Combining the Power of Meta-Programming and Meta-Modeling within the OMG MDA Framework.

Jean Bézivin & Nicolas Ploquin

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

2nd Workshop on UML for Enterprise Applications: Model Driven Solutions for the Enterprise

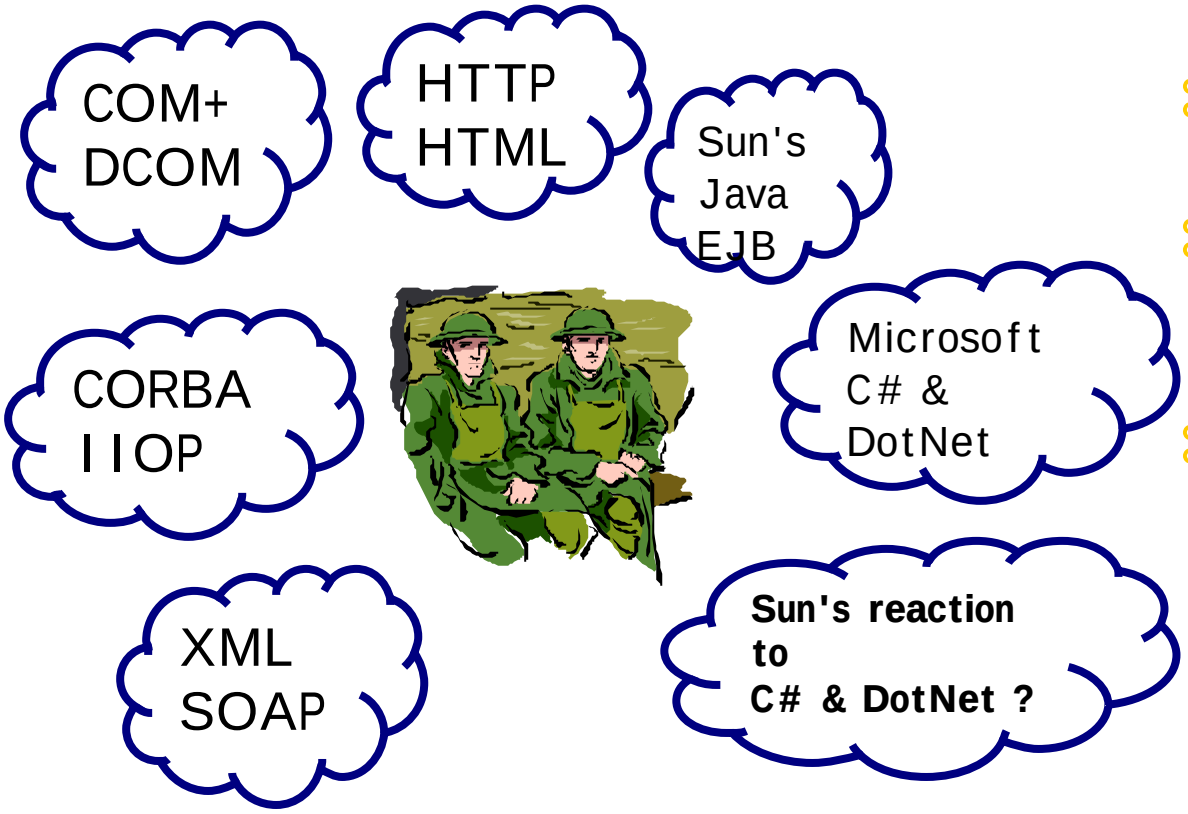
Outline

- Why the MDA?
  - ✓ Rapid paradigm shift from objects to models
- Basic concepts of the MDA
  - ✓ Revisiting the 4-layer architecture
- Tooling the MDA
  - ✓ A tour of models and tools
- Model extraction
  - ✓ Static MX
  - ✓ JIT/MP
- Conclusion

Why the MDA?

- Objects failed to bring conceptual simplicity
- Platform migration is too frequent and too costly
- New models are emerging
- Models for humans and models for computers
- The unique (object) model is replaced by the multiple model
- Consequence : The middle-war is over

The middleware war is over

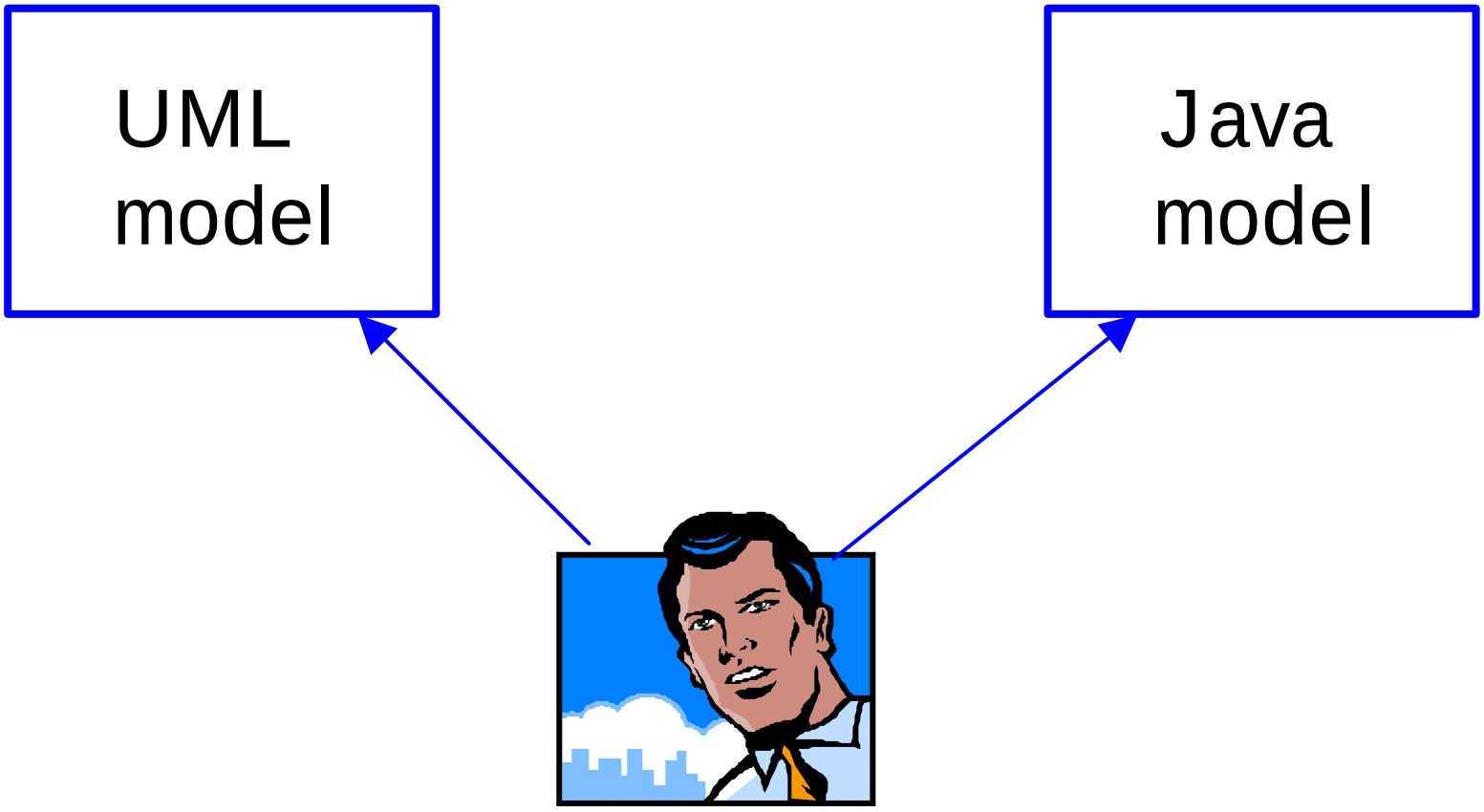


- There is no clear winner nor loser
- The next battlefield will be model transformation
- The OMG's Model Driven Architecture (MDA) initiative is aimed at using modelling and meta-modelling to drive the design and implementation of distributed systems.

