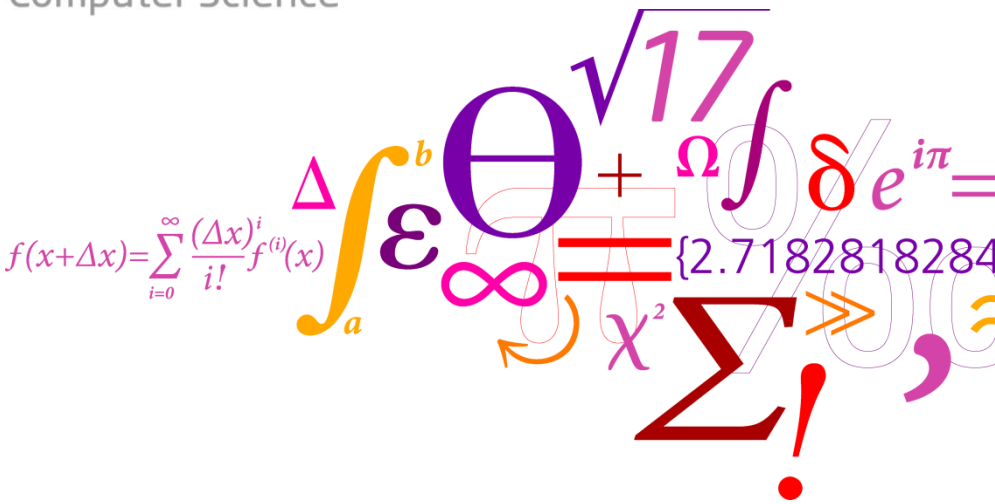
Model-based Software Engineering
(02341, spring 2016)

Ekkart Kindler

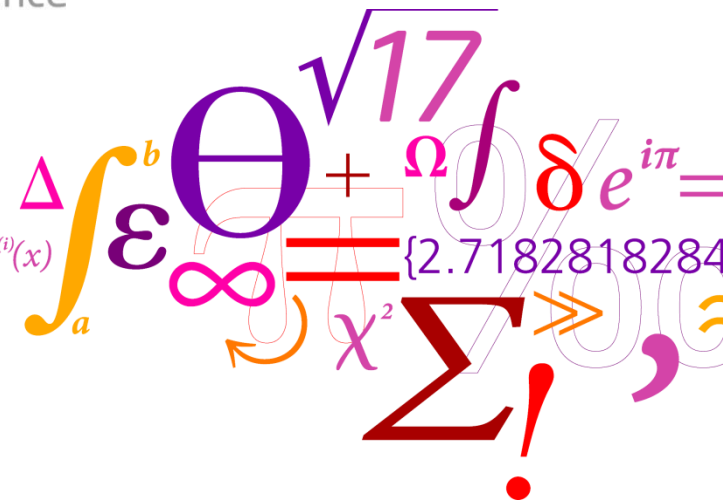
DTU Compute
Department of Applied Mathematics and Computer Science



Meta-modelling and Domain Specific Languages (DSLs) and Outlook

DTU Compute
Department of Applied Mathematics and Computer Science

Recapitulation and more detailed discussion on "meta" in MOF (EMOF)



DSL(s)

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- DSL (singular):
  A single domain specific language,
  designed and realised according to some principles
  and for a specific purpose or a specific domain

- DSLs (plural):
  - Disipline and principles for designing and realising a DSL
  - A technology or set of technolgies for designing and realising a DSL (mostly from MBSE)
  - A way of ”thinking” software design (idioms)

Parts of a DSL definition

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- **Abstract syntax** (see L01):
  language concepts and their relation
  (*API / domain model / framework)*

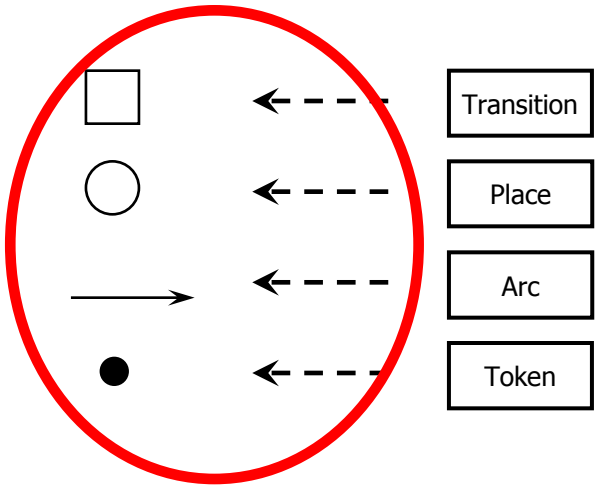
- **Concrete syntax** (see L01):
  syntactical representation of concepts
  (graphical or textual)

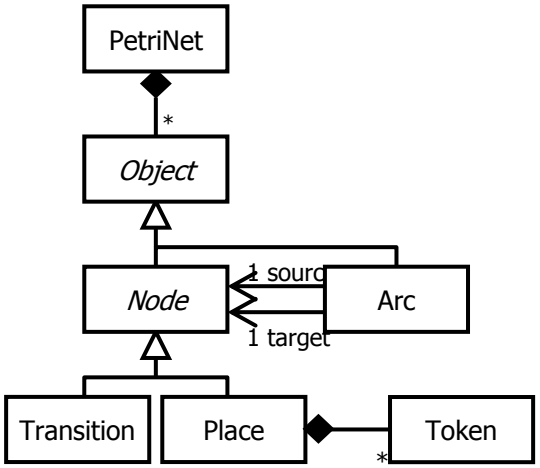
Actually, there could be different concrete syntax for the same abstract syntax

- **Semantics** (what it does):
  Code generation or interpretation, which enacts what an instance of the DSL says

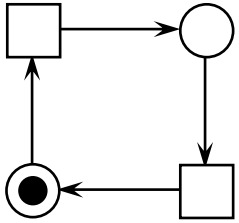
DSL Technolgies typically support the first two steps; and might help a bit with the last!

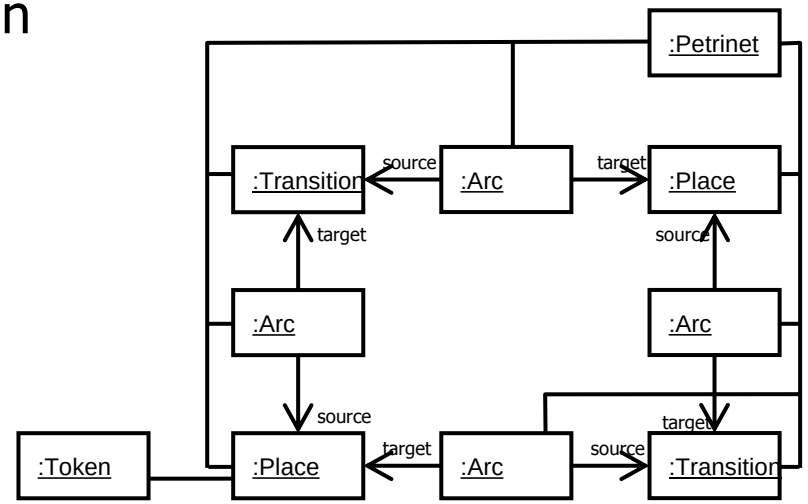
DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler





generate an editor





Philosophy

- A DSLs should help decrease redundancy and unnecessary work
- A DSL should help separating the variable or generic parts of a software product from parts which do not change
- A DSL should increase reuse
- A DSL should support abstraction form irrelevant technical details
- A DSL should emphasize the domain's idioms

