SEN9110 Simulation Masterclass
Lecture 11: Multi-Paradigm Simulation

Prof.dr.ir Alexander Verbraeck
a.verbraeck@tudelft.nl

Brightspace: SEN9110

With slides from Deniz Cetinkaya, Mamadou D. Seck

1.

Papers: Vangheluwe & De Lara

Multi-formalism modeling

Read:

H. Vangheluwe, J. de de Lara, P.J. Mosterman. An introduction to multi-paradigm modelling and simulation. In: F. Barros and N. Giambiasi (eds.), Proceedings of the AIS'2002 Conference (AI, Simulation and Planning in High Autonomy Systems), Lisboa, Portugal, 2002a. pp. 9-20.

H. Vangheluwe, H. and J. de de Lara. "Meta-models are models too", In: E. Yücesan, C.-H. Chen, J.L. Snowdon, and J.M. Charnes (eds). Proceedings of the 2002 Winter Simulation Conference. New York: ACM Press. 2002b. pp. 597-605.

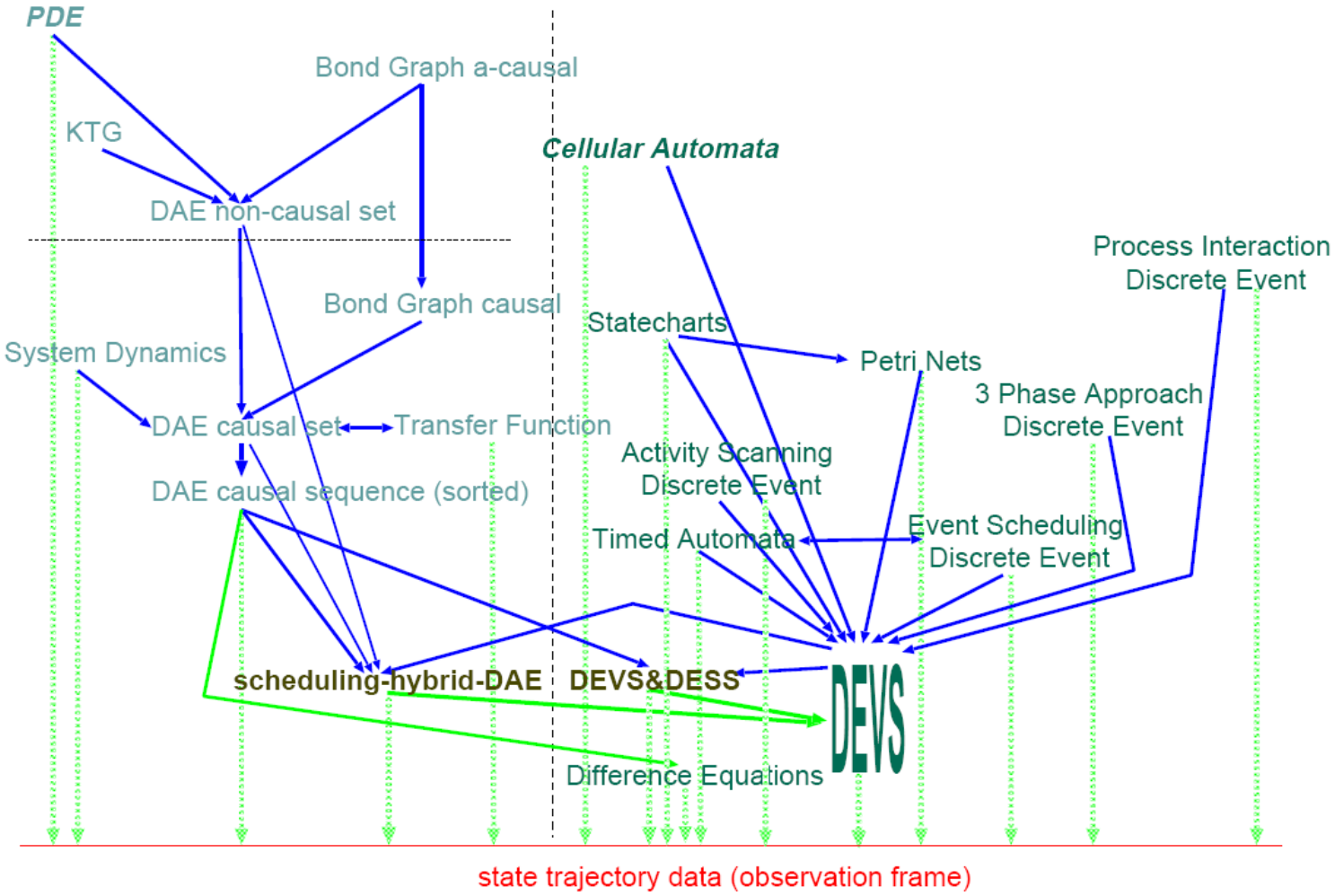
Multi-paradigm modeling
[Vangheluwe et al., 2002a]

Topics:

- **multi-formalism modeling**, concerned with the coupling of and transformation between models described in different formalisms
- **model abstraction**, concerned with the relationship between models at different levels of abstraction
- **meta-modeling**, concerned with the description (models of models) of classes of models, which allows formalism specification
- **model-transformation**, concerned with transforming a model from one formalism into another

Formalism transformation graph

Source: Vangheluwe & De Lara, 2002b, p.599



Meta-modeling levels

Source: Vangheluwe & De Lara, 2002b, p.599

Level	Description	Example
Meta-Meta-Model	Model used to specify modelling languages	Entity-Relationship Diagrams, UML class Diagrams.
Meta-Model	Model used to specify simulation models	Finite State Automata, Ordinary differential equations (ODE).
Model	The description of an object in a certain formalism	$f'(x) = -\sin x$, $f(0) = 0$ (in the ODE formalism)

Morphisms

- **(Homo)morphism**: A transformation of one set into another that preserves in the second set the operations between the members of the first set (but not necessarily the other way around)

- **Isomorphism**: A one-to-one correspondence between the elements of two sets such that the result of an operation on elements of one set corresponds to the result of the analogous operation on their images in the other set (e.g. Arabic and Roman numerals: sum and product give the same result in both)

Combining multiple formalisms

For example:

- Combined DEVS/DESS (or DES/DESS) models
- Combines DEVS/Agent-based (or DES/ABM) models

- Think about issues
  - time management
  - state management

- Relation with distributed simulation

- Hierarchical DEVS can combine different types of model components without any problem!