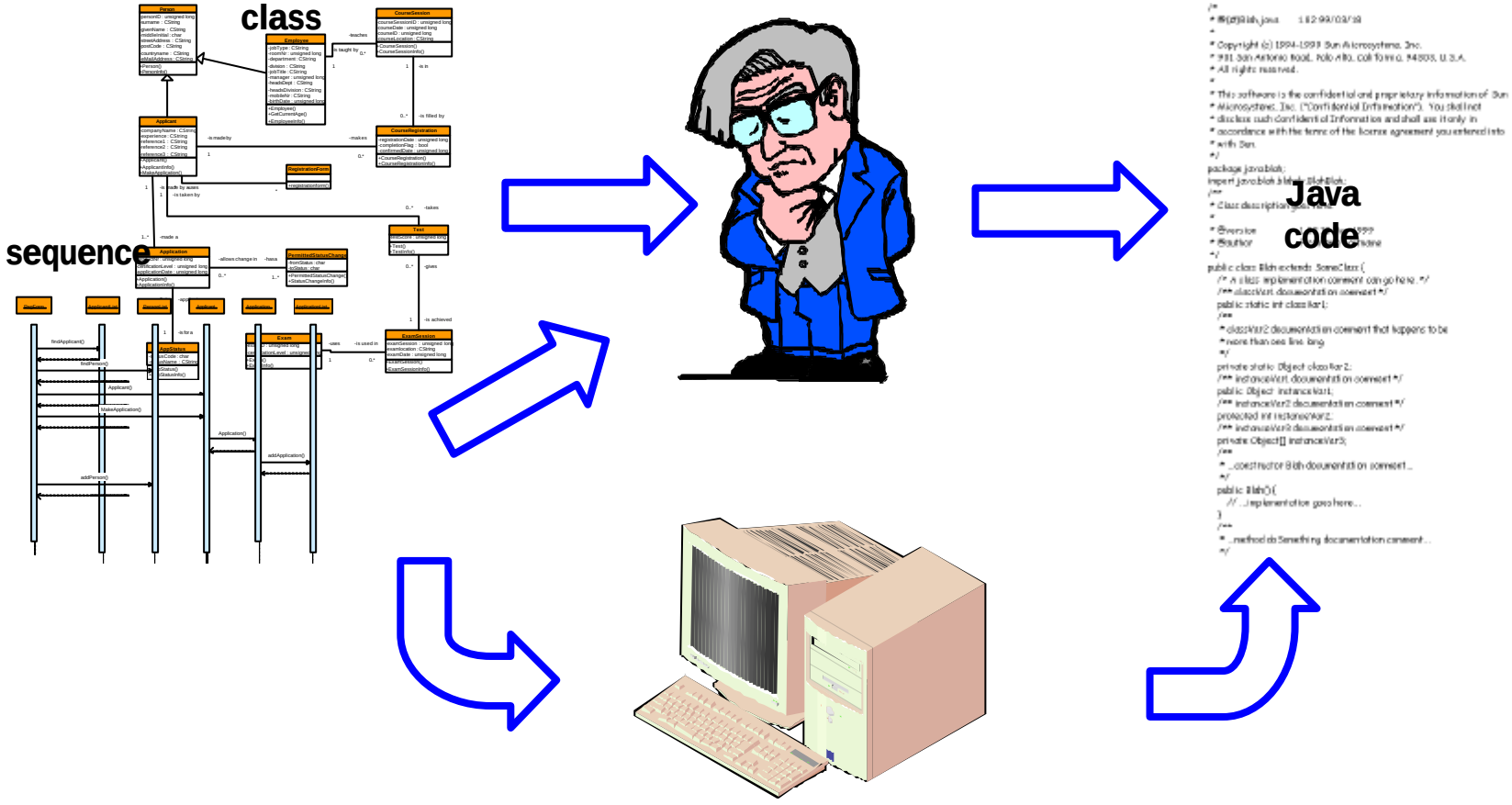
New models are emerging

- From object to components …
- … and then to:
  - ✓ Rules,
  - ✓ Workflow,
  - ✓ Services,
  - ✓ …

Consequence: having to deal simultaneously with several models of different semantics



From contemplative to productive

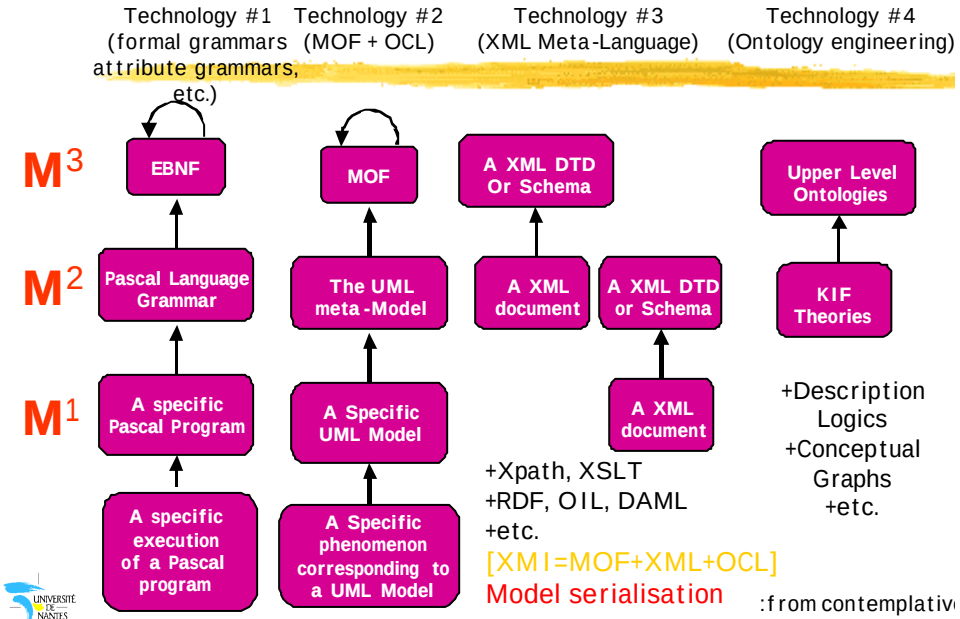


From human-readable to computer-understandable (XMI)

Basic concepts of the MDA

⌘Revisiting the 4-layer organization

Abstract Syntax Systems Compared



Technologies evaluation grid

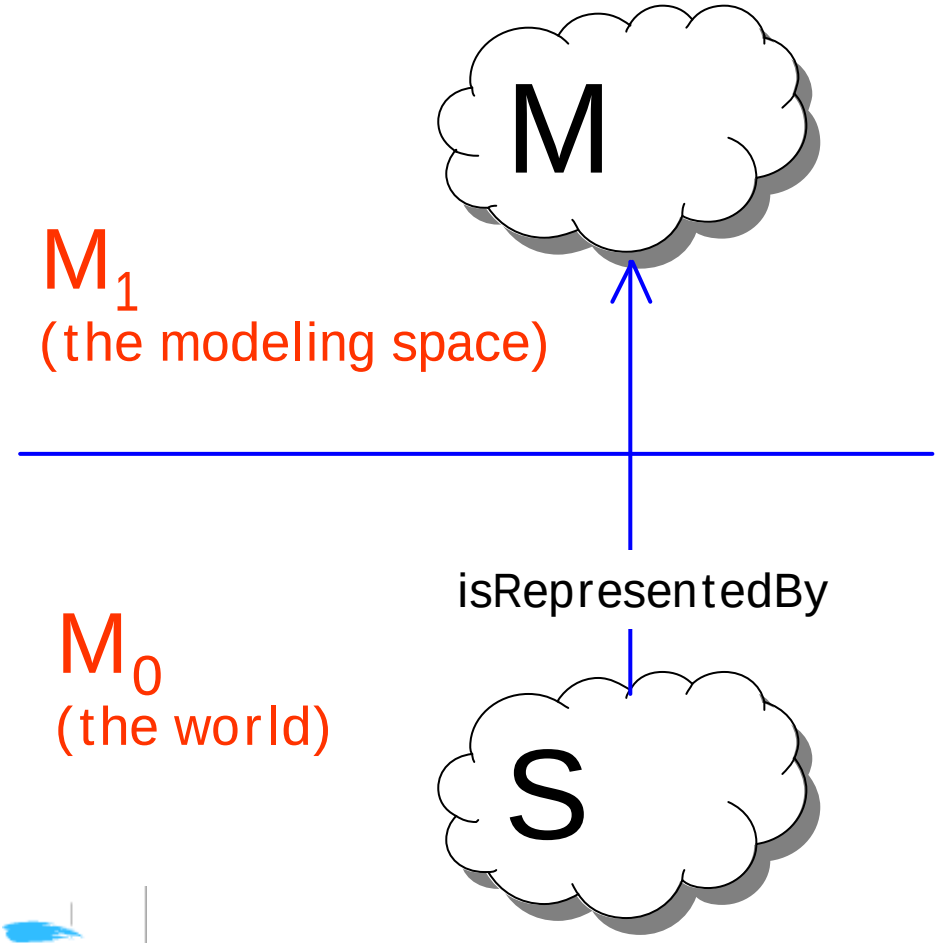
Different problems

Different technological families

	Abstract syntax systems	Meta-models	XML	Ontologies	Data bases
modularity					
transformation					
rendering					
executability					
verification					
navigation					
etc.					

2nd Workshop on UML for Enterprise Applications: Model Driven Solutions for the Enterprise

Systems and models



A model M is a simplified representation of the world, as a matter of fact of only a part S of the world called the system.

