

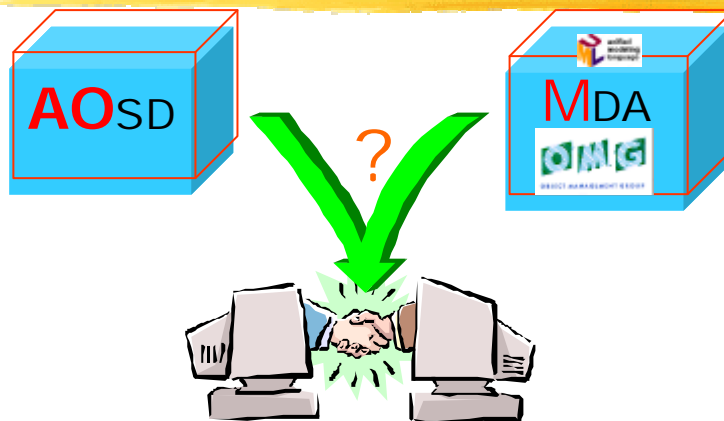
# Aspect-Oriented Modeling: Oxymoron or Pleonasm?

**Jean Bézivin**

Université de Nantes - CRGNA  
Faculté des Sciences et Techniques  
2, rue de la Houssinière BP 92208  
44322 Nantes cedex 3, France  
[Jean.Bezivin@sciences.univ-nantes.fr](mailto:Jean.Bezivin@sciences.univ-nantes.fr)

AOM?

## Conditions for a Possible Synergy?



[Aspect-Oriented Modeling?]



# Main point of the presentation

~~⌘ Objects everywhere~~

## Models

- ⌘ How will model engineering help capturing aspect management in software development?
- ⌘ How this could fit in the OMG MDA initiative?
- ⌘ Which kind of aspects could be captured by separate models?
- ⌘ Which kind of operations (separation, weaving, etc.) could be defined on models
- ⌘ How may these different concerns, represented by coordinated non executable abstract models, be operationalized (i.e. mapped on equivalent executable models)



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

# Agenda

- ⌘ A point of vocabulary
- ⌘ Paradigm evolution : from objects & components to models
- ⌘ What is the MDA?
- ⌘ Models as an explicit specification of aspects
- ⌘ Conclusions: AOSP vs MDA/MDE



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

# Oxymoron

⌘ A rhetorical figure in which an epigrammatic effect is created by the conjunction of incongruous or contradictory terms

⌘ A rhetorical figure in which incongruous or contradictory terms are combined, as in a *deafening silence* and a *mournful optimist*.

⌘ *The American Heritage® Dictionary of the English Language, Fourth Edition*

⌘ A figure in which an epithet of a contrary signification is added to a word - e. g., *cruel kindness* - *laborious idleness*.

⌘ *Webster's Revised Unabridged Dictionary, © 1996, 1998 MICRA, Inc.*

⌘ n : conjoining contradictory terms (as in 'deafening silence')

⌘ *WordNet® 1.6, © 1997 Princeton University*



Jean Bézin - AOPDCS'2002, Vienna, July 2nd 2002

# Oxymora

⌘ active waiting - acute dullness - Advanced BASIC - almost exactly - alone together - clearly ambiguous - clearly confused - clearly misunderstood - conciliation court - constant variable - dangerously safe - deafening silence - definite maybe - diet ice cream - even odds - exact estimate - extensive briefing - extinct life - fish farm - flexible ethics - found missing - fresh-frozen - friendly fire - genuine imitation - good shit - hells angels - holy war - home office - idiot savant - industrial park - intense apathy - jumbo shrimp - linear curve - liquid gas - little giants - living dead - long sleeved t-shirt - minor crisis - new classic - non-alcoholic wine - non-dairy creamer - non-working mother - old news - only choice - open secret - original copies - paid volunteer - passive aggression - peace offensive - plastic glasses - plastic silverware - pretty ugly - randomly organized - real potential - resident alien - sad clown - scheduled spontaneity - seriously funny - silent scream - sweet sorrow - synthetic natural gas - temporary tax increase - tragic comedy - uncrowned king - vaguely aware - virtual reality - working vacation - etc.



Jean Bézin - AOPDCS'2002, Vienna, July 2nd 2002

# Pleonasm

- ⌘ Pleonasms are the **opposites** (antonyms) of **oxymora**. A pleonasm consists of several concepts (usually several words) that are redundant, e.g. "**at this moment in time**." (presumably these five words mean "now.")
- ⌘ **pleonasm** \PLEE-uh-naz-uhm\, *noun*:
  1. The use of more words than are necessary to express an idea - as, "**I saw it with my own eyes**."
  2. An instance or example of pleonasm.
  3. A superfluous word or expression.
- ⌘ from Latin *pleonasmus*, Greek *pleonasmós* ('more-ness') excess, redundancy]. A traditional term for the use of more words than necessary, either for effect or more usually as a fault of style, and any instance of that use, as in: *Could you repeat that again?* rather than *Could you say that again?* or *Could you repeat that?* - *They both got one each* rather than *They both got one* or *They got one each* - *That's a more superior product* (*superior* already denotes 'more') - *It's a really new innovation* (an *innovation* is already *new*). Some common pleonasms attract little comment, such as *free gift* (gifts are by definition free) and *plans for the future* (plans cannot be about the present or past).
- ⌘ *The Oxford Companion to the English Language*, © Tom McArthur 1992



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

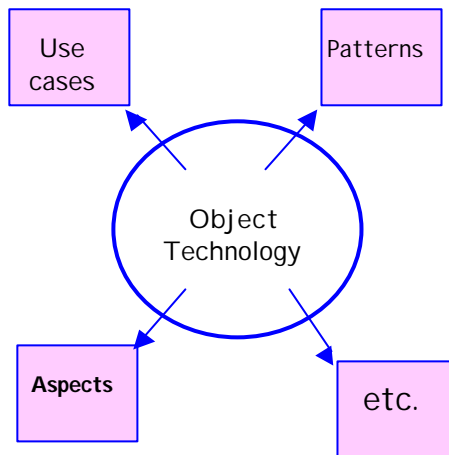
# Pleonasms

- ⌘ absolutely essential - a cappella without music - advance warning - affirmative yes - A.M. in the morning - attach together - autobiography of my life - bad evil - classic tradition - classify into groups - climb up - close proximity - cold ice - collaborate together - combined together - dark night - DOS operating system - empty hole - foreign imports - frozen ice - grateful thanks - handwritten manuscript - hot fire - imminent at any moment - individual person - invited guests - join together - joint collaboration - joint cooperation - knowledgeable experts - little baby - long litany - major breakthrough - malignant cancer - may possibly - mental thought - merge together - mutual cooperation - near proximity - new discovery - new innovations - new neophyte - new recruit - nostalgia for the past - old senior citizens - oral conversation - original founder - original source - pair of twins - past history - positive yes - postponed until later - potentially capable - repeat again - round circle - specific examples - top priority - unexpected surprise - wordy and verbose - youthful teenagers



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

## Paradigm evolution

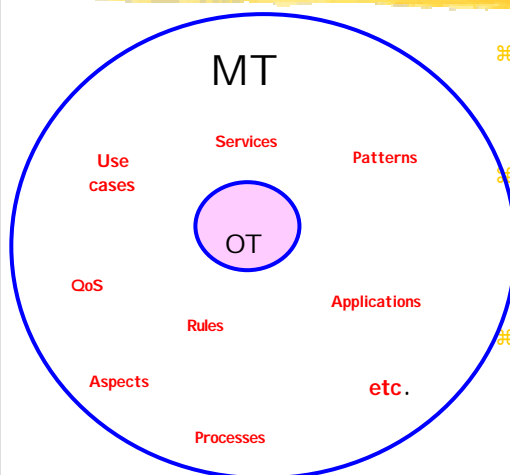


- ⌘ OT (Object Technology) has got to answer many difficult problems since its initial industrial discovery in the 80's.
- ⌘ Generally OT did not succeed very well in finding good internal solutions to these challenges.
- ⌘ As a consequence, the proposed solutions are often ad-hoc, informal, and even baroque. They are difficult to generalize and hard to combine. They often don't scale up.



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

## Paradigm evolution



- ⌘ MT (Model Technology) seems to be in a position to meet some of these challenges to which OT was not able to find an internal solution.
- ⌘ Among these we may quote:
  - ✓ Aspect separation
  - ✓ Homogeneous handling of functional and non-functional attributes
  - ✓ Integration of different paradigms (rules, services, processes, architecture, etc.)
- ⌘ MT seems able to subsume OT and to offer a realistic migration path from present object and component solutions to more ambitious regular and scalable organizations.