MBSE & DSLs

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- MBSE Technologies help implementing DSLs in a fast and efficient way (mostly concerning abstract and concrete syntax)

- Therefore, the terms MBSE and DSL are often used in the same context (and sometimes mixed up)

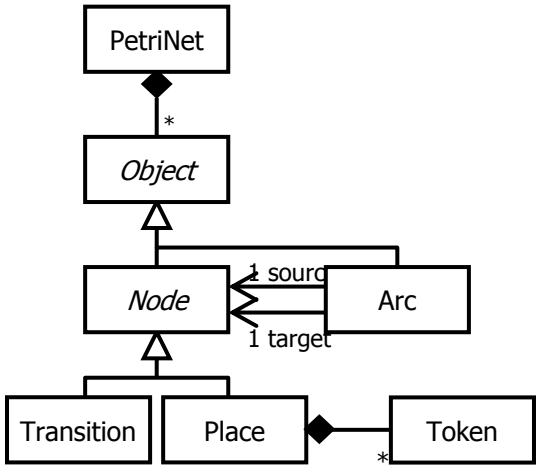
Meta-modelling (and MOF)

Meta-modelling is a core part of DSL design: Defining the abstract syntax

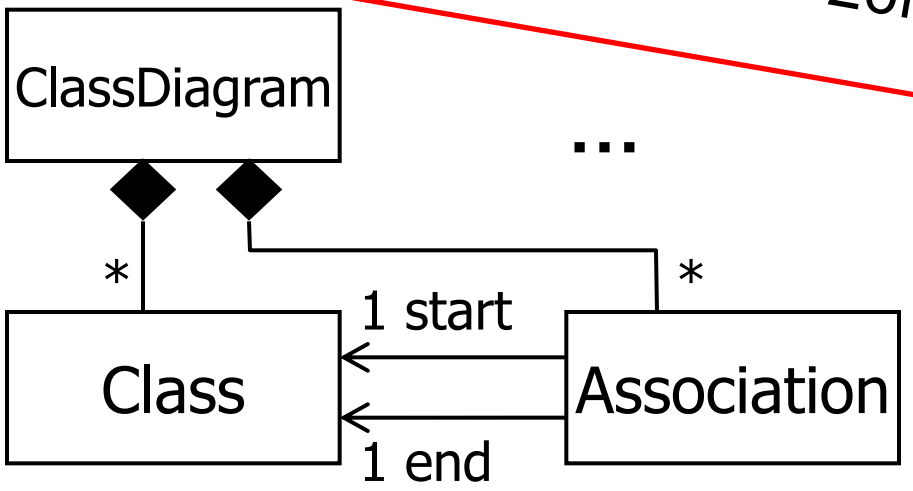
Class Diagrams are models too

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

Note: We will see later that this is **much more** involved (slide 26ff)!



UML model

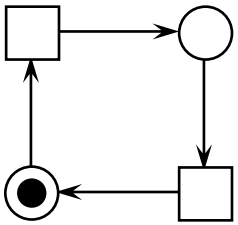


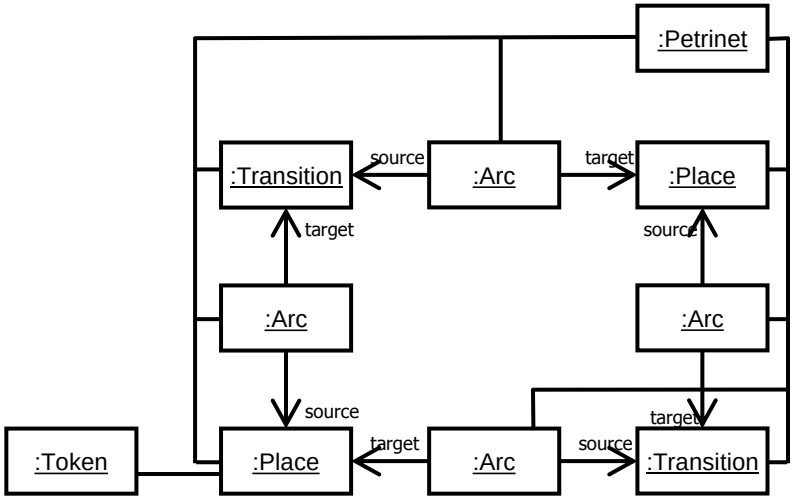
Meta-model for UML (class diagrams)

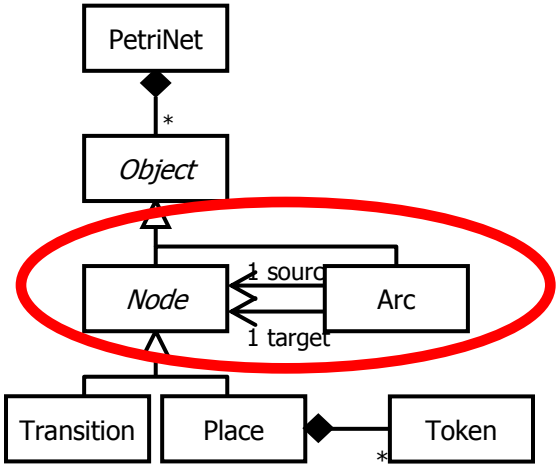
Now, the term “meta” model makes sense!

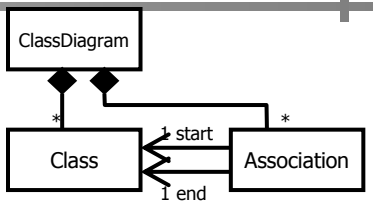
Meta-levels

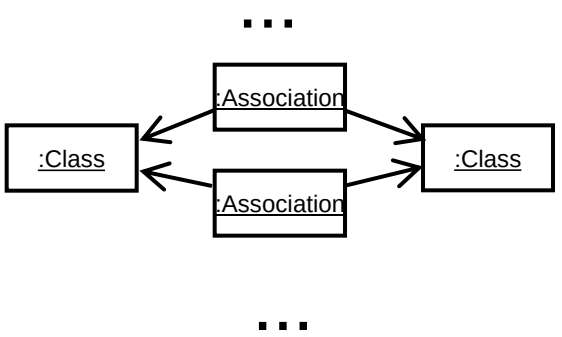
DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler









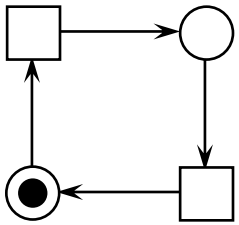


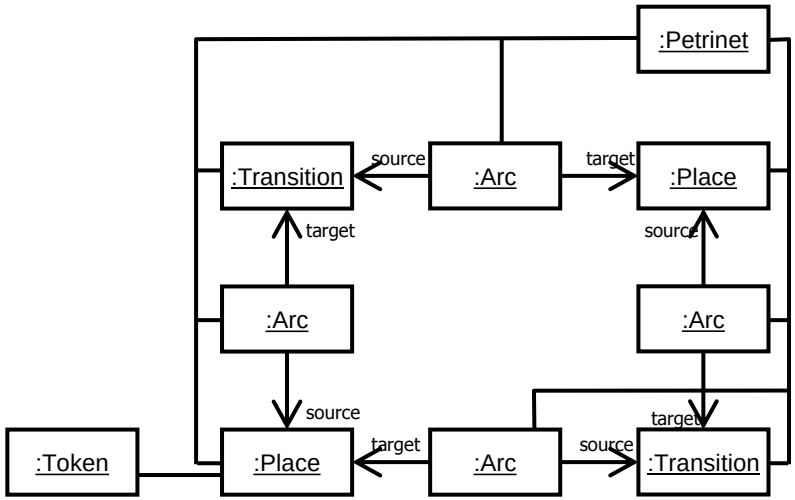
Levels of models

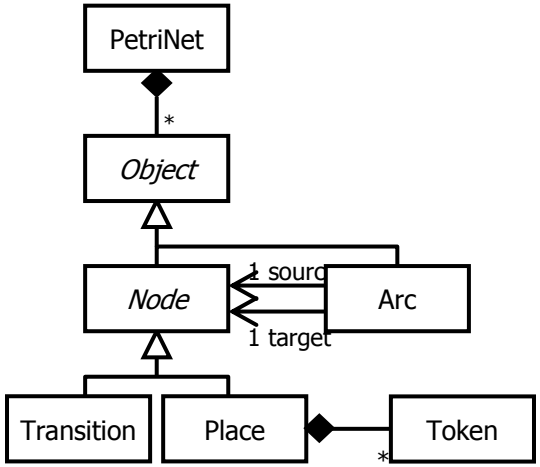
DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

