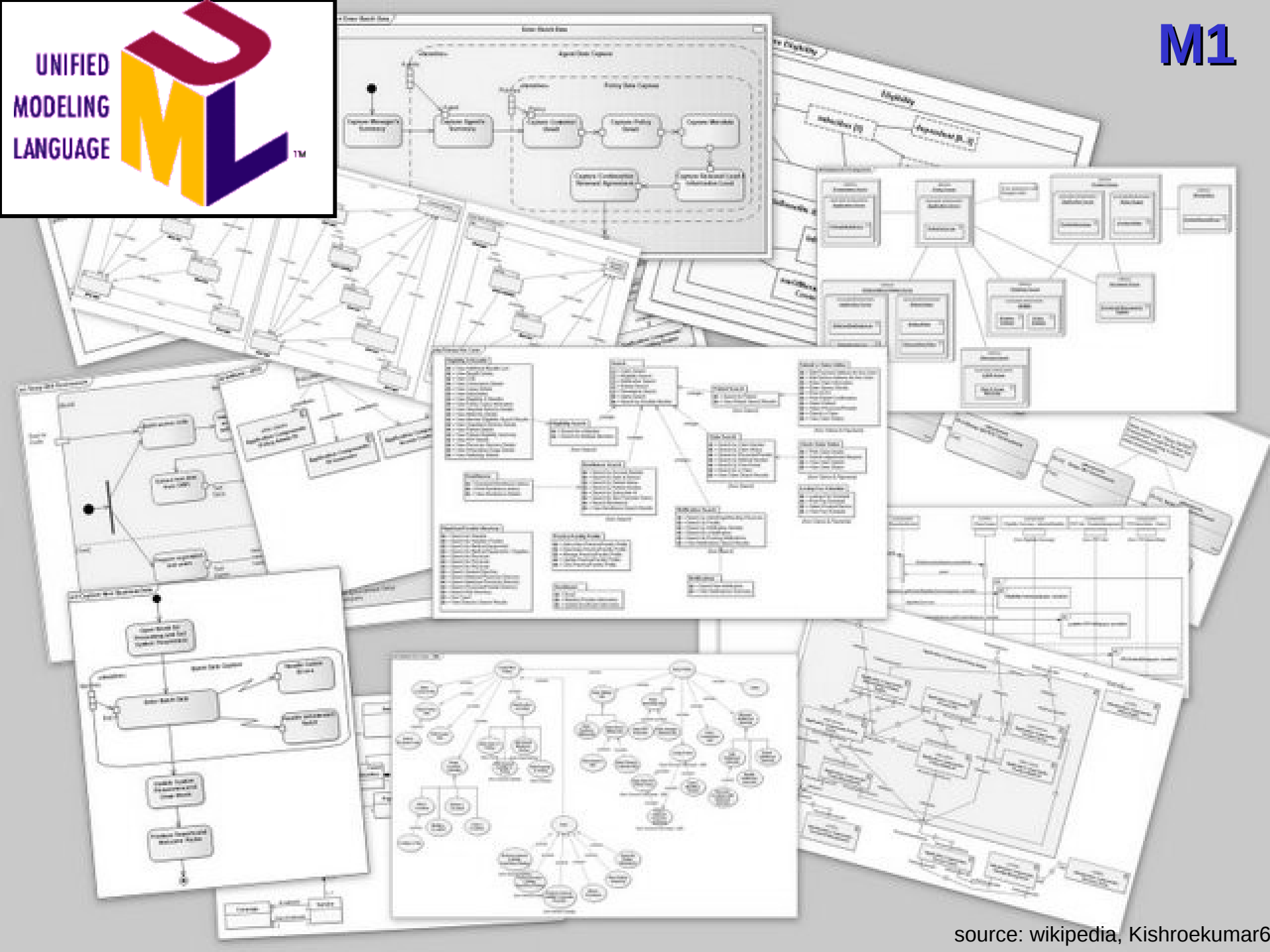
2454

133

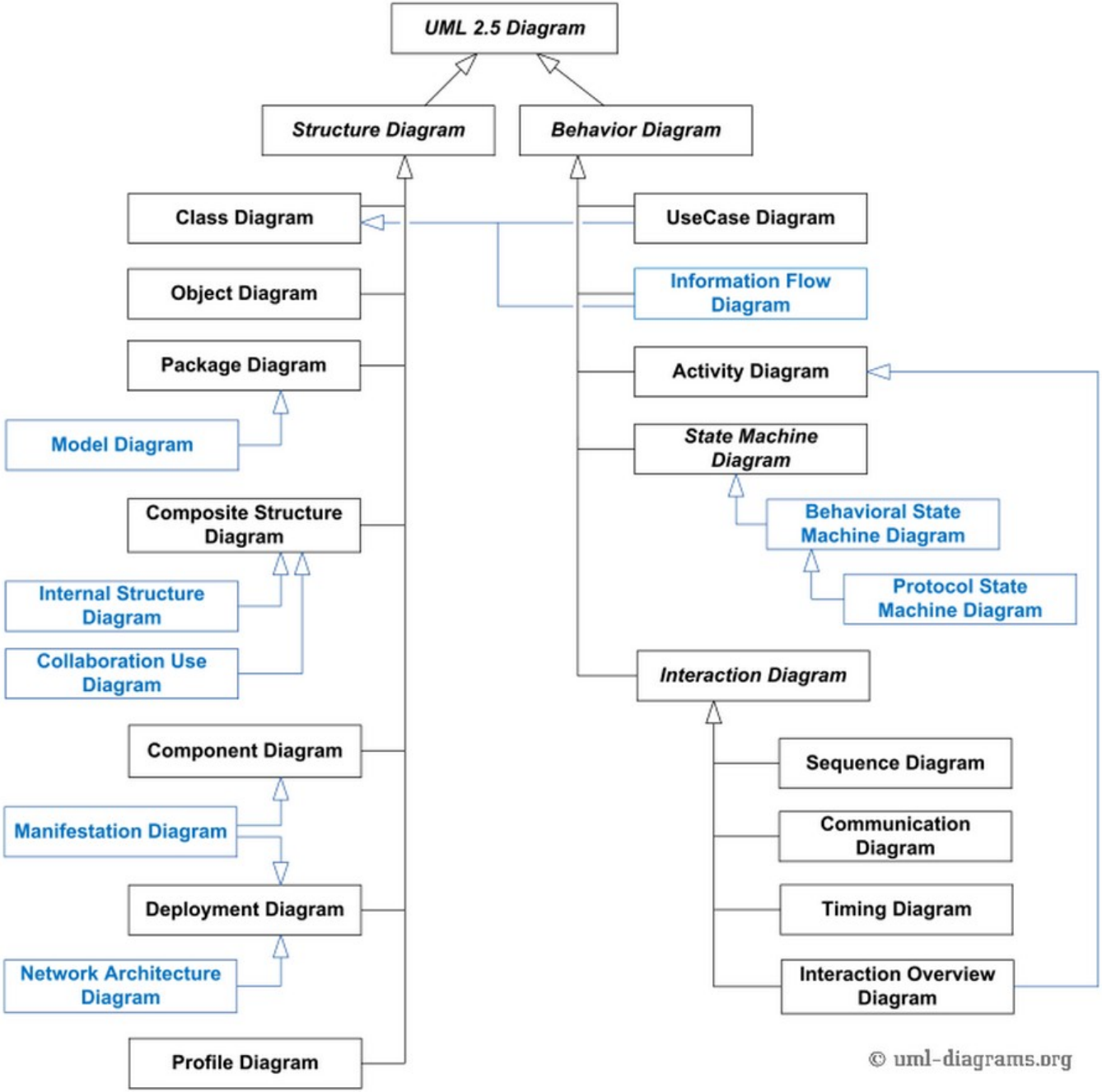
72

28

7



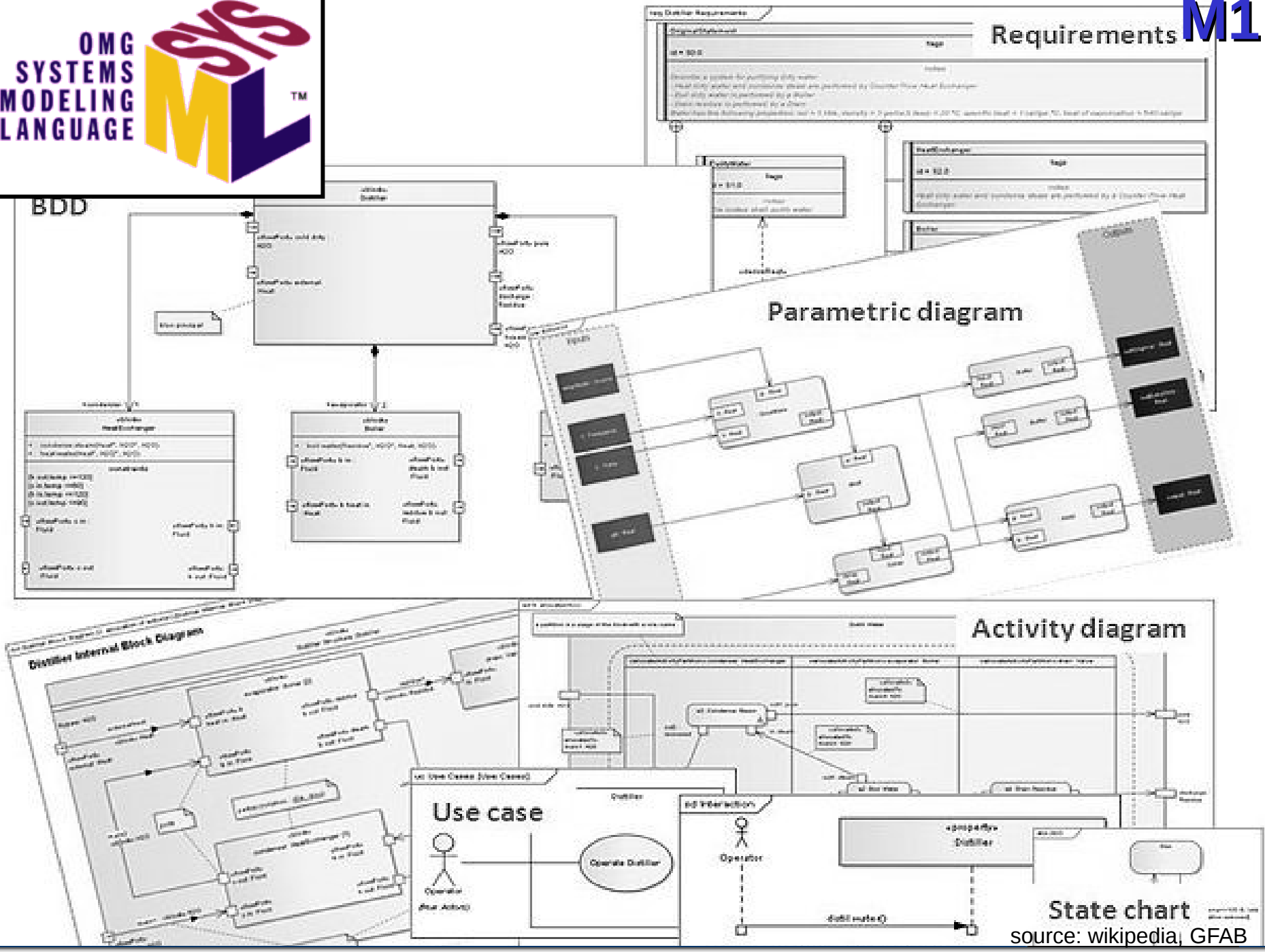
source: wikipedia, Kishroekumar6



UML 2.5 Diagrams Overview.
Note, items in blue are not part of official taxonomy of UML 2.5 diagrams.

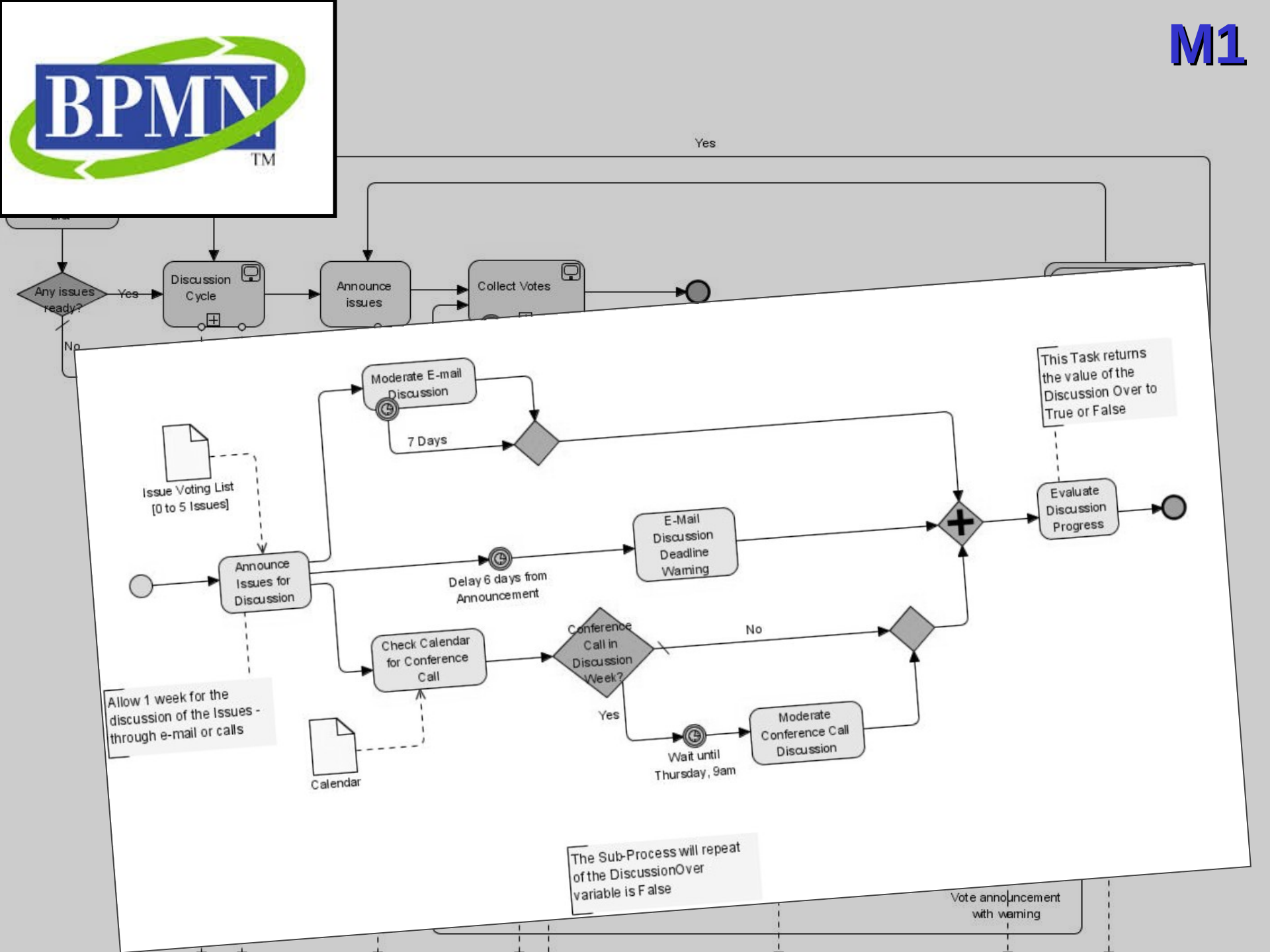
OMG SYSTEMS MODELING LANGUAGE™

M1



source: wikipedia, GFAB

BPMN™


Yes
Any issues ready?
Yes
Discussion Cycle
Announce issues
Collect Votes
No
Moderate E-mail Discussion
7 Days
This Task returns the value of the Discussion Over to True or False
Issue Voting List [0 to 5 Issues]
Announce Issues for Discussion
Delay 6 days from Announcement
E-Mail Discussion Deadline Warning
Evaluate Discussion Progress
Check Calendar for Conference Call
Conference Call in Discussion Week?
No
Yes
Moderate Conference Call Discussion
Wait until Thursday, 9am
Allow 1 week for the discussion of the Issues - through e-mail or calls
Calendar
The Sub-Process will repeat of the DiscussionOver variable is False
Vote announcement with warning