Limited Substituability Principle

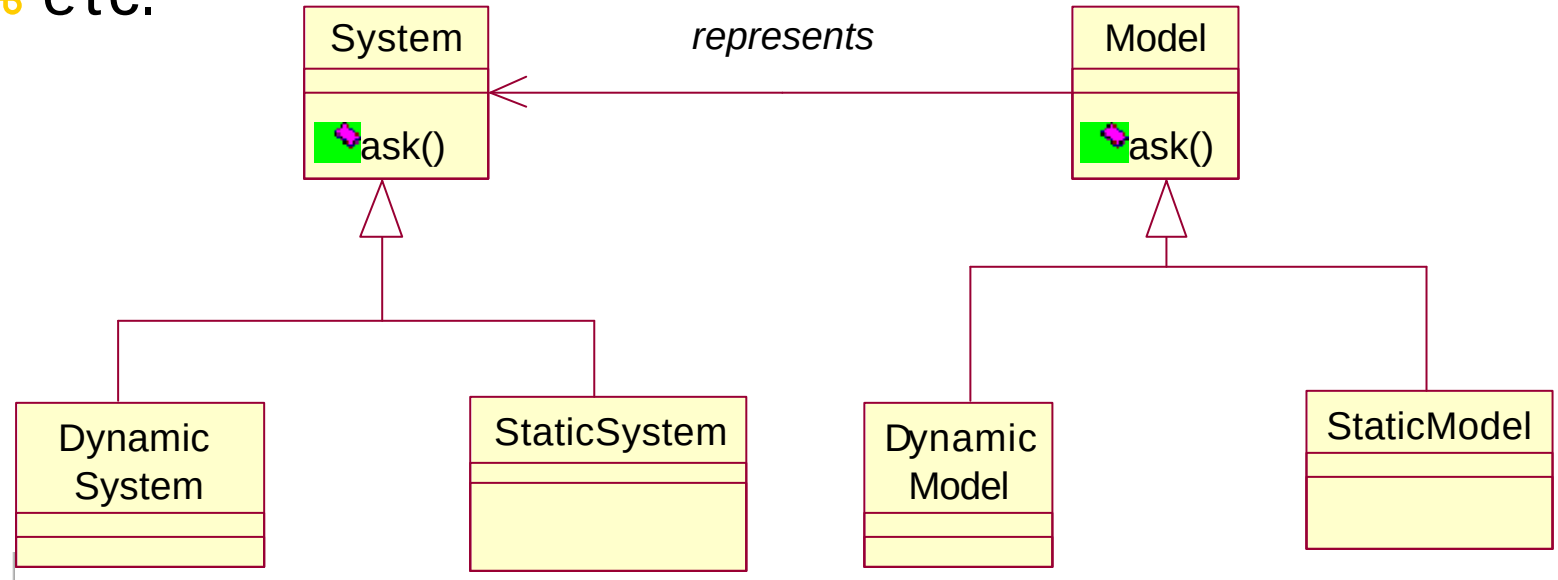
- The purpose of a model is always to be able to answer some questions in place of the system, exactly in the same way the system itself would have answered similar questions.

The global MDA model space

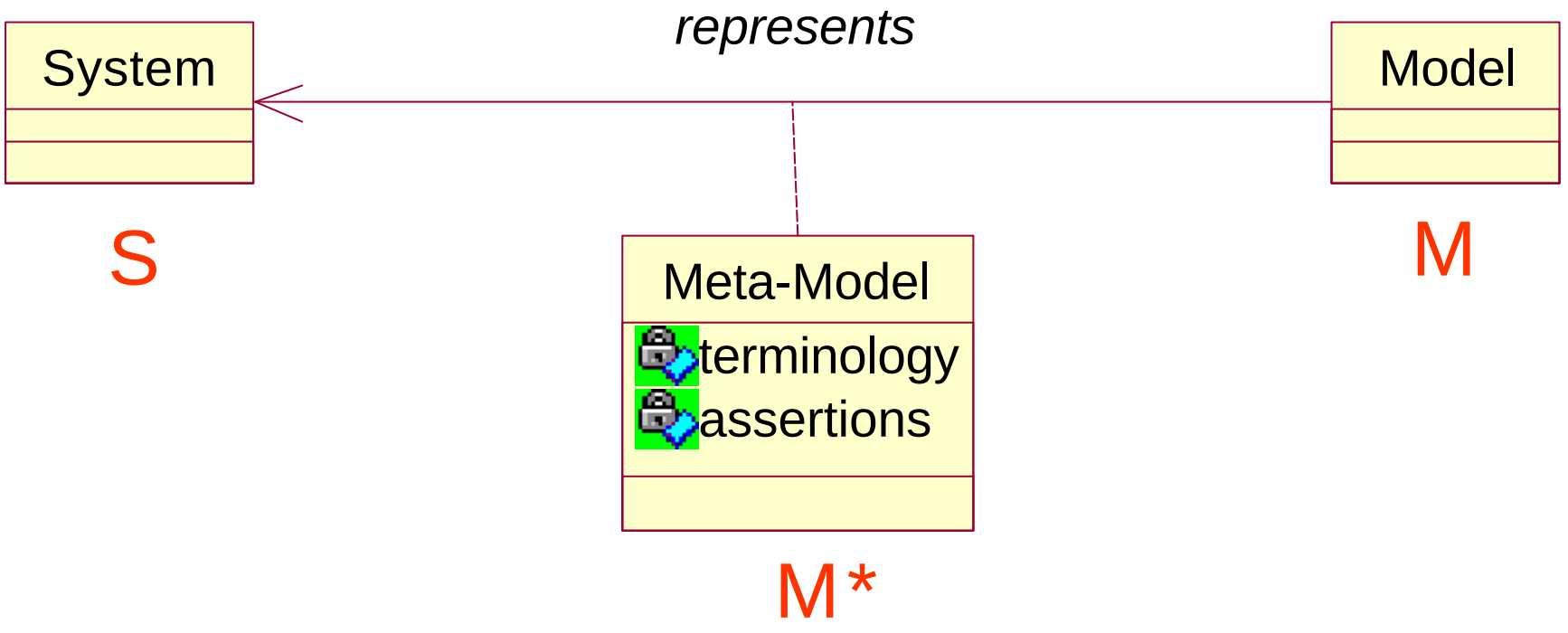
- The development software cycle is populated with models
  - Models are of unequal importance
  - The model space is structured
  - Models are linked in a complex organization network
  - The content of each model is defined (constrained) by a corresponding meta-model (ontology)
  - The model space is constantly broadening starting from the essential models (Domain, Service, Resource)
- Many different kinds of models
  - Business models and computer models
  - Models of product & models of processes
  - Object, component, rule, workflow, service models among others
  - Legacy (Cobol, RDB) and NT (Web, SOAP, etc) models
  - PSMs , PIMs , PDMs,
  - etc.

Various kinds of models

- Products and processes
- Legacy and components
- Static and dynamic
- etc.

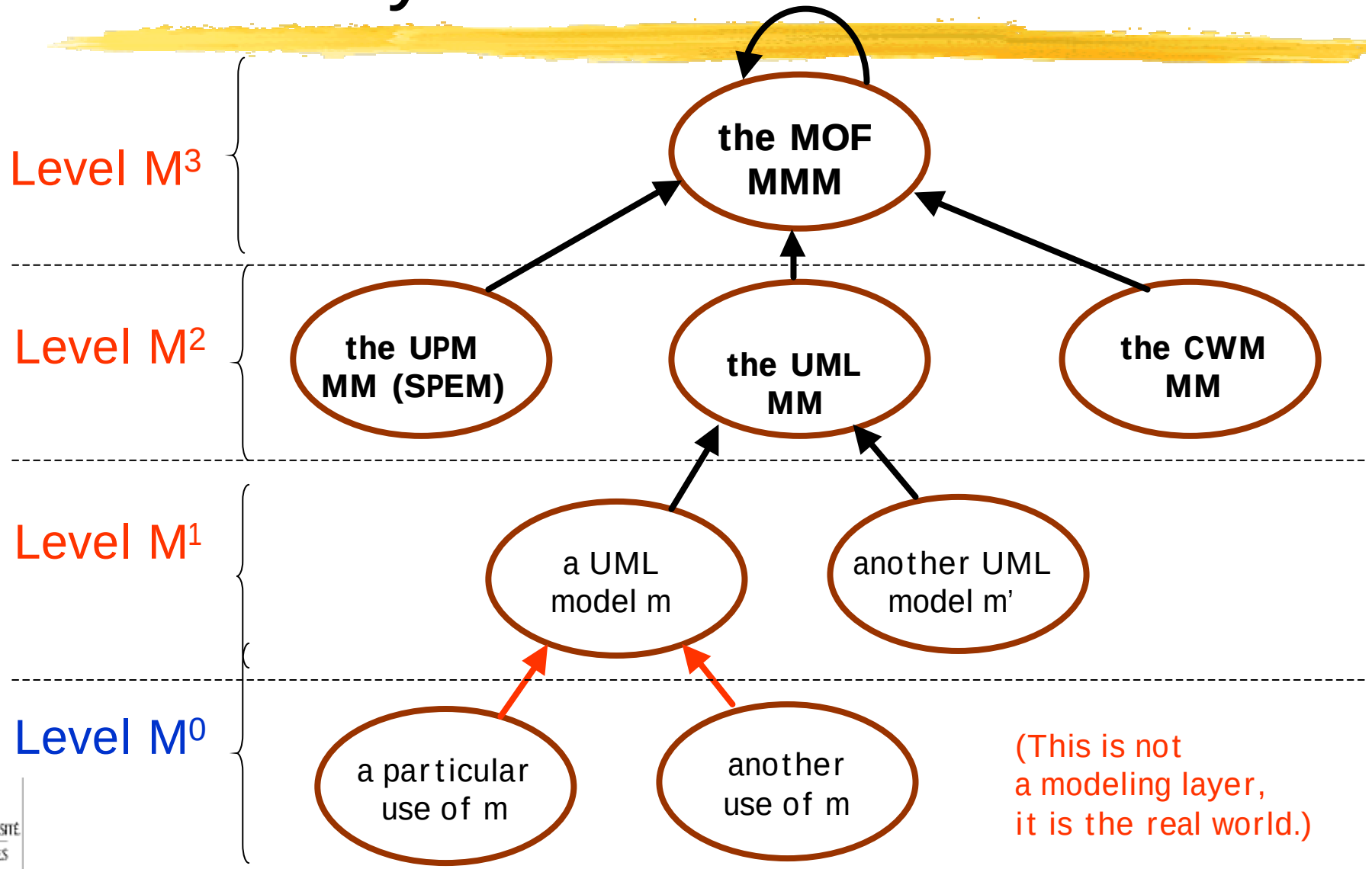


What is a Meta-Model?