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

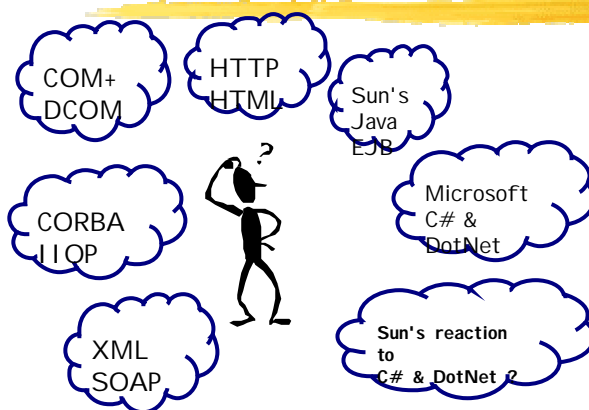
## Angry end-users

We don't want anymore to pay such a high price for simply moving our information system to a new middleware platform (COM, CORBA, Java, HTML, XML, DotNet, etc.) when our business system stays stable.  
We are prepared to pay a last price for building the abstract models of our business and services that will guarantee us against technological obsolescence.  
From there, any platform provider will also have to provide the mapping solutions from standard business models before we buy.



Jean Bézin - AOPDCS'2002, Vienna, July 2nd 2002

## The middleware war is over



- ⌘ There is no clear winner nor loser
- ⌘ The next battlefield will be model transformation
- ⌘ The OMG's Model Driven Architecture (MDA) **initiative** is aimed at using modeling **and** meta-modeling to drive the design and implementation of distributed systems.

+ the next wonderful  
Middleware platform (~2005)



Jean Bézin - AOPDCS'2002, Vienna, July 2nd 2002

# Mapping to multiple and evolutive platforms



UML and MOF compliant  
platform independent models



COM+  
DCOM

Java  
EJB

HTTP  
HTML

CORBA

C#  
DotNet

XML  
SOAP

⌘ MOF along with UML is a core technology for MDA.

⌘ Technology neutral models of systems can be mapped to implementations that use a variety of middleware technologies (easy to say, more difficult to implement)

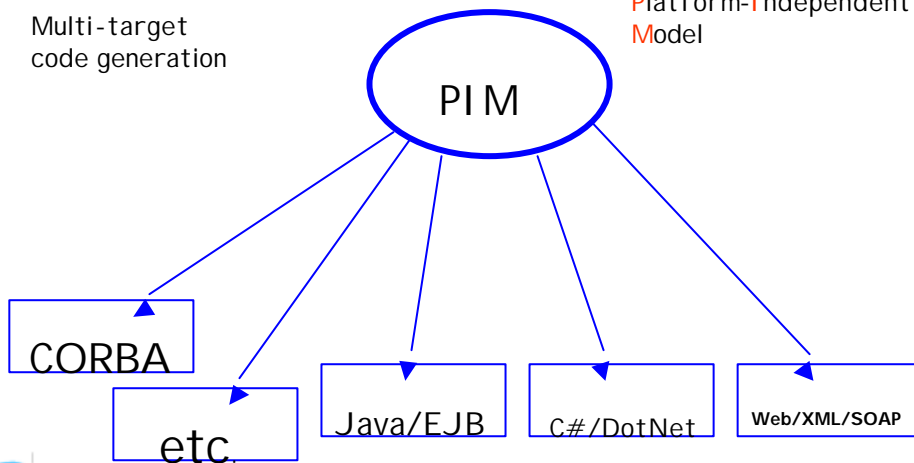


Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

## Various platforms

Multi-target  
code generation

Platform-I  
Model

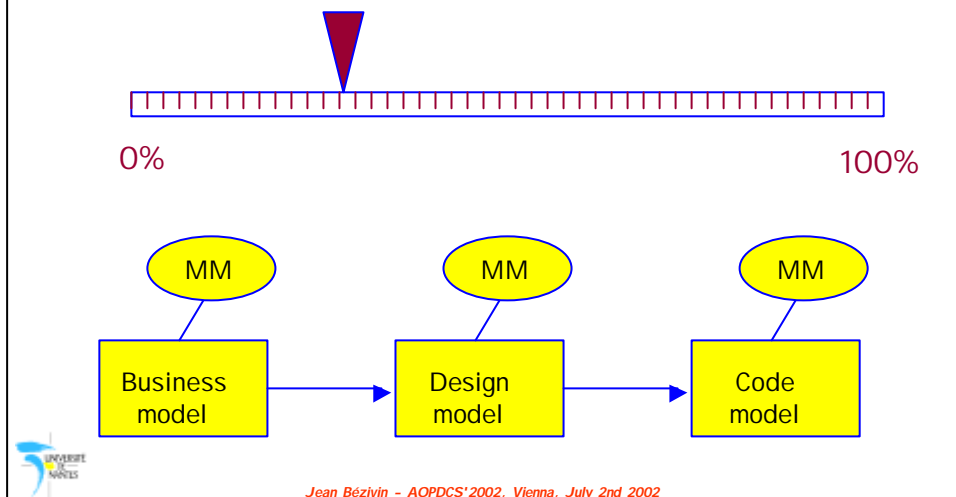


+ SVG, GML, Delphi, etc.



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

## Manual/Automatic code generation



## MDA: PIMs and PSMs

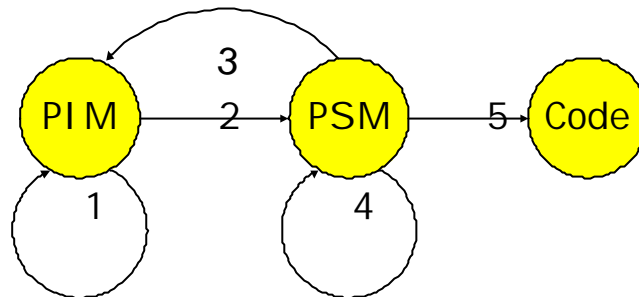
### ⌘ MDA models

- ✓ **PIM**: Platform Independent Models
  - ☒ Formal specification of a system that abstracts away technical detail
  - ☒ Example: Billing system expressed in UML
- ✓ **Platform Description Models (PDMs)**
  - ☒ e.g. of component constructs (CCM), of Eiffel, C#, EJB, ...
- ✓ **PSM**: Platform Specific Models
  - ☒ Expressed in terms of the specification model of the platform
  - ☒ Example: Billing system expressed in "UML profile for CORBA"



## Basic MDE operations

- ⌘ OMG standards are specified in terms of a PIM and normally one or more PSMs, all in UML or UML profiles or based on any other kind of MOF meta-model.
- ⌘ The MDA defines consistent relationships across these models.
- ⌘ It makes it easier to produce implementations on different platforms while conforming to the same essential and precise structure and behavior of the system. As a consequence, the business model may define business goals and policies in a computation independent manner.



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

## Some of the OMG successes

1. CORBA

2. IDL

3. IIOP

4. UML

5. MOF

6. XMI

7. CWM

8. UPM/SPERM

yesterday  
OMA

today

?

tomorrow  
MDA

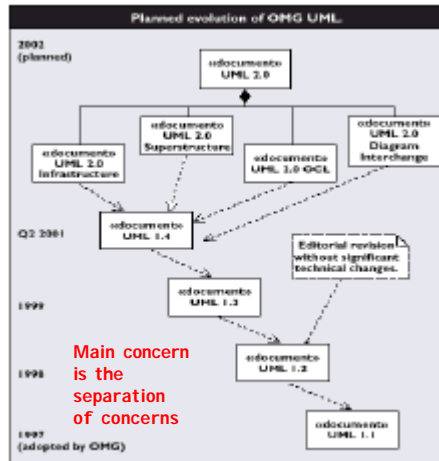


Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

# UML 2.0 : not a language, but a family of languages

- ☼ **UML 2.0 Infrastructure RFP** - - A UML 2.0 RFP issued September 15, 2000 that is primarily concerned with architectural alignment, restructuring and extension mechanisms.
- ☼ **UML 2.0 Superstructure RFP** - - A UML 2.0 RFP issued September 15, 2000 that is primarily concerned with the refinement and extension of UML 1.x semantics and notation.
- ☼ **UML 2.0 OCL RFP** - - A UML 2.0 RFP issued September 15, 2000 that is primarily concerned with defining an OCL metamodel.
- ☼ **UML 2.0 Diagram Interchange RFP** - - A UML 2.0 RFP issued March 2, 2001 that is primarily concerned with defining a metamodel for diagram interchange using the XMI facility.

from "Will UML 2.0 Be Agile or Awkward?" by Cris Kobryn



Jean Bézin - AOPDCS'2002, Vienna, July 2nd 2002

# UML opened the road, several roads indeed...

From Object-Oriented Programming  
to  
Model-Based Software Engineering.

