



Modeling and metamodeling in Model Driven Development

What is a metamodel: the OMG's metamodeling infrastructure

Warsaw, May 14-15th 2009

Gonzalo Génova

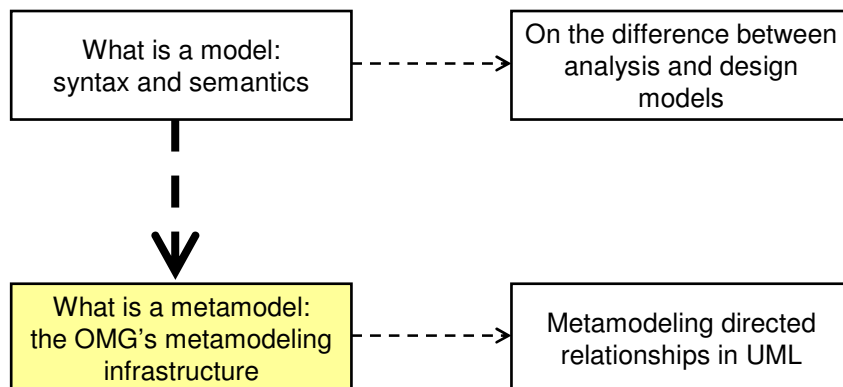
ggenova@inf.uc3m.es

<http://www.kr.inf.uc3m.es/ggenova/>

Knowledge Reuse Group
Universidad Carlos III de Madrid



Structure of the seminar



By the way, what does this diagram mean, what is its syntax?



Sources

- Jean Bézivin
 - **Model Engineering for Software Modernization.**
 - The 11th IEEE Working Conference on Reverse Engineering, Delft, November 8th-12th 2004.
 - **On the unification power of models.**
 - Software and Systems Modeling 4(2): 171–188, May 2005.
- Colin Atkinson, Thomas Kühne
 - **Model-Driven Development: A Metamodeling Foundation.**
 - IEEE Software 20(5): 36-41, Sep-Oct 2003.
 - **Reducing Accidental Complexity in Domain Models.**
 - Software and Systems Modeling 7(3): 345-359, July 2008.
- My own ideas and elaboration.



Table of contents

1. Introduction: definitions of metamodel
2. Representation and conformance
3. The four metamodeling layers
4. Metamodel and semantic domain
5. A case of metamodel/domain mismatch
6. Conclusions



Introduction: definitions of metamodel



What is a metamodel (according to Google definitions)

- If someone still believes there is a commonly accepted definition...
 - Metamodeling, or meta-modeling, is the analysis, construction and development of the **frames, rules, constraints, models and theories** applicable and useful for modeling a predefined class of problems.
 - A meta-model typically defines the **languages and processes** from which to form a model.
 - The meta-model in **neuro-linguistic programming** (or meta-model of therapy) is a heuristic set of questions intended to elaborate and clarify...
 - According to the Meta-Object Facility (MOF) standard, a metamodel is a model that defines the **language for expressing a model**.
 - A model that explains a set of related models.
 - Metamodel is model's model that serves for explanation and definition of relationships among the various components of the applied model itself.
 - **A CIM component** that describes the entities and relationships representing managed objects. For example, classes, instances, and associations are included in the metamodel.
 - **A concept map** showing all the main classes of concepts and relationships between them. Used for setting up a k-base ontology and templates.



What is a metamodel (according to “authorized” UML)

- Unified Modeling Language Superstructure, version 2.2 (2009-02-03), p. 1.
- UML meets the following requirements:
 - A formal definition of a common MOF-based metamodel that **specifies the abstract syntax of the UML**. The abstract syntax defines the set of UML modeling concepts, their attributes and their relationships, as well as the rules for combining these concepts to construct partial or complete UML models.
 - A detailed explanation of the **semantics of each UML modeling concept**. The semantics define, in a technology-independent manner, how the UML concepts are to be realized by computers.
 - A specification of the **human-readable notation** elements for representing the individual UML modeling concepts as well as rules for combining them into a variety of different diagram types corresponding to different aspects of modeled systems.
 - A detailed definition of ways in which UML tools can be made compliant with this specification. This is supported (in a separate specification) with an XML-based specification of corresponding **model interchange formats** (XMI) that must be realized by compliant tools.