SBVR

M1

Rental Industry
inition: the semantic community that *is* the group of people who work in the business of renting cars

al Business
inition: the body of shared meanings that *is* the set *of* meanings that are generally accepted as important across **The Car Rental Industry**

Necessity: **Car Rental Business** *unites* **The Car Rental Industry**.

EU-Rent
Definition: the international car rental company that trades as 'EU-Rent'

The EU-Rent Community
Definition: the semantic community that *includes* all employees of EU-Rent and all others who share their body of meanings and use their vocabularies

Necessity: **The EU-Rent Community** *is* a subcommunity *of* **The Car Rental Industry**.

EU-Rent Car Rental Business
Definition: the body of shared meanings that *is* the set *of* meanings that are important to **The EU-Rent Community**

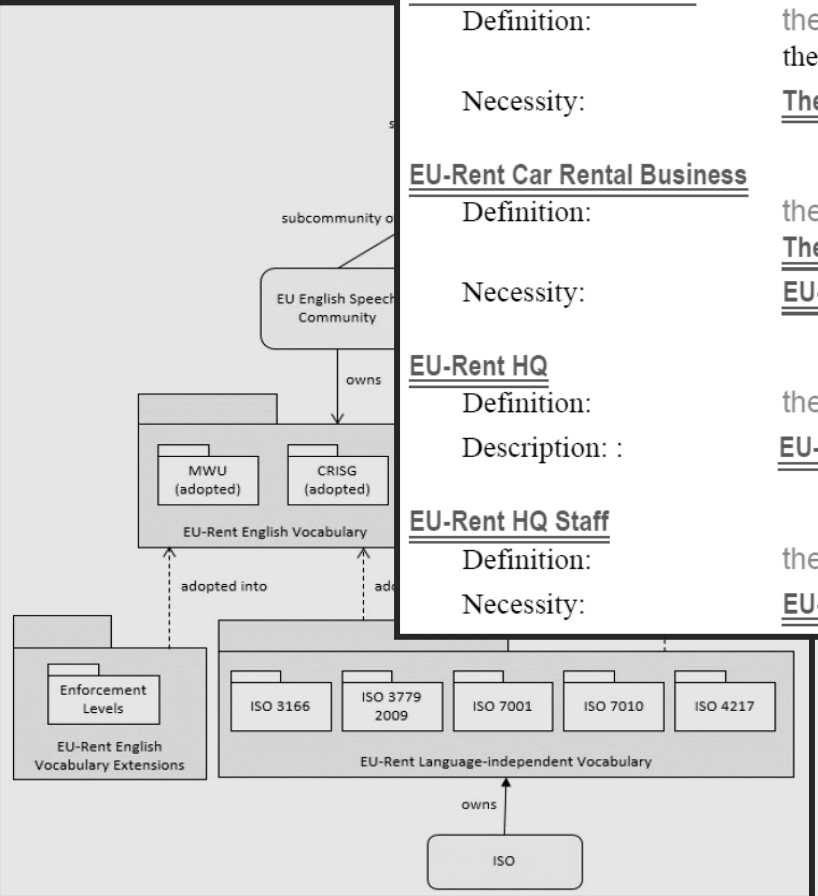
Necessity: **EU-Rent Car Rental Business** *unites* **The EU-Rent Community**.

EU-Rent HQ
Definition: the organization unit that *is* **EU-Rent**'s world headquarters and management company

Description: : **EU-Rent HQ** sets global policy and owns **EU-Rent**'s worldwide reservations system.

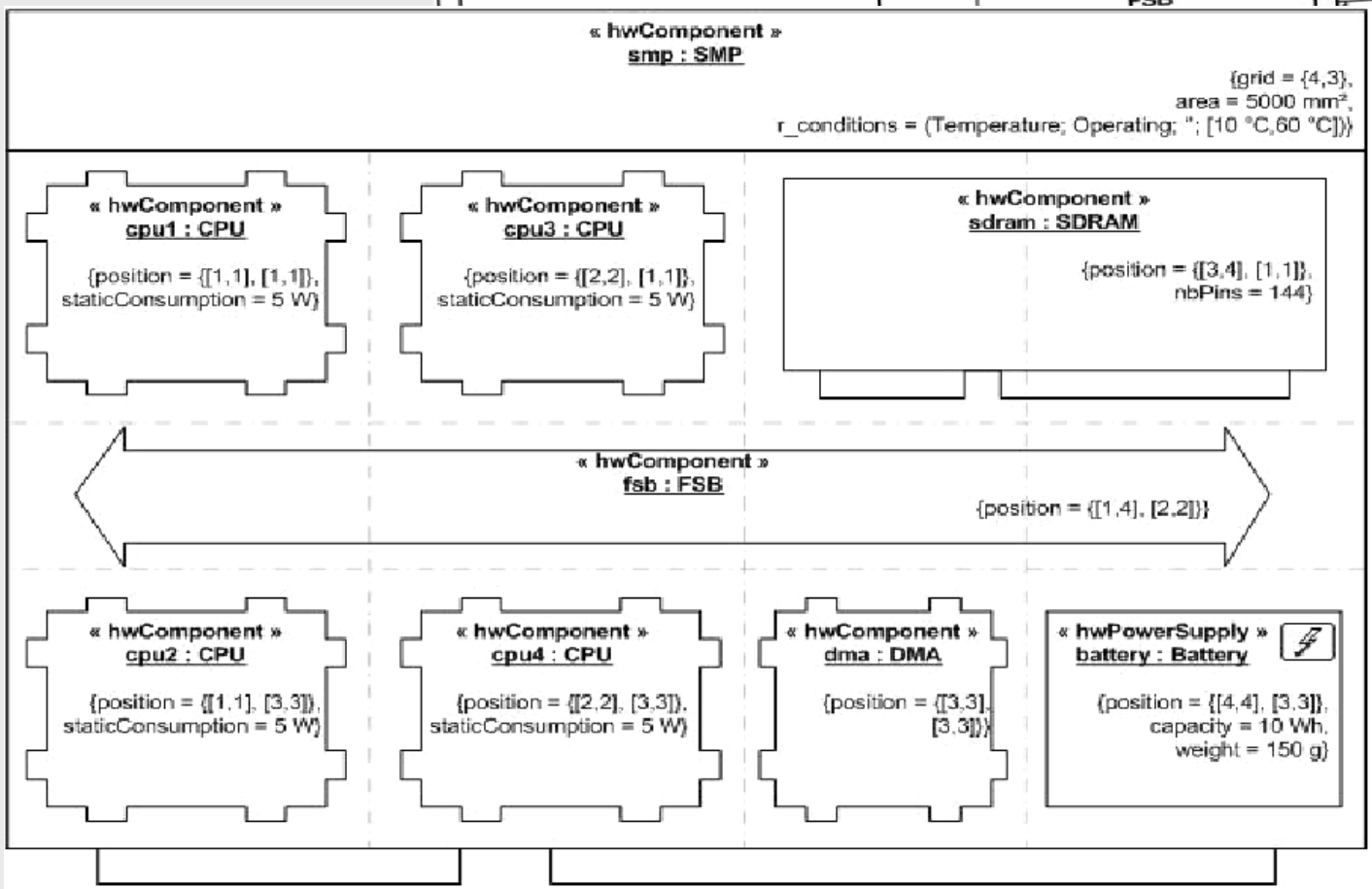
EU-Rent HQ Staff
Definition: the community that *is* the set of employees of **EU-Rent HQ**