2.

What is Metamodeling?

History

- The word and the concept of metamodel has been used in the scientific literature for a long time
- The concept of developing meta-constructs for modeling process can be traced back to the 1970s
- However, metamodeling became popular with the introduction of 4-layer metamodeling architecture in UML specification by OMG in 1999

Epistemological Analysis

meta – model(ing)

a prefix which means 'about' ('beyond', 'above' or 'of') its own category
used to refer to a higher abstraction level

a model is a representation of a system (specified in a modeling language)
modeling is the process of developing and specifying a model

A metamodel is a model *of* a model

Metamodeling is modeling *of* a model

?

By the way…

- **Epistemology: What is knowledge?**
  From Greek ἐπιστήμη - epistēmē, meaning "knowledge, understanding", and λόγος - logos, meaning "study of". What is knowledge and how can it be acquired.

- **Ontology: How do we structure reality?**
  From Greek ὄν - on (genetive ὄντος, ontos), i.e. "being; that which is". What entities exist or can be said to exist, how such entities can be grouped, related within a hierarchy, and subdivided according to similarities and differences.

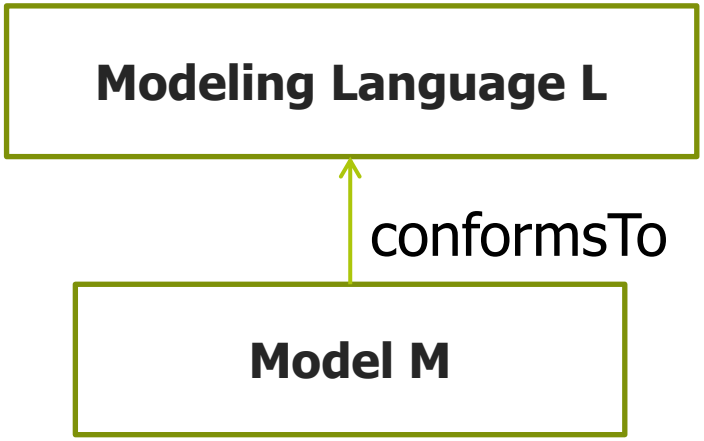
- **Methodology: How do we create knowledge?**
  The systematic, theoretical analysis of the methods applied to a field of study, or the theoretical analysis of the body of methods and principles associated with a branch of knowledge.

What is Metamodeling?

- Metamodeling is "the process of developing a metamodel" (a model) of "another model"
- A metamodel is generally a model that defines the concepts, relations, rules and constraints (or the language) for specifying a model
- Hence, metamodeling is the most commonly used method to describe **modeling languages** in a formal way

Modeling Languages

- A modeling language is a means of expressing models in a precise way by using diagrams, rules, symbols, signs, letters, numerals, etc.
- The relation between a modeling language and a model expressed in that language is called the 'conformsTo' relation, such as:
  - the model M conformsTo
  - the modeling language L.



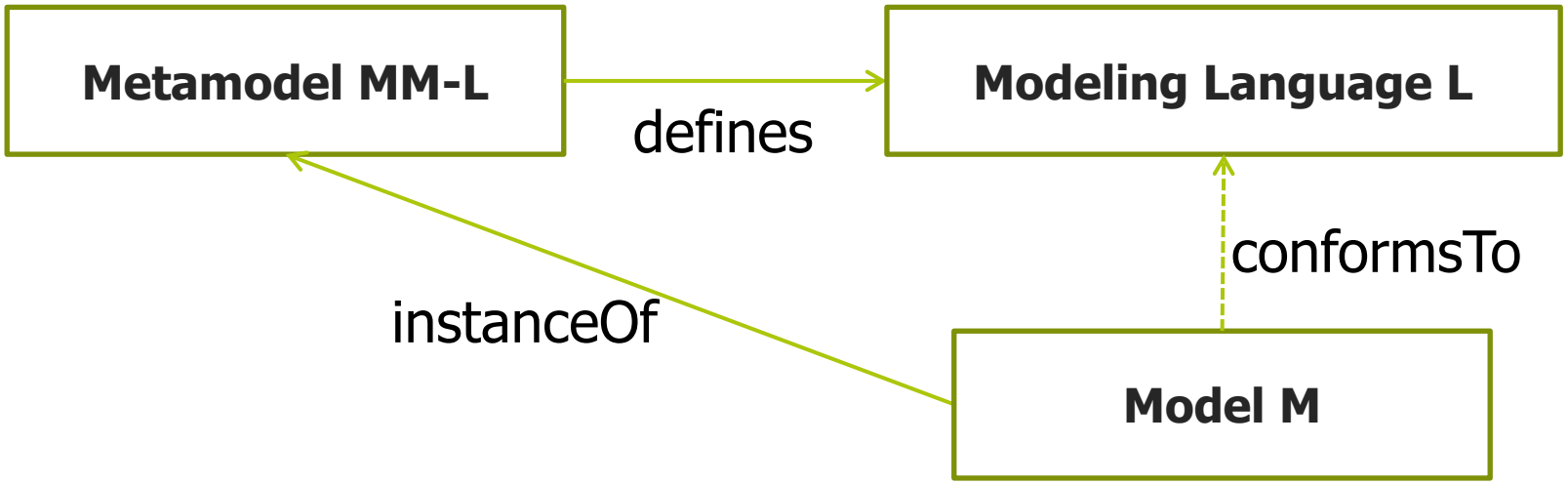
Modeling Languages

- A modeling language consists of the
  - **abstract syntax**: describes the vocabulary of the concepts and the grammar of the language,
  - **concrete syntax**: describes the representation method either in a diagram form (visual syntax) or in a structured textual form (textual syntax),
  - and **semantics**: explains what the abstract syntax actually means