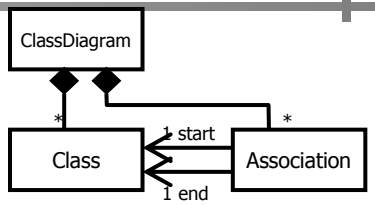
is an instance of

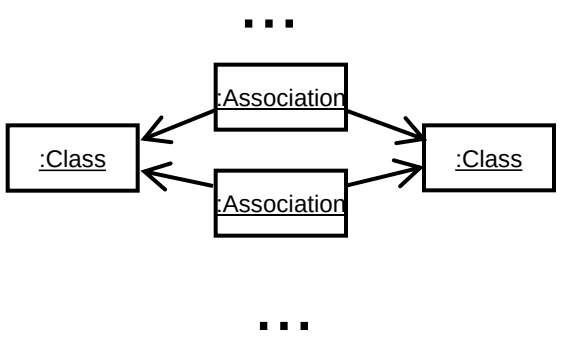
concrete syntax reprs. for





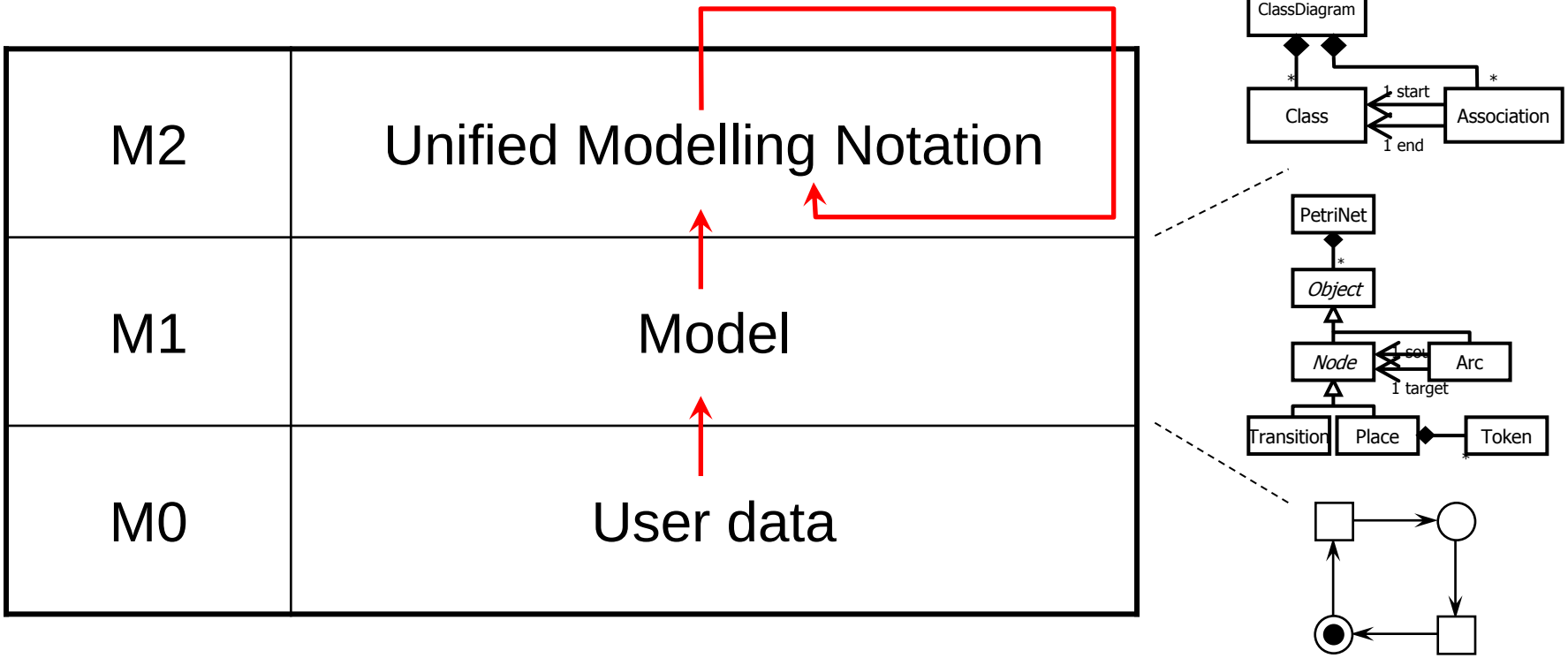






Outset

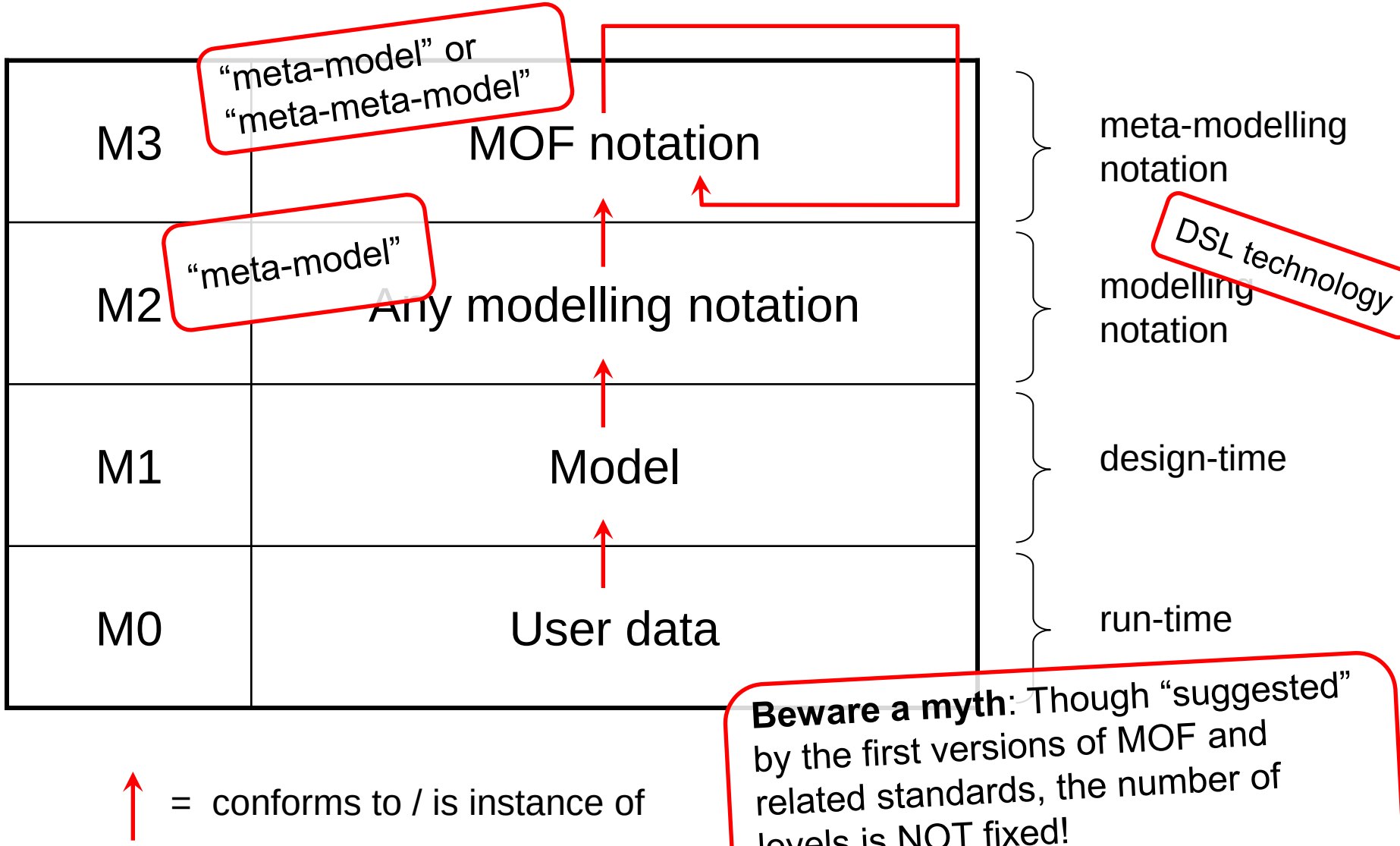
DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler



= conforms to / is instance of

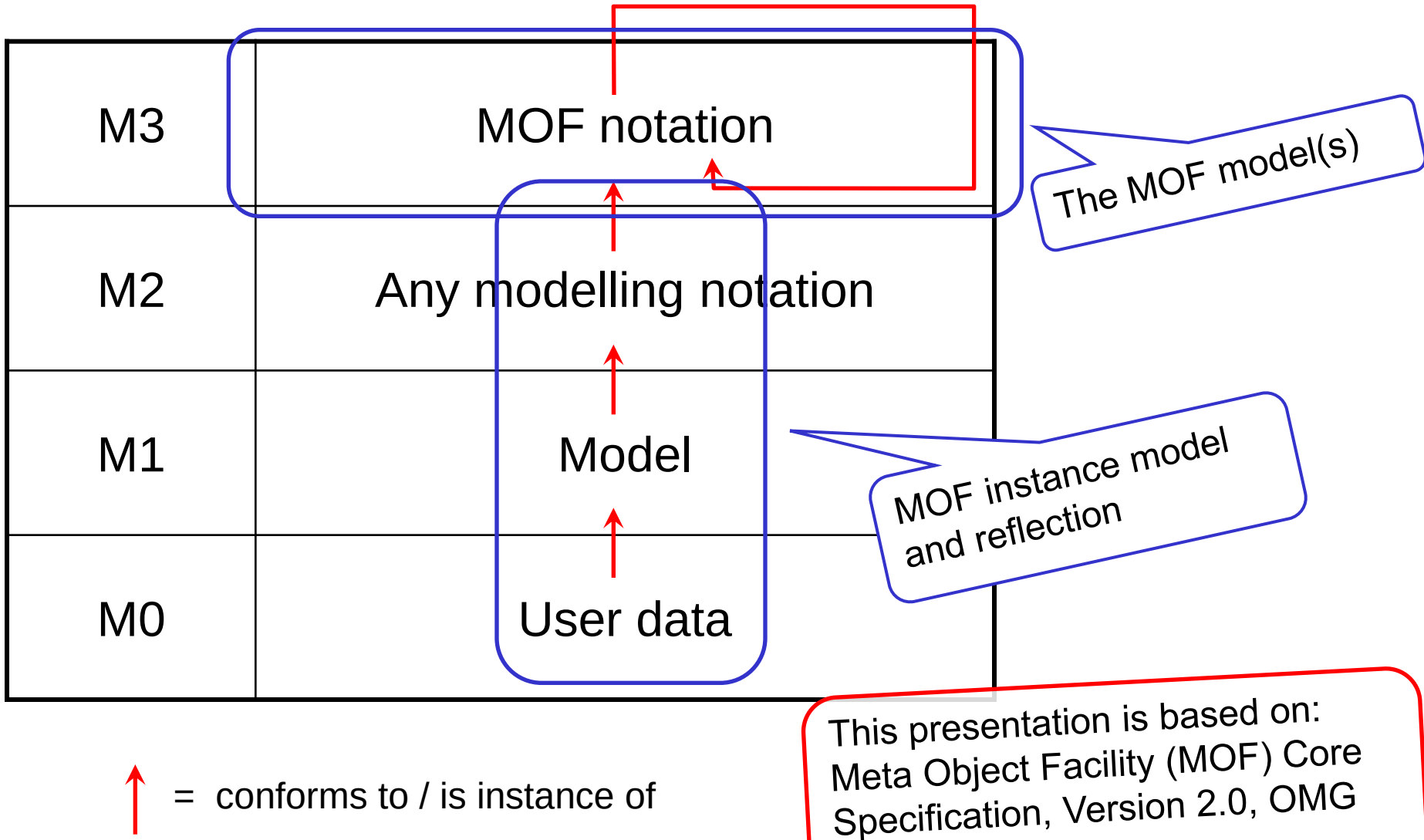
Meta Object Facility (MOF)