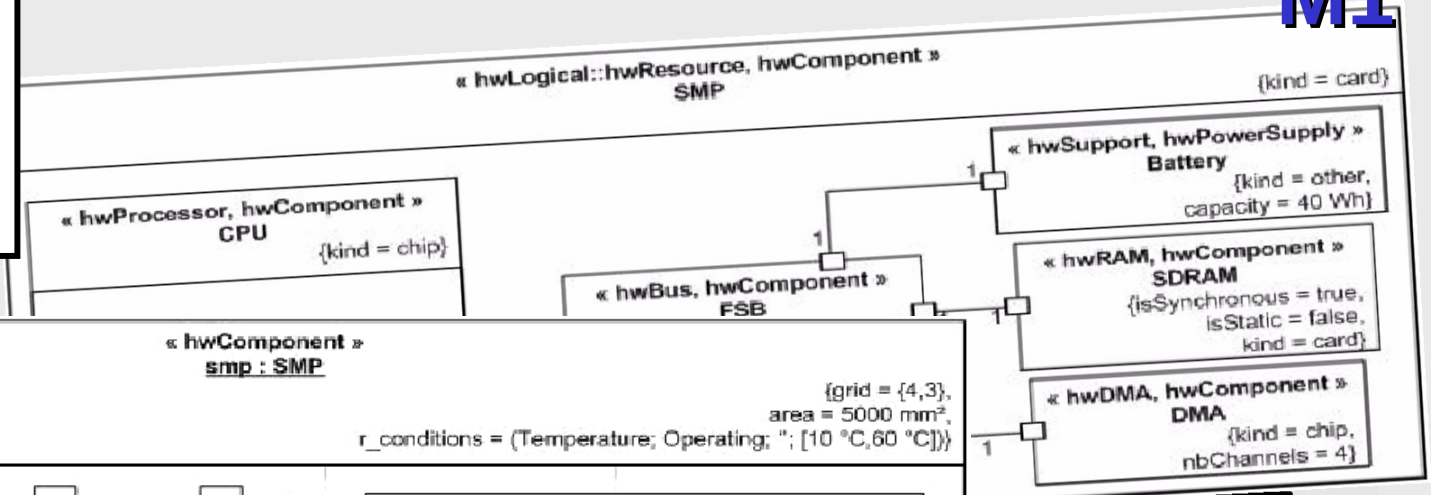
Necessity: **EU-Rent HQ Staff** *is* a subcommunity *of* **The EU-Rent community**.

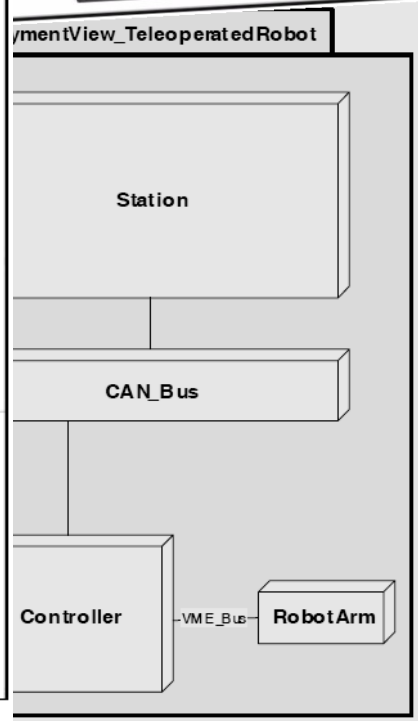


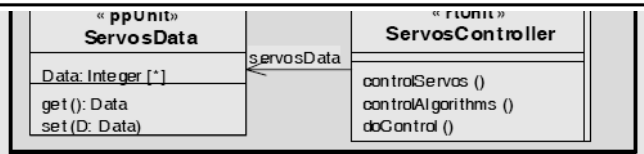
UML MARTE

M1


« hwComponent »
smp : SMP
{grid = {4,3},
area = 5000 mm²,
r_conditions = (Temperature; Operating; ''; [10 °C,60 °C])}
« hwComponent »
cpu1 : CPU
{position = {[1,1], [1,1]},
staticConsumption = 5 W}
« hwComponent »
cpu3 : CPU
{position = {[2,2], [1,1]},
staticConsumption = 5 W}
« hwComponent »
sdram : SDRAM
{position = {[3,4], [1,1]},
nbPins = 144}
« hwComponent »
fsb : FSB
{position = {[1,4], [2,2]}}
« hwComponent »
cpu2 : CPU
{position = {[1,1], [3,3]},
staticConsumption = 5 W}
« hwComponent »
cpu4 : CPU
{position = {[2,2], [3,3]},
staticConsumption = 5 W}
« hwComponent »
dma : DMA
{position = {[3,3], [3,3]}}
« hwPowerSupply »
battery : Battery
{position = {[4,4], [3,3]},
capacity = 10 Wh,
weight = 150 g}

