Software Engineering I: Software Technology

WS 2008/09

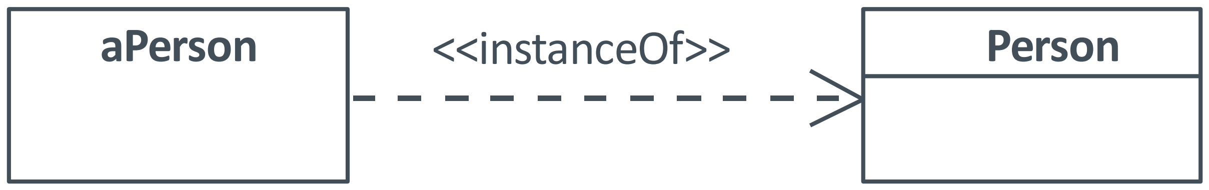
The UML 2.0 meta model

Prof. Bernd Bruegge, Ph.D.
Florian Schneider
Applied Software Engineering
Technische Universitaet Muenchen

Outline for today

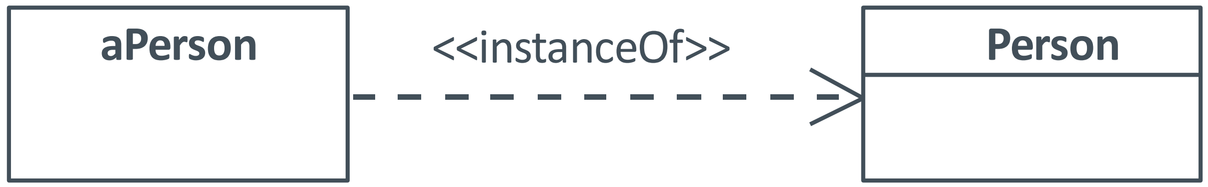
- From model instances to meta models
- MOF meta model hierarchy
- How UML relates to MOF
  - Example: Use case diagram meta model
  - Example: Class diagram meta model
- Different notations for the UML meta model describe the same language
- UML Profiles: Adding new members to the family

From model instances to meta models



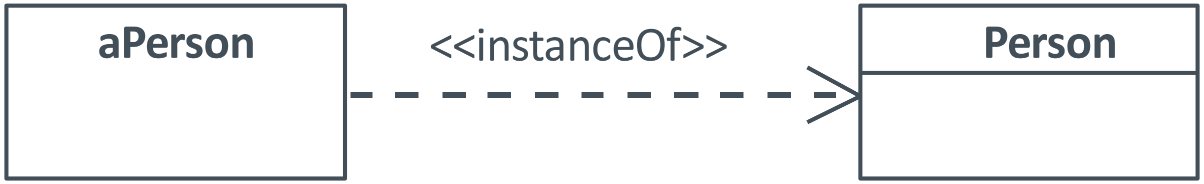
From model instances to meta models

- Canonical model-instance-relationship:



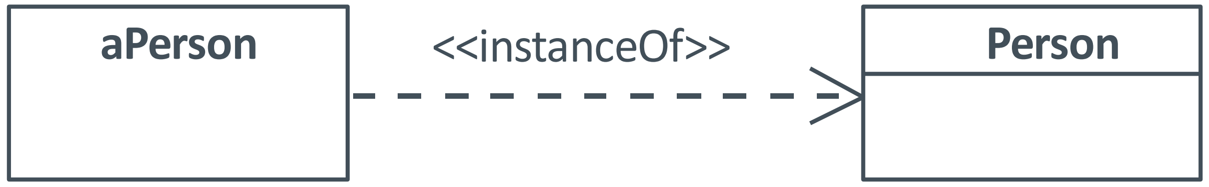
From model instances to meta models

- Canonical model-instance-relationship:



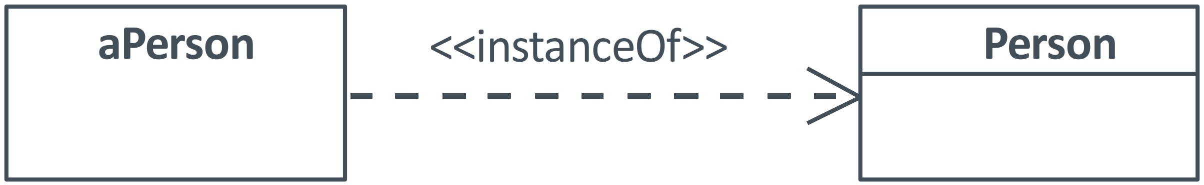
From model instances to meta models

- Canonical model-instance-relationship:



From model instances to meta models

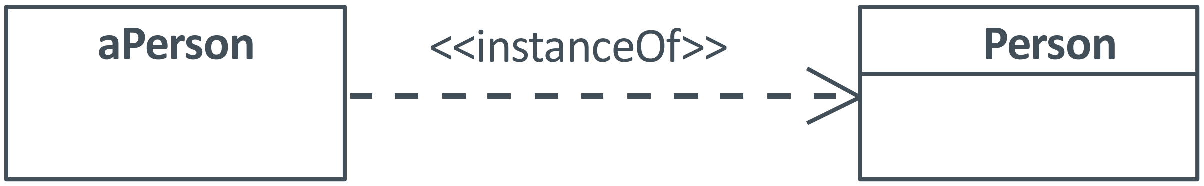
- Canonical model-instance-relationship:



aPerson is an instance of the class Person.

From model instances to meta models

- Canonical model-instance-relationship:

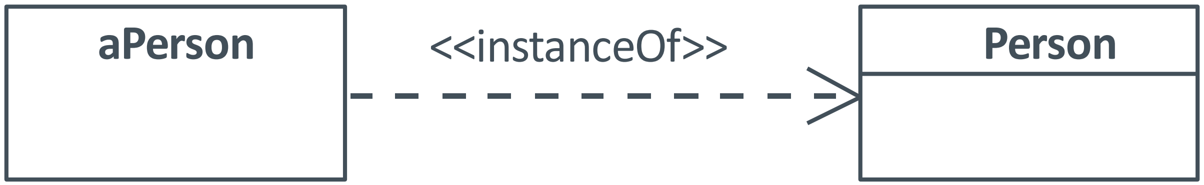


aPerson is an instance of the class Person.

Thus the class Person is a model for aPerson.

From model instances to meta models

- Canonical model-instance-relationship:

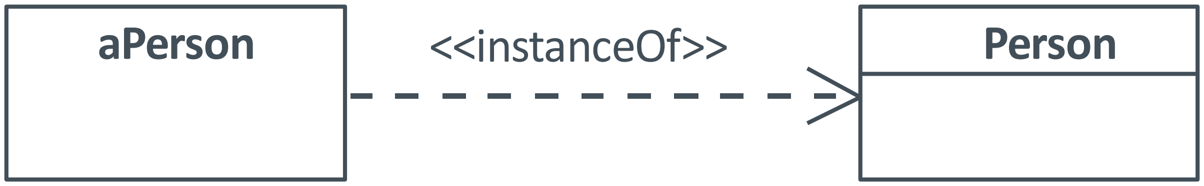


aPerson is an instance of the class Person.

Thus the class Person is a model for aPerson.

From model instances to meta models

- Canonical model-instance-relationship:



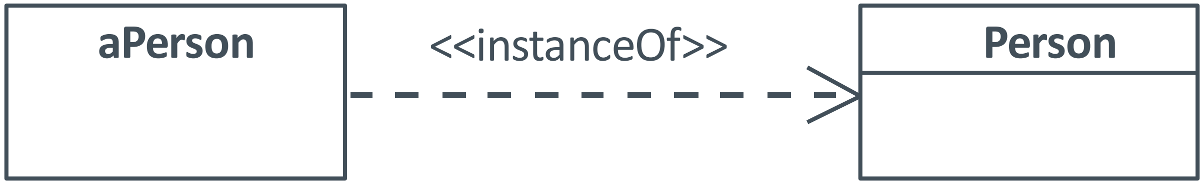
aPerson is an instance of the class Person.

Thus the class Person is a model for aPerson.

- Can we generalize this relationship?