What is a metamodel (according to “authorized” MDA)

- MDA Guide Version 1.0.1 (2003-06-01), p. 7-4:
 - In Language specifications the **abstract syntax** of the language is specified as a MOF-compliant metamodel.
- A Proposal for an MDA Foundation Model (2005-04-01), p. 2:
 - A metamodel is a special kind of model that specifies the **abstract syntax** of a modeling language. It can be understood as the representation of the class of all models expressed in that language. Metamodels in the context of MDA are expressed using MOF.
- So, according to OMG standards:
 - **metamodel = abstract syntax**.
 - a data model to store, manipulate and interchange models.
 - the metamodel has very little semantics content (very little meaning).
- However...



What is a metamodel (according to other authors)

- Mellor, Scott, Uhl & Weise. MDA Distilled. Principles of Model-Driven Architecture, 2004.
 - A metamodel is a model of a modeling language. The metamodel defines the **structure**, **semantics** and **constraints** for a family of models.
- Clark, Sammut & Willans. Applied Metamodelling. A Foundation for Language Driven Development, 2008.
 - A metamodel is a model of a language that captures its essential properties and features. These include the **language concepts** it supports, its textual and/or graphical **syntax** and its **semantics** (what the models and programs written in the language mean and how they behave).
- It seems even among experts there is **no agreement...**
- Difference in terminology is an issue if it provokes **confusion**.



Representation and conformance



What is a model (revisited)

A model is **a simplified representation of a certain reality**.

There can be many different maps of the same territory, depending on the **purpose**:

- riding a bike,
- visiting museums,
- analyzing votes,
- etc.

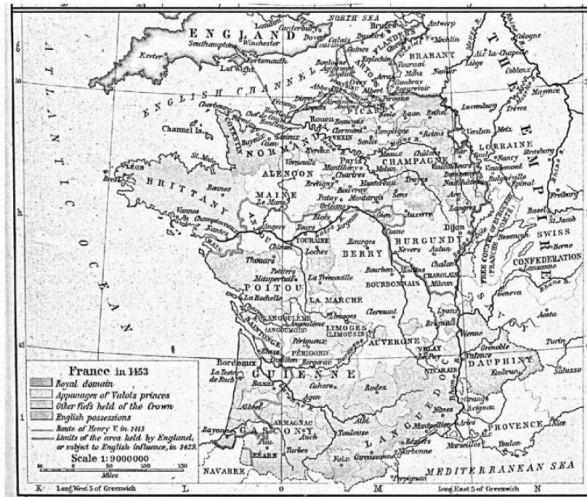
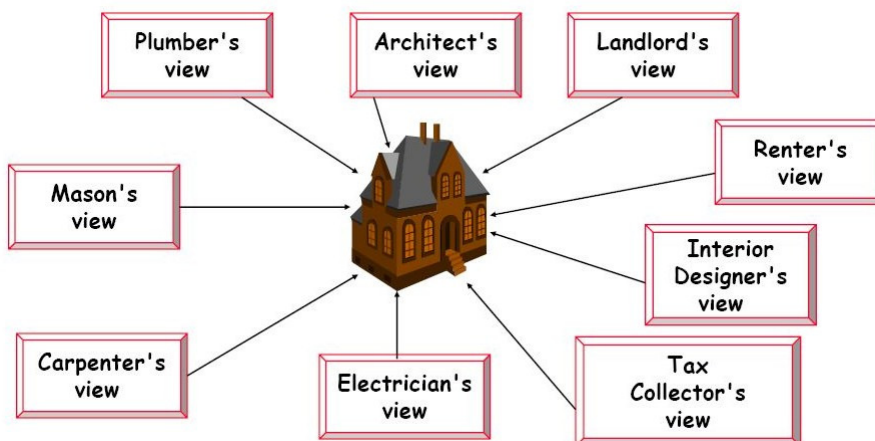


Fig. 3. Map of France in 1453 from the Historical Atlas by R. Shepherd

Jean Bézivin, *On the unification power of models*, 2005.



Each model captures a certain aspect/view of the system



Jean Bézivin, *Model Engineering for Software Modernization*, 2004.