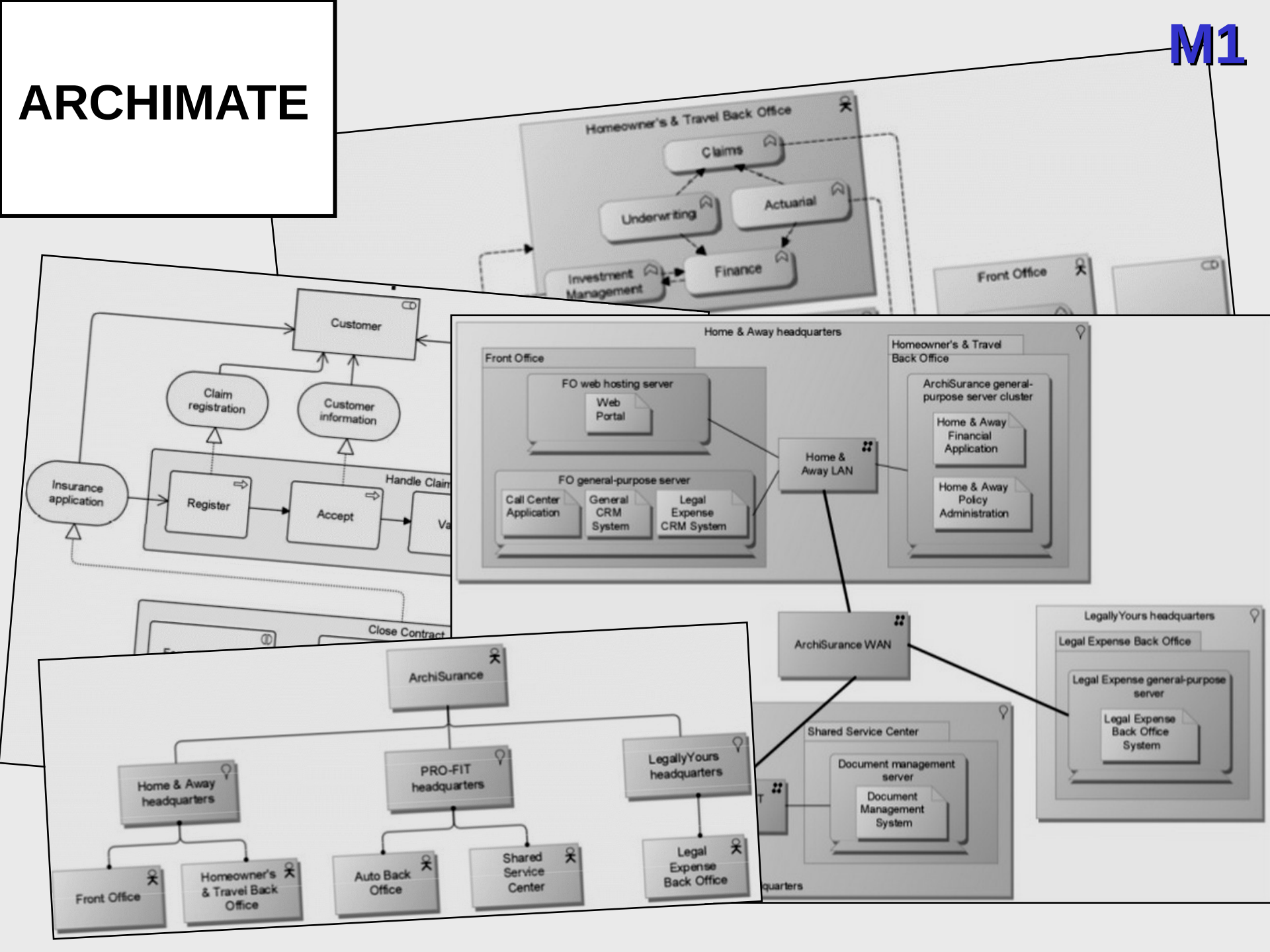

« hwLogical::hwResource, hwComponent »
SMP
{kind = card}
« hwProcessor, hwComponent »
CPU
{kind = chip}
« hwBus, hwComponent »
FSB
« hwSupport, hwPowerSupply »
Battery
{kind = other,
capacity = 40 Wh}
« hwRAM, hwComponent »
SDRAM
{isSynchronous = true,
isStatic = false,
kind = card}
« hwDMA, hwComponent »
DMA
{kind = chip,
nbChannels = 4}
1
1
1
1


ymentView_TeleoperatedRobot
Station
CAN_Bus
Controller
VME_Bus
RobotArm


« ppUnit»
ServosData
Data: Integer [*]
get (): Data
set (D: Data)
servosData
« rUnit »
ServosController
controlServos ()
controlAlgorithms ()
doControl ()

