IRISA

Model-Driven Engineering:

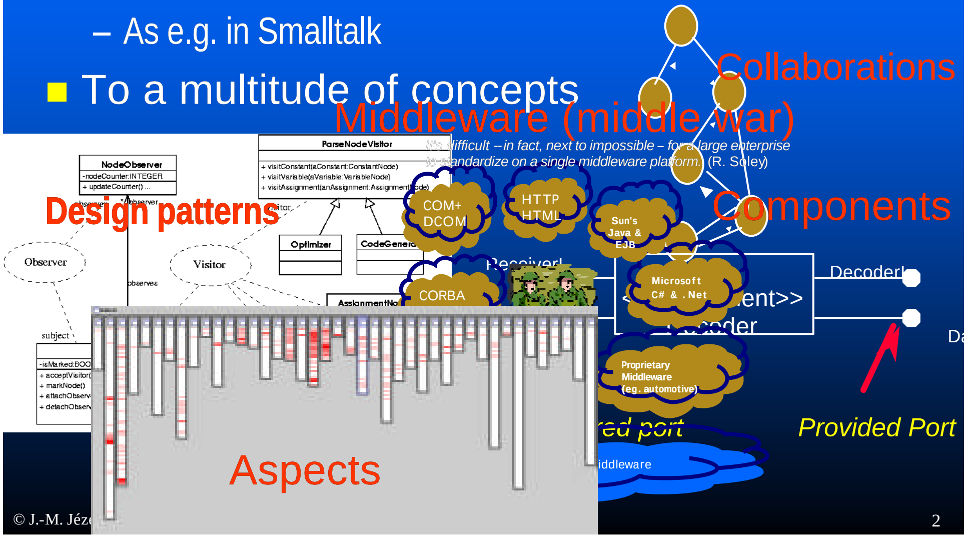
Core Principles and Challenges*

Prof. Jean-Marc Jézéquel
(Univ. Rennes 1 & INRIA)
Triskell Team @ IRISA
Campus de Beaulieu
F-35042 Rennes Cedex
Tel : +33 299 847 192 Fax : +33 299 847 171
e-mail : jezequel@irisa.fr
http://www.irisa.fr/prive/jezequel

**Joint work with J. Bézivin (INRIA/Atlas-Univ. Nantes) and B. Rumpe (Irisa & TUM)*

Once upon a time… software development looked simple

- From the object as the *only* one concept
  - As e.g. in Smalltalk
- To a multitude of concepts



It's difficult --in fact, next to impossible – for a large enterprise to standardize on a single middleware platform. (R. Soley)

Why modeling: master complexity

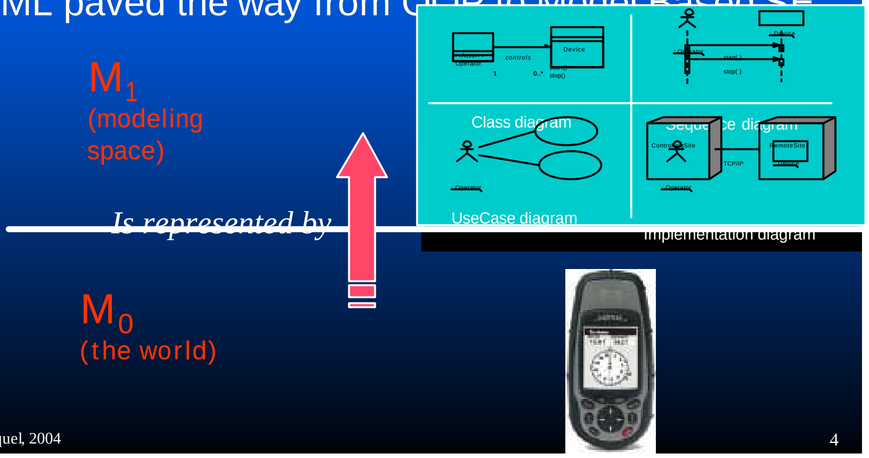
- Modeling, in the broadest sense, is the *cost-effective use of something in place of something else for some cognitive purpose*. It allows us to use something that is *simpler*, *safer* or *cheaper* than reality instead of reality for some purpose.
- A model represents reality for the given purpose; 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.