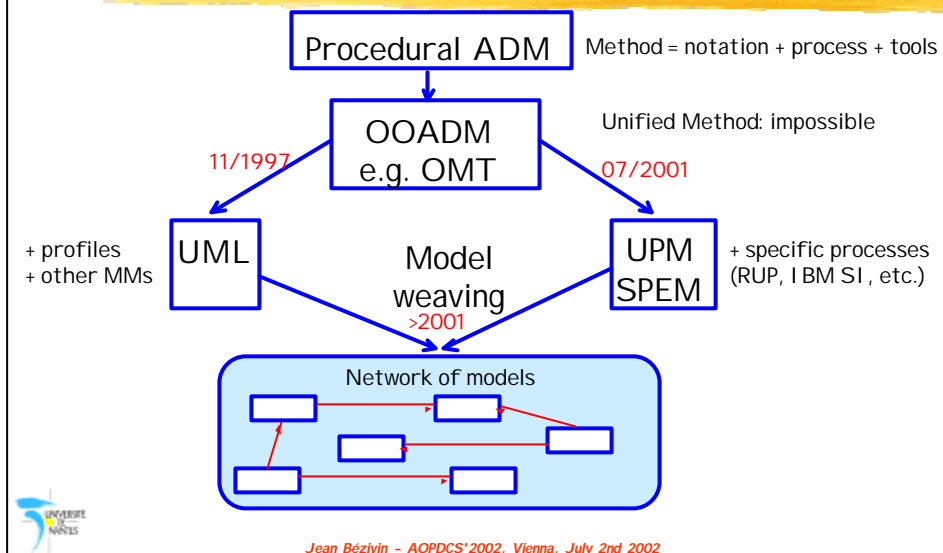
product  
and  
process  
models  
+  
model  
engineering



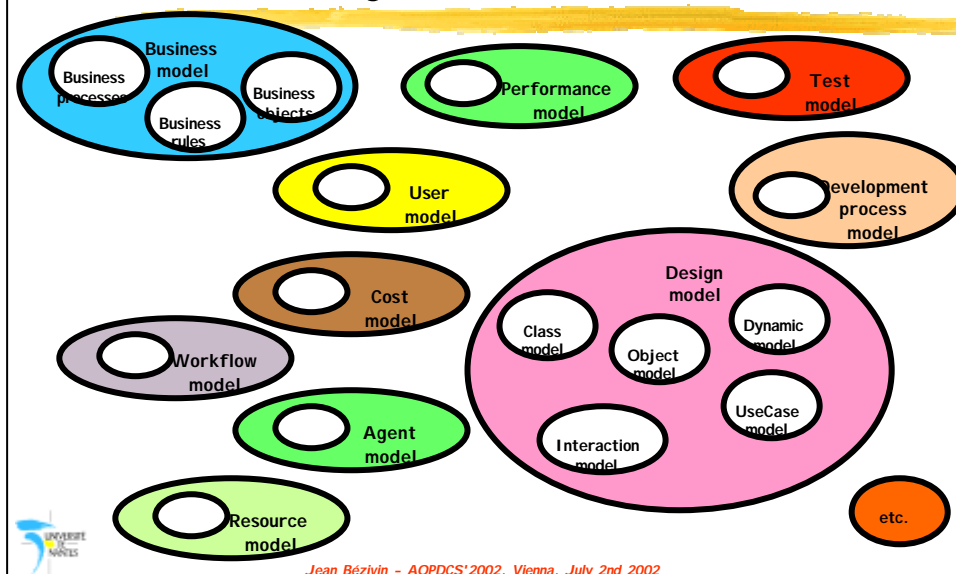
OMT  
SA/RT  
SADT  
ERD  
Merise  
DFD  
UFD  
JSD  
etc.

Jean Bézin - AOPDCS'2002, Vienna, July 2nd 2002

## The initial roadmap: separation of artifact and process aspects



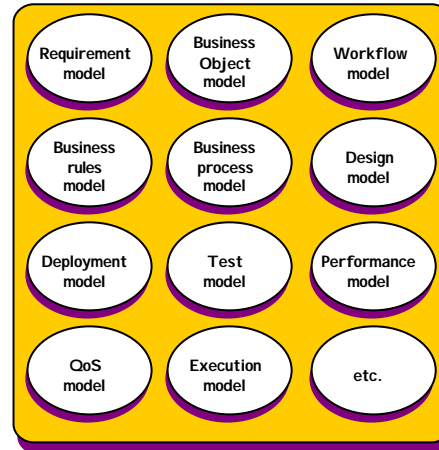
## Models Everywhere



## 1 aspect = 1 model

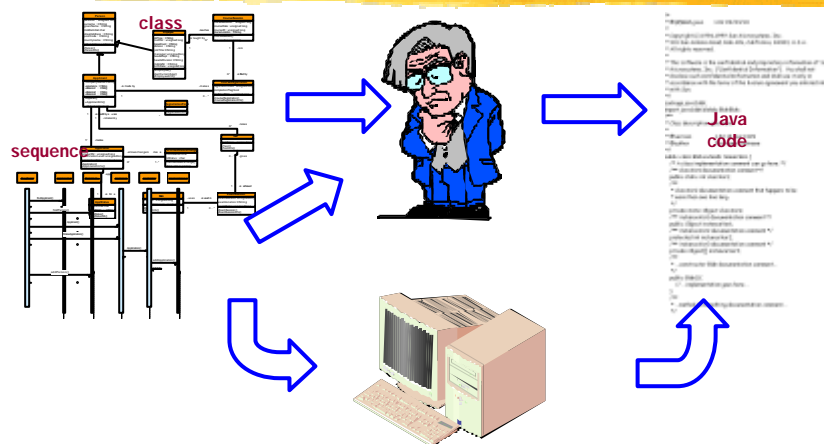
The software life cycle is populated with a lot of models

- ⌘ The development software cycle is populated with models
- ⌘ Models are of unequal importance
- ⌘ The model space is structured (composite and atomic models)
- ⌘ Models are related in a complex network of production/consumption separation/weaving, transformation, etc.
- ⌘ The content of each model is defined (constrained) by a corresponding meta-model (ontology)
- ⌘ The model space is constantly broadening starting from the essential models (Domain, Service, Resource)
- ⌘ All models are not of equal importance



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

## From contemplative to productive

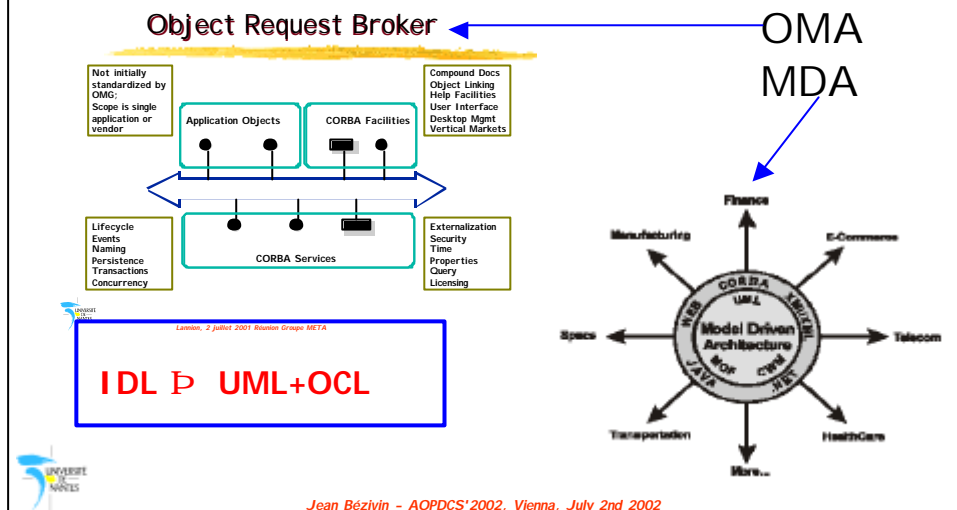


"from human-readable to computer-understandable"