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler



The Meta Object Facility

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler



Discussion: meta

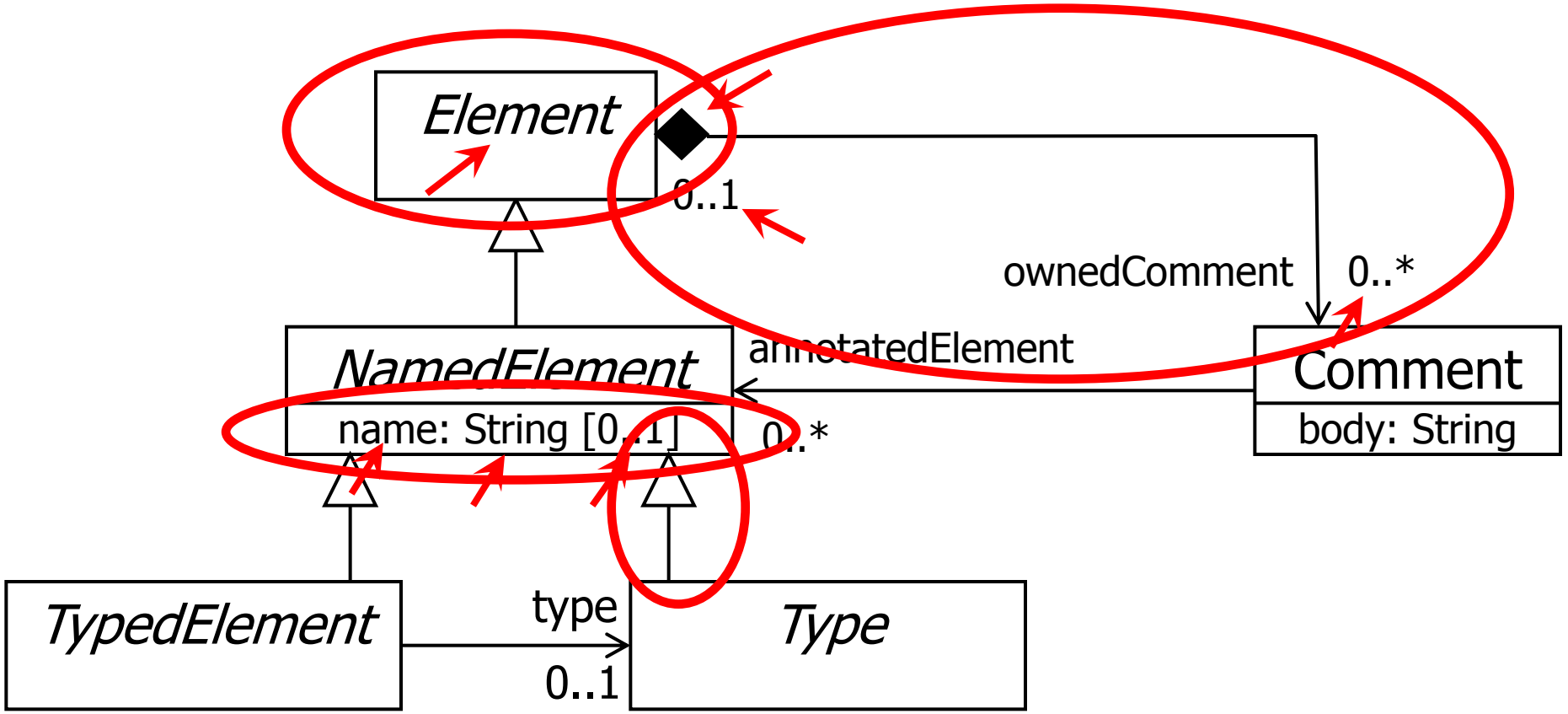
DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- EMOF as instance of itself

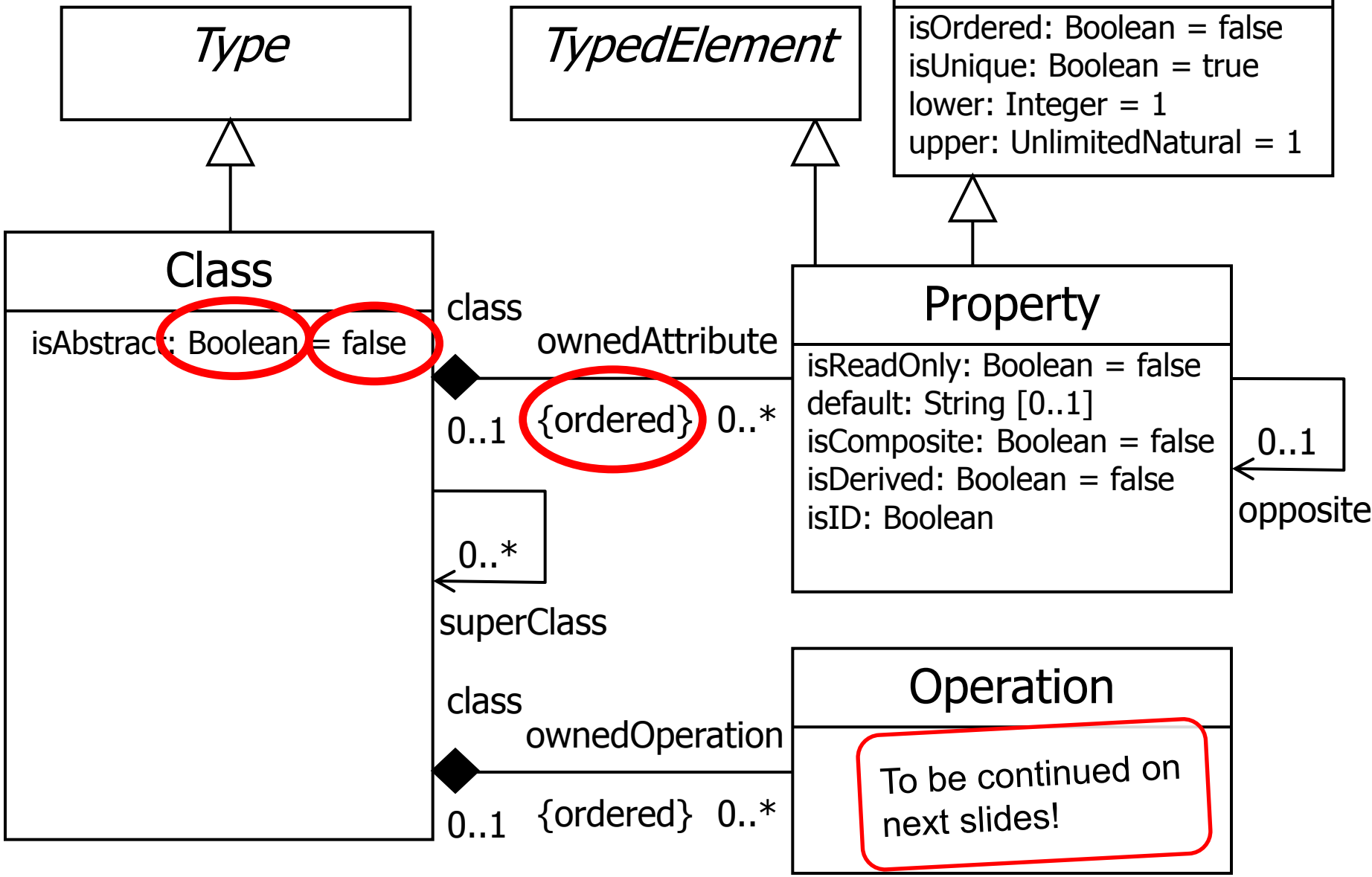
- Where are the different features of the EMOF model represented in EMOF

EMOF Types

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler



EMOF Classes



EMOF Classes (cntd.)