From model instances to meta models

- Canonical model-instance-relationship:



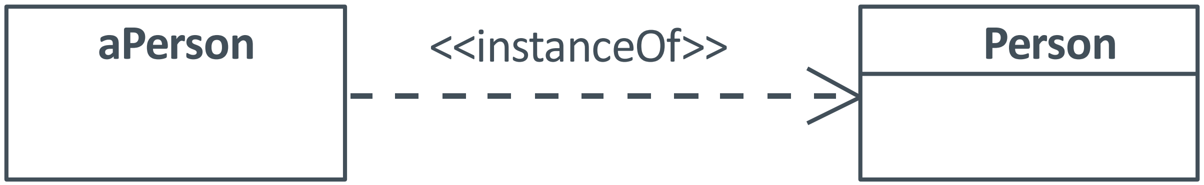
aPerson is an instance of the class Person.

Thus the class Person is a model for aPerson.

- Can we generalize this relationship?

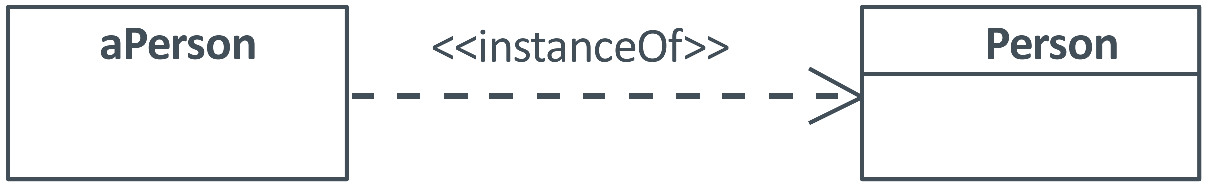
  ➔ What is the model for the class Person?

From model instances to meta models



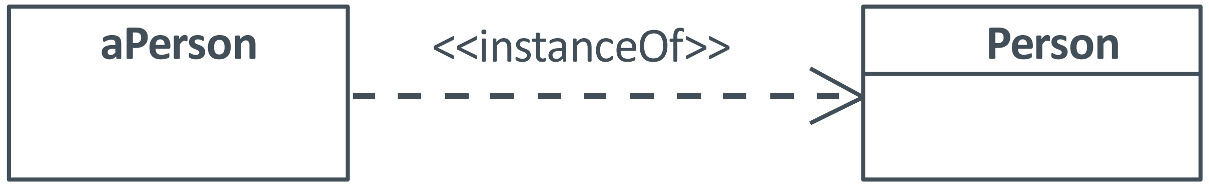
From model instances to meta models

- Canonical model-instance-relationship:



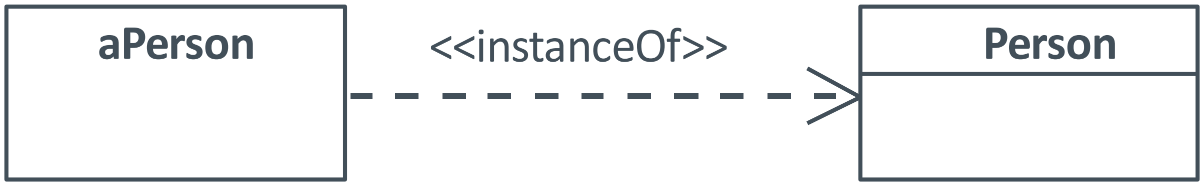
From model instances to meta models

- Canonical model-instance-relationship:



From model instances to meta models

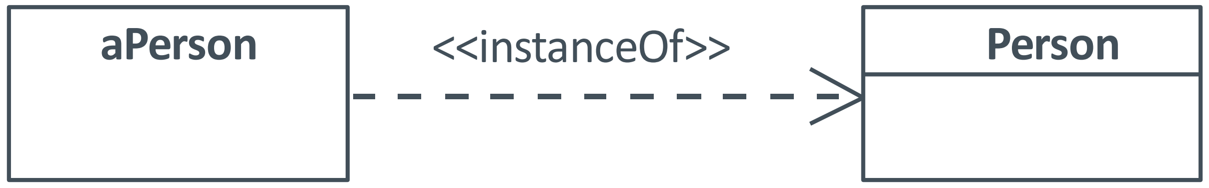
- Canonical model-instance-relationship:



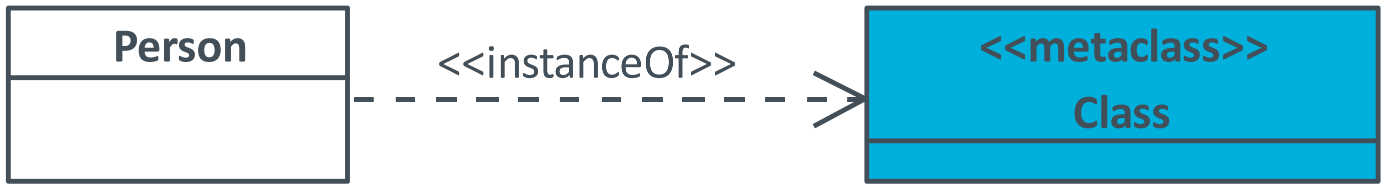
  – The instance aPerson and the class Person are on different levels of abstraction
  – The class Person specifies features that characterize the structure and behavior of *Persons*

  ➔The model for the class Person must characterize the structure and behavior of *classes*

From model instances to meta models

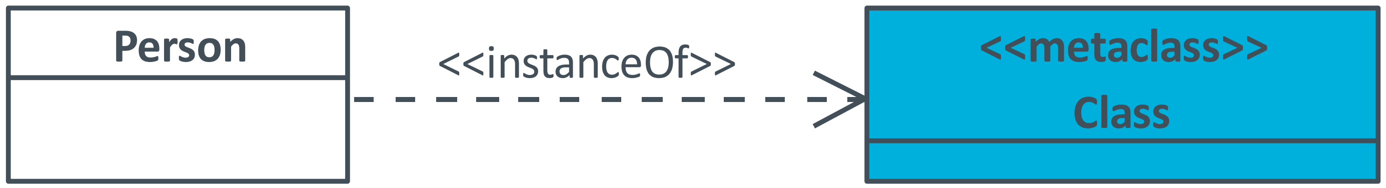


From model instances to meta models



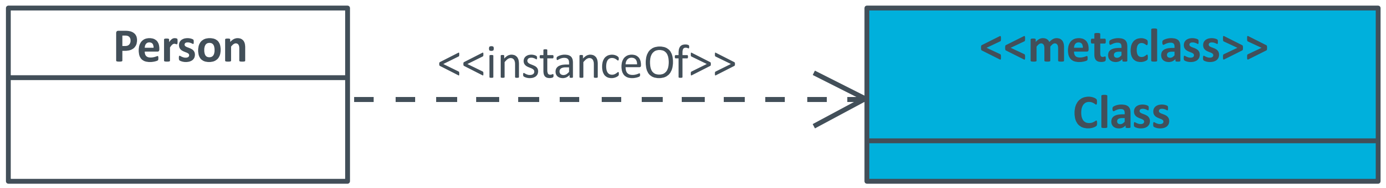
From model instances to meta models

- Relationship between model and meta-model:



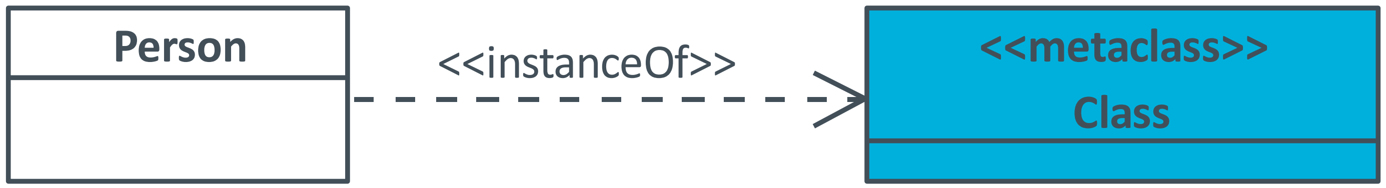
From model instances to meta models

- Relationship between model and meta-model:



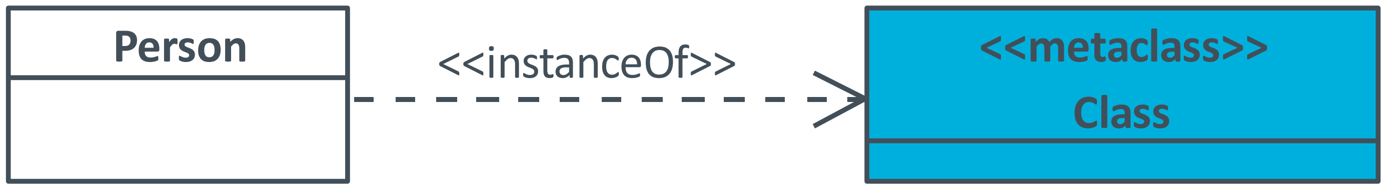
From model instances to meta models

- Relationship between model and meta-model:



From model instances to meta models

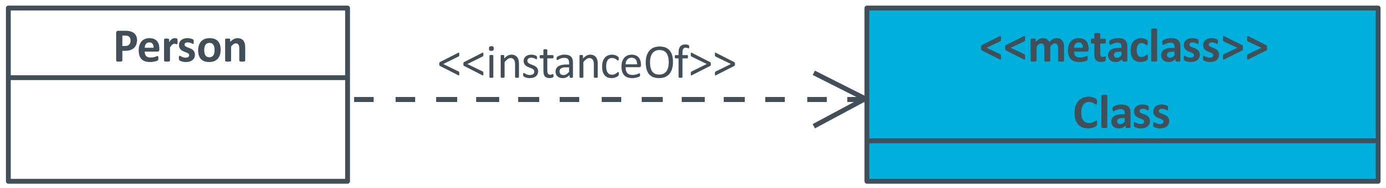
- Relationship between model and meta-model:



  – The meta class Class is a model for the class Person

From model instances to meta models

- Relationship between model and meta-model:



  - The meta class Class is a model for the class Person
  - Since Person is a model (for the instance aPerson), Class is a meta model (model for models)

From model instances to meta models

From model instances to meta models

At first this might be confusing, so:

From model instances to meta models

At first this might be confusing, so:

- Think of the different layers of abstraction:

From model instances to meta models

At first this might be confusing, so:

- Think of the different layers of abstraction:
  - Instances are concrete

From model instances to meta models

At first this might be confusing, so:

- Think of the different layers of abstraction:
    - Instances are concrete
    - Models are an abstract description of the instances

From model instances to meta models

At first this might be confusing, so:

- Think of the different layers of abstraction:
  - Instances are concrete
  - Models are an abstract description of the instances
  - Meta models are an abstract description of models
  - …

Meta models

Meta models

Why do we need them?

Meta models

Why do we need them?

- Meta models can be used for instance to formalize UML notations:

Meta models

Why do we need them?

- Meta models can be used for instance to formalize UML notations:
  - The UML is a Language, meta models are used to describe the grammar

Meta models

Why do we need them?

- Meta models can be used for instance to formalize UML notations:
  - The UML is a Language, meta models are used to describe the grammar
  - The UML meta model describes all models one can create using UML

Meta models

Why do we need them?

- Meta models can be used for instance to formalize UML notations:
  - The UML is a Language, meta models are used to describe the grammar
  - The UML meta model describes all models one can create using UML
  - The meta model allows to talk about semantics

Meta models

Meta models

OK, so meta models are cool, but how do I create a meta model?

Meta models

OK, so meta models are cool, but how do I create a meta model?

To approach this problem, we will look at the history of UML's meta model first.

The history of UML’s meta model

The history of UML's meta model

- First there was UML which had semantic problems

The history of UML's meta model

- First there was UML which had semantic problems
- The OMG tried to formalize meta models

The history of UML’s meta model

- First there was UML which had semantic problems
- The OMG tried to formalize meta models
- They realized that all they needed to describe meta models was a subset of UML class diagram elements

The history of UML's meta model

- First there was UML which had semantic problems
- The OMG tried to formalize meta models
- They realized that all they needed to describe meta models was a subset of UML class diagram elements

➔ To describe any meta model, we can use the UML class diagram notation!

Meta Object Facility (MOF)

Meta Object Facility (MOF)

- The OMG introduced the MOF to create a common approach to meta modeling

Meta Object Facility (MOF)

- The OMG introduced the MOF to create a common approach to meta modeling
- A meta model which is defined using MOF is called “MOF compliant”

Advantages of MOF compliant meta models

- They can easily be compared
- Their instances (models) can be exchanged in a standardized way (XML Metadata Interchange)
- Their instances can live in the same metadata repository (data warehousing)

Meta Object Facility (MOF) - Facts

Meta Object Facility (MOF) - Facts

In general, the sequence

instance → *model* → *meta model* → *meta-meta model* →…

could be continued infinitely.