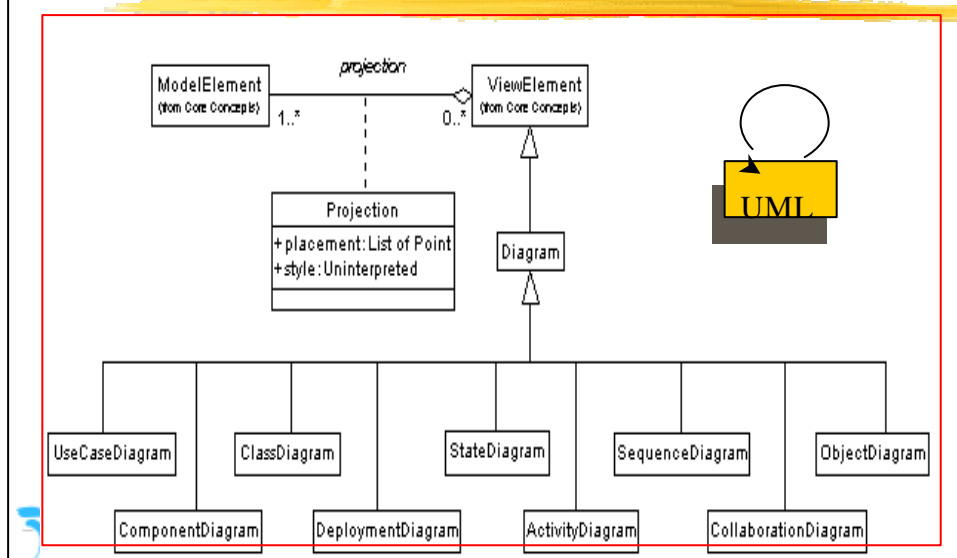
Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

# Compare code-centric and model-centric approaches

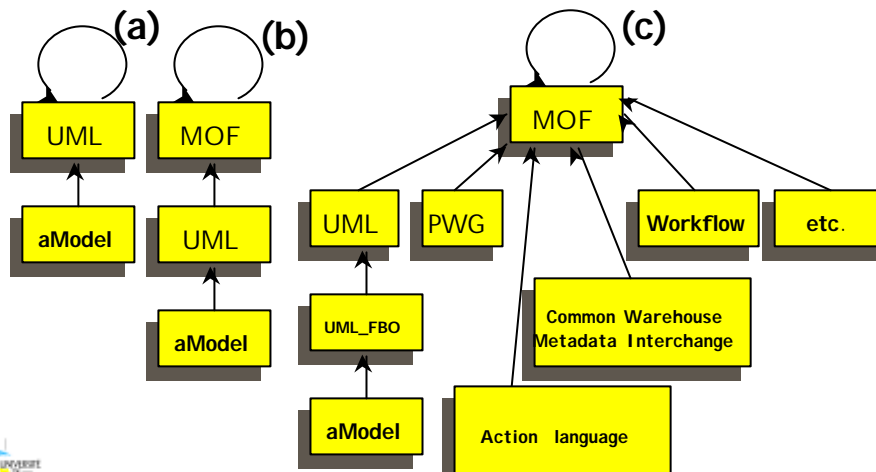


## Fragments of a UML meta-model

not 1.4

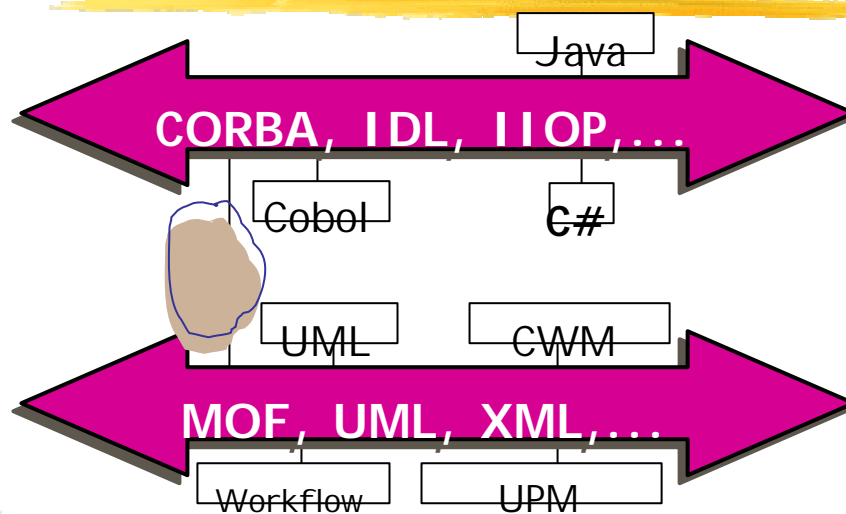


Three stages in the evolution of modeling techniques at OMG.



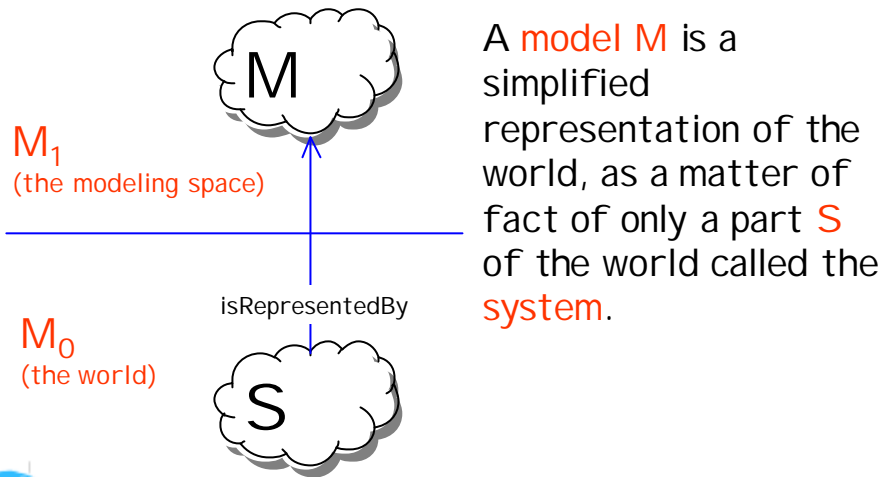
Jean Bézin - AOPDCS'2002, Vienna, July 2nd 2002

OMG : the software bus and the knowledge bus.



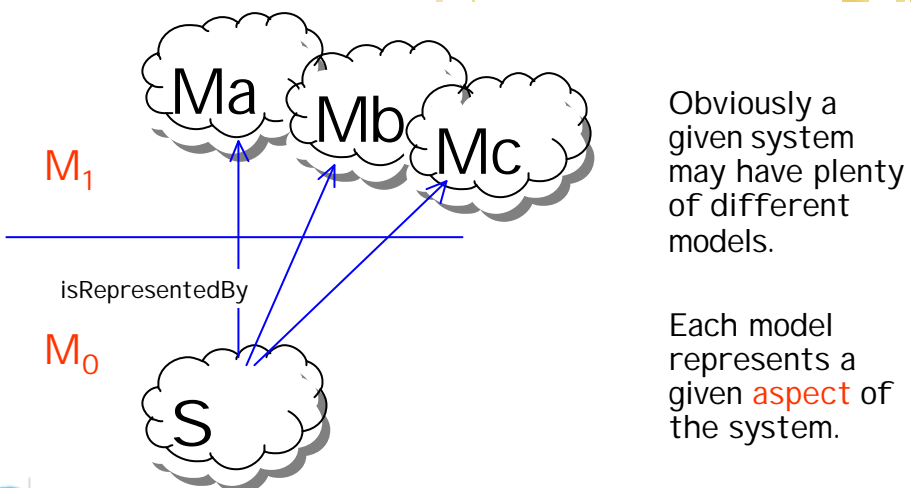
Jean Bézin - AOPDCS'2002, Vienna, July 2nd 2002

## Systems and models



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

## Aspect-Oriented Modeling



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

## Limited Substitutability Principle

- ⌘ The purpose of a model is always to be able to answer some questions in place of the system, exactly in the same way the system itself would have answered similar questions.



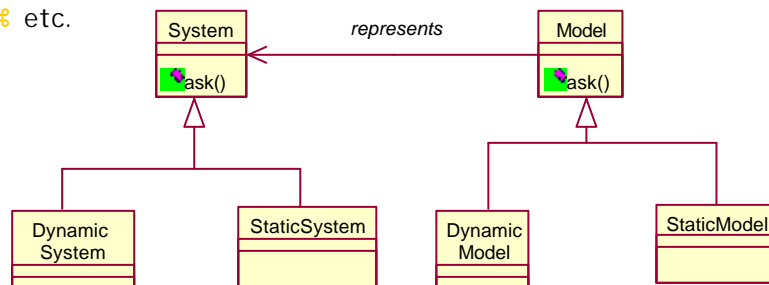
- ⌘ A model represents certain specific aspects of a system and only these aspects.
- ⌘ But how to make this assumption explicit, automatically checkable, etc?