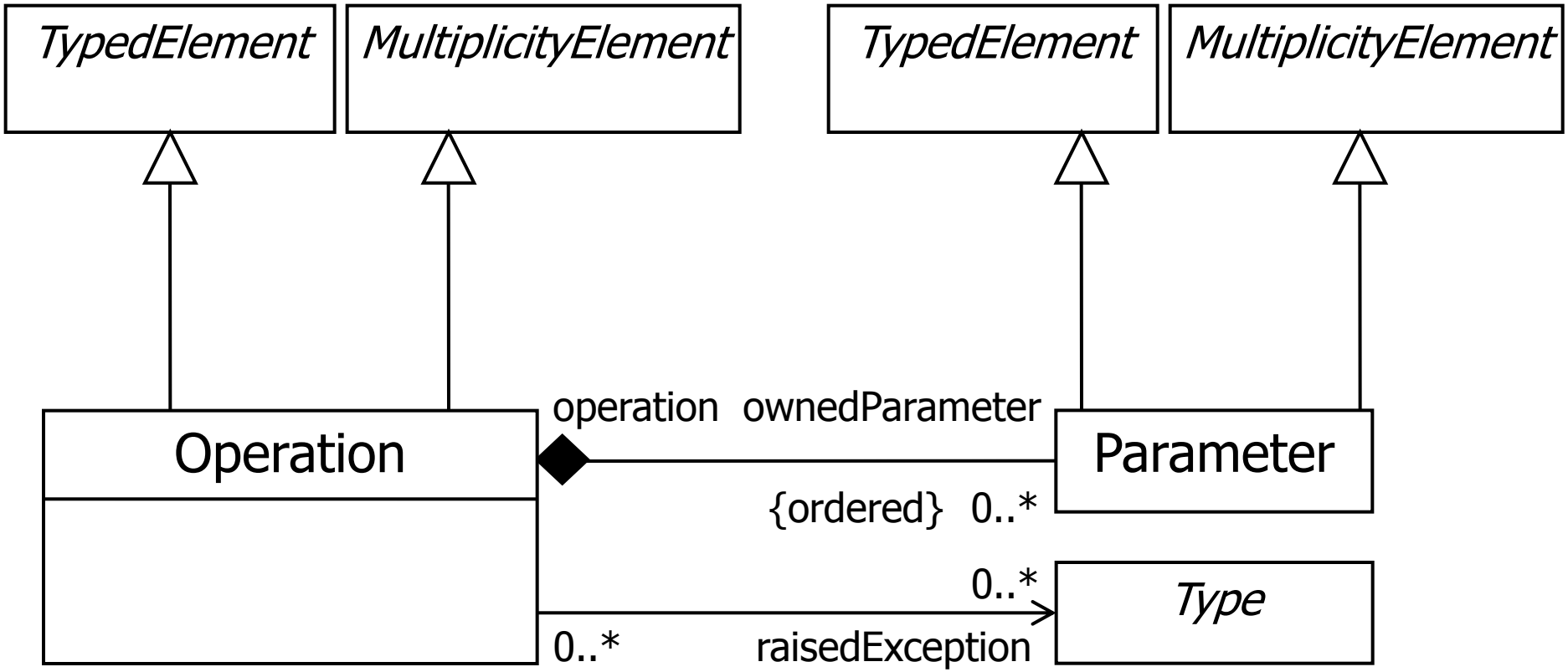
DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

Additional constraints (e.g.):

- opposite properties are properly paired
- no cycles in inheritance structure
- an object can be contained in at most one container

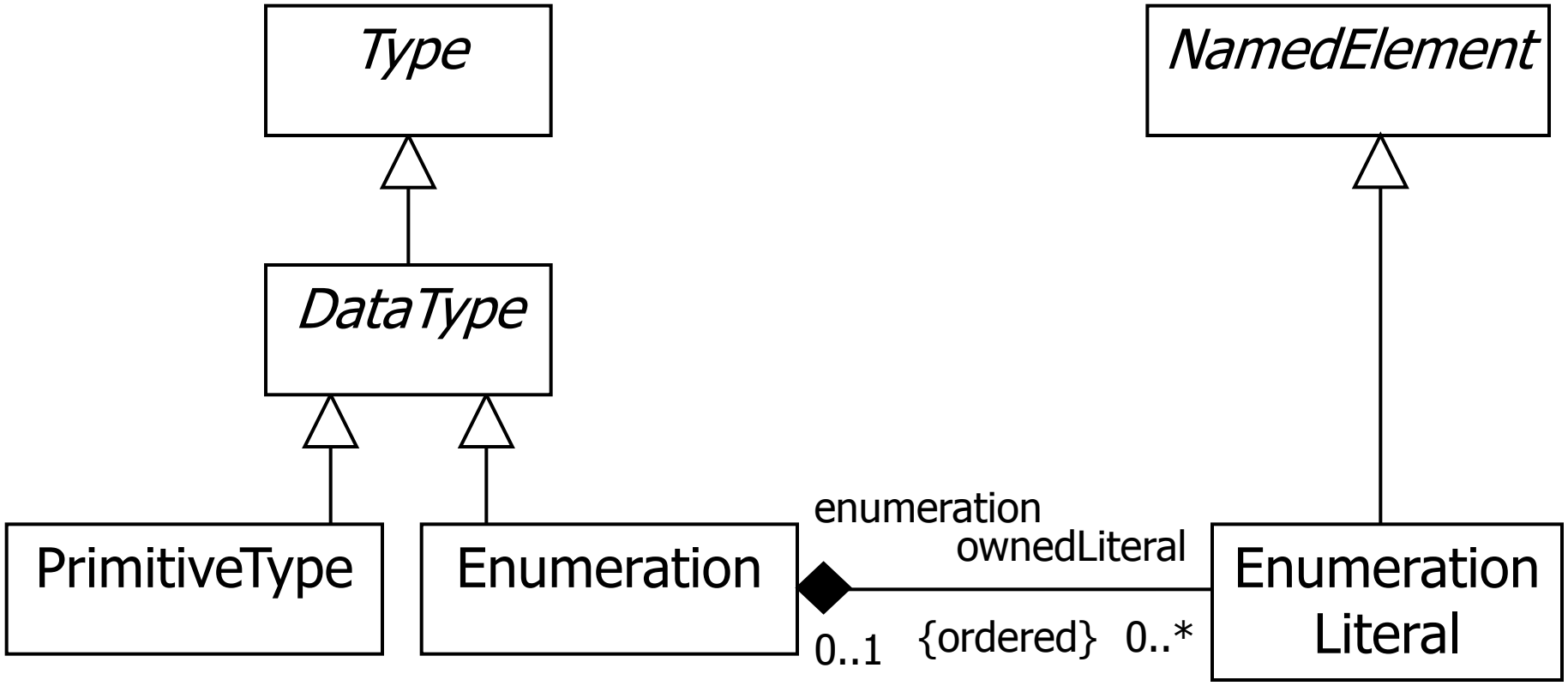
EMOF Classes (cntd.)