What is modeling

- Rothenberg, The nature of modeling, 1989.
 - Modeling, in the broadest sense, is the cost-effective use of something in place of something else for some cognitive **purpose**. It allows us to use something that is simpler, safer or cheaper than reality instead of reality for some **purpose**. A model represents reality for the given **purpose**; the model is an abstraction of reality in the sense that it cannot represent all aspects of reality. This allows us to deal with the world in a simplified manner, avoiding the complexity, danger and irreversibility of reality.
- A model is focused in one particular **aspect** of a system:
 - A model is not intended to capture all the aspects of a system, but mainly to **abstract out** only some of these characteristics.
 - A system is usually represented by **a set of different models**, each one capturing some specific aspects.
 - Which aspects to capture depends on the **purpose** of the model.
 - A model **must not** represent the system with absolute preciseness.
 - Think of a map at the 1:1 scale!
 - A UML model must not try to capture the totality of aspects of a Java program.
 - A model is useful **because** it is a simplified copy.



The representation relationship



repOf



Characterized by **contextual substitutability**: a model should be able to answer a given set of questions in the same way the system would answer these same questions.

You can ask the map **certain questions**, but not others.

What is the distance from Madrid to Warsaw?

How many whales are there in the oceans?

Jean Bézivin, *Model Engineering for Software Modernization*, 2004.



A model is a partial analogy of a system



Ceci n'est pas une pipe.

The analogy between the model and the represented reality is partial.
The properties of the model are **not identical** to the properties of the reality.
I can't smoke with this pipe!

Jean Bézivin, *Model Engineering for Software Modernization*, 2004.

René Magritte
"This is not a pipe"



The conformance relationship

A model is a simplified representation of a certain reality, **according to the rules of a certain modeling language**.

The map conforms to its **legend**.

The legend defines the **language** to read the map.



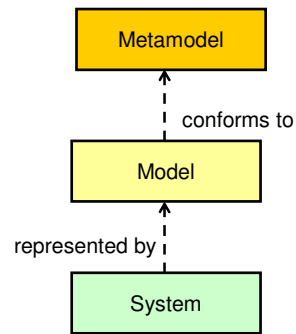
Fig. 3. Map of France in 1453 from the Historical Atlas by R. Shepherd

Jean Bézivin, *On the unification power of models*, 2005.



The two basic metamodeling relationships

- **What is a model?**
 - A model is a simplified representation of a certain reality,
 - according to the rules of a certain modeling language.



A model **represents** a system and **conforms** to a metamodel.



The four metamodeling layers



A metamodel is the model of a language

- Let's recall (from "A Proposal for an MDA Foundation Model"):
 - A metamodel is a special kind of model that specifies the **abstract syntax** of a modeling language. It can be understood as the representation of the **class of all models** expressed in that language. Metamodels in the context of MDA are expressed using MOF.
- What does a metamodel **represent**?
 - Not a model, or a set of models, but the abstract syntax of a modeling language.
 - The expression "model of a model" is particularly confusing.
 - A metamodel is a model of a (modeling) language.
- What does a metamodel **conform to**?
 - To a meta-metamodel, of course.

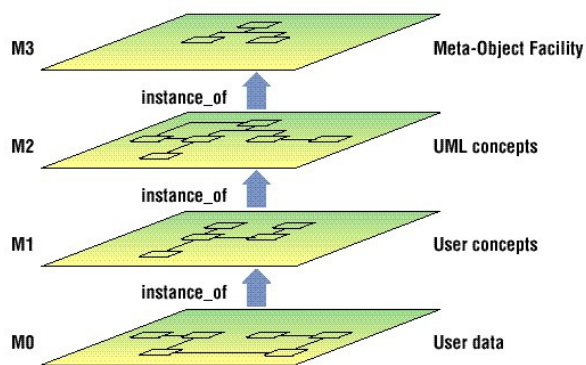


The four metamodeling layers

A metamodel is a "class of models".

Is a model an instance of a metamodel?

Many meanings of the "instance-of" relationship.



Traditional OMG's metamodeling infrastructure.
Are all **instance-of** relationships fundamentally of the same kind?

Colin Atkinson, Thomas Kühne,
Model-Driven Development: A
Metamodeling Foundation, 2003.



Bézivin's version of metalayers: 3+1

The M0 layer is the real **system**.