- The abstract syntax consists of the concepts, the relationships and well-formedness rules, where
  - **well-formedness** rules state how the concepts may be combined

Metamodels

- In a metamodeling approach, a metamodel is a definition of a modeling language in the form of a model
- The 'conformsTo' relation is formalized via '**instanceOf**' and '**defines**' relations



Loose vs. Strict Metamodeling

- While the 'conformsTo' relation only guarantees a valid model, the 'instanceOf' relation requires that a model must be an instance of the metamodel. Without any constraints this is known as **loose** metamodeling.

- However, in practice the 'instanceOf' relation requires that every element of a model must be an instance of some element in the metamodel. This type of metamodeling is known as **strict** metamodeling

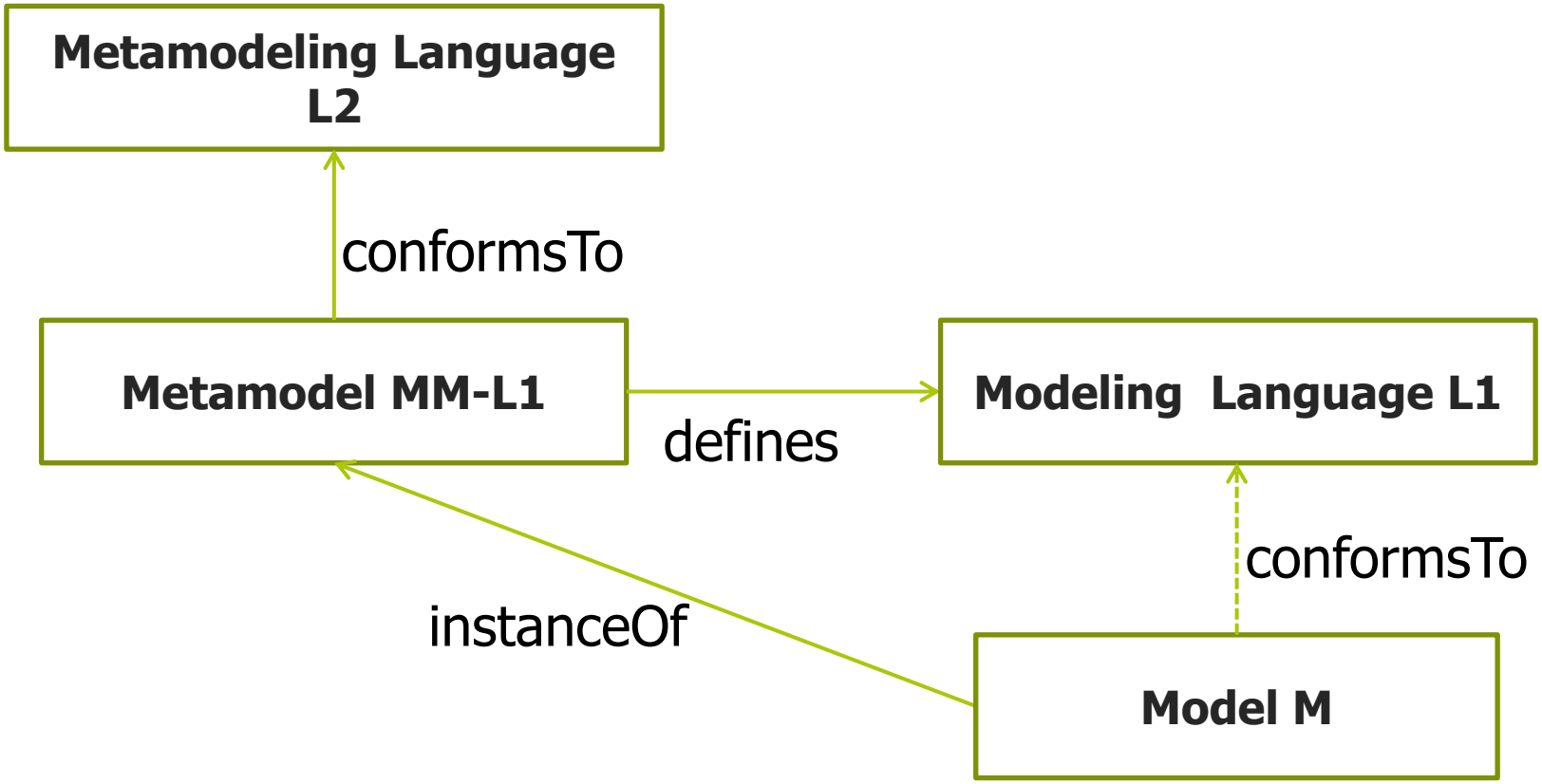
3.