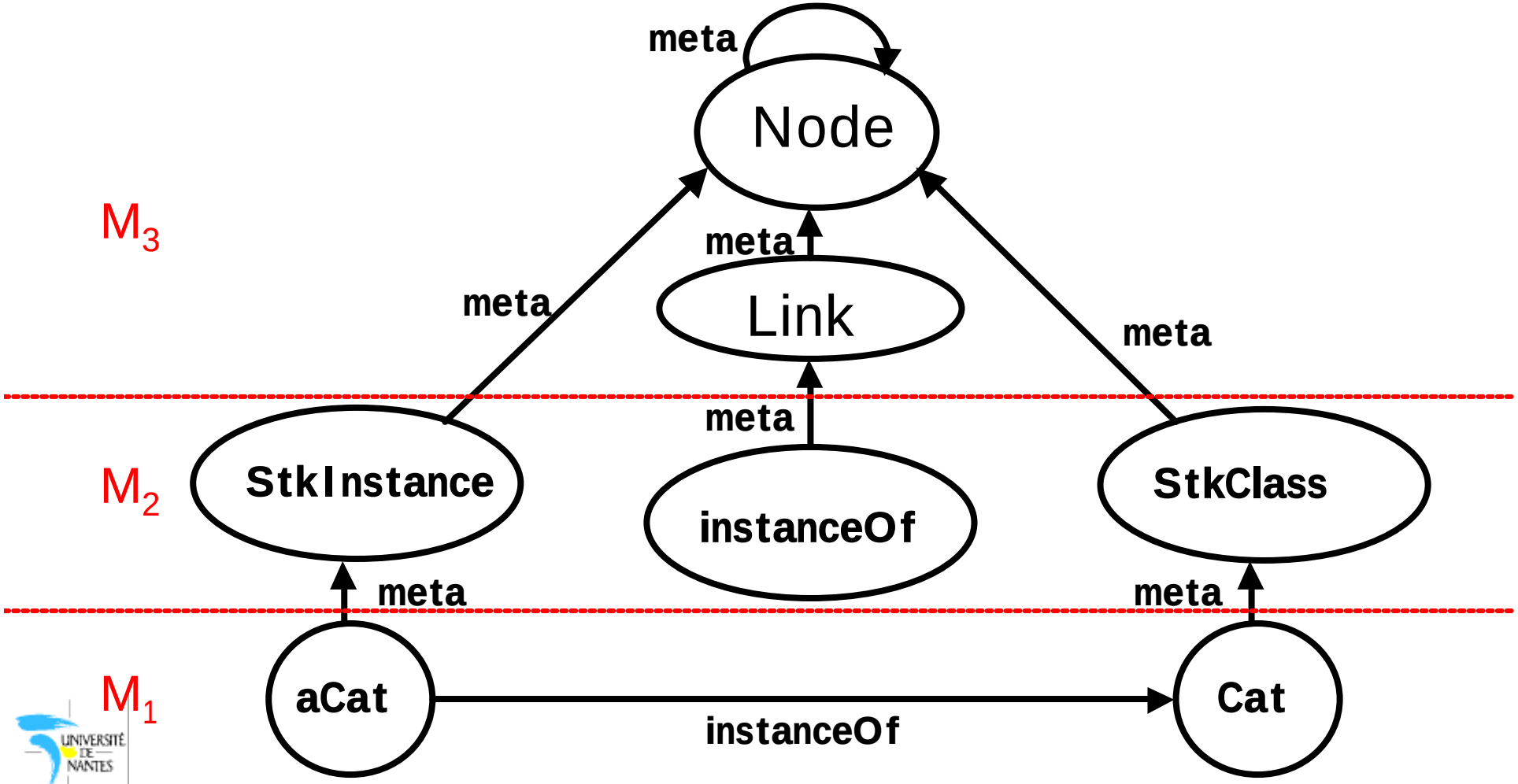


The correspondence between a model and a system is defined by a meta-model.