ARCHIMATE

M1



Modeling Language Ecosystems

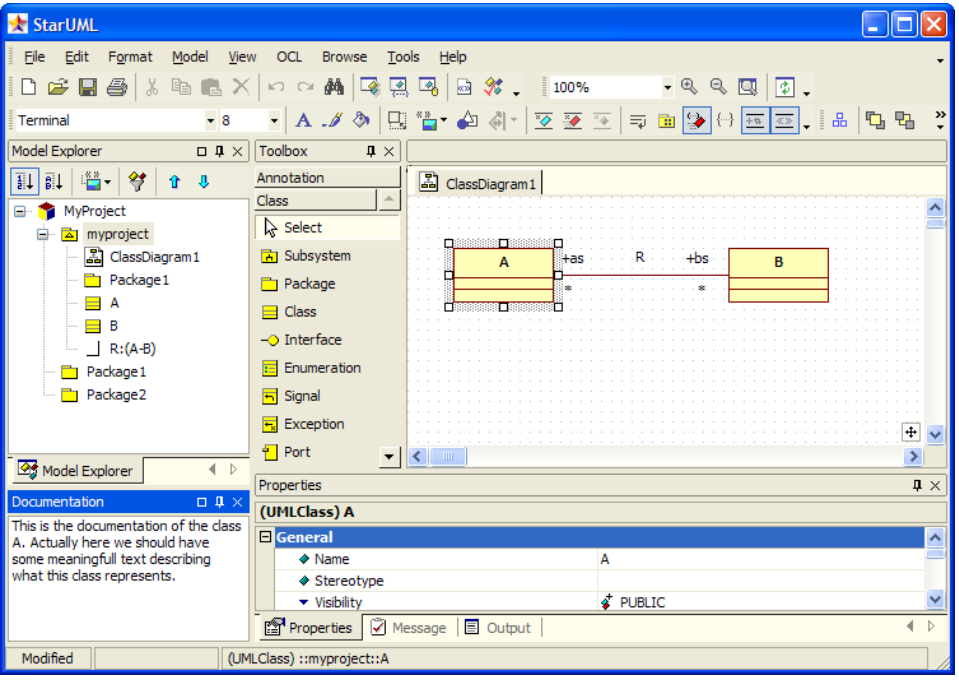
2454

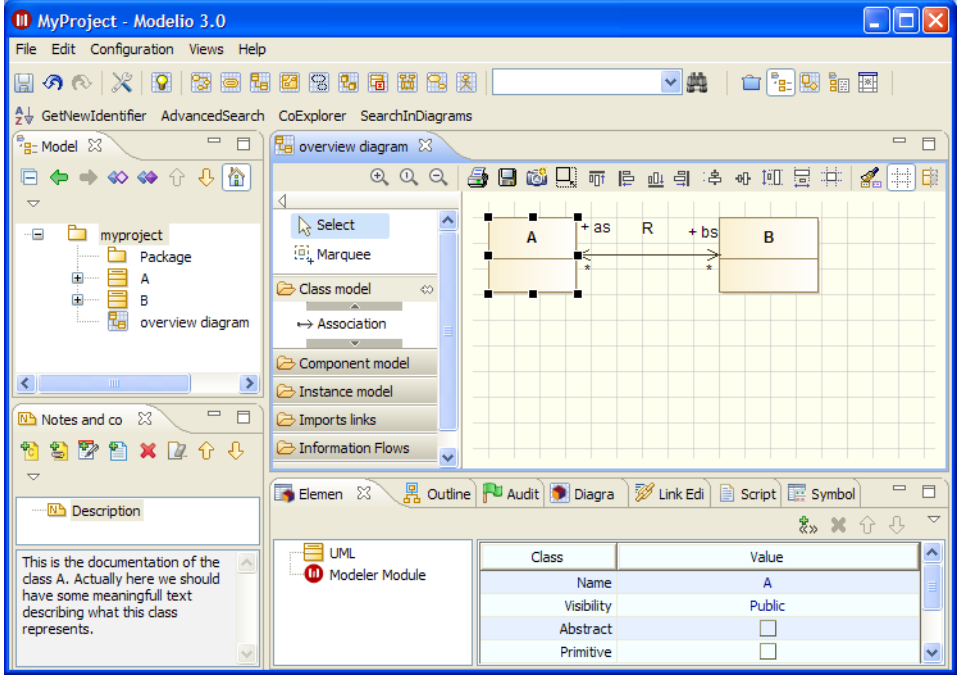
133

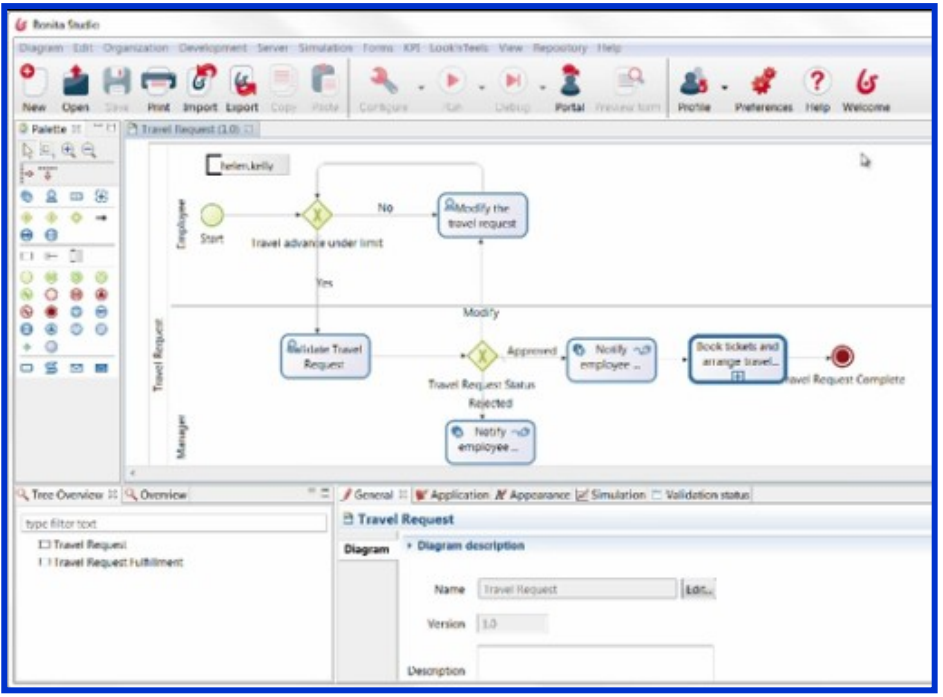
72

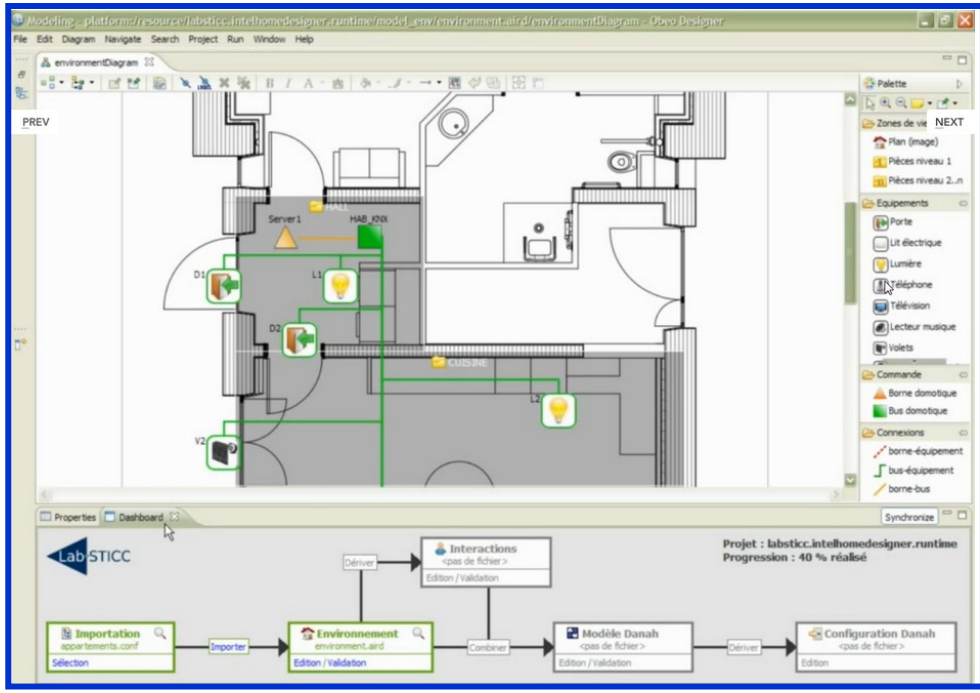
28

7


StarUML
File Edit Format Model View OCL Browse Tools Help
Model Explorer
Toolbox
Annotation
Class
Select
Subsystem
Package
Class
Interface
Enumeration
Signal
Exception
Port
ClassDiagram1
MyProject
myproject
ClassDiagram1
Package1
A
B
R:(A-B)
Package1
Package2
A
+as
R
+bs
B
Documentation
This is the documentation of the class A. Actually here we should have some meaningfull text describing what this class represents.
Properties
(UMLClass) A
General
Name
A
Stereotype
Visibility
PUBLIC
Properties
Message
Output
Modified
(UMLClass) ::myproject::A


MyProject - Modelio 3.0
File Edit Configuration Views Help
GetNewIdentifier AdvancedSearch CoExplorer SearchInDiagrams
Model
myproject
Package
A
B
overview diagram
Notes and co
Description
This is the documentation of the class A. Actually here we should have some meaningfull text describing what this class represents.
overview diagram
Select
Marquee
Class model
Association
Component model
Instance model
Imports links
Information Flows
A
+as
R
+bs
B
Elemen
Outline
Audit
Diagra
Link Edi
Script
Symbol
UML
Modeler Module
Class
Value
Name
A
Visibility
Public
Abstract
Primitive


Bonita Studio
Travel Request
Travel Request Fulfillment
Travel Request
Diagram description
Name
Version
Description


PREV
NEXT
Server1
HALL
HAB_KNX
CUISINE
LabSTICC
Interactions
Importation
Environnement
Modèle Danah
Configuration Danah
Projet : labstic.intelhomedesigner.runtime
Progression : 40 % réalisé


food for thought
food for thought
Metamodeling

Metamodels

"A **metamodel** is a **model** of a (**modeling) language**"

best known example:

"the" UML metamodel

(terms and relations to be defined! --> megamodeling)

Metamodeling

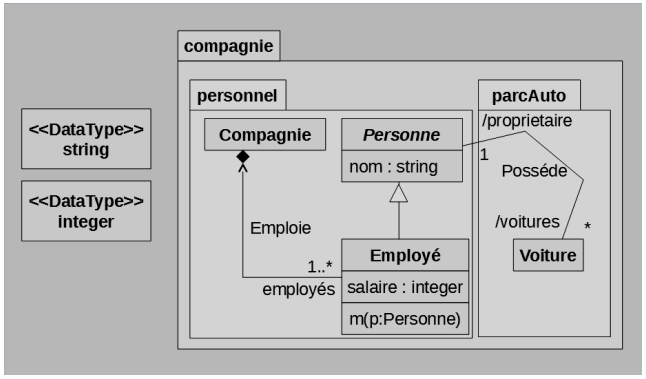
Purposes of **metamodels**

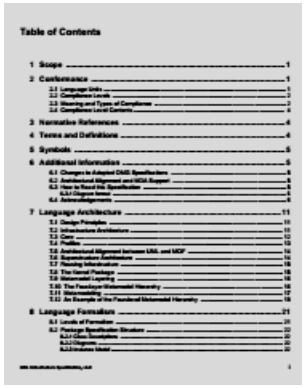
- "inform" about a modeling language
- "well-defined language"

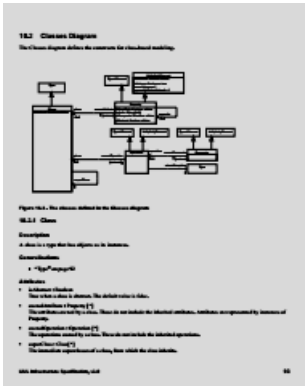
Useful for

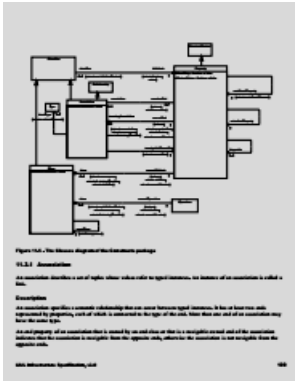
- **conceptualization** of languages
- **standardization** of languages
- **implementation/extension** of modeling tools
- model **transformation** / **code** generation