Metamodeling Languages vs. Meta-metamodels

Metamodeling Languages

- By definition, a metamodel is itself a model and has to be defined in a modeling language
- The higher level modeling language that is used to specify metamodels is called metamodeling language
- In other words, a metamodeling language is a higher-level language to describe new modeling languages
- Popular metamodeling languages are: MOF, Ecore, KM3 and MetaGME

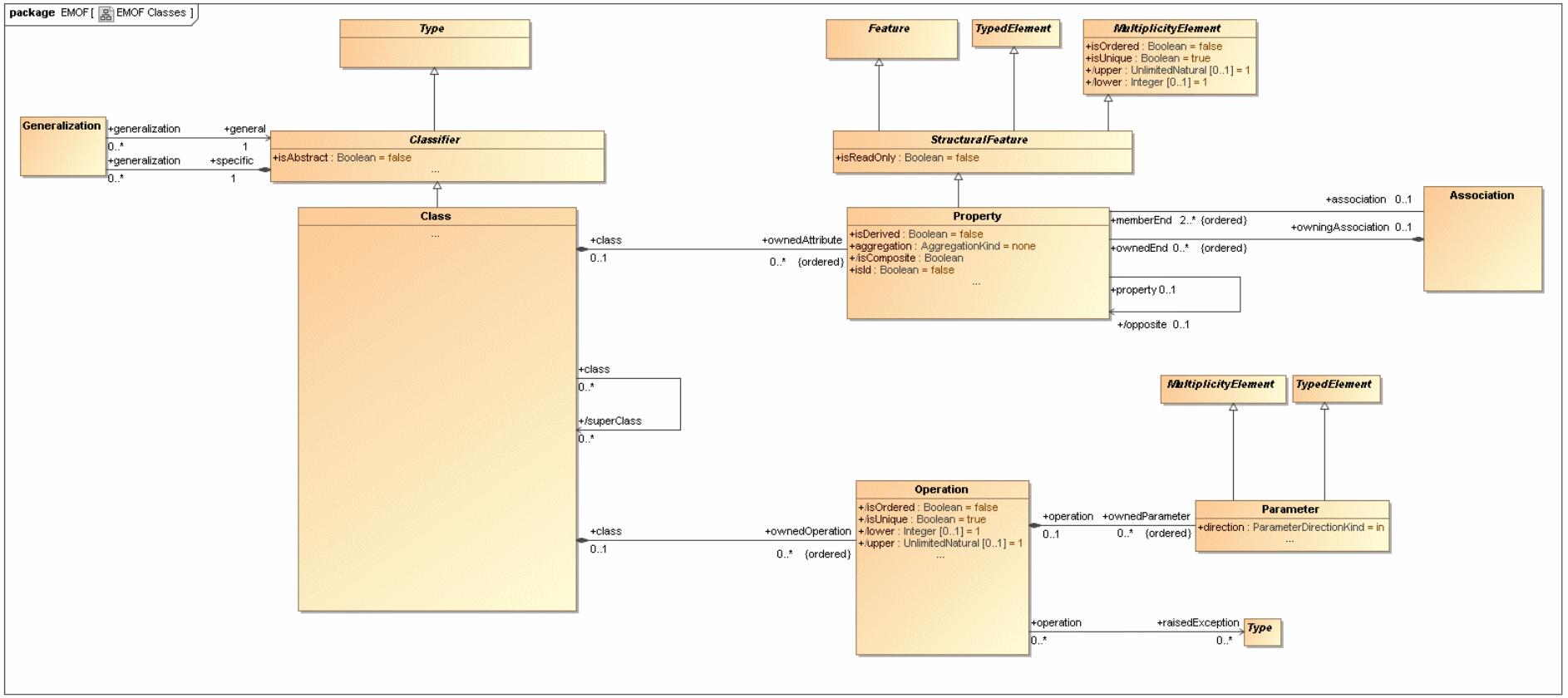
Metamodeling Languages



Meta-metamodels

- The metamodeling language can also be defined with a metamodel
- The metamodel of the metamodeling language is called a meta-metamodel
- To avoid an infinite stack in the number of metamodeling levels, meta-metamodels are often designed to be **self-reflexive** via loose metamodeling (example: MOF)