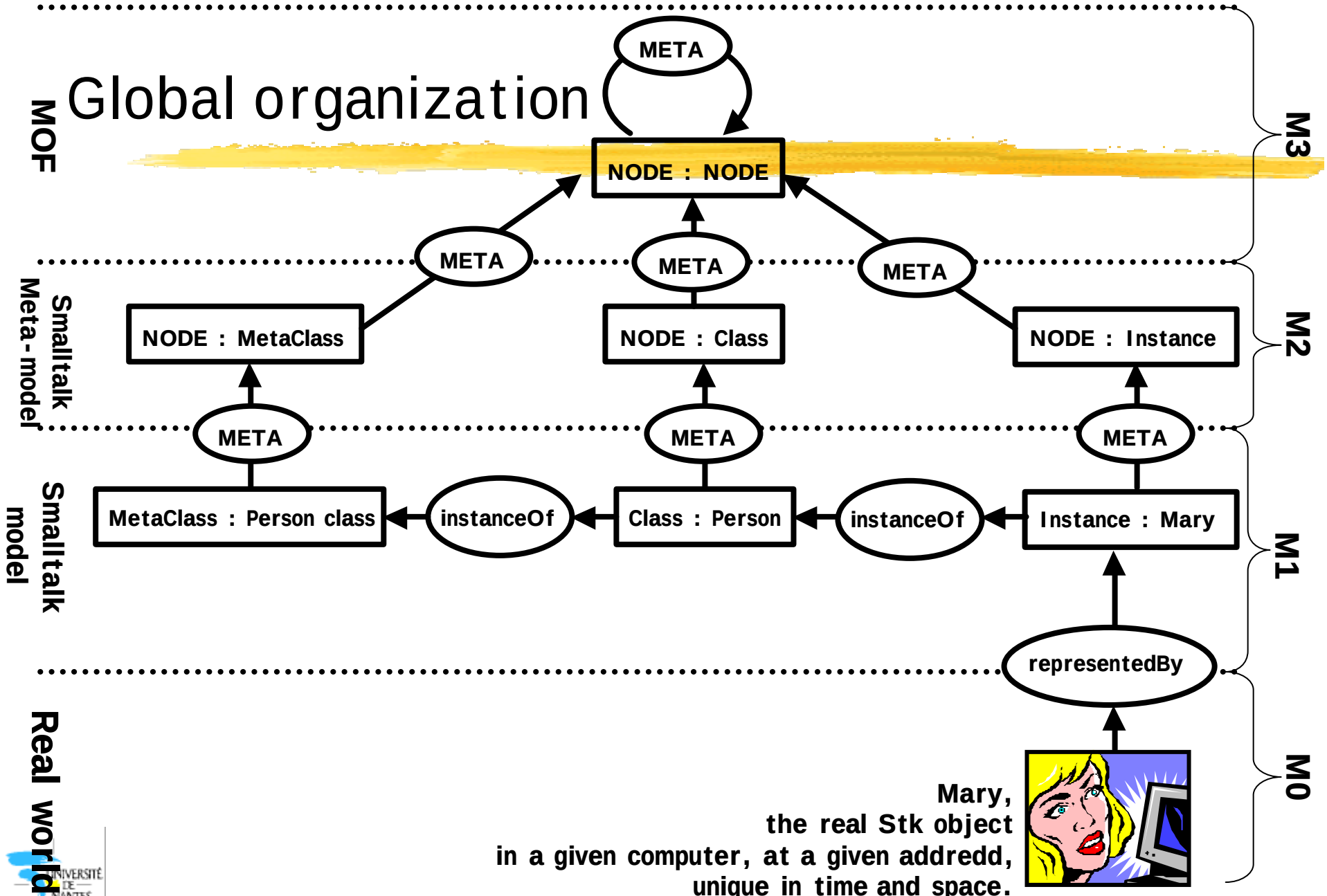
The 3+1 Layers



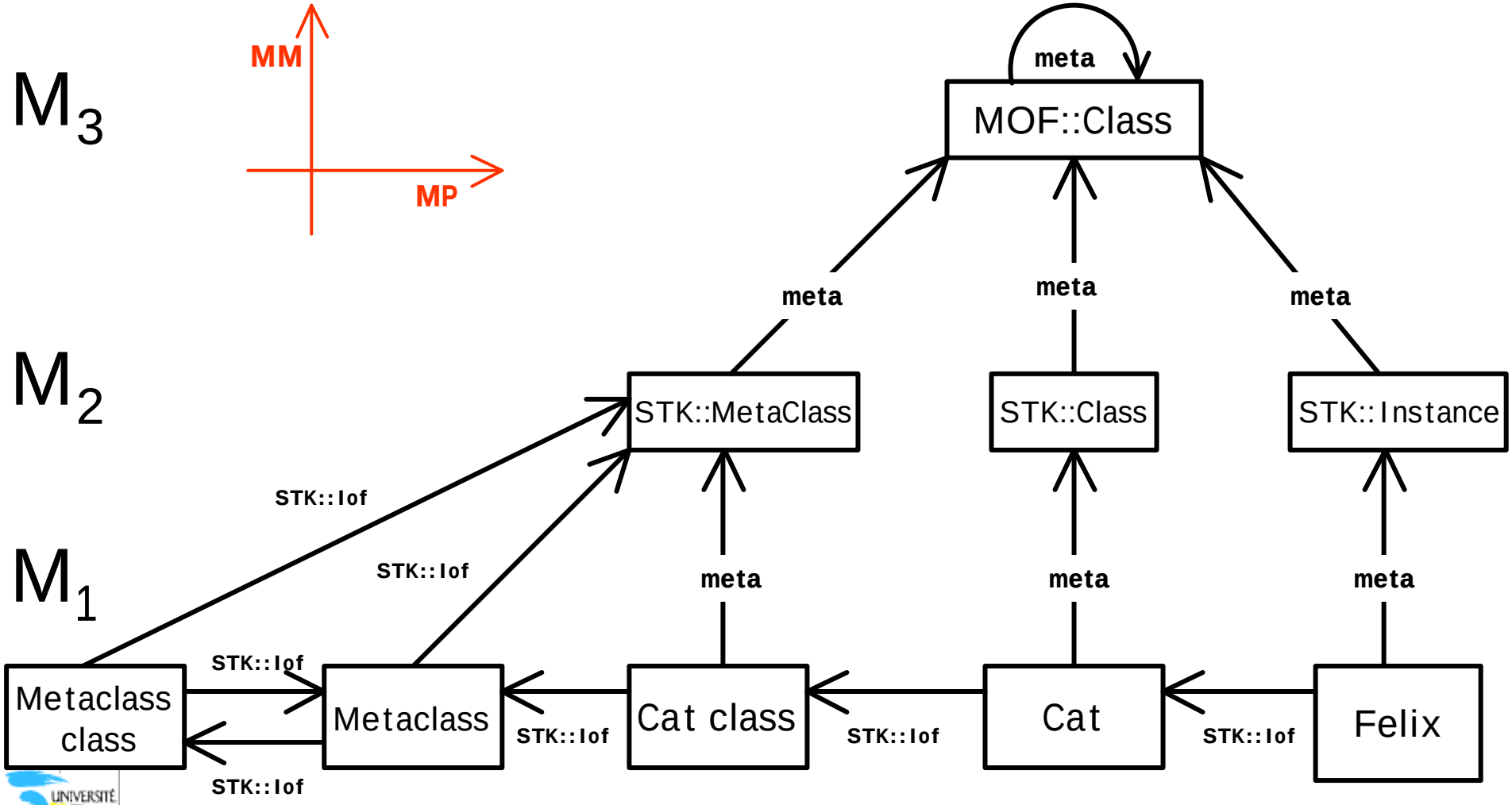
Local and global definitions



Global organization



Meta-programming vs. Meta-modeling

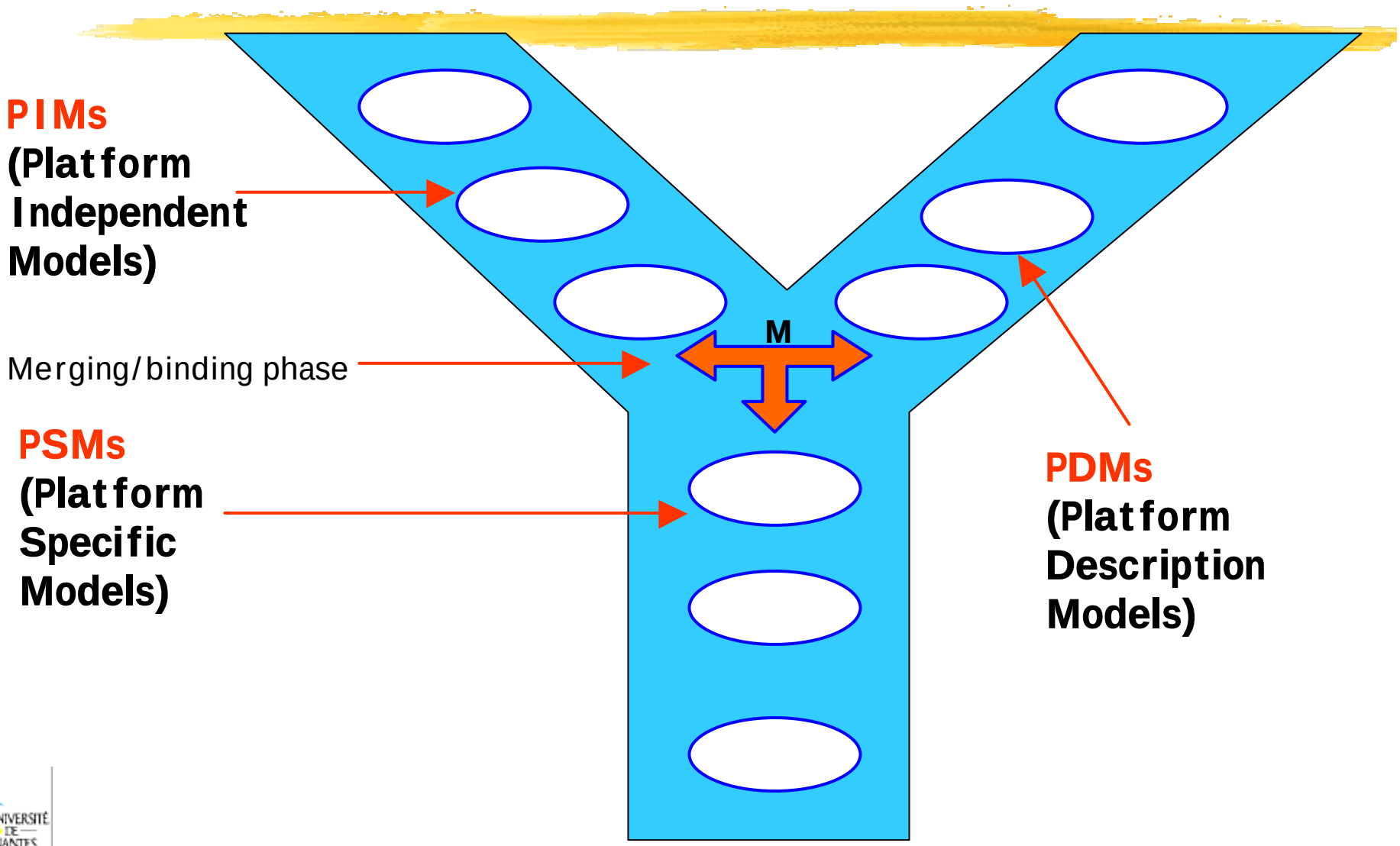


Tooling the MDA : Sample

- 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.codigoxpress.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.objectspark.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...

