Part 0 - MOST
1. Modeling

Prof. Dr. rer. nat. Uwe Aßmann

Institut für Software- und Multimediatechnik

Lehrstuhl Softwaretechnologie

Fakultät für Informatik

Technische Universität Dresden

http://st.inf.tu-dresden.de/teaching/most
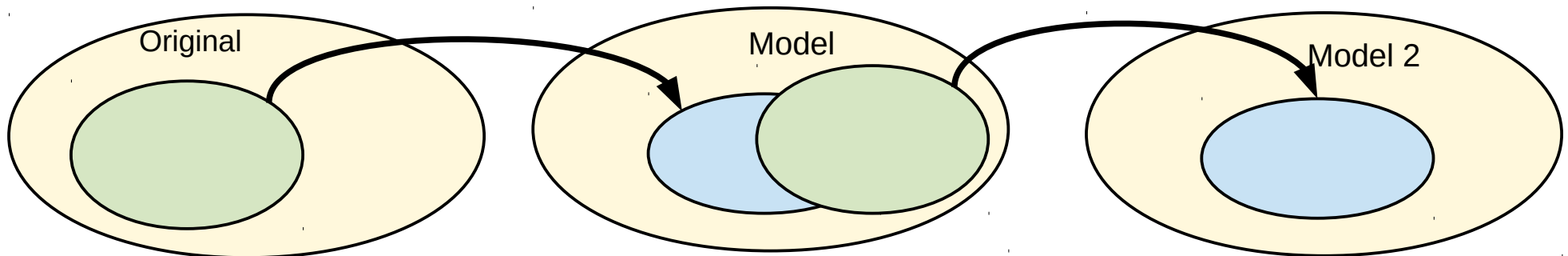
Version 15-0.4, 10.10.15

Literature

▶ Obligatory:

- [HesseMayr] Wolfgang Hesse and Heinrich C. Mayr. Modellierung in der Softwaretechnik: eine Bestandsaufnahme. Informatik Spektrum, 31(5):377-393, 2008.

▶ References:

- Stachowiak, Herbert. Allgemeine Modelltheorie. Springer, Wien, 1973

Original and Model
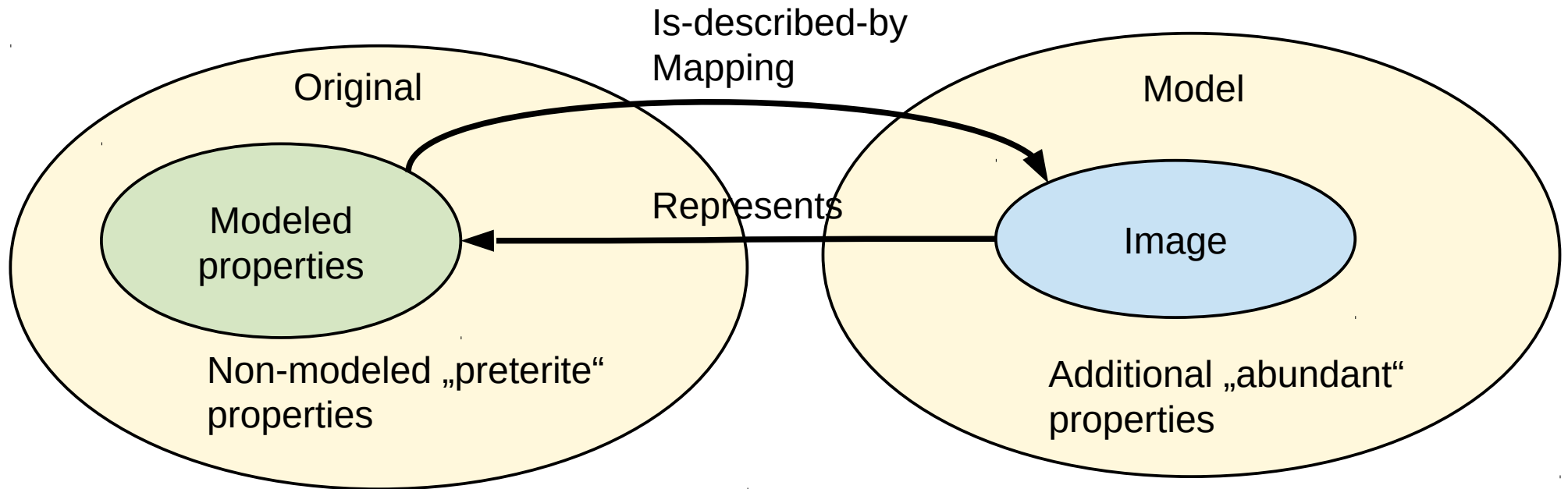
- ▶ [HesseMayr, Stachowiak]
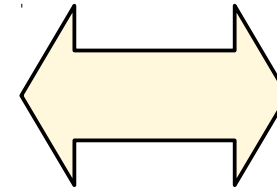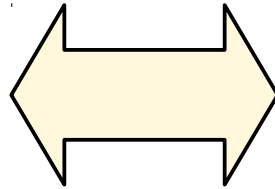- ▶ Model mappings can be sequenced:

A **model** is an abstraction of an original [Stachowiak]

A **system model** is an abstraction of a system

A direct **model** is an abstraction of a reality

A **world model** is an abstraction of a world

A **domain model** is an abstraction of a domain of the world

© Prof. U. Aßmann

Descriptive
Modeller
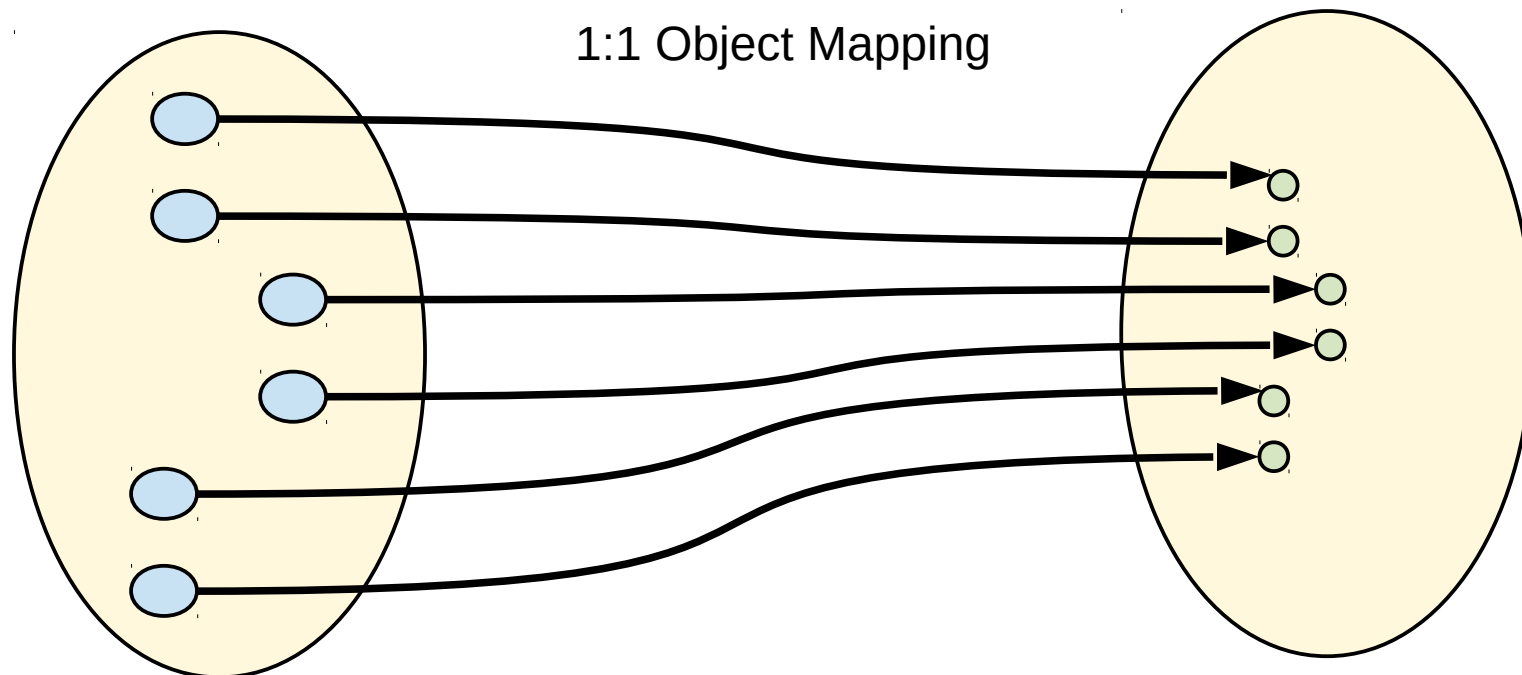
Prescriptive
Modeler;
Specifier;
Implementer