Jeff Rothenberg. **"The Nature of Modeling." in Artificial Intelligence, Simulation, and Modeling, L.E. William, K.A. Loparo, N.R. Nelson, eds. New York, John Wiley and Sons, Inc., 1989, pp. 75-92**

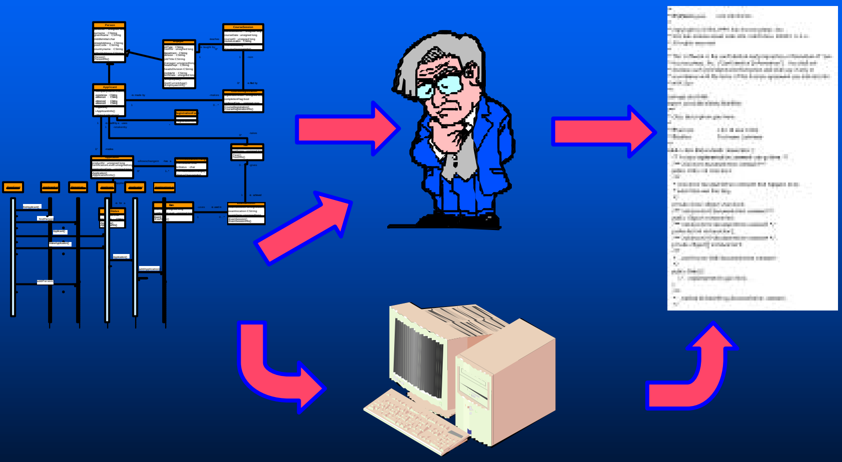
http://poweredge.stanford.edu/BioinformaticsArchive/PrimarySite/NIHpanelModeling/RothenbergNatureModeling.pdf

The World and the Model

- A Model is a *simplified* representation of an *aspect* of the World for a specific *purpose*
  - Consider modeling both the machine & its environment (M. Jackson)
- UML paved the way from OOP to Model Based SE



Models: from contemplative to productive



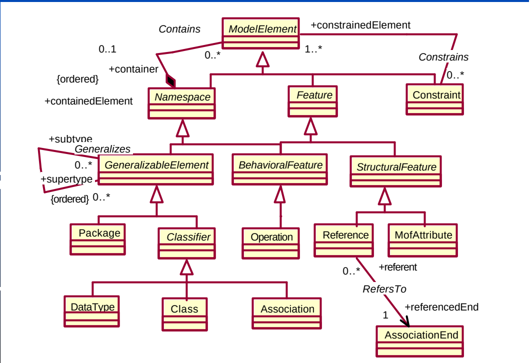
"from human-readable to computer-understandable"

© J.-M. Jézéquel, 2004

From J. Bézivin

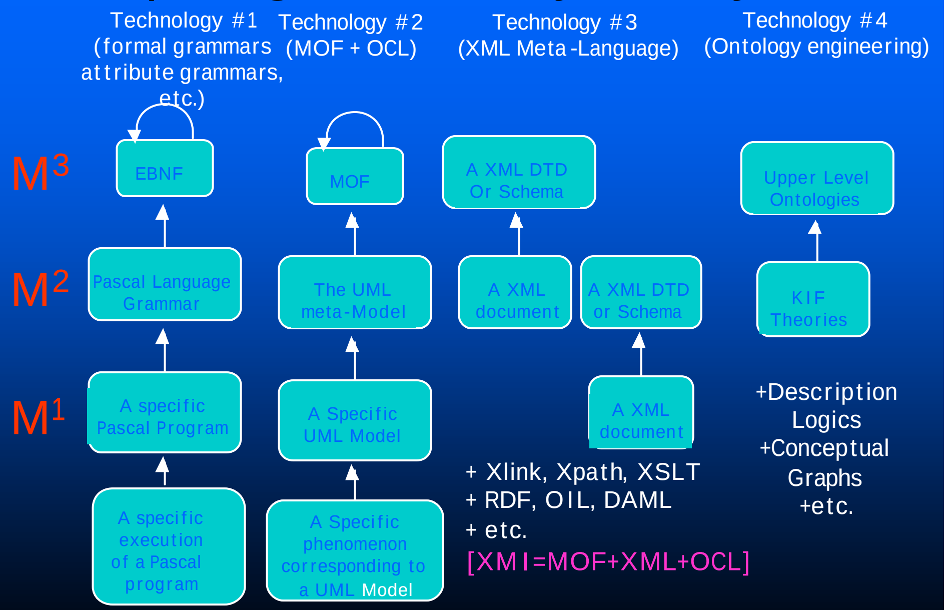
Assigning Meaning(s) to Models

- If a UML model *is no longer* just
  - – fancy pictures to decorate your room
  - – a graphical syntax for C++/Java/C#/Eiffel...
- Then tools must be able to manipulate models
  - – Let's make a model of what a model is!
  - - semantic variation points
  - – => ***meta-modeling***
    - » & meta-meta-modeling..