Example : The Y cycle

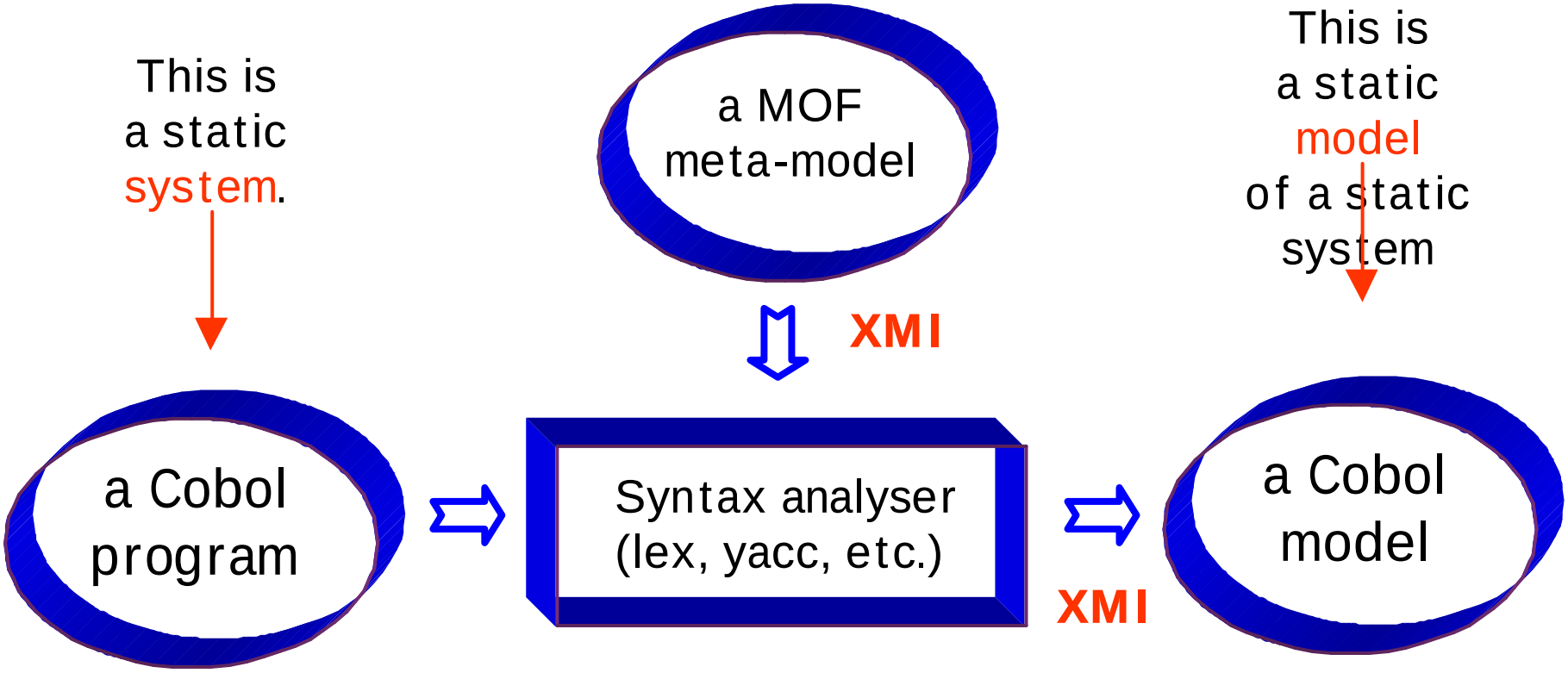


What is a pure "MDA tool"?

- It implements some operations on models or meta-models
- It is compliant with the main MDA recommandations (UML, MOF, XMI, etc.)
- It is interoperable with other MDA tools
- It is compatible with the MDA vision (meta-model driven)

- Operations on M. & MM.
  - ✓ M. checking
  - ✓ M. transformation
  - ✓ M. merging
  - ✓ M. presentation
  - ✓ M. & MM. browsing
  - ✓ Code generation
  - ✓ Reverse engineering
  - ✓ MM. Alignment

Source code model extraction

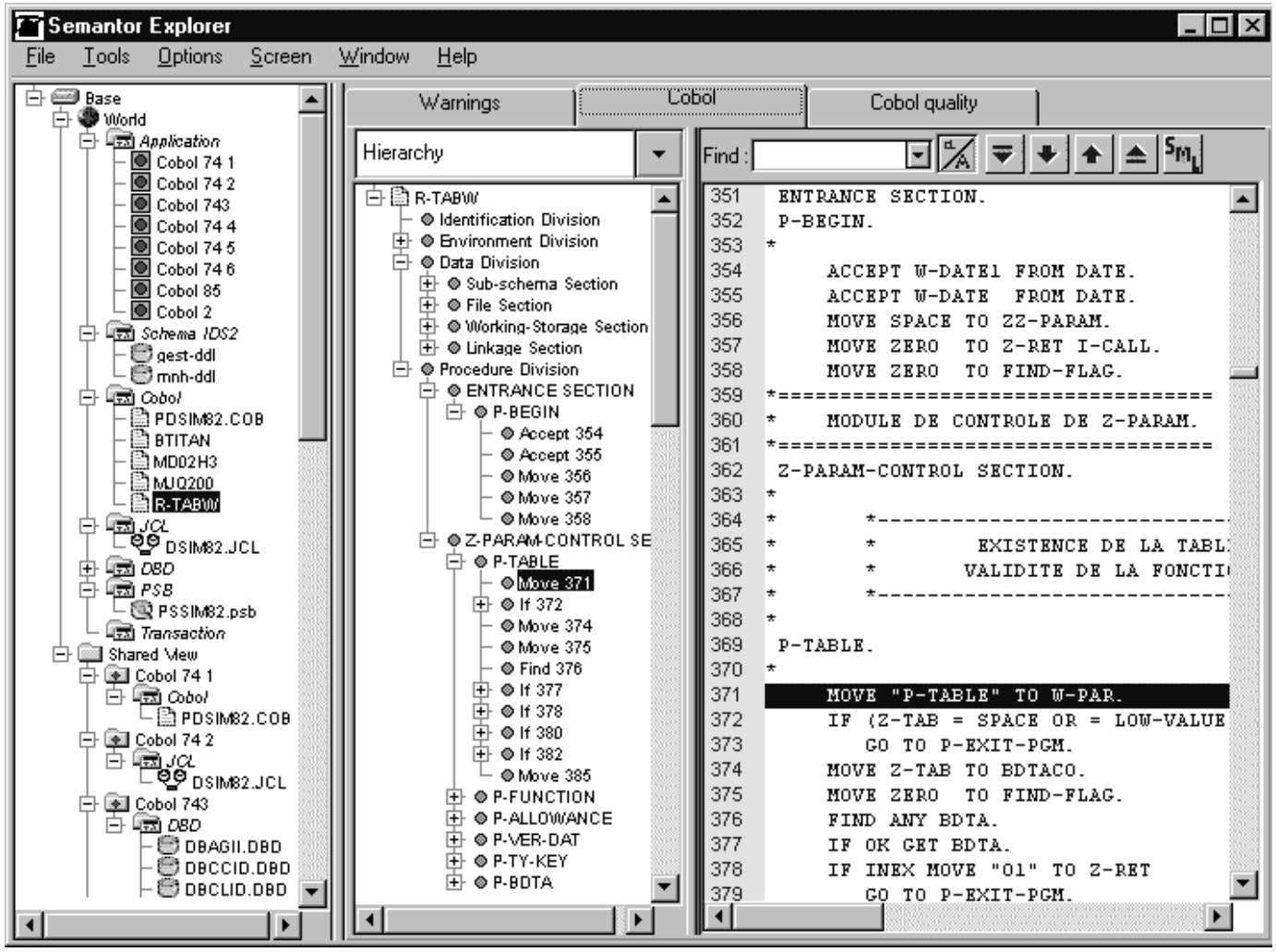


Meta-model driven SC (source code) model extraction

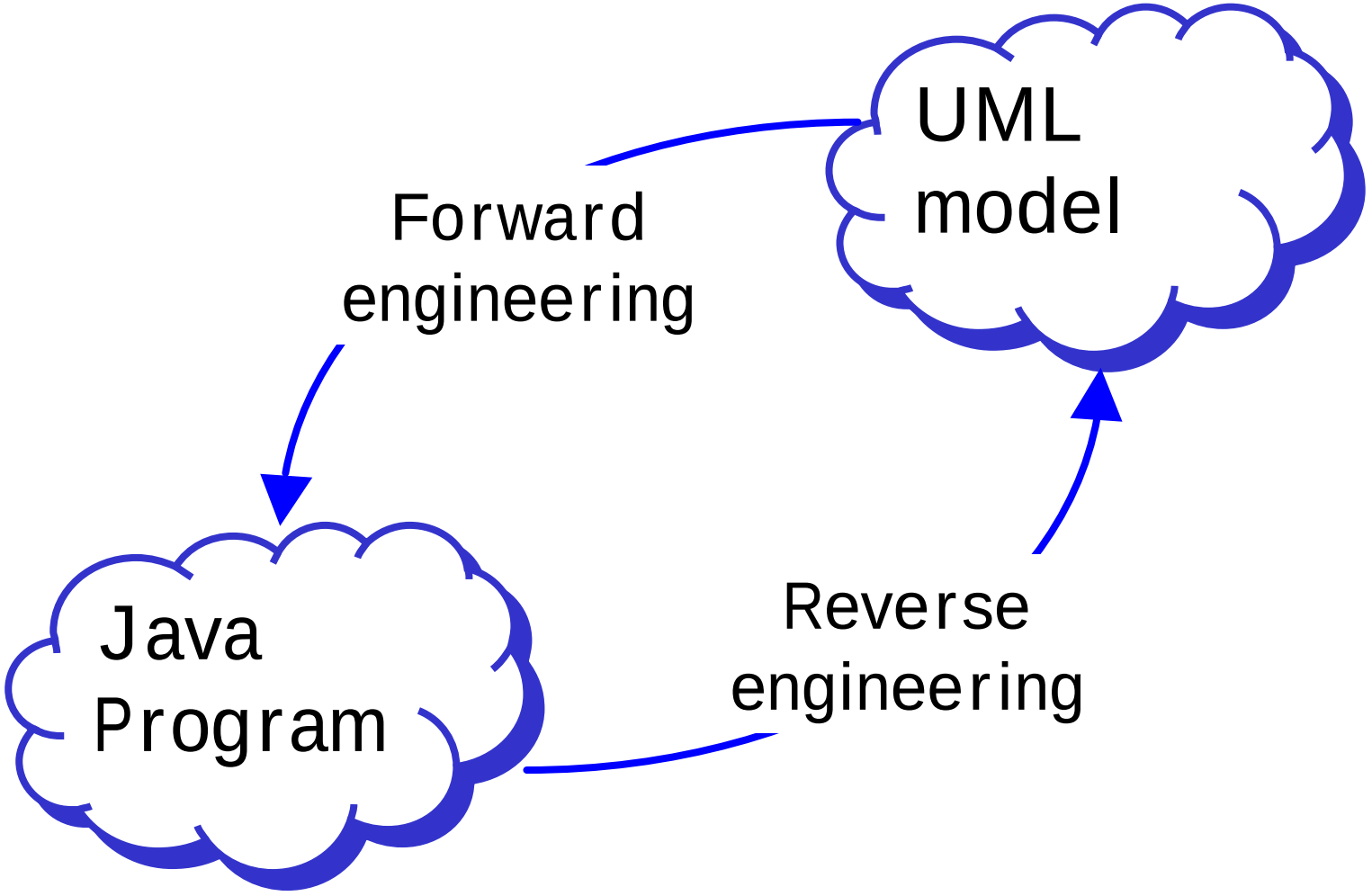
Example :Semantor Explorer

Full
Syntax
Source
Code
Model
Extraction

(once extracted, models can be worked on)