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

## Various kinds of models

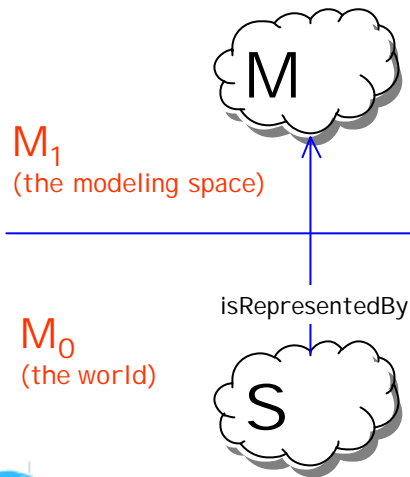
- ⌘ Products and processes
- ⌘ Legacy and components
- ⌘ Static and dynamic
- ⌘ Functional and non-functional aspects
- ⌘ etc.



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002



# Systems and models



⌘ What is exactly this relation?

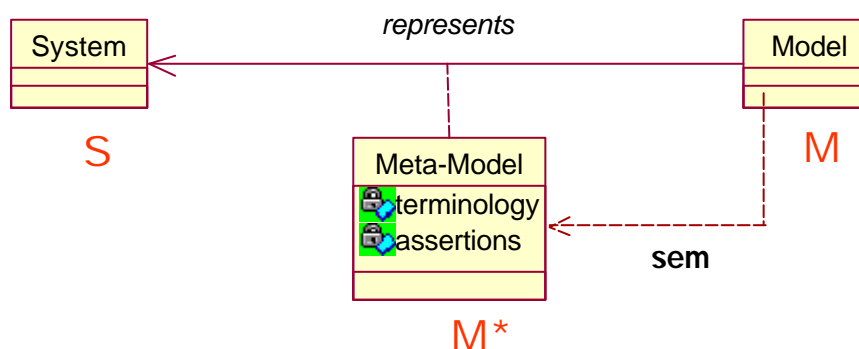
⌘ Can we specify this "aspect selection" with precision?

⌘ How to combine several such relations? How to characterize them?



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

## Solution: Meta-Model

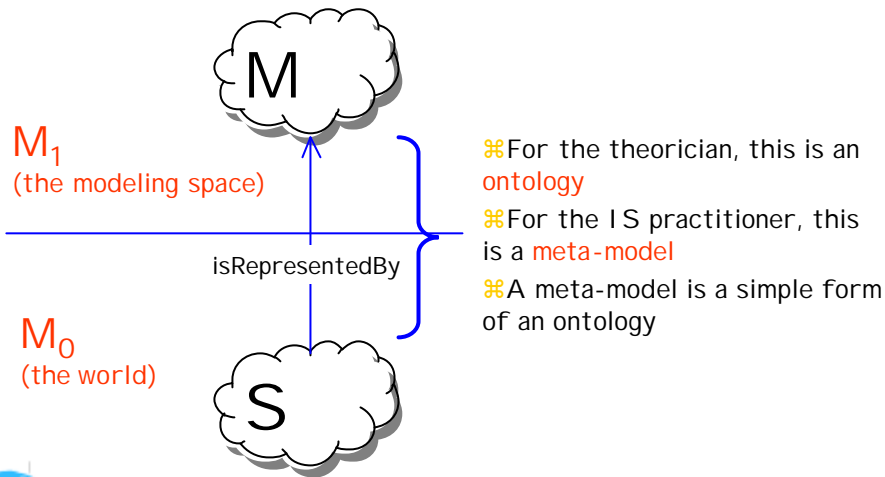


The correspondence between a model and a system is explicitly and precisely defined by a meta-model.



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

# Ontologies and meta-models



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

# Ontology: definition

"A body of formally represented knowledge is based on a **conceptualization**: the **objects**, **concepts**, and other **entities** that are presumed to exist in some **area of interest** and the **relationships** that holds them.

A conceptualization is an **abstract, simplified view** of the world that we wish to represent **for some purpose**.

An ontology is an **explicit specification of a conceptualization**. The term is borrowed from philosophy, where an ontology is a systematic account of Existence. For knowledge-based systems, what "exists" is exactly that which can be represented. When the knowledge of a domain is represented in a declarative formalism, the set of objects that can be represented is called the universe of discourse. This set of objects, and the describable relationships among them, are reflected in the representational vocabulary with which a knowledge-based program represents knowledge. Thus, we can define the ontology of a program by defining a set of representational terms. In such an ontology, definitions associate the names of entities in the universe of discourse (e.g. classes, relations, functions or other objects) with human-readable text describing what the names are meant to denote ..."

Gruber, T.G.

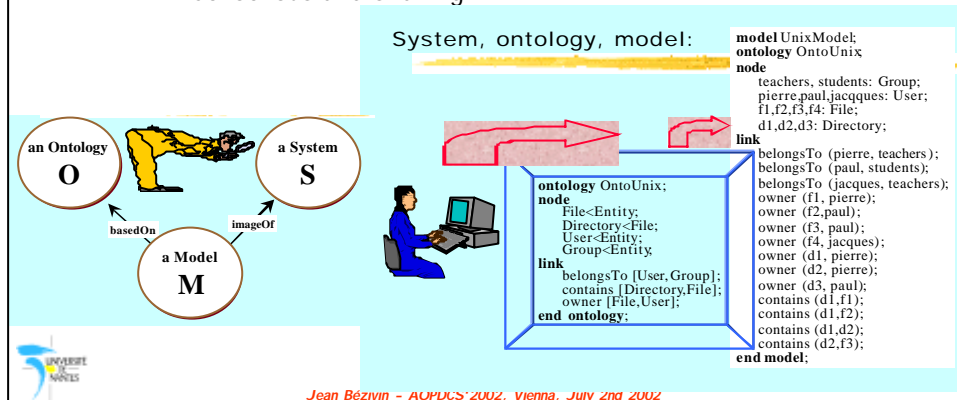
A Translation Approach to Portable Ontology Specifications  
Knowledge Acquisition, V.5, N.2, (1993)

Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

# Meta-models and ontologies

✓ Ontologies bring:

- ☒ Abstraction
- ☒ Consensus and sharing



# Ontologies

⌘ Normative consensus

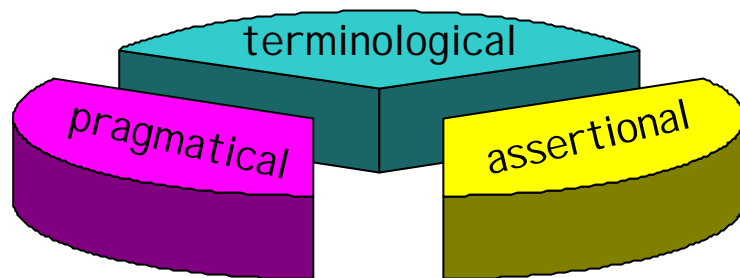
⌘ Consensual norms

⌘ Because:

- ✓ There are non-normative consensus
- ✓ There are non-consensual norms

## Layered ontologies

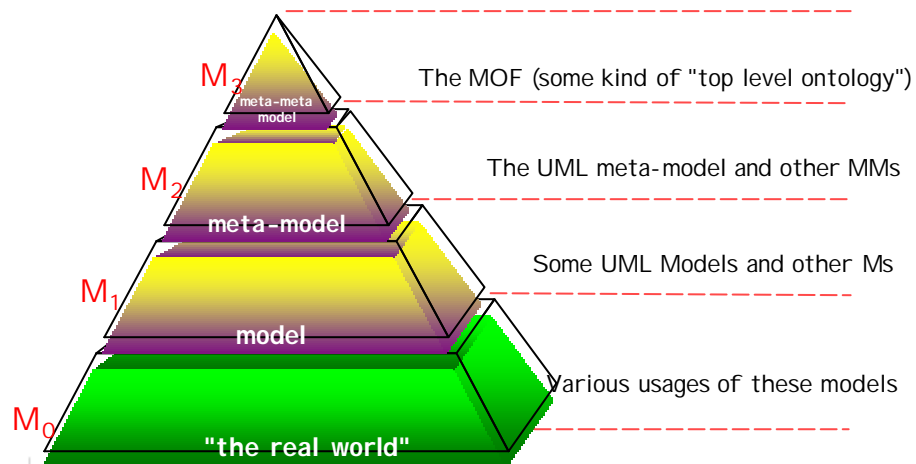
Concepts and Relations  
e.g. UML diagrams