A **model** represents this system at level M1.

This model conforms to its **metamodel** defined at level M2.

The metamodel itself conforms to the **meta-metamodel** at level M3

The meta-metamodel conforms to itself.

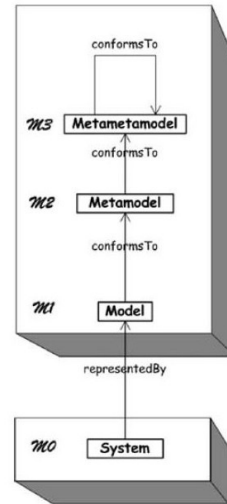


Fig. 7. The 3+1 MDA organisation

Jean Bézivin, *On the unification power of models*, 2005.



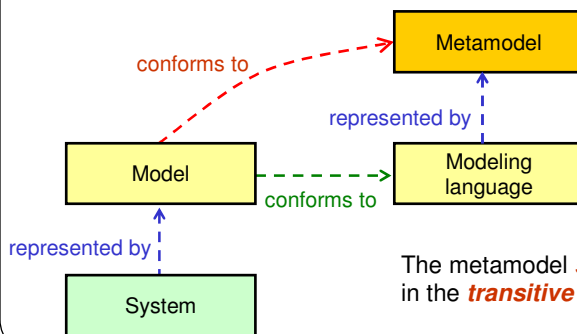
Are they really "layers"?

A **model** conforms to a language whose abstract syntax is represented by a **metamodel**.

Transitively, we can say that a model conforms to a metamodel.

The represented-by and conforms-to relationships are **essentially different**.

Arranging them in the same direction may be confusing.



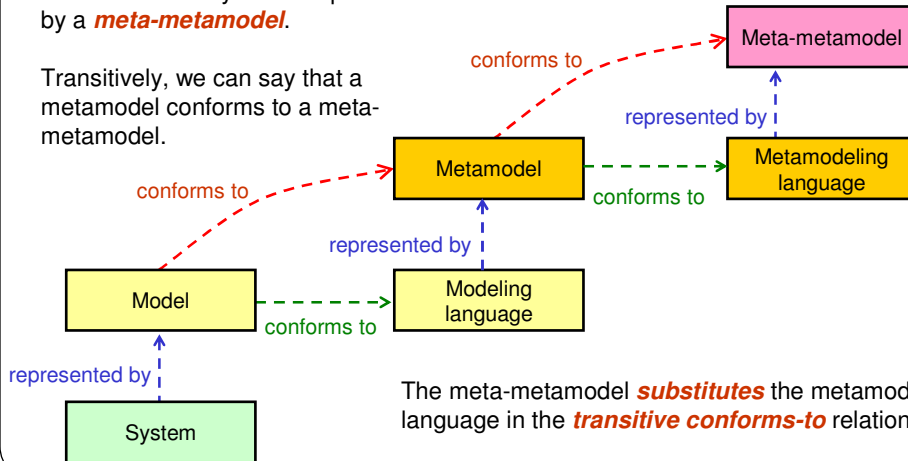
The metamodel **substitutes** the modeling language in the **transitive conforms-to** relationship.



Are they really "layers"?

A **metamodel** conforms to a language whose abstract syntax is represented by a **meta-metamodel**.

Transitively, we can say that a metamodel conforms to a meta-metamodel.