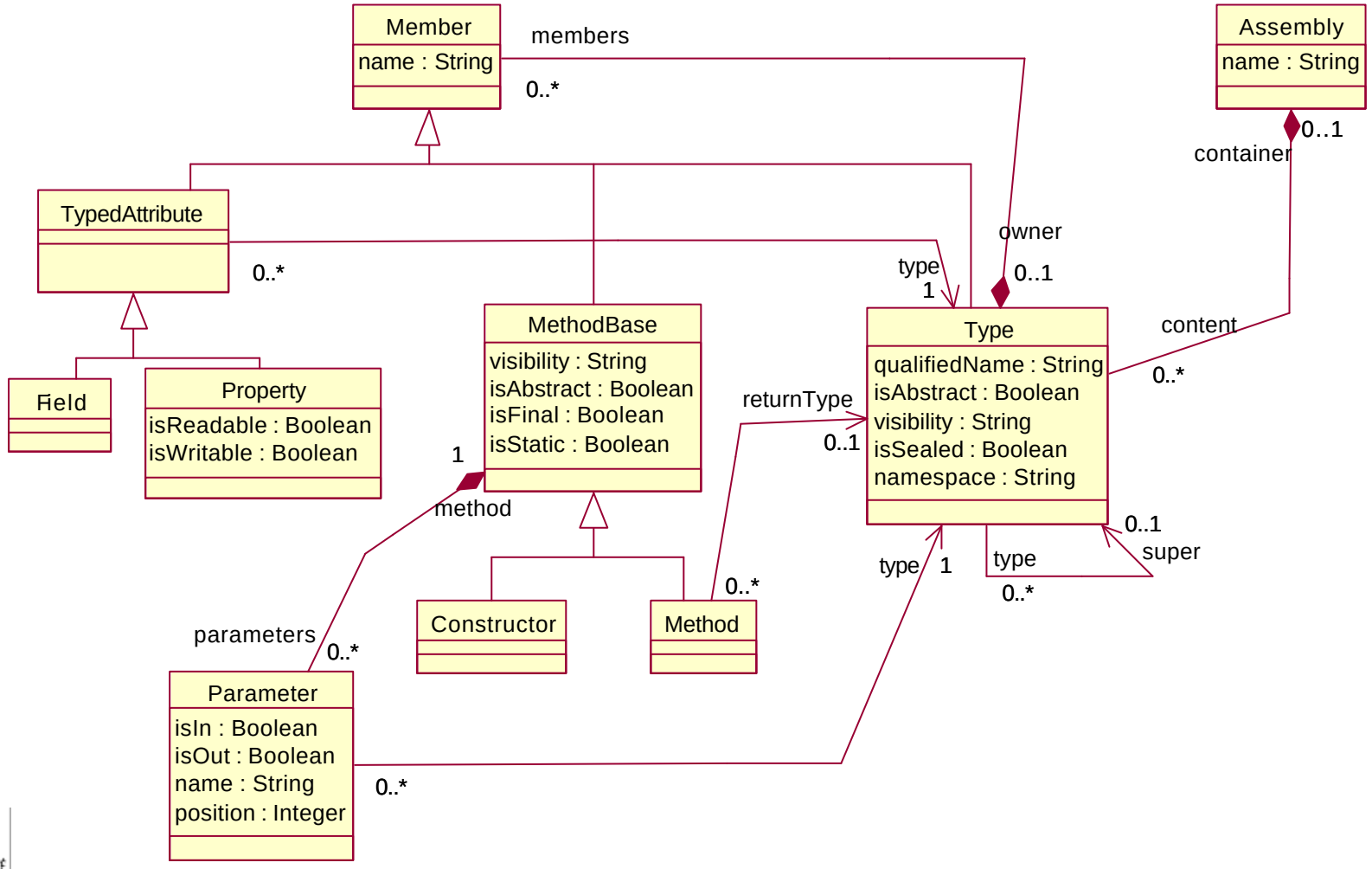
Round-Trip Engineering



MM-driven SC model extraction

- What kind of MM for source model extraction ?
  - ✓ Full syntax
    - ☒ 100% of the info extracted
    - ☒ Operation is reversible
  - ✓ Partial syntax
    - ☒ Non reversible operation
  - ✓ Semantic
    - ☒ Need some heuristics
    - ☒ Examples
      - **Process classes**
      - **Exception classes**
      - **Interface classes**
      - **Business classes**
- Important (and obvious) remark:
  - ✓ **Class instances only exist at run-time (usual instances, processes, exceptions, etc.)**

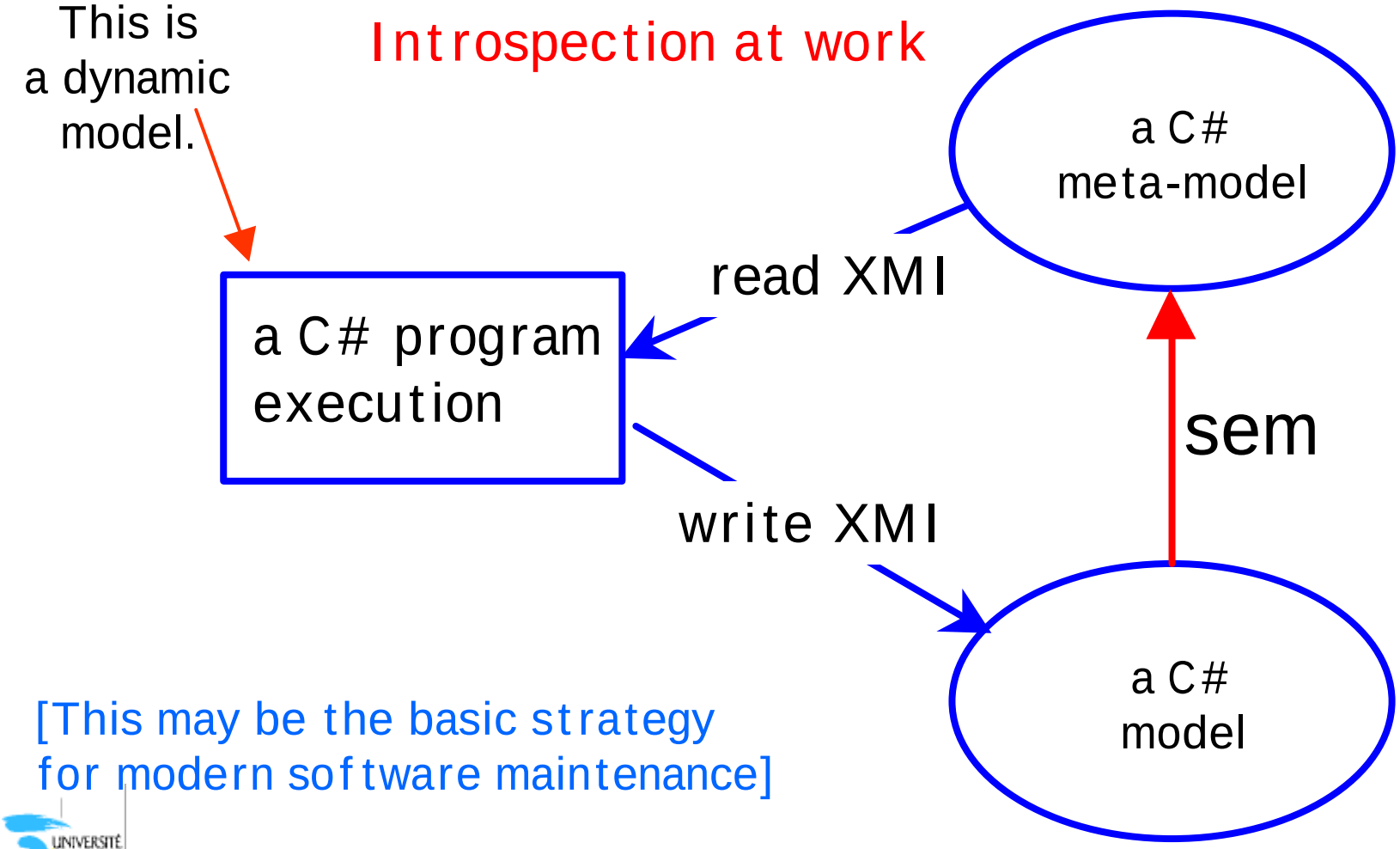
Part of the C# MetaModel (grammar)