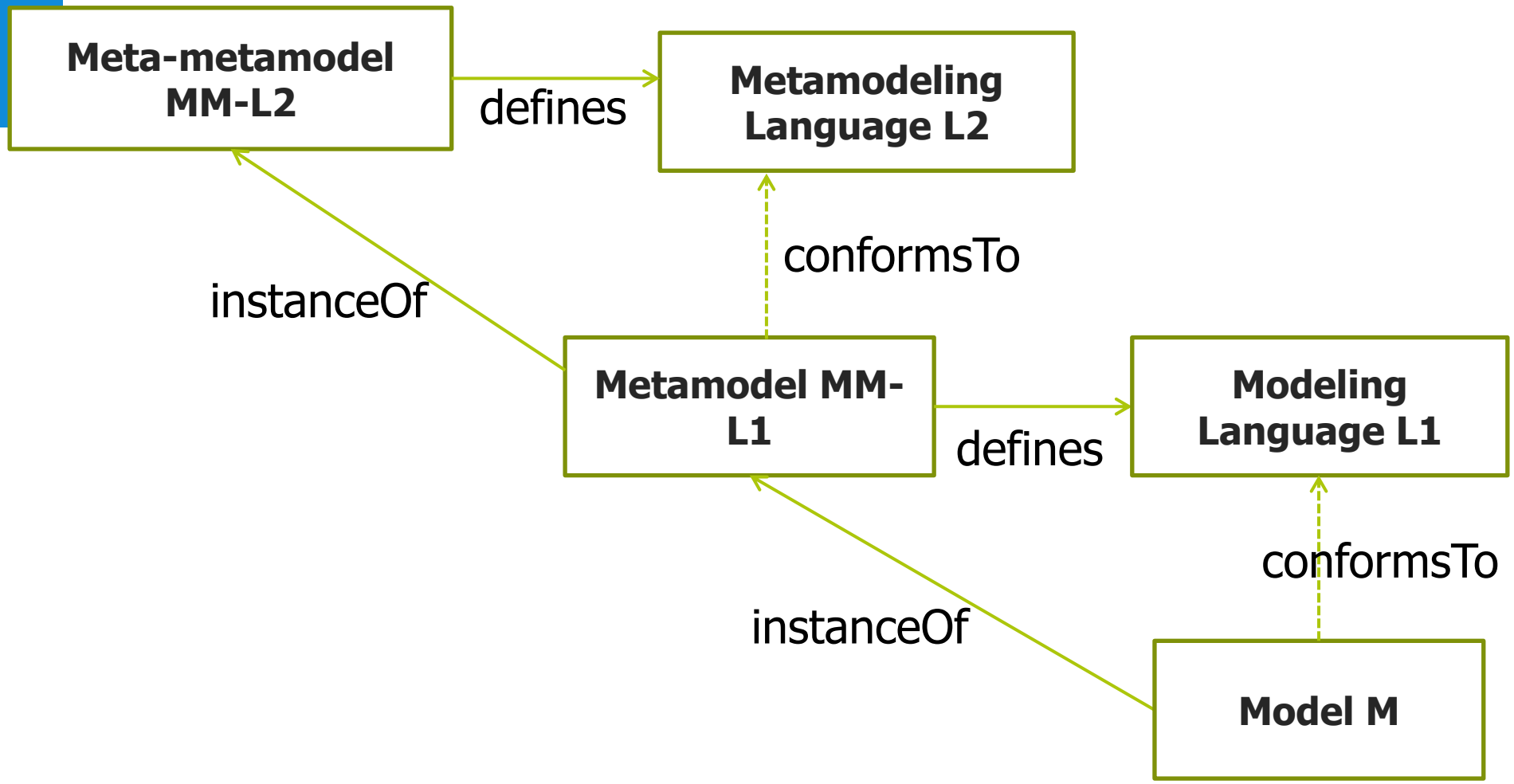
Example MOF diagram



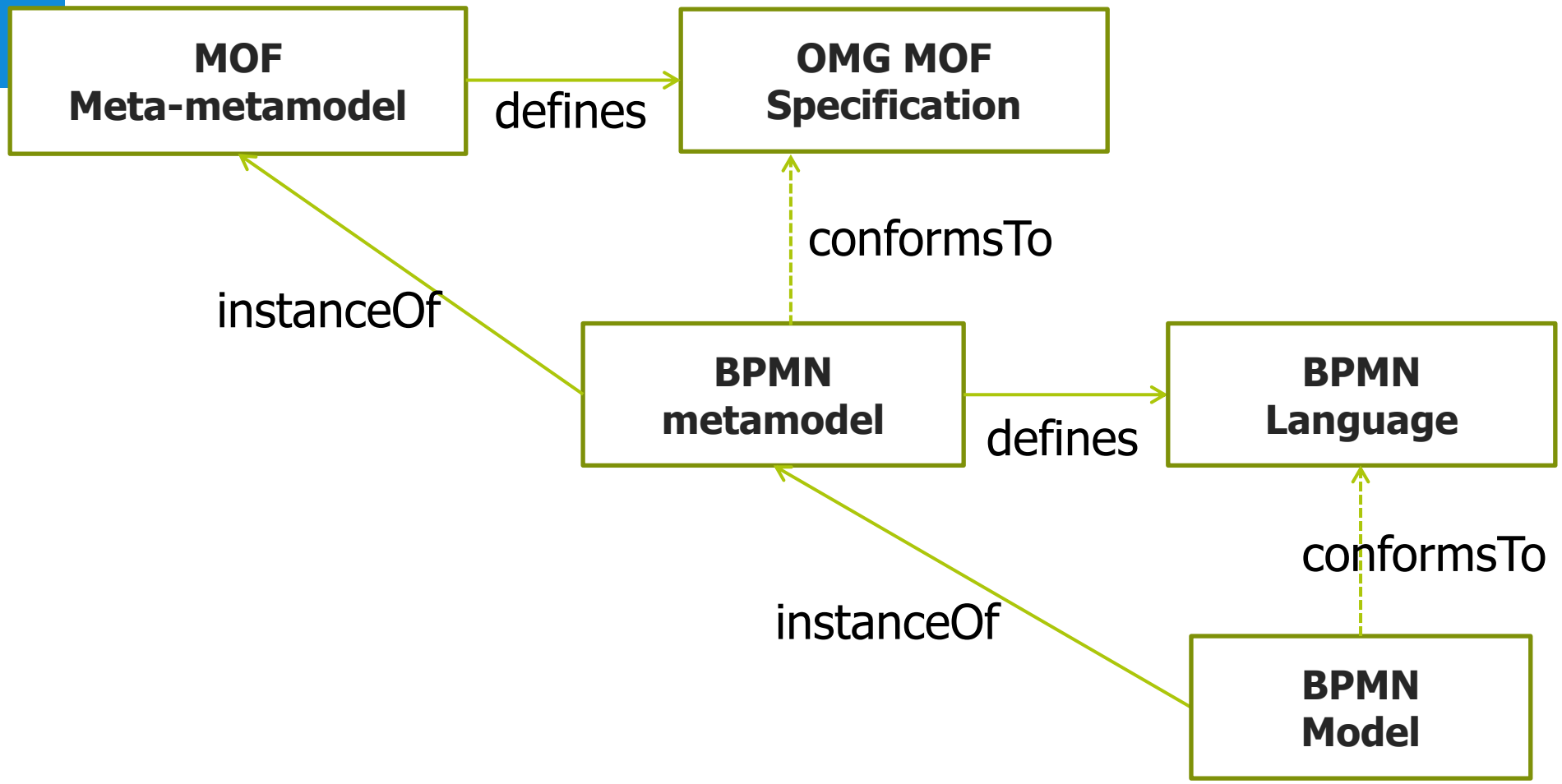
"Recursive" description

OMG Meta Object Facility (MOF) Core Specification, v2.4.1, http://www.omg.org/spec/MOF/2.4.1

Meta-metamodels



Meta-metamodels: example



Four-Level Metamodeling Architecture

Levels	Description	Example
M3 level: meta-metamodel	Defines the language for specifying metamodels.	**MetaClass** and **MetaAttribute** in the MOF Specification
M2 level: metamodel	Defines the language for specifying models. An instance of the meta-metamodel.	**Class** and **Attribute** in the UML Specification
M1 level: model	Defines the model without user data. An instance of the metamodel.	**Car** Class and **Name** Attribute in a UML Model
M0 level: user data	Defines a specific user model with user data. An instance of the model.	**Car.3** Instance with **Name="abc"** in a specific user model

[Ref: OMG, "UML Specification Version 1.3", 1999]

4.