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler



EMOF Data Types

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler



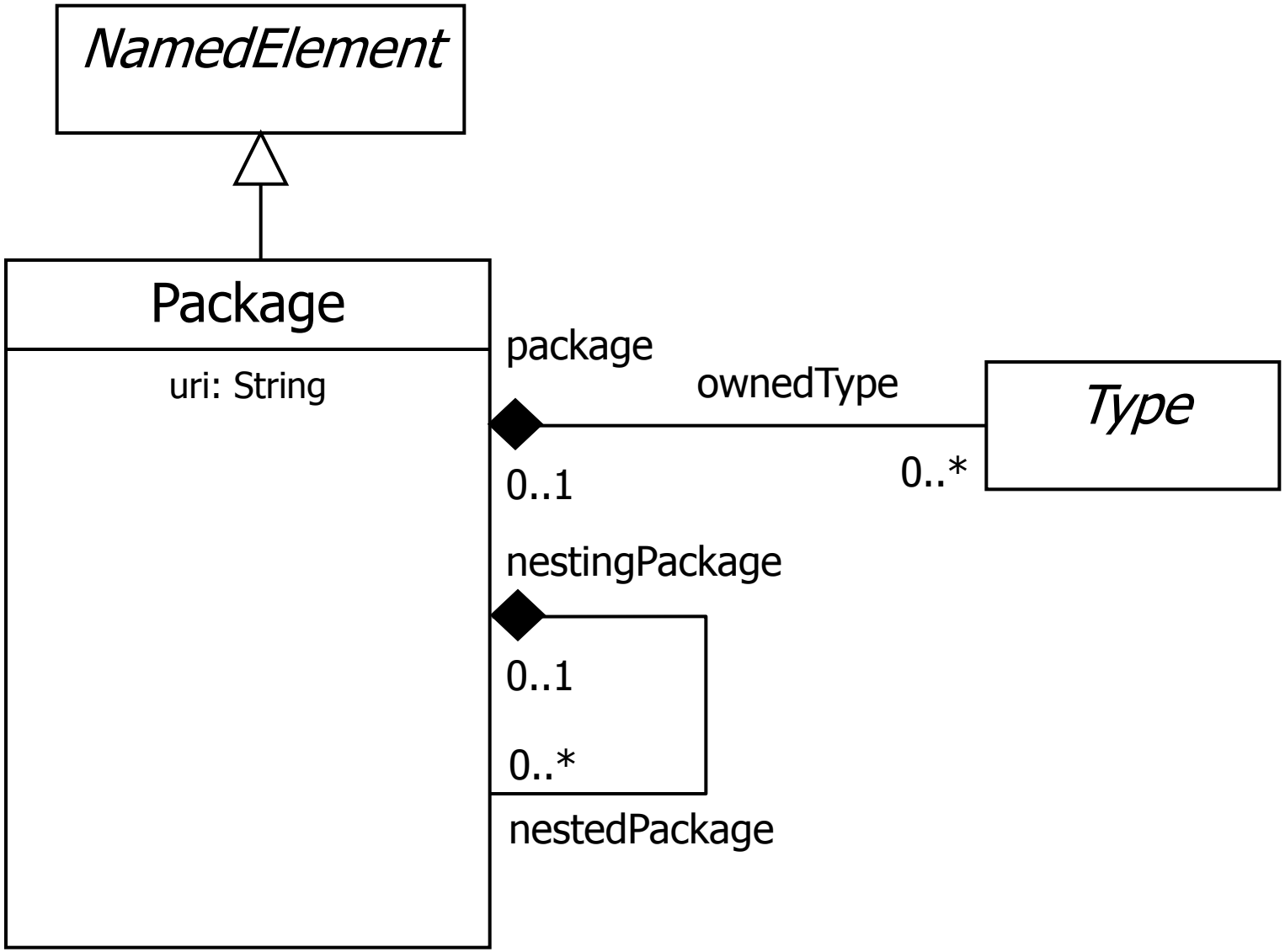
EMOF Primitive Types

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- Boolean
- String
- Integer
- UnlimitedNatural (* for ”infinity”)