© J.-M. Jézéquel, 2004

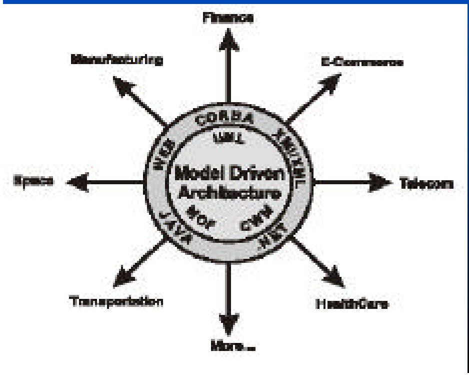
Comparing Abstract Syntax Systems



(From J. Bézivin)

MDA: the OMG new vision

"OMG is in the ideal position to provide the model-based standards that are necessary to extend integration beyond the middleware approach… Now is the time to put this plan into effect. Now is the time for the Model Driven Architecture."

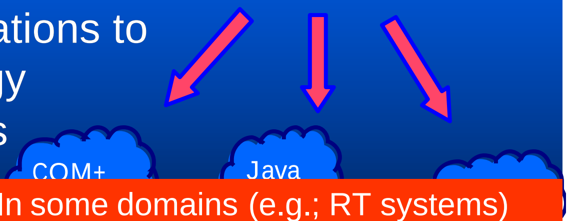


Richard Soley & OMG staff,
MDA Whitepaper Draft 3.2
November 27, 2000

Mappings to multiple and evolving platforms

- Organization assets expressed as models
  - MOF & UML as the core

Platform neutral models based on UML & MOF



- Model transformations to map to technology specific platforms
  - "Just" *weave* the platform aspect !

In some domains (e.g.; RT systems) transformations can get more complex than initial model!
=> Many organizations already have accumulated huge piles of LOC of transformations

Why complex transformations?

- Example: Air Traffic Management
  - “business model” quite stable & not that complex
- Various modeling languages used beyond UML
  - As many points of views as stakeholders
- Deliver software for (many) variants of a platform
  - Heterogeneity is the rule
- Reuse technical solutions across large product lines (e.g. fault tolerance, security…)
- Customize generic transformations
- Compose reusable transformations
- Evolve & maintain transformations for 15+ years!

The 3 ages of Transformations

- Awk-like (inc. *sed, perl*…)

```
BEGIN {action)
pattern #1 {action #1}
…
pattern #n {action #n}
END {action)
```

SE Limit: ~100 LOC

- XSLT
  - W3C standard for transforming XML
  - Operates on tree structures
  - syntactical & inefficient

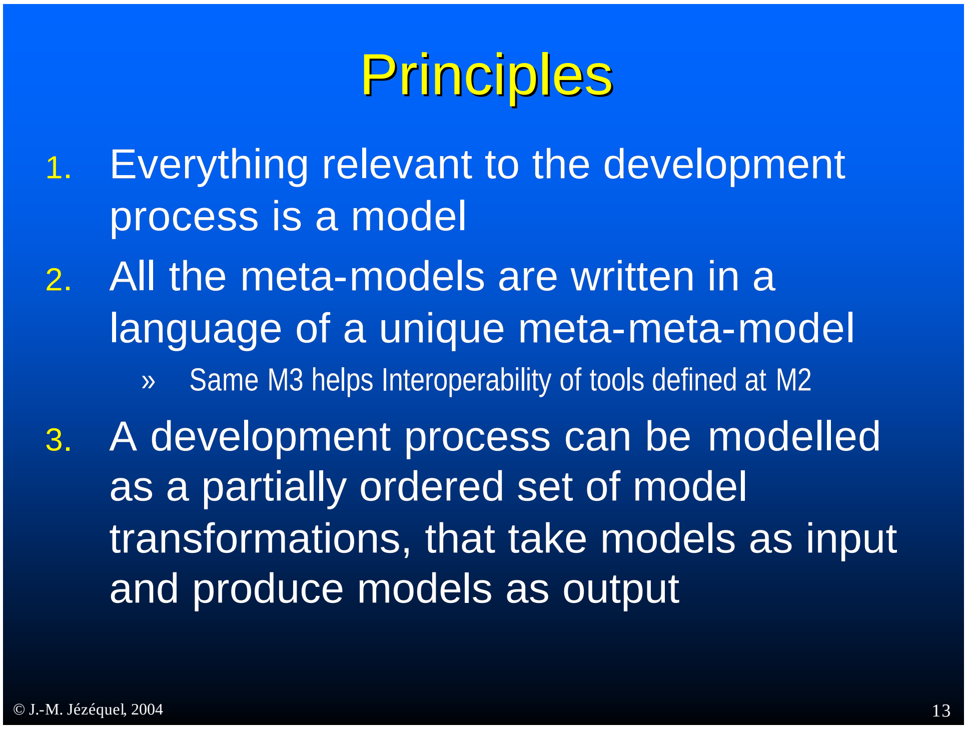
SE Limit: ~1000 LOC

- QVT-like
  - Now hot topic at OMG with RFP Q/V/T
    - » Query/View/Transformation
  - Operates on graphs

SE Limit: ?
Standard?
Which/When?