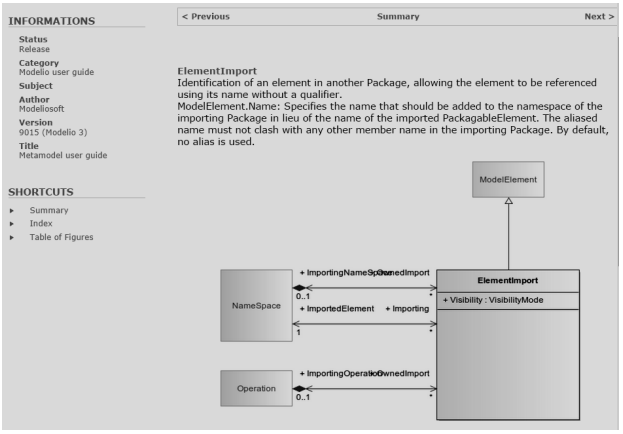
Metamodeling with/of UML

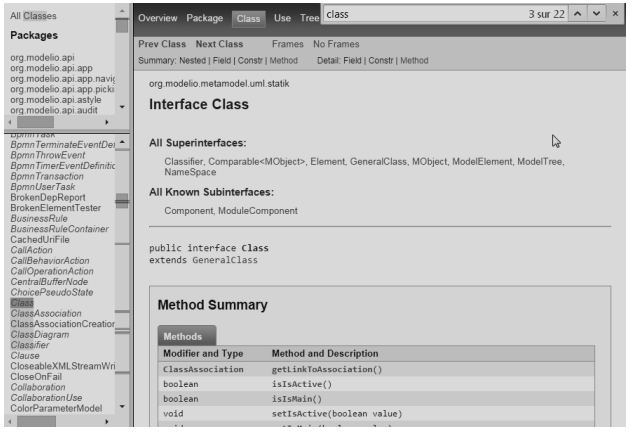


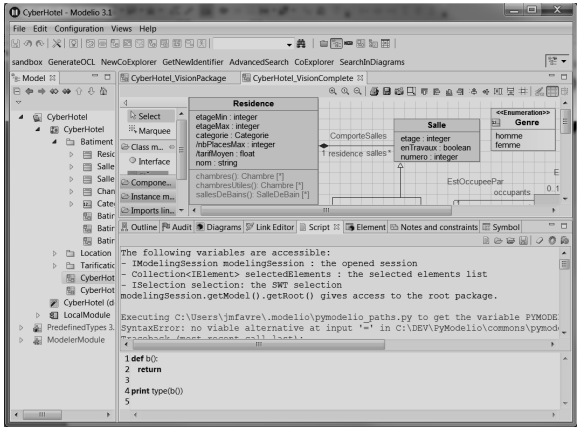








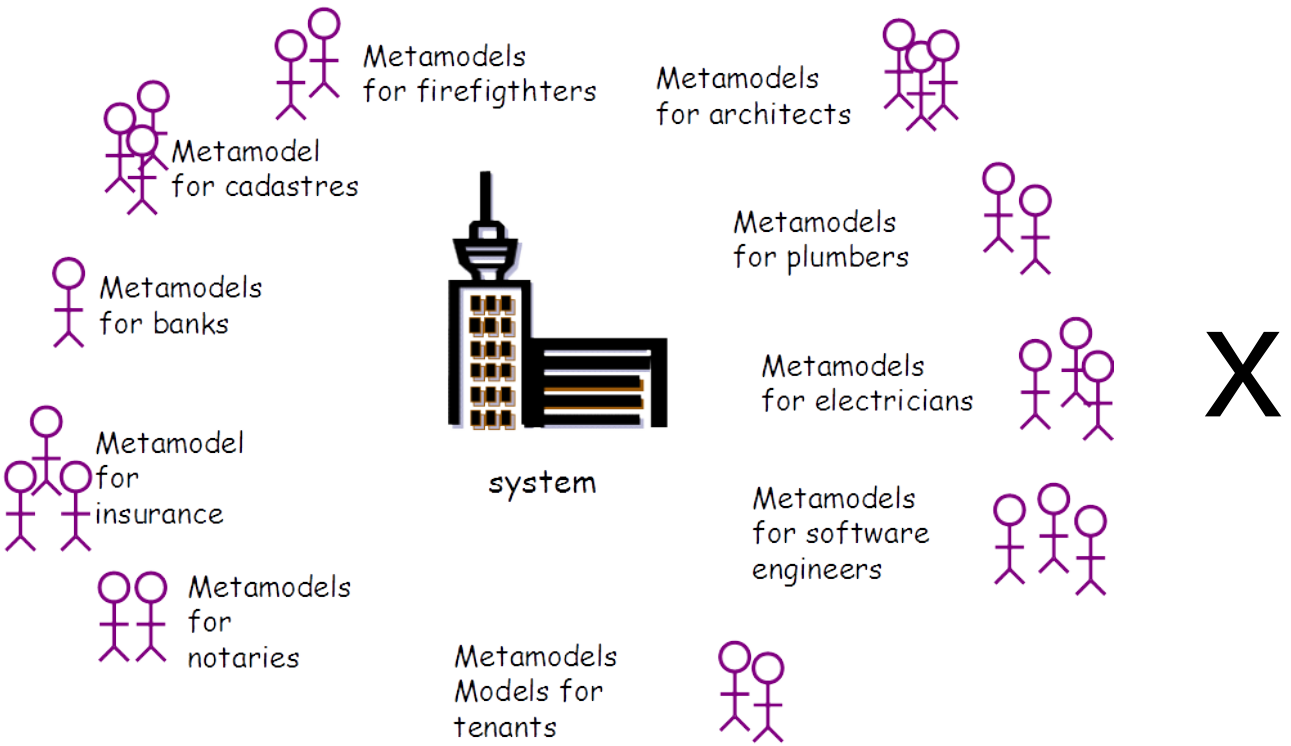




Metamodels for who?

- standardization bodies
- (modeling) tools builders
- language engineers
- instructors
- modeling tool users