[HesseMayr]

Token Modeling

▶ In **Token modeling**, some features of the objects in original domain O are forgotten, but never the objects themselves

- Abstraction over features

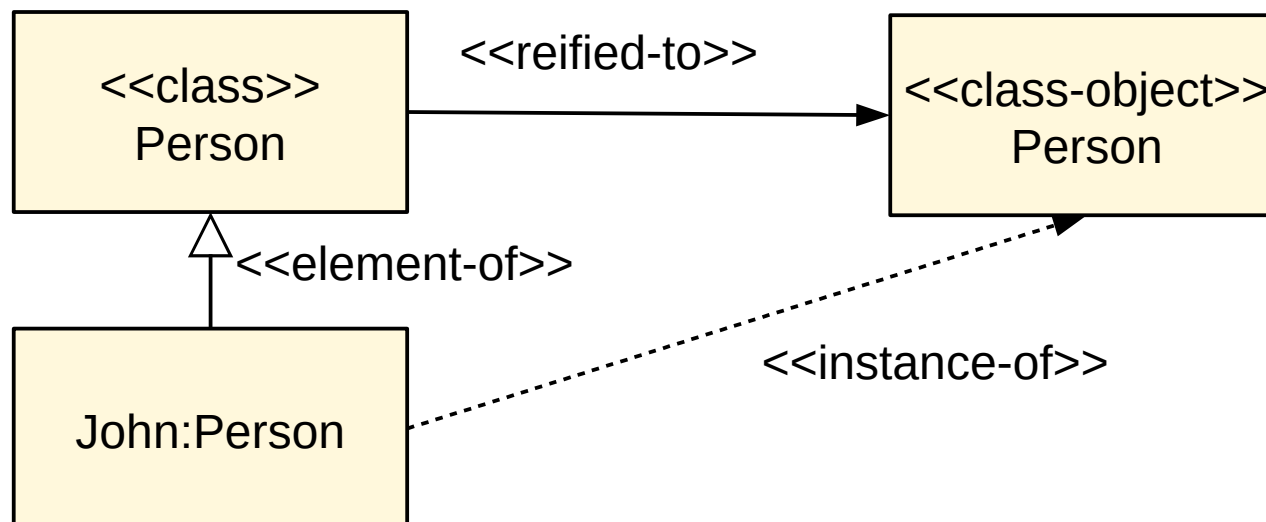- Leading to view-based modeling, aspect-oriented modeling