JIT/MP : a new concept

- Just In Time Model Production
- An execution of a program dynamically produces an (XMI) model of its current situation
  - on demand,
  - Periodically,
  - on given events (internal or externals), …
- The produced model corresponds to an (XMI) MM that has been directly read by the program
- Produced models may be concurrently used and combined
- The nature of the produced models may change in time (MM-driven)

Combining the power of meta-modeling and meta-programming

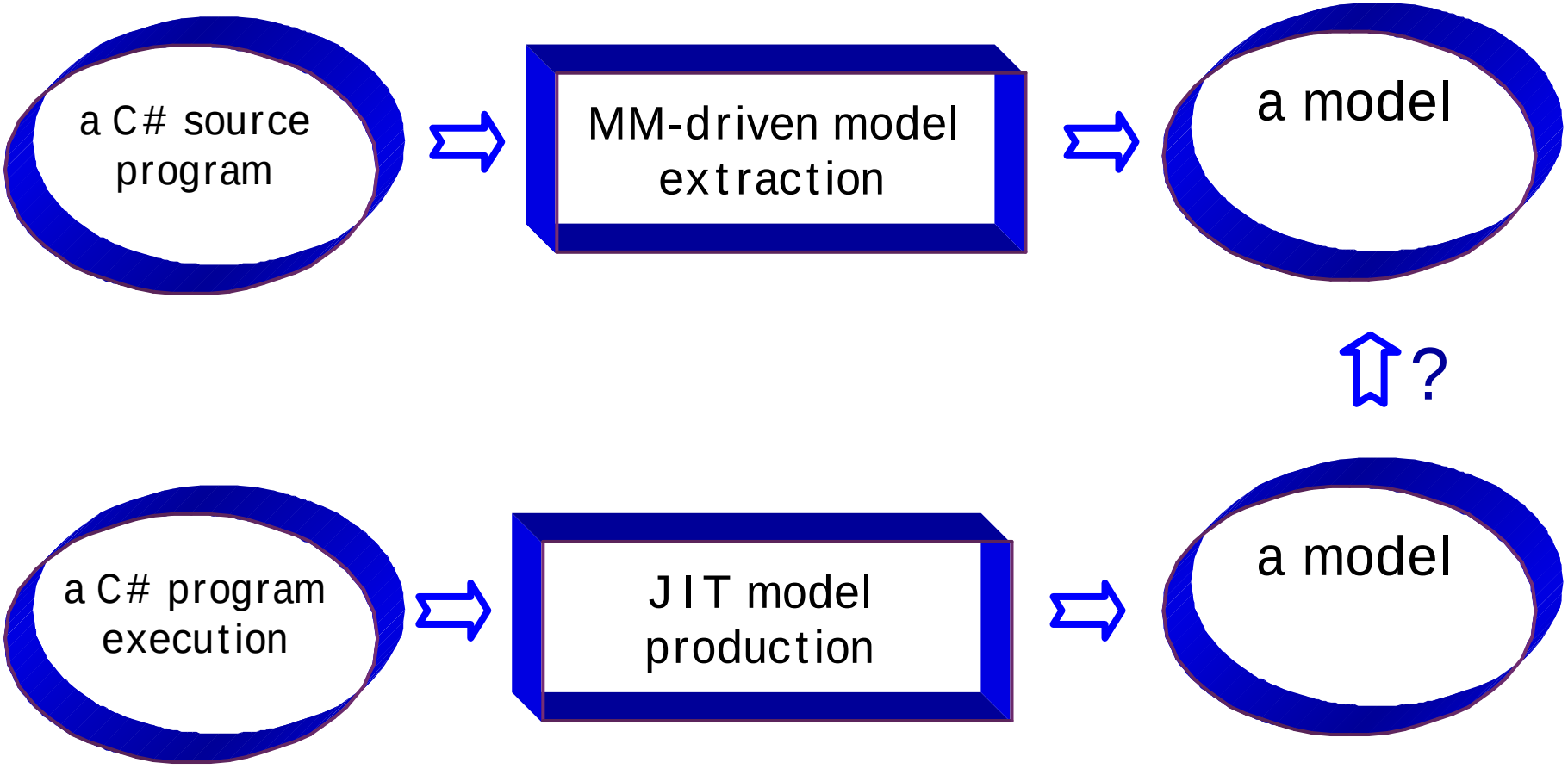


[This may be the basic strategy
for modern software maintenance]

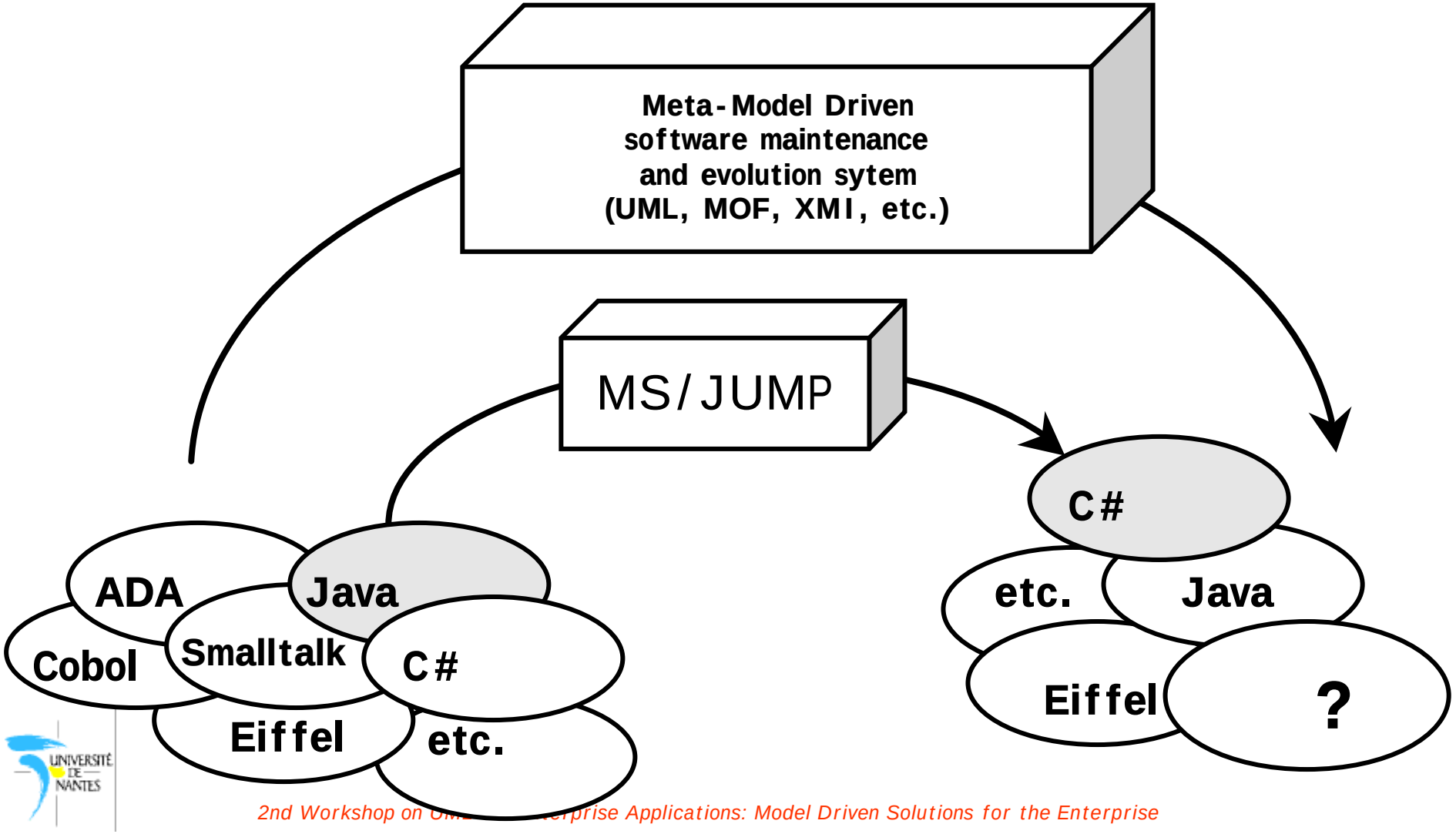
Challenge

- How to associate instructions to different elements of the meta-model that will specify how the model may be extracted
- Hint : these instructions are C# code that uses the introspection API of C#
- This allows defining complex heuristics (e.g. finding patterns, etc.)

Interesting question



Several migration paths



Conclusions

- **Meta-modeling and meta-programming are two orthogonal solutions**
- **They may be combined**
- **JIT/MP is a very powerful technique with tremendous protential**