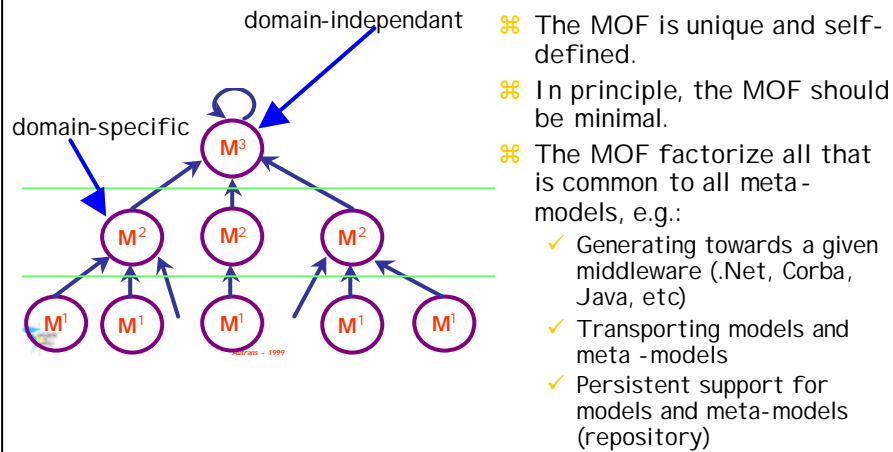
e.g. How to draw a class?  
presentation issues, etc.

e.g. OCL statements

## Egyptian architecture [Inspired by IRDS, CDIF, etc.]

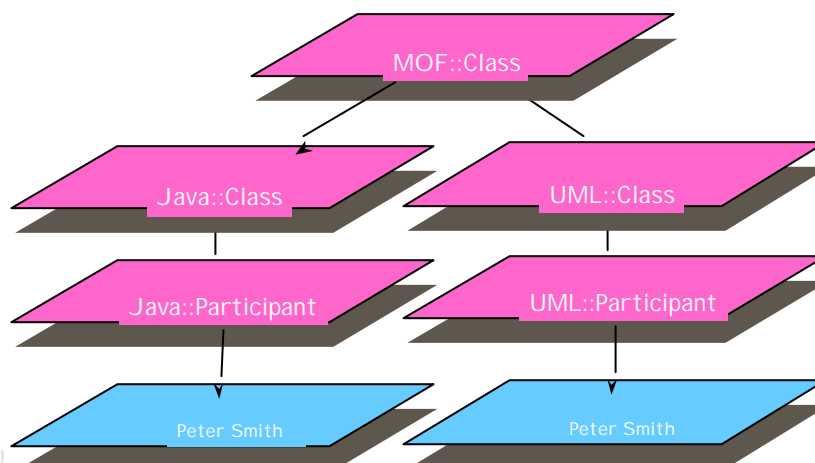


## MOF : some definitions



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

## Multiple meta-models



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

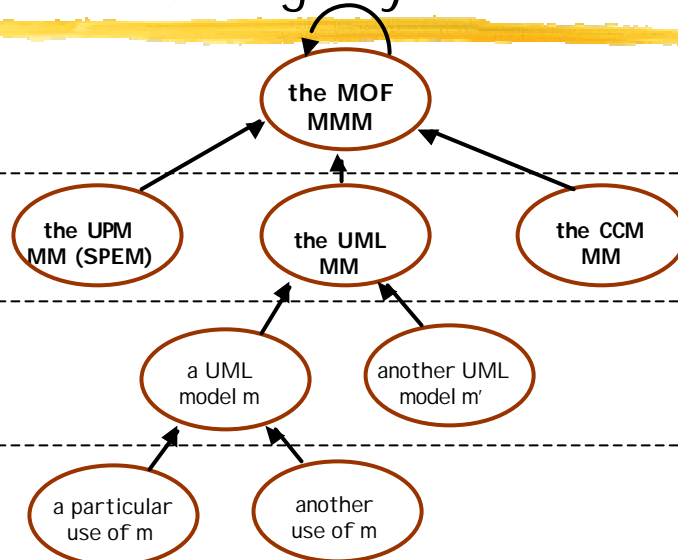
## The Three Modeling Layers

Level  $M^3$

Level  $M^2$

Level  $M^1$

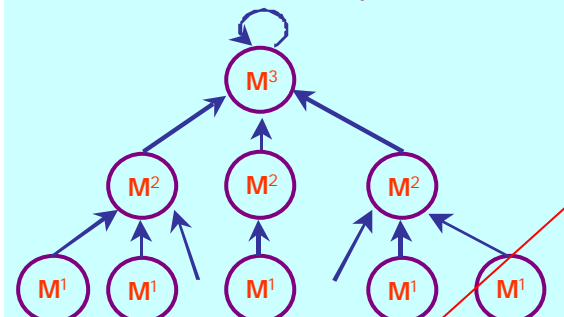
Level  $M^0$



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

## The OMG/MOF Meta-Model Stack

$M^1$ ,  $M^2$  &  $M^3$  spaces

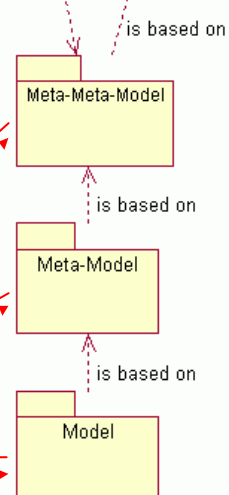


- One unique Meta-Meta-model (the MOF)
- An important library of compatible Meta-Models
- Each of the models is defined in the language of its unique meta-model

$M^3$

$M^2$

$M^1$



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

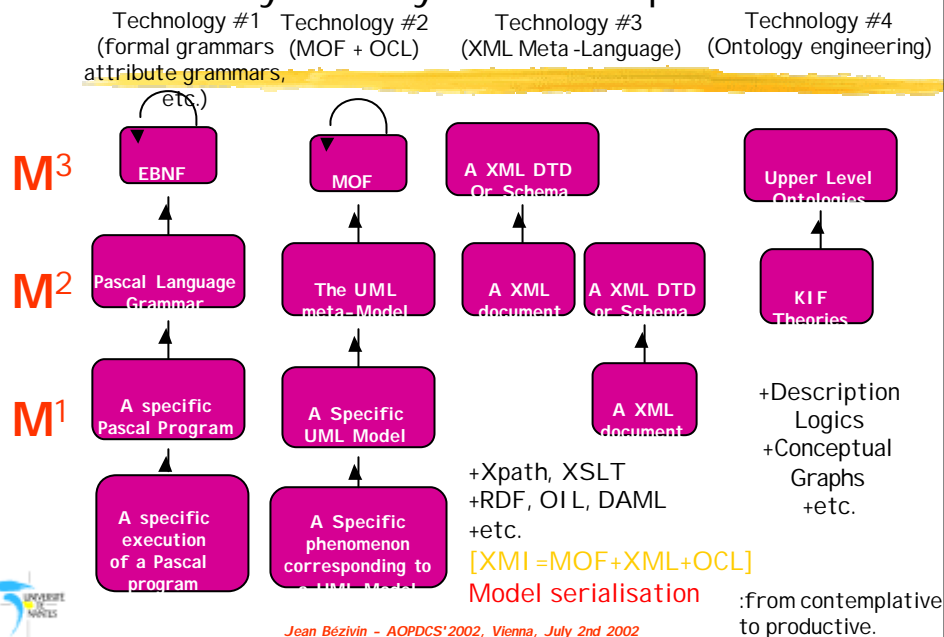
# UML profiles

- ⌘ A UML profile is a grouping construct for UML model elements that have been customized for a specific domain or purpose using extension mechanisms such as stereotypes, tagged values and constraints. For example, the UML Profile for CORBA RFP customizes UML for specifying CORBA IDL.
- ⌘ A meta-model defines a **domain-specific language**. A profile is a variant of a meta-model. It allows to define a **dialect** of a given language. There are a dozen of UML profiles that are currently being defined.