Metamodel-based Model Transformations

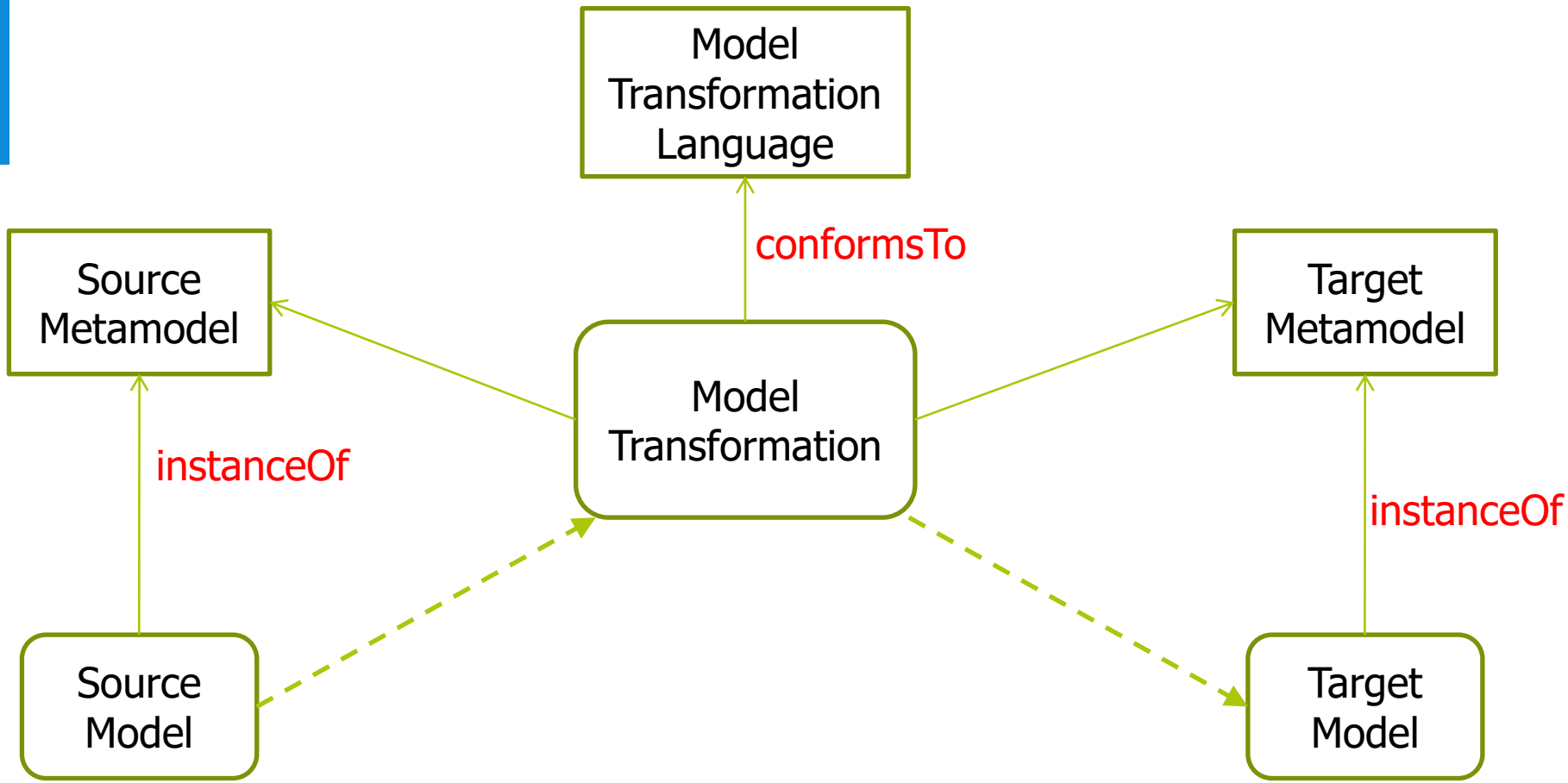
Model Transformations

- Once the models are constructed conforming to their specific metamodels, model transformations can be performed to generate new models based on the existing information in the models
- A model transformation is the process of converting a source model into a target model according to a set of transformation rules
- The rules are defined with a model transformation language