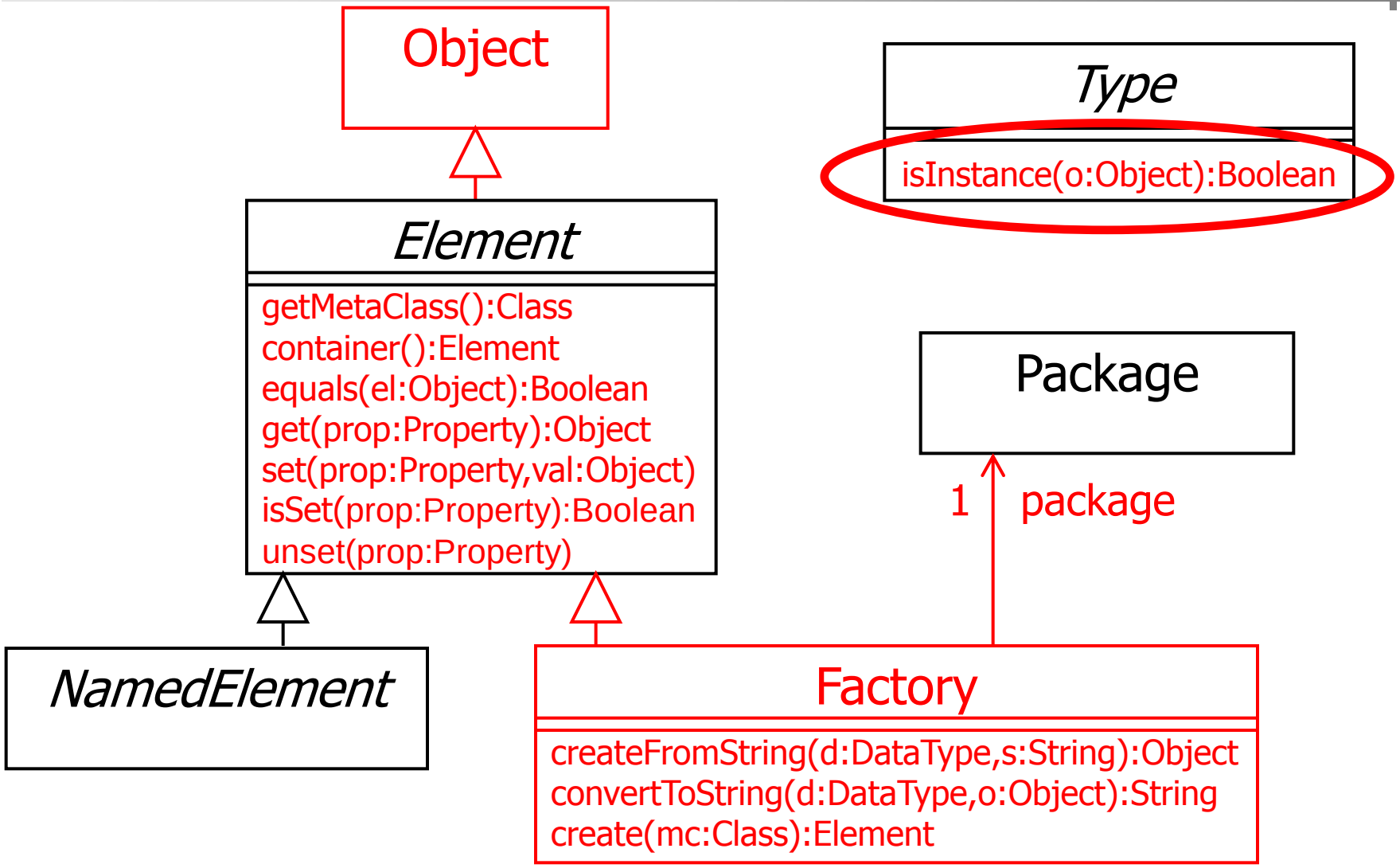
EMOF Packages

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler



Reflection package

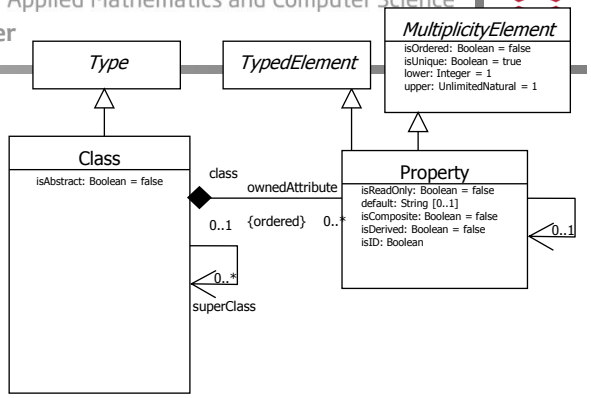
DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler



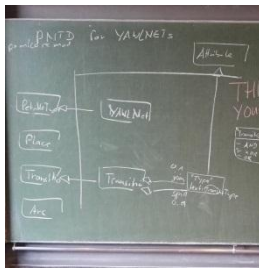
For properties with more than one value, there exist ReflexiveCollection and ReflexiveSequence (similar to Java Collections)!

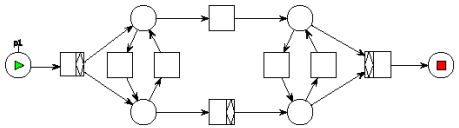
MOF in our project

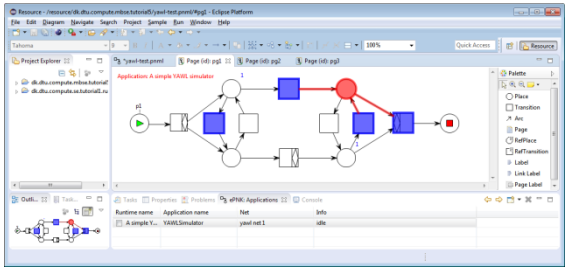
DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler



M3	Ecore (~ EMOF)	
M2	ePNK and YAWL meta models	+ Annotation model (runtime simulator)
M1	YAWL model	
M0	YAWL case (instance of a simulation)	



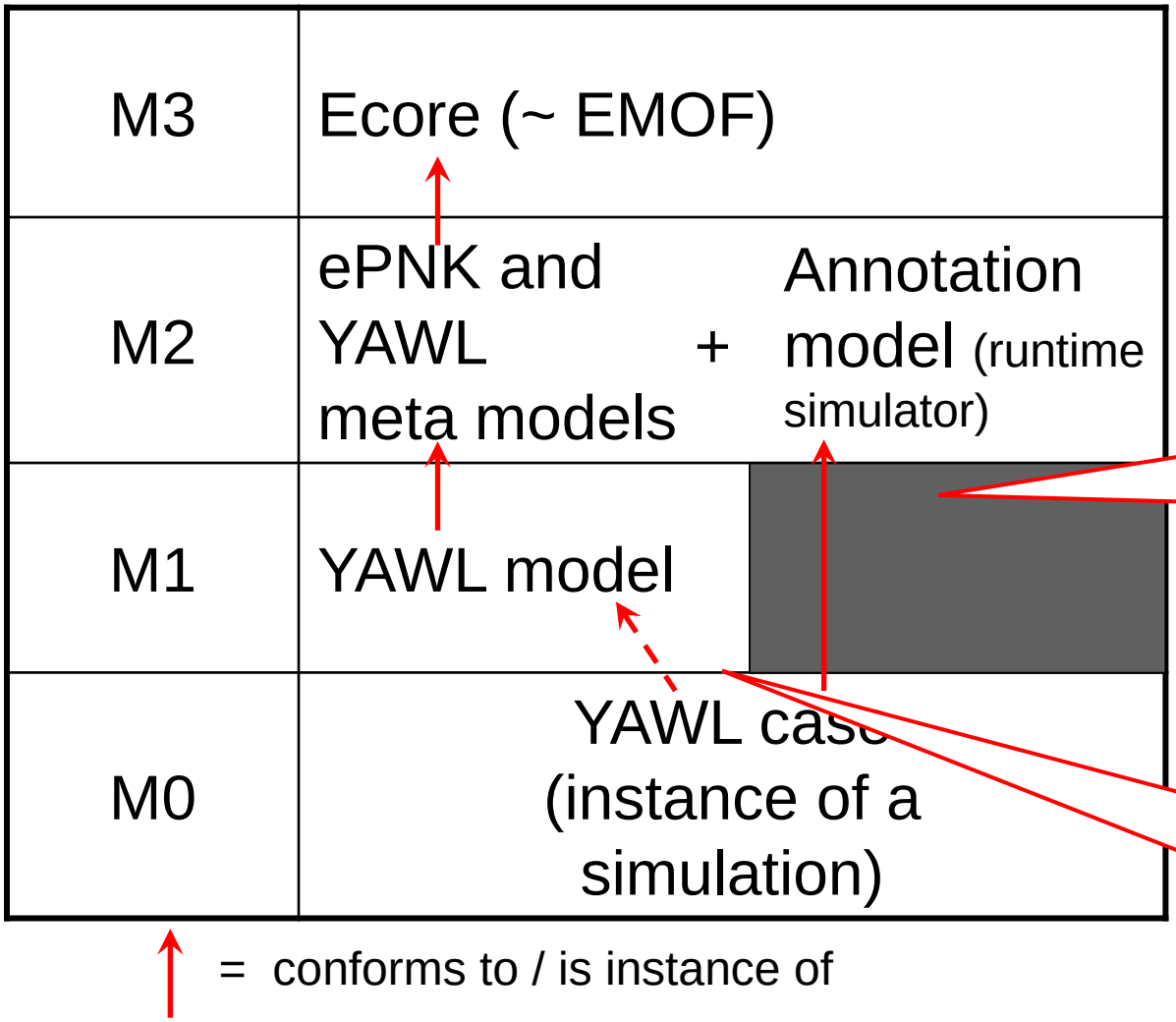




= conforms to / is instance of

MOF in our project

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler



Technically the runtime model instances jump one conceptual level

Not a technical (Java / OO) instance; just a conceptual instance of YAWL

MBSE

Feel the consequences of wrong models.

In addition to models and automatic code generation,

MBSE provides techniques for structuring/extending (big) software (framework behind the scenes):

- Factories
- Interfaces
- Listeners / observers
- Commands
- Handlers
- Extension points

MBSE: Not covered

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- Customizing code generation
- Defining own code generators
  (defining model to text transformations M2T)
- Transforminga model into another model
  (class diagram to database scheme, M2M)
- Other MBSE technologies (e.g. Microsofts Entity Framework)
- Techniques for developping embedded DSLs

Tomorrow

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- Guest lecture by Rasmus Petersen from Netcompany:

  How does Netcompany use models