1:1 Object Mapping

Type Modeling

▶ In **type modeling**, sets of objects are abstracted

N:m<n Mapping
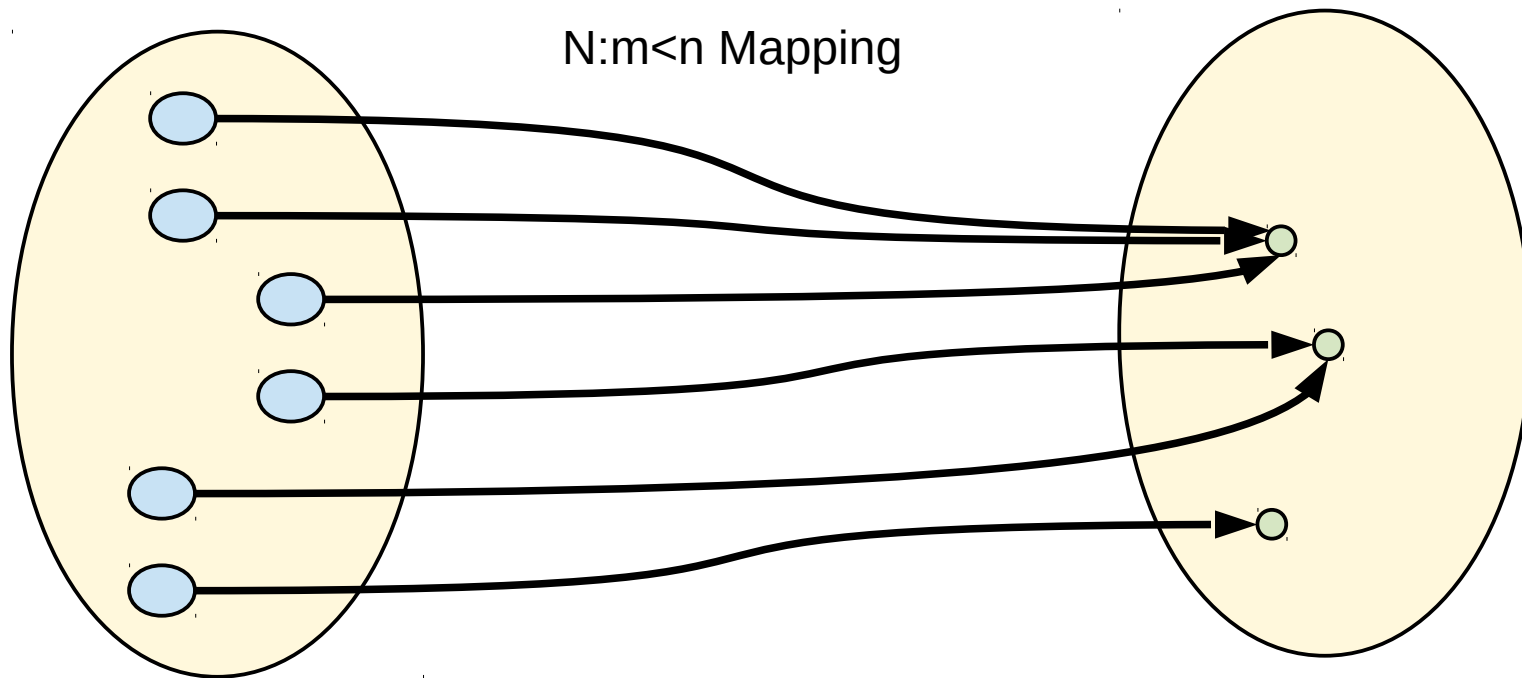
Type Modeling

- ▶ **Clabjects (class-objects)** are classes reified as *representant objects* on the metalevel.
    - ▪ In an object-oriented program, clabjects are objects that represent classes of other objects.
- ▶ Russells Paradox "The set of all sets containing themselves as elements" forbids infinitely many reifications
- ▶ <<instance-of>> is a composition of <<element-of>> with <<reified-to>>

```
    ┌──────────────┐   <<reified-to>>   ┌──────────────────┐
    │  <<class>>   │ ─────────────────▶ │ <<class-object>> │