Source and Target Models

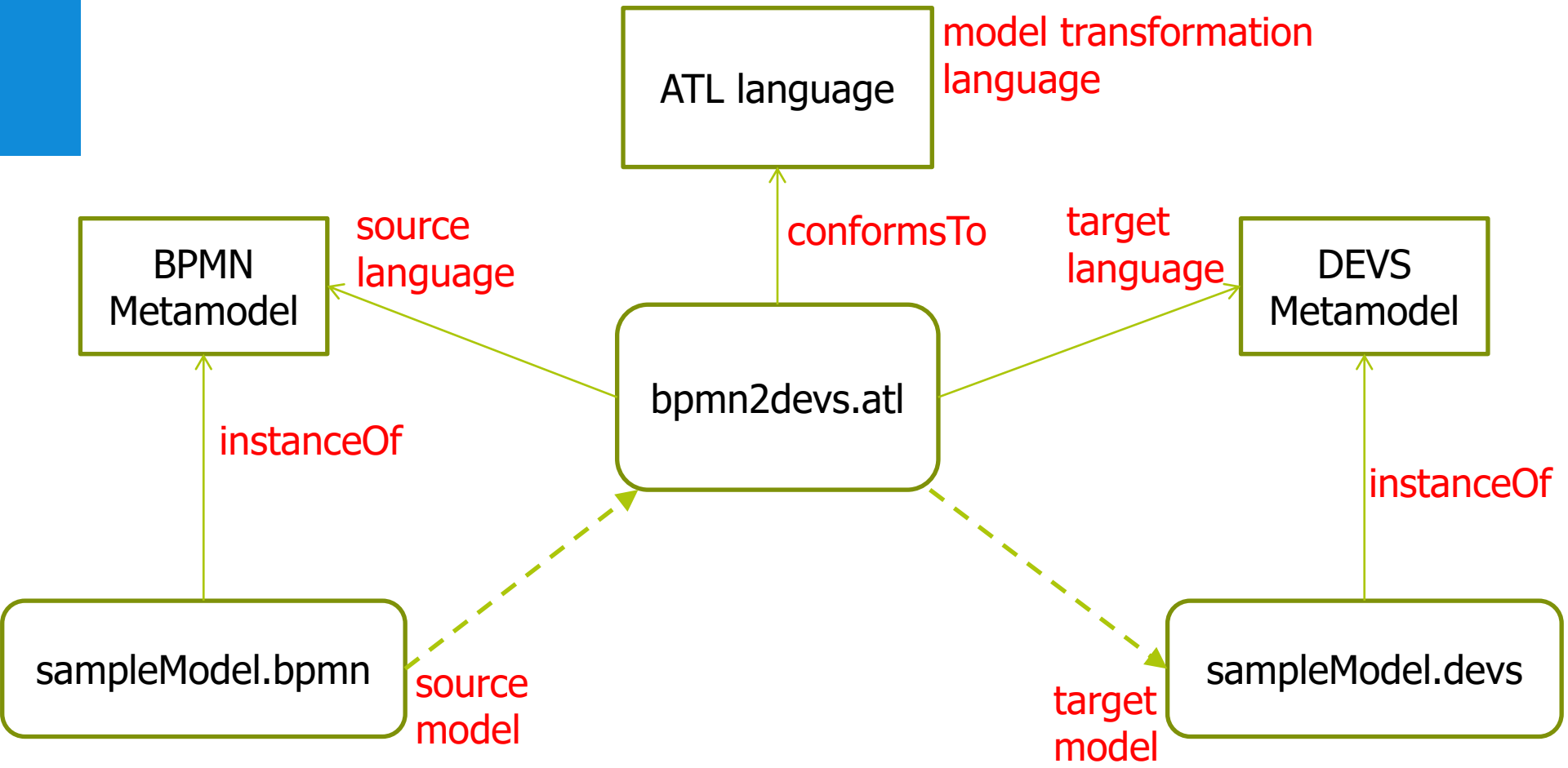
- A transformation rule consists of two parts: a left-hand side that accesses the source model; and a right-hand side that generates the target model
- The source model conforms to the source metamodel and the target model conforms to the target metamodel
- If the source and target metamodels are identical, the transformation is called endogenous; otherwise it is called exogenous

Model Transformations



[Ref: OMG, "MDA Version 1.0.1", 2003]

Model Transformations: example



Types of Model Transformations

- According to the type and complexity of the process, the model transformation can be named model-to-model (M2M) transformation, model-to-text (M2T) transformation, model merging, model linking, model synthesis or model mapping

- Popular M2M transformation languages are:
  - QVT (Query/View/Transformation)
  - ATL (ATLAS Transformation Language)
  - GReAT (Graph Rewriting and TransformationLanguage),
  - and the Xtend language

Model Transformations: Why?

- The goal of model transformations is to automatically generate different views of a system at different abstraction levels and to enable the reuse of information that was once modeled
- If the level of abstraction does not change, the transformation is called a horizontal transformation
- If the level of abstraction does change, the transformation is called a vertical transformation

5.

Metamodeling in Modeling and Simulation

Past

- The metamodeling term has been used in simulation for many years in a different context
- Metamodeling referred to constructing simplified models for simulation models that were quite complex and therefore slow to execute
- In this context, a metamodel is known as a "surrogate" model. Surrogate models mimic the complex behavior of the underlying simulation model
  - R. R. Barton. 1998. ”Simulation Metamodels”
  - W. J. H. Van Groenendaal and J. P. C. Kleijnen. 1996. “Regression metamodels and design of experiments”

Metamodel according to Kleijnen

- A model of a model (surrogate model)
- Reproducing input-output mapping (X -> Y)
  - black-box modeling, approximation
- Describing the behavior of a model with linear equations:

$$y_1 = a_{11}x_1 + a_{12}x_2 + \ldots + a_{112}x_1x_2 + \ldots$$
$$y_2 = a_{21}x_1 + a_{22}x_2 + \ldots + a_{212}x_1x_2 + \ldots$$
$$\ldots$$

interaction effects

- Design Of Experiments (DOE) needed to estimate a_{ij}
- Several other approaches described in Barton (1998)
- Papers in background reading on Bb

Past

- Metamodeling and model transformations in the presented context have been introduced to simulation in the last decade
  - A. Bakshi, V.K. Prasanna, and A. Ledeczi. 2001. "MILAN: A model based integrated simulation framework for design of embedded systems"
  - H. Vangheluwe and J. de Lara. 2002. "Meta-models are models too"
  - A. Tolk. 2002. "Avoiding another green elephant – a proposal for the next generation HLA based on the Model Driven Architecture"