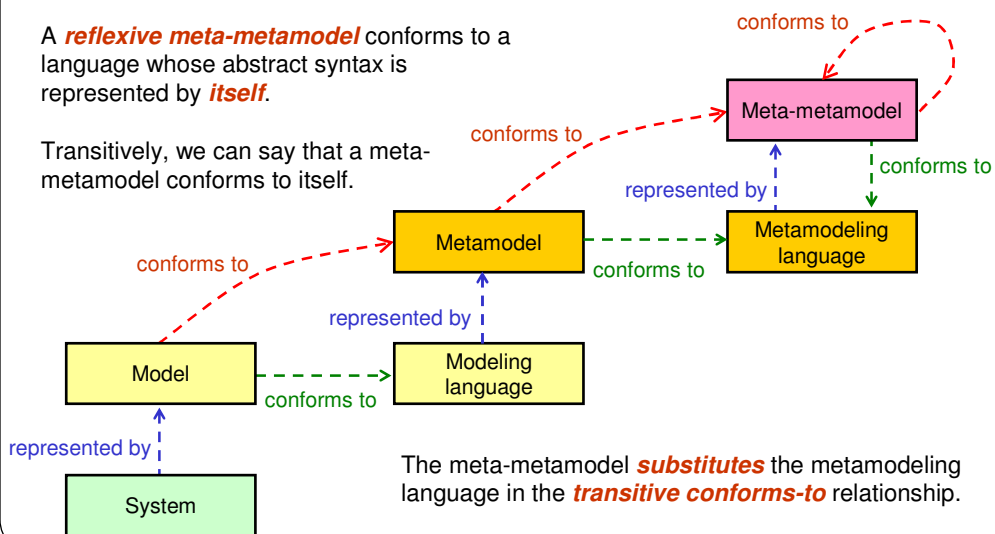
The meta-metamodel **substitutes** the metamodelling language in the **transitive conforms-to** relationship.



Are they really "layers"?

A **reflexive meta-metamodel** conforms to a language whose abstract syntax is represented by **itself**.

Transitively, we can say that a meta-metamodel conforms to itself.



The meta-metamodel **substitutes** the metamodelling language in the **transitive conforms-to** relationship.



Metamodel and semantic domain



Semantic domain

- What is a semantic domain?
 - **Semantic domain = Real world?** Well, not exactly...
 - A semantic domain is **focused on certain entities and aspects** of the “real world” in which we are interested, excluding others.
 - The selection of relevant entities and aspects is driven by our modeling **purpose**.
- The semantic domain is:
 - An abstraction (a representation, **a model**) of the real world.
 - A conceptualization of some part of the “real world”, a closed world of concepts, with rules governing the relationships between those concepts (**a rational world**).
- We, software engineers, don't need to worry about the “true structure of the world”.
 - The correspondence between the concepts and the world beyond is a **philosophical issue** (most interesting, indeed!). Do universal concepts exist?
 - We are concerned only about **the world as we speak about it** (the “universe of discourse”), the information world.



Metamodel and semantic domain

- The semantic domain is a closed world of concepts and relationships.
- The metamodel gives **names** to these concepts and relationships.
 - The metamodel (= abstract syntax) gives **no meaning**, no interpretation.
 - Safe for the meaning **conveyed by the names** (so, be careful with the selection of names).
- Each language defines, or is associated with, a particular semantic domain.
- **What is the domain of Natural Language?**
 - The entire world: we can speak about everything.
 - Or, better, there is a domain for each language (Polish, Spanish, Japanese...).
 - Each language conceptualizes the world in its own way (Weltanschauung).
- The semantic domain is defined in the metamodel, **not in the model**.
 - Each model uses the language of its particular semantic domain / metamodel.
 - Each model receives its meaning from the **semantic mapping** MM → SD.
- **What about domain models?**
 - Either they are metamodels, or they do not properly define a new domain.



The metamodel / semantic domain of UML

- What is the **metamodel** of UML?
 - Read the Standard.
- What is the **semantic domain** of UML?
 - Again, read the Standard, but this time it is not so easy to answer.
 - Informally described in (rather inconnected) natural language discussions.
 - More or less, **the world of object orientation**: classes, objects, messages...
 - It does not correspond to any particular programming language.
 - In fact, it has defined **a new object-oriented paradigm**.
 - UML does not specify clearly and rigorously its semantic domain.
 - Too many “semantic variation points”.
 - Trying to accommodate everybody, it accommodates nobody.
- Source for the proliferation of **Domain Specific Languages**.
 - Each DSL has its own metamodel and semantic domain.
 - Interoperability is easier between MOF-compliant DSLs.