Complex Systems & Stakeholders



X

users

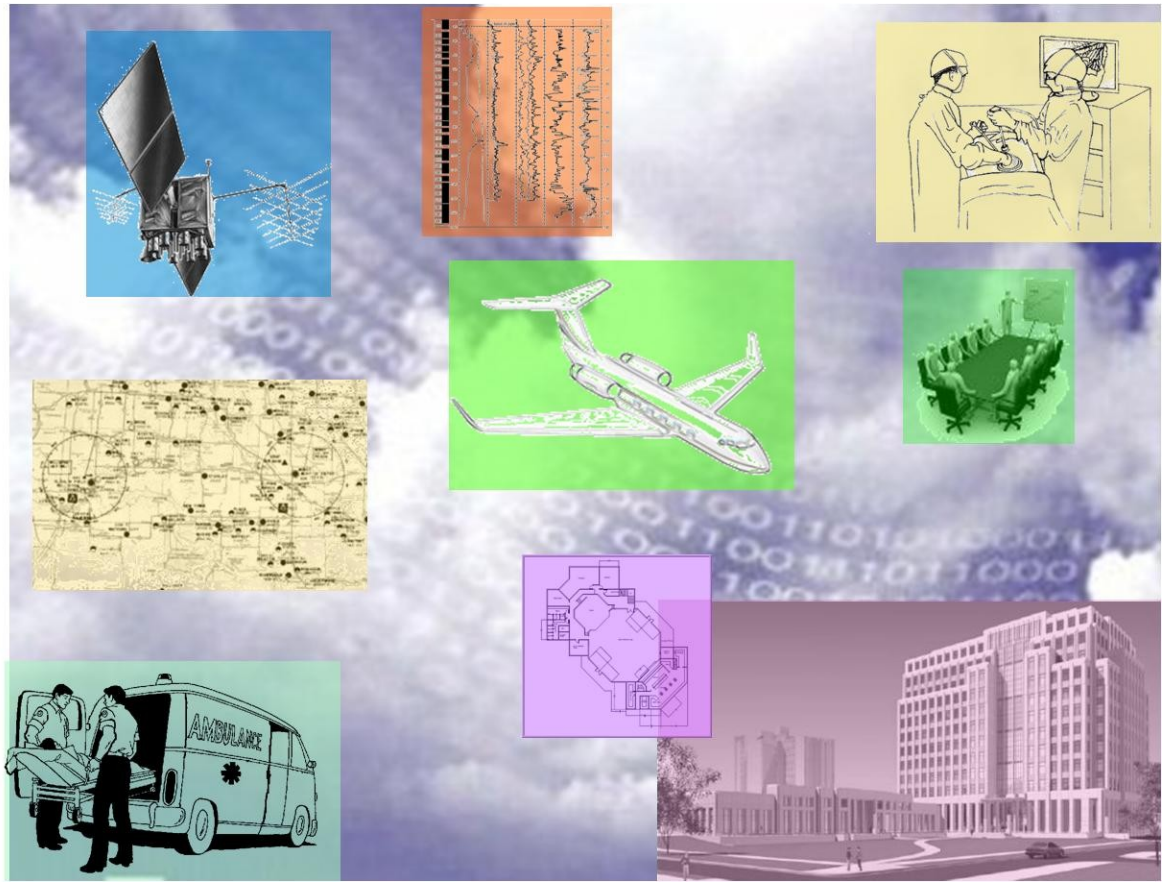
standardization bodies

language engineers

tool builders

instructors

Ultra-large Scale Systems



X

- users
- standardization bodies
- language engineers
- tool builders
- instructors


Executable UML
A Foundation for Model-Driven Architecture
Stephen J. Mellor
Marc J. Balcer
Foreword by Ivar Jacobson
OMG PRESS
The Object Constraint Language Second Edition
Getting Your Models Ready for MDA
MDA Distilled
Principles of Model-Driven Architecture
Stephen J. Mellor
Kendall Scott
Axel Uhl
Dirk Weise
Foreword by Dr. Richard Mark Soley
Executable UML
How to Build Class Models
Leon Starr
Enterprise Patterns and MDA
Building Better Software with Archetype Patterns and UML
Richard Mark Soley
Mastering XML
Java Programming
WILEY
Software Factories
Assembling Applications with Patterns, Models, Frameworks, and Tools
Jack Greenfield and Keith Short with Steve Cook and Stuart Kent
Foreword by John Crupi
Featuring the IBM MDA Manifesto and the Cook-Guttman debate on the Microsoft and OMG approaches to model-driven systems
The MDA Journal
Model Driven Architecture
Straight from the Masters
Oliver Sims, Stephen Mellor, David Frankel, Jorn Bettin, Steve Cook, Mike Rosen, Michael Guttman, Patrick Hayes, Elisa Kendall, Deborah McGuinness, and Alan Brown, Bran Selic, Sridhar Iyengar and other members of the IBM Rational Team
Foreword by Dr. Richard Soley, CEO, OMG
David S. Frankel & John Parodi, Editors
OMG PRESS
Model Driven Architecture™
Applying MDA™ to Enterprise Computing
David S. Frankel
Foreword by Michael Guttman
Model Driven Architecture with Executable UML™
Executable UML
Models Are the Code
A Case Study
MDA Explained
The Model Driven Architecture™: Practice and Promise
Applied Metamodelling
A Foundation for Language Driven Development
Version 0.1
Tony Clark, Andy Evans, Paul Sammut, James Willans
©Xactium, 2004.
Eric Evans
Foreword by Martin Fowler
eclipse
Modeling Framework
Frank Budinsky · David Steinberg
Ed Merks · Ray Ellersick · Timothy J. Grose
SERIES EDITORS · Erich Gamma · Lee Nackman · John Wiegand

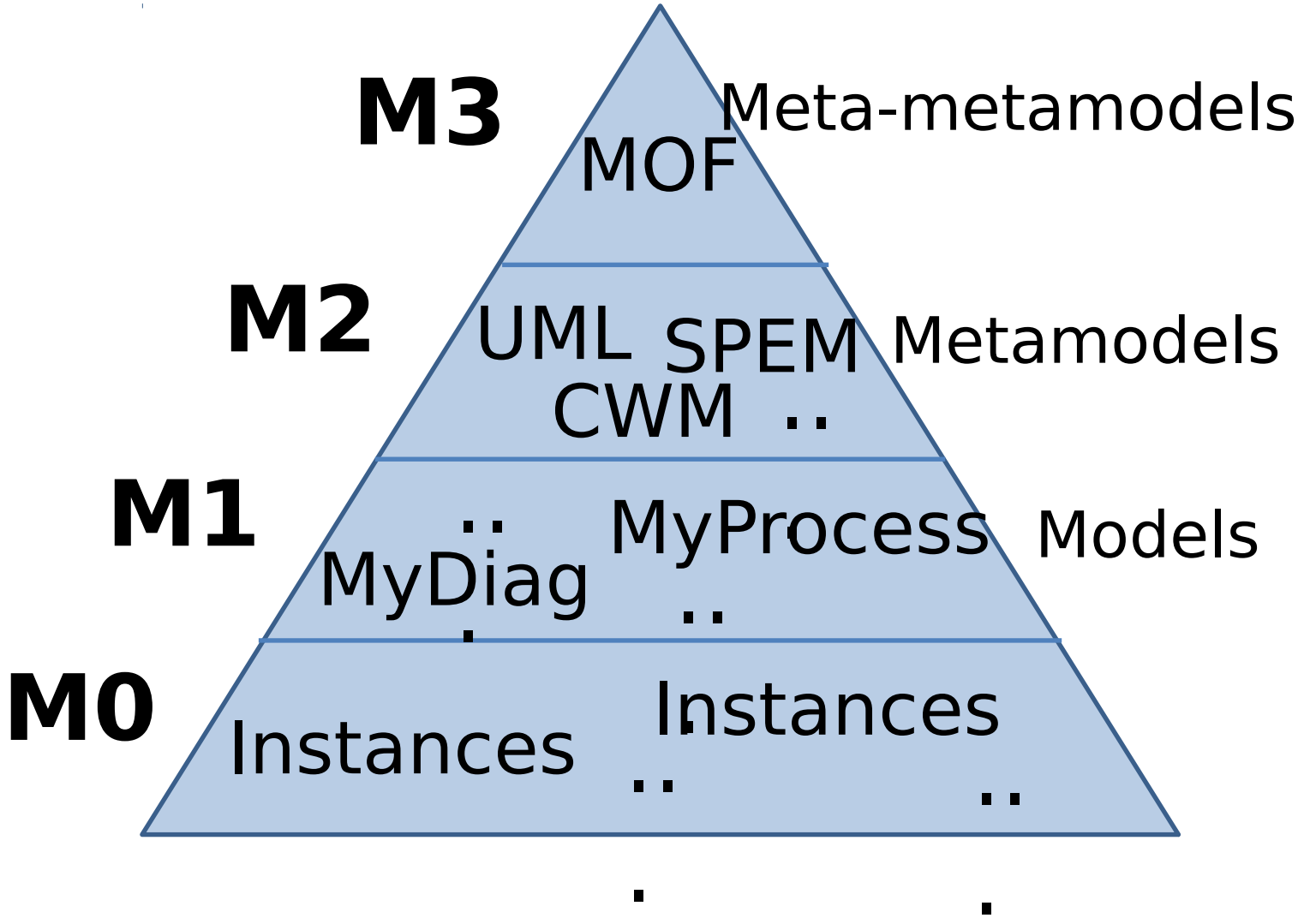

food for thought
food for thought
egamodeling

Megamodels

A **megamodel** is a model that represents models, metamodels and/or other MDE entities along with their relationships.

Megamodeling

The OMG "Mega" Pyramid



Pyramid of Actors