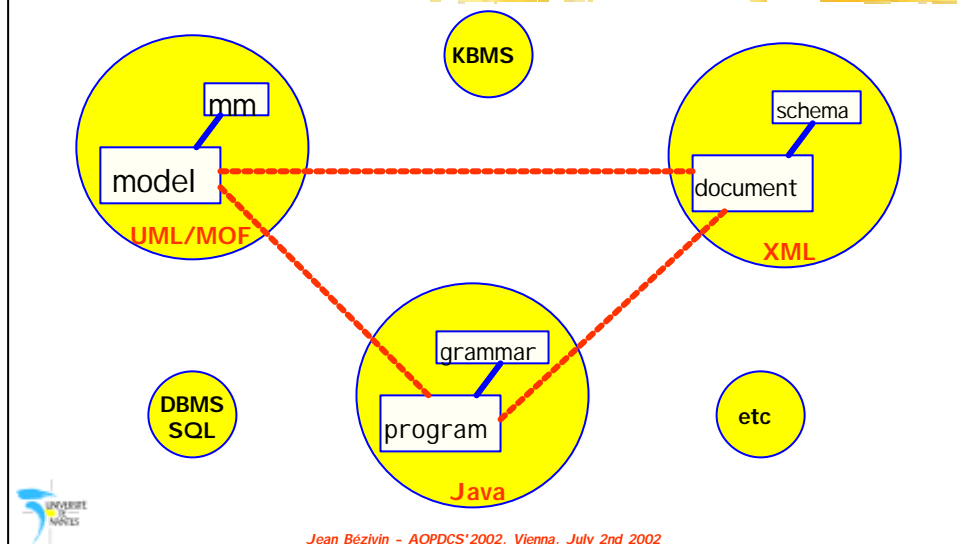
Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

## Abstract Syntax Systems Compared

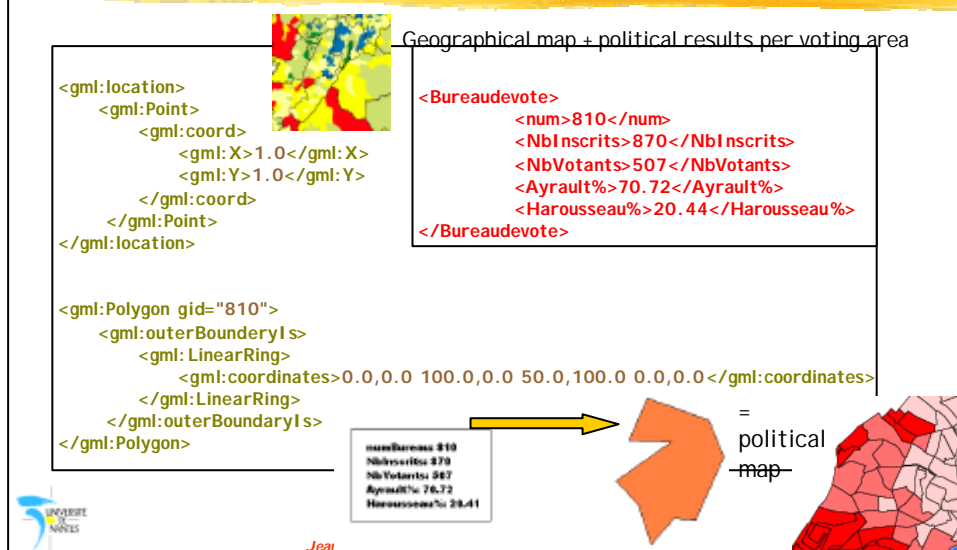


Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

## Some technological spaces

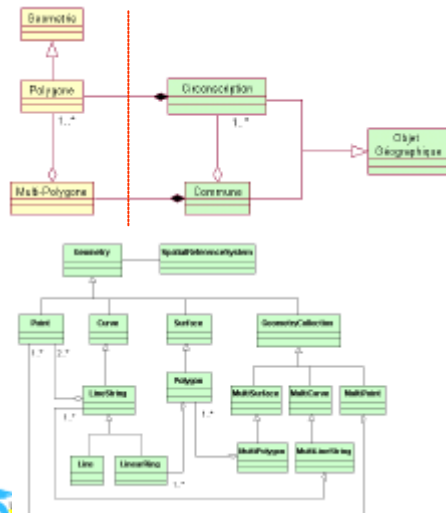


## Ex: Weaving geographical and political aspects in XML





# Model weaving

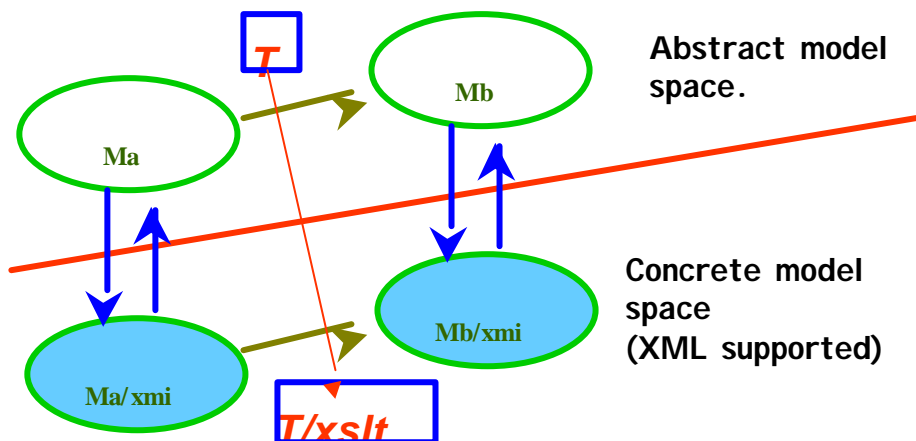


✂ XML technology offers tools for document merging (Xlink, Xpointer, etc.)

✂ It is also possible to perform the weaving operation within the model space and then to project to the XML space (SVG, GML, etc.)

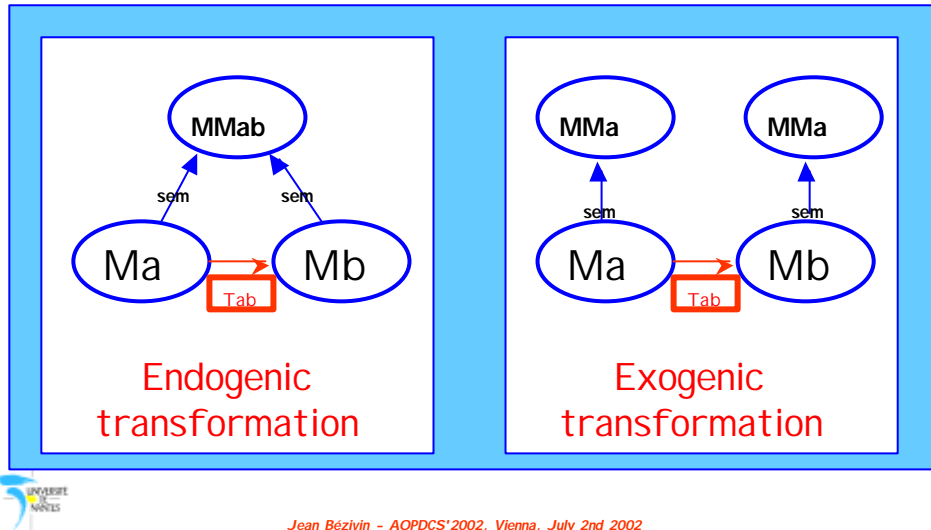
Jean Bézin - AOPDCS'2002, Vienna, July 2nd 2002