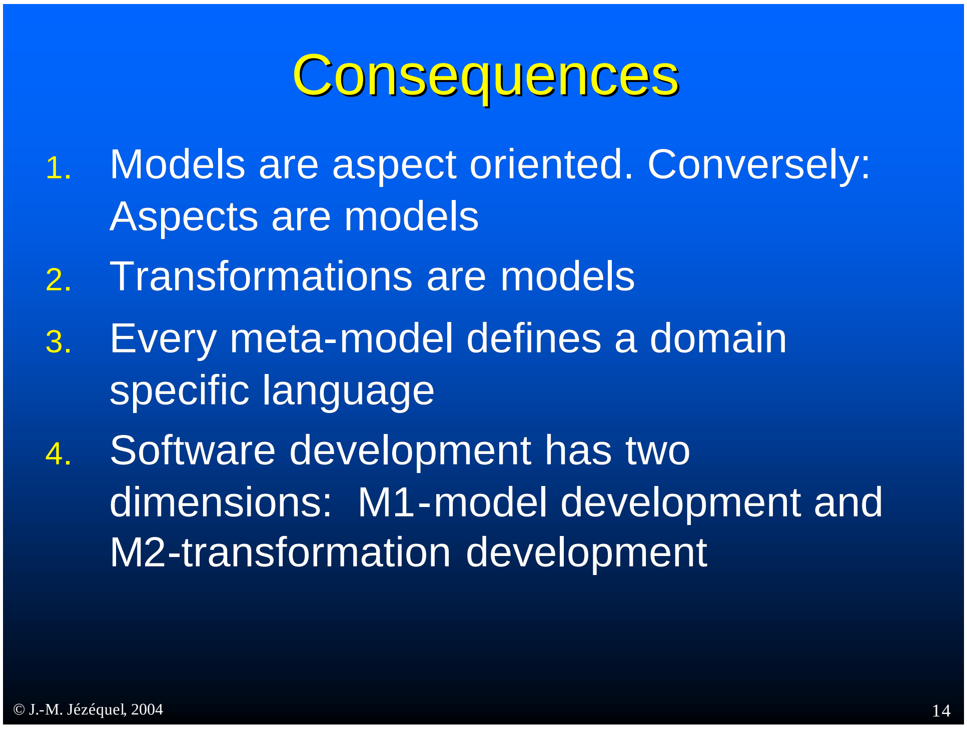
Transformations are Assets => apply sound SE principles

- Must be Modeled
  - with the UML, using the power of OO
- Must be Designed
  - Design by Contract, using OCL
- Must be Implemented
  - Made available through libraries of components, frameworks…
- Must be Tested
  - test cases
    - » input: a UML Model
    - » output: a UML Model, + contract checking
- Must be Evolved
  - Items of Configuration Management
  - Transformations of transformations



Principles

1. Everything relevant to the development process is a model
2. All the meta-models are written in a language of a unique meta-meta-model
   - » Same M3 helps Interoperability of tools defined at M2
3. A development process can be modelled as a partially ordered set of model transformations, that take models as input and produce models as output



Consequences

1. Models are aspect oriented. Conversely: Aspects are models
2. Transformations are models
3. Every meta-model defines a domain specific language
4. Software development has two dimensions: M1-model development and M2-transformation development

Challenges

- Language definition problems (Q/V/T)
  - Expressive, easy to use language(s) for transformations
- Technological Issues
  - Tool set, interoperability…
- Software Engineering Issues
  - From requirements to tests and SCM
- Leverage Formal Methods to go from one model to another one:
  - A transformation is a *mapping* between semantic domains
  - Beyond Code Generation: Proof, Performance Evaluation, Test Synthesis …

References

- Triskell Project
  - www.irisa.fr/triskell
- Modelware @ INRIA (Triskell, Atlas, Jacquard)
  - modelware.inria.fr
- CARROLL (Inria/CEA/Thalès)
  - www.carroll-research.org
- OFTA working group on MDA
  - Book in preparation, due in May, collection Arago
- Action Spécifique CNRS MDA
  - www-adele.imag.fr/mda/as
- OMG
  - www.omg.org