    │   Person     │                    │     Person       │
    └──────────────┘                    └──────────────────┘
           △                                      ▲
           │ <<element-of>>                       ⋰
    ┌──────────────┐                          ⋰
    │              │              <<instance-of>>
    │ John:Person  │ ⋰⋰⋰⋰⋰
    └──────────────┘
```
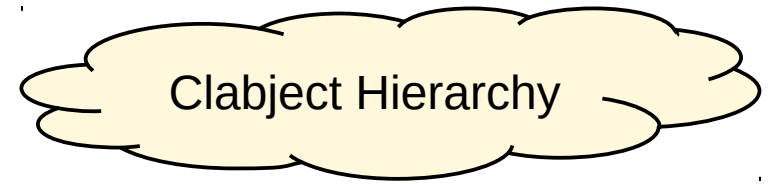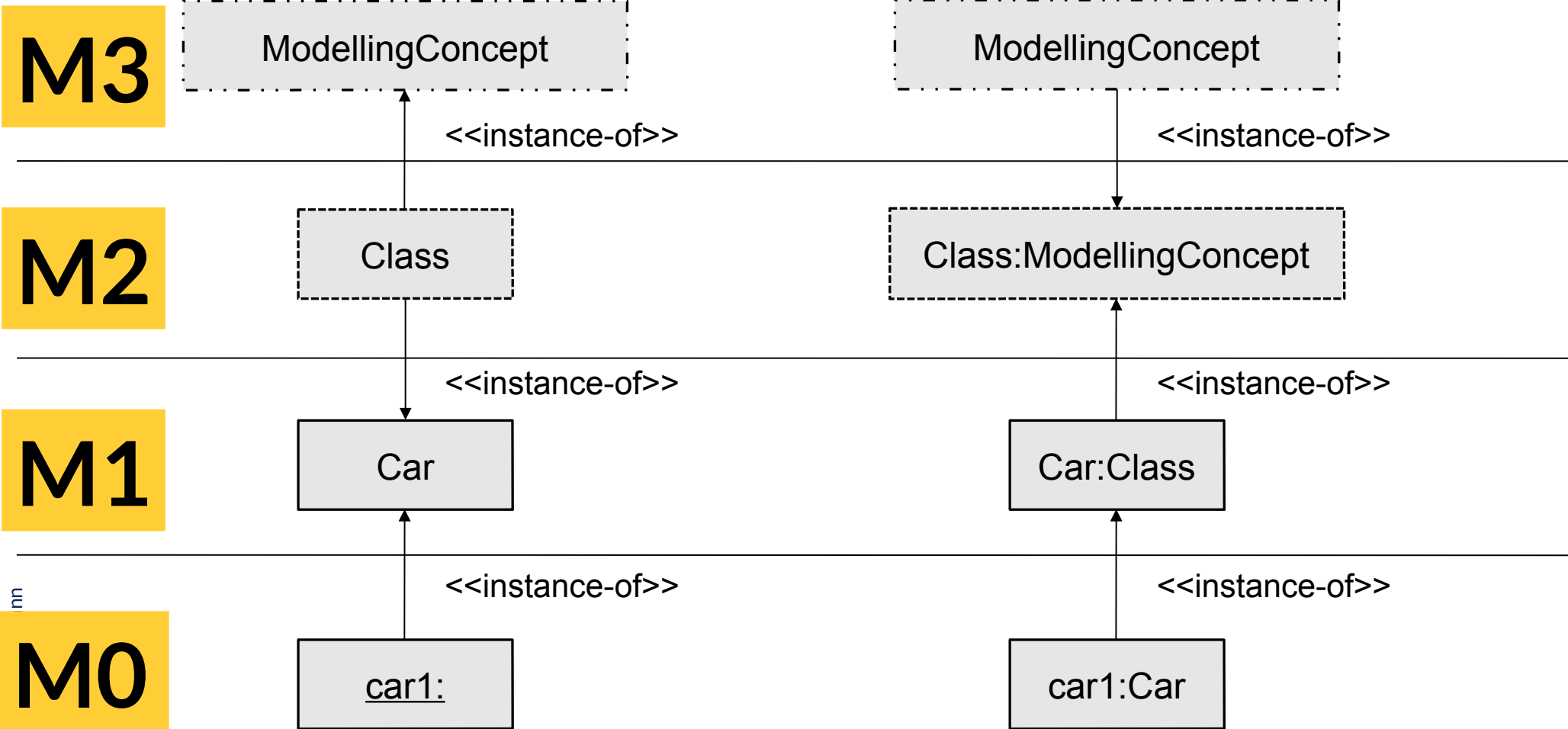
The Smalltalk Metaclass

▶ Smalltalk-80 was the first language to introduce metamodeling

▶ It introduced **clabjects** as **class-objects** and as **metaclass.**

▶ Changing the Smalltalk metaclass changes the semantics of all classes and all objects.