Metamodel / semantic domain and the real world

- To be useful, the concepts and rules defined in a modeling language must adequately **correspond with reality**.
 - Why is **Newtonian physics** so useful?
 - Do **personal debts** behave in the RW as we have conceptualized them?
- A bad conceptualization of the “real world” in a semantic domain (and metamodel) has several drawbacks:
 - It can produce models that are **useless** for predictions.
 - It can produce models that are **twisted representations** of reality.
- **Everything is an object...**
 - does not express a property of the “real world”,
 - but a property (a limitation?) of the domain modeler.
- For a person who holds a hammer, **everything is a nail...**

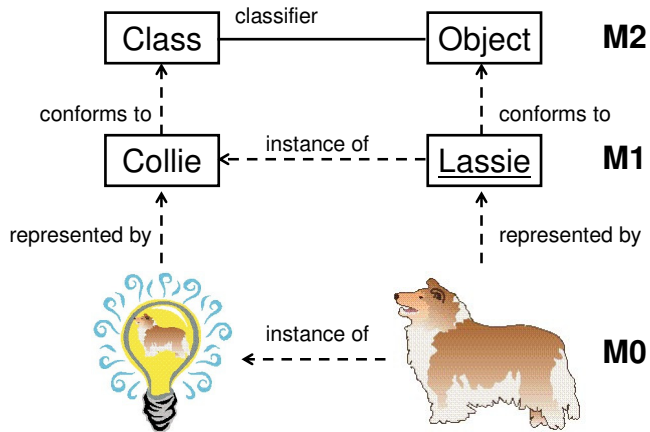


A case of metamodel/domain mismatch



The “two level” modeling paradigm

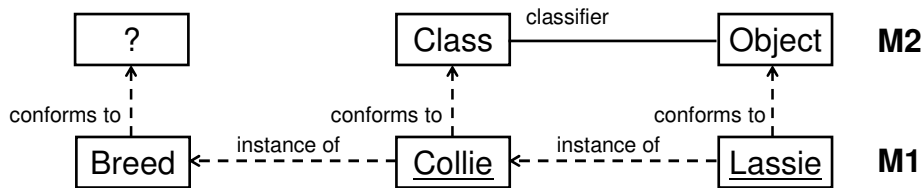
The UML metamodel (following many OOPs) consecrates the principle that *the universe of discourse consists of classes and objects* (types and instances).



Adapted from: Colin Atkinson, Thomas Kühne, Model-Driven Development: A Metamodeling Foundation, 2003.



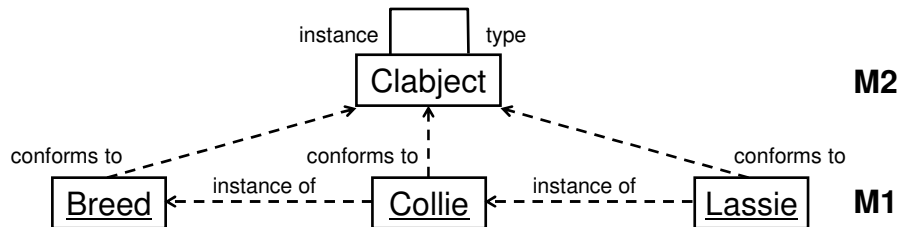
Instance and type at the same time



- How can I represent that Collie is an instance of Breed?
 - In the “real world”, an entity can be *an instance and a type at the same time*.
- **Twisted solutions:**
 - Stereotype Collie as «Breed».
 - Use the Type-Object pattern.
 - Extend the metamodel with new metaclasses.



Breaking the “two level” modeling paradigm



- Allows **an arbitrary long chain** of instance-of relationships at M0.
- More **difficult to translate** into traditional OOPs.
- Some solutions:
 - Powertypes.
 - Deep instantiation (see details in Atkinson & Kühne).
- UML allows other constructs difficult to implement (**multiple generalization**).



Conclusions



Conclusions

- **Metamodel = abstract syntax** (with very little semantics content).
- Two basic metamodeling relationships:
 - A model **represents** a system and **conforms to** a metamodel.
 - A metamodel is a special kind of model that represents the abstract syntax of a modeling language and conforms to a meta-metamodel.
- The represented-by and conforms-to relationships are essentially different.
 - Arranging them in the same direction may be confusing.
 - Instead of a **stack of layers**, a **stair of layers** may be more clarifying.
- A metamodel is closely related to its semantic domain.
 - A semantic domain is a conceptualization of some part of the “real world”.
 - A metamodel **gives names** to the concepts of the semantic domain.
 - But **the meaning of names** is properly out of the metamodel.
 - A bad conceptualization of the RW is the source of **modeling problems**.
- Example: **the “two-level” modeling paradigm**.



Questions?