Present

- Metamodeling have gained the attention of both industry and research communities in the M&S field
  - MDD4MS: A Model Driven Development Framework for Modeling and Simulation (research Deniz Cetinkaya)
    - http://simulation.tudelft.nl/MDD4MS/
    - dissertation: http://doi.org/10.4233/uuid:3db45913-1662-429f-a385-ed53f5ac41fd
  - ATOM3: A Tool for Multi-formalism and Meta-Modeling
    - http://msdl.cs.mcgill.ca/projects/projects/AToM3/
  - eUDEVS: Executable UML with DEVS Theory of Modeling and Simulation
    - http://duniptechnologies.com/jm/research.html
  - International Workshop on Model-driven Approaches for Simulation Engineering (also in '12, '13, '14, '15, '16 – change address below)
    - http://www.sel.uniroma2.it/mod4sim17/

Present

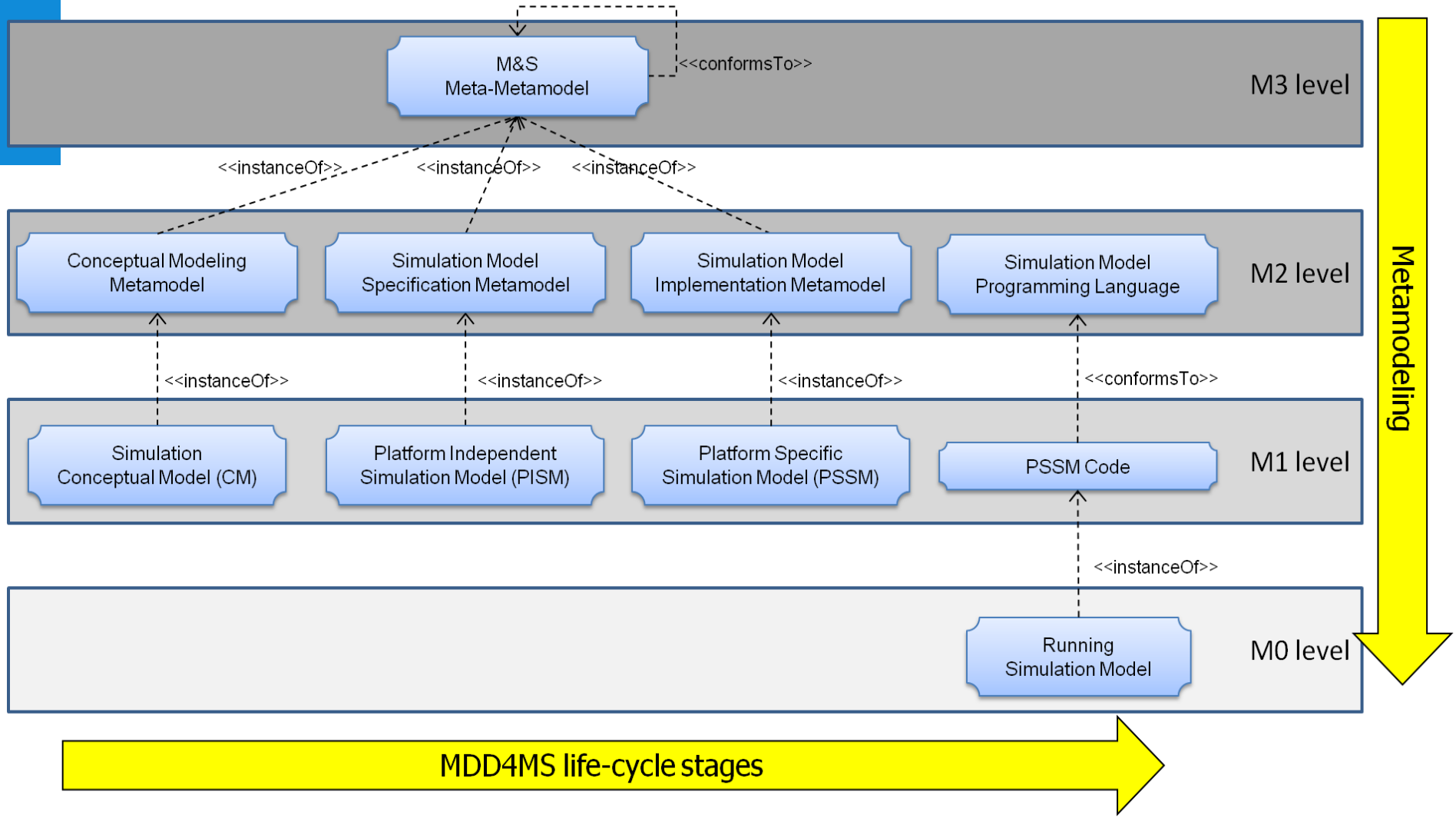
- In many applications, metamodeling is used to specify a DSL (Domain Specific Language) and model transformation is used for automatic code generation from the models that were specified in that DSL

- Metamodeling and model transformations can help more in M&S domain with the following subjects:
  - having tool support during the entire simulation model development life-cycle
  - using conceptual models in the further stages
  - producing well-structured and maintainable systems

MDD4MS Research Project

- The MDD4MS Research Project consists of three parts:
  - The MDD4MS Framework: concepts
  - The MDD4MS Case Studies: applications
  - The MDD4MS Prototype: tools
- The MDD4MS Framework addresses the steps in a simulation study from conceptual design through final implementation by applying metamodeling and model transformations
- It introduces three different metamodels to represent the simulation model implementation at different abstraction levels

Metamodeling in MDD4MS



Future

- Describing the simulation model accurately and completely, including the dynamics of the simulation model via metamodeling
- Adding component based approach in order to increase the portion of the auto-generated code
- Adding library approach and search facilities in order to allow reusing already defined components (implemented building blocks)
- Proposing a comprehensive theoretical framework for the metamodeling concepts in M&S domain