▶ In Java, class `Class` is the metaclass, but it is immutable

Notation

Clabject Hierarchy
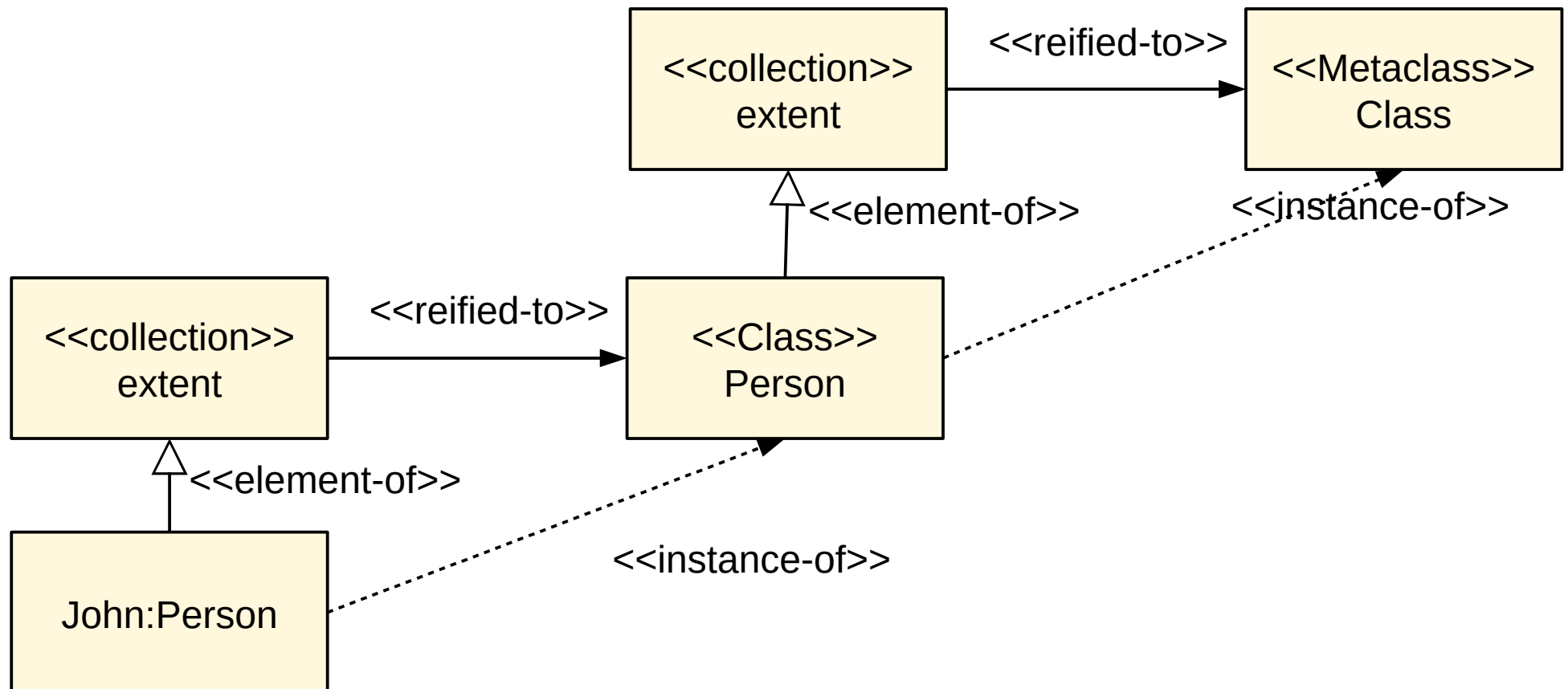
▶ We write metaclasses (clabjects) with dashed lines, metametaclasses (clabjects) with dotted-dashed lines