## Ex: Crossing the MOF to XML boundary



Jean Bézin - AOPDCS'2002, Vienna, July 2nd 2002

## Various kinds of transformations



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

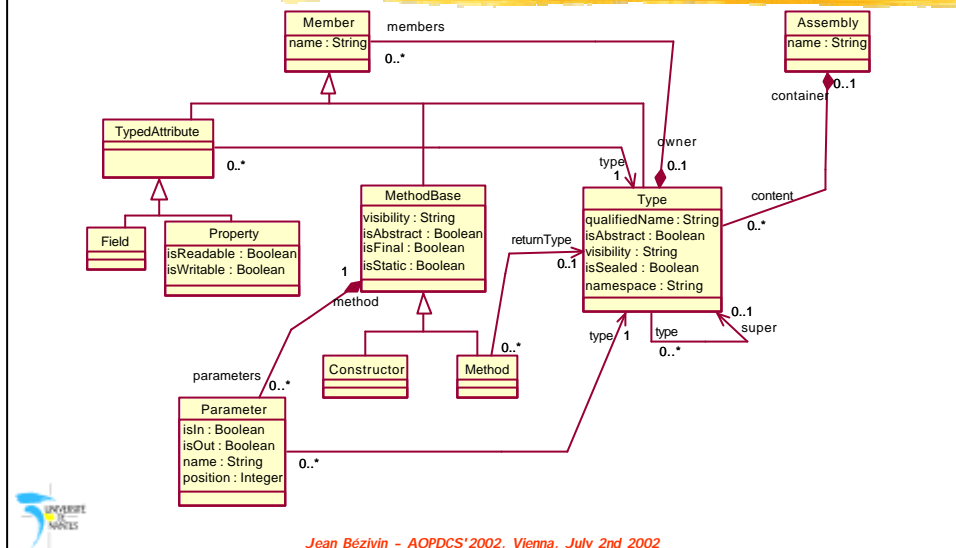
## Tooling the MDA : Sample

- %% Adaptive's Framework <http://www.adaptive.com/>
- %% France- Telecom Universalis <http://universalis.elibel.tm.fr/>
- %% Codagen Gen-it <http://www.codagen.com/>
- %% Codigo CodigoXpress <http://www.codigoexpress.com/>
- %% DSTC dMOF <http://www.dstc.edu.au/Products/CORBA/MOF/>
- %% Interactive Objects ArcStyler <http://www.io-software.com/>
- %% Kabira Business Accelerator <http://www.kabira.com/>
- %% Kennedy Carter iUML and ICCG <http://www.kc.com/>
- %% Metamatrix MetaBase <http://metamatrix.com/>
- %% NetBeans Meta Data Repository MDR <http://www.netbeans.org/>
- %% ONTOS ObjectSpark <http://www.objectspace.com/>
- %% ObjectRad Java Metadata Server <http://www.objectrad.com/>
- %% ObjeXion Software Netsilon <http://www.netsilon.com/>
- %% Project Technology BridgePoint/DesignPoint <http://www.projtech.com/>
- %% Secant Technologies ModelMethods <http://www.modelmethods.com/>
- %% Soft-Maint Scriptor & Semantor <http://www.sodifrance.fr/>
- %% Tata Research Development ADEX <http://www.tcs.com/>
- %% University of Berne MOOSE <http://www.iam.unibe.ch/>

and much more...

Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

## Part of the C#/DotNet MetaModel



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

## The Y cycle : weaving PIMs & PDMs to produce PSMs

**PIMs**  
(Platform  
Independent  
Models)

Merging/binding phase

**PSMs**  
(Platform  
Specific  
Models)

binding

**PDMs**  
(Platform  
Description  
Models)



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002



# MDA: beyond the buzzword

- ⌘ Modern model engineering techniques are ready for prime time in software engineering. They are based on:
  - ✓ A four level architecture (3+1)
  - ✓ A unique meta-meta-model (MOF),
    - ⊗ with transfer and exchange mechanisms
    - ⊗ with transformation mechanisms
    - ⊗ with standard projection mechanisms on a variety of middlewares (CORBA first, Java and DotNet next, ...)
  - ✓ A growing collection of specialized meta-models (evolutive)
    - ⊗ Object meta-models (Java/EJB/J2EE, CLR, CCM, etc.)
    - ⊗ Legacy meta-models (Relational, CWM)
    - ⊗ Enterprise meta-models : Business objects, Healthcare, Transportation, Process & Rules, and much more
    - ⊗ Product and process meta-models (e.g. workflow, RUP)
- ⌘ Automatic and semi-automatic generation tools, from high abstraction standardized models to various middleware platforms will progressively appear in the coming years.



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002



# Conclusion

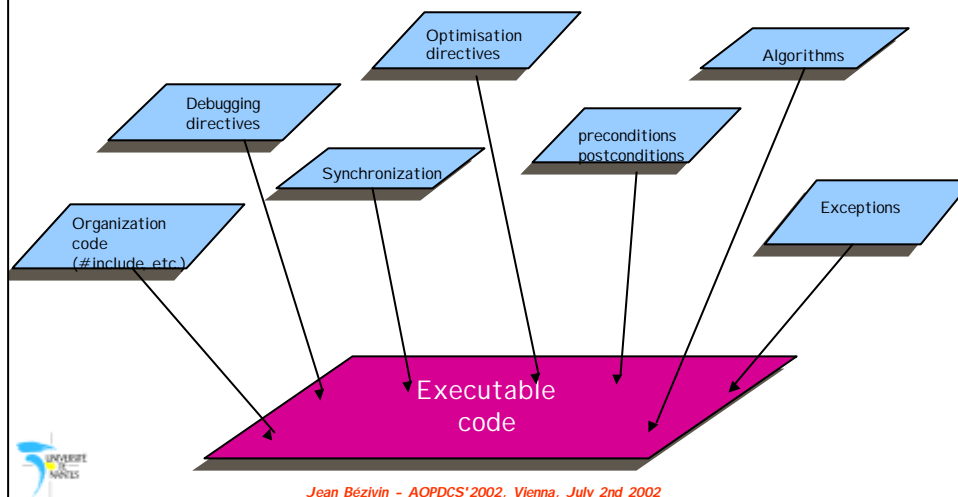
## ⌘ Model engineering is the future of object technology

- ✓ As object and classes were seen in the 80's as "first class entities", with libraries of several hundred of classes hierarchically organized, models and meta-models are beginning to be considered alike in the 2000's.
- ✓ Libraries (lattices) of hundreds of meta-models (ontologies) of high abstraction and low granularity are beginning to appear. Each such meta-model may contain several hundreds of concepts and relations.
- ✓ Tools will be needed to work with these vast libraries of models and meta-models.
- ✓ This will have a rapid impact on the daily work of the information engineer.
- ✓ More research is urgently needed to bring together the people involved in the theory and practice of model engineering (ontologists, methodologists, software practitioners, information system builders, database specialists, etc.).

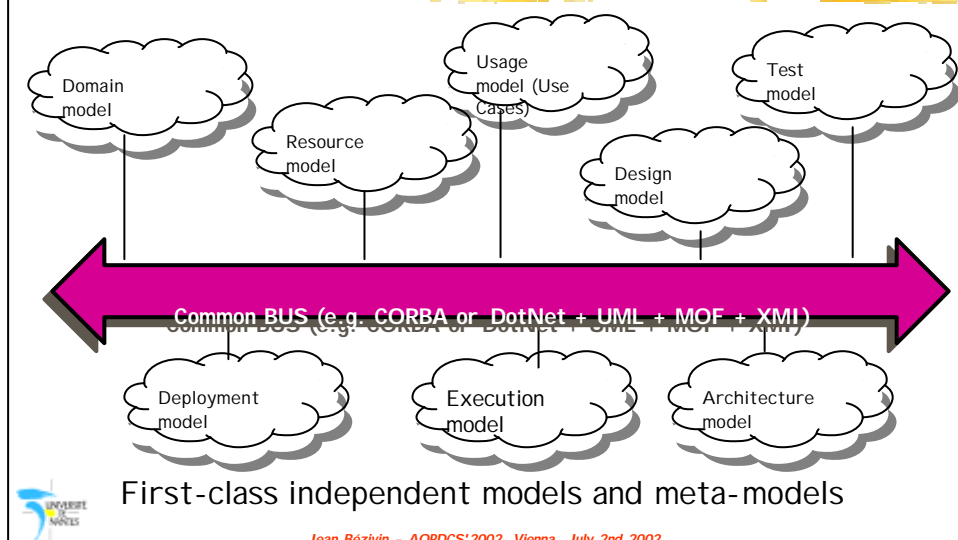


Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

## AOP: the code as unique reference



## The on-line separate models organization (aspect-oriented software engineering)



## Where are models coming from?

- ⌘ Essential models (resource, business logic, service)
- ⌘ Other development sources (exception handling, testing, user behaviour, enforcing contracts, performance improvement, deployment, security, authentication, concurrency management, etc.),
- ⌘ Legacy systems,
- ⌘ Derived models,
- ⌘ Executable code extraction (code is a model)
- ⌘ Multiple other sources, like Just in Time Model Production
- ⌘ etc.



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002

## AOM: oxymoron or pleonasm?



- ⌘ The central issue in AOSD is knowledge management (aspects + operations)
  - ✓ AOP gives a key role to the code (executable code) for this representation problem.
    - ⊗ As a consequence the problem of final mapping to executable code is solved
    - ⊗ But the problem of precise definition, capture and maintenance of the various aspects is made more difficult
  - ✓ On the contrary, MDA/MDE starts from the a priori of an independent and homogeneous representation of aspects in non-executable "external spaces"
    - ⊗ The conceptual handling of separate aspects and operations on these aspects is considerably simplified
    - ⊗ The practical handling of mapping to executable platforms (operationalization) still requires considerable research and development efforts that will go much beyond compiler technology and rewriting systems.
- ⌘ Therefore, the conclusion:
  - ✓ From the point of view of representation systems, AOM is an oxymoron
  - ✓ From the point of view of the problem to be solved (separation of concerns), AOM is a pleonasm.
  - ✓ AOM is more a pleonasm than an oxymoron  $\Rightarrow$  There seems to be more place for cooperation than competition between both approaches



Jean Bézivin - AOPDCS'2002, Vienna, July 2nd 2002