M3

| ModellingConcept | ModellingConcept |

<<instance-of>>   <<instance-of>>

M2

| Class | Class:ModellingConcept |

<<instance-of>>   <<instance-of>>

M1

| Car | Car:Class |

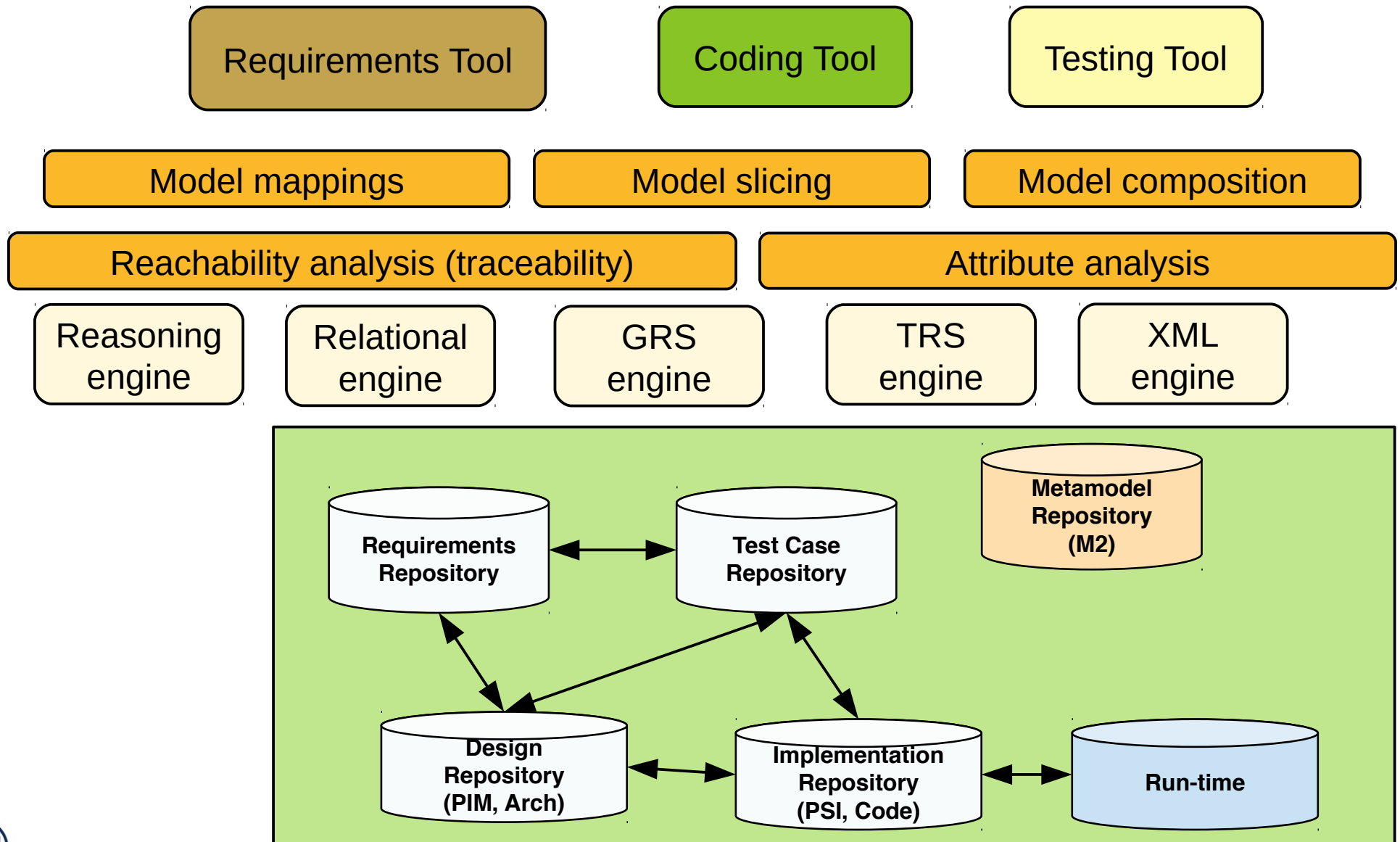<<instance-of>>   <<instance-of>>

M0

| car1: | car1:Car |

Q1: IDE and Model-Driven Software Development

▸ MDSD systematically connects the customer's problems, the system's requirements, testing, design, coding, and documentation and develops these models in coordination

▸ MDSD relies on model mappings between requirements, test cases, design, and code

▸ Integrated Development Environments (IDE) provide tools for all singular aspects, as well as model mappings

Model-Driven Software Development in Technical Spaces (MOST) © Prof. U. Aßmann

| Requirements Tool | Coding Tool | Testing Tool |

| Model mappings | Model slicing | Model composition |

| Reachability analysis (traceability) | Attribute analysis |

| Reasoning engine | Relational engine | GRS engine | TRS engine | XML engine |

Requirements Repository ⟷ Test Case Repository

Metamodel Repository (M2)

Design Repository (PIM, Arch) ⟷ Implementation Repository (PSI, Code) ⟷ Run-time

The End

14