Meta Object Facility (MOF) - Facts

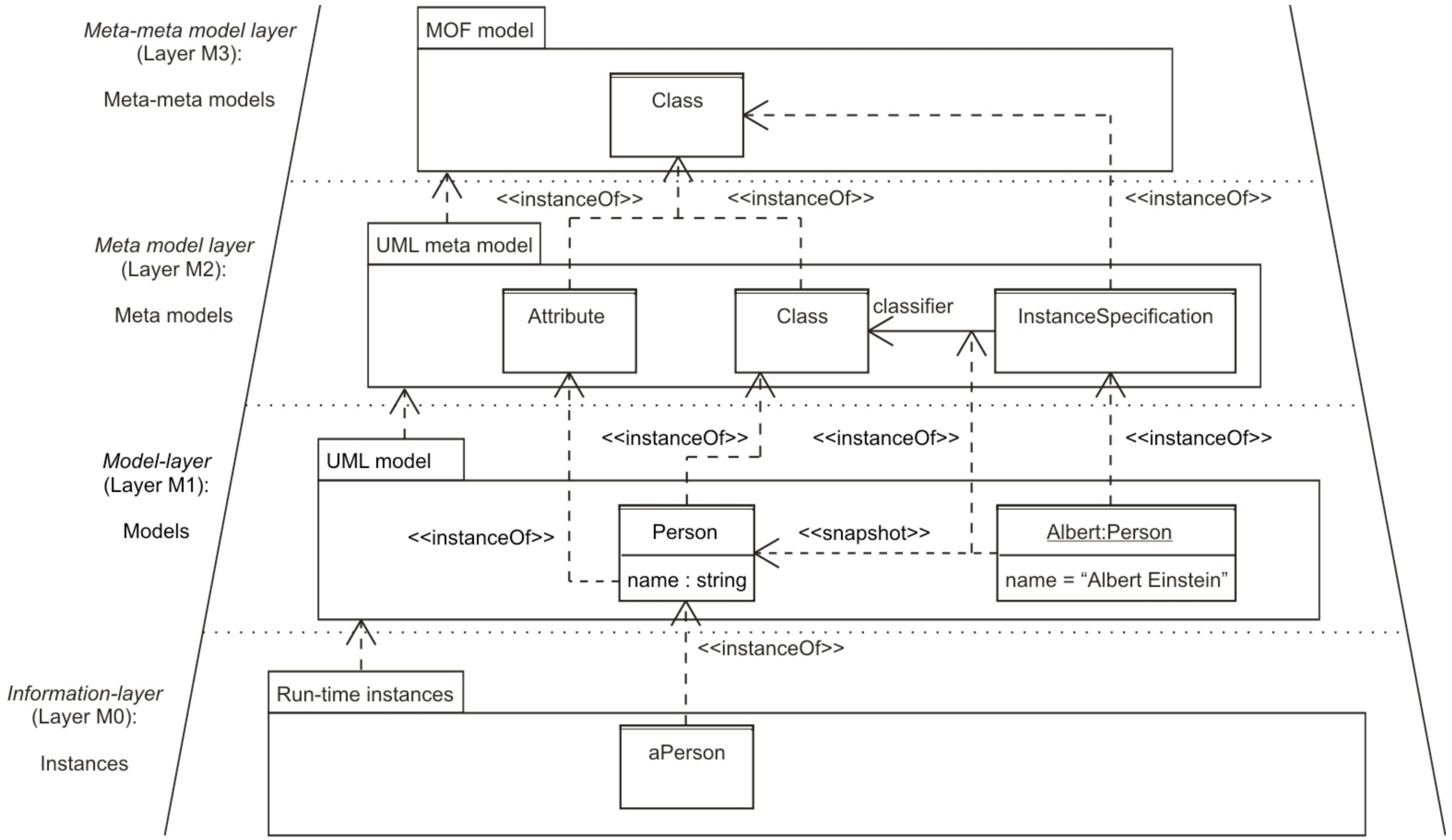
In general, the sequence

instance → *model* → *meta model* → *meta-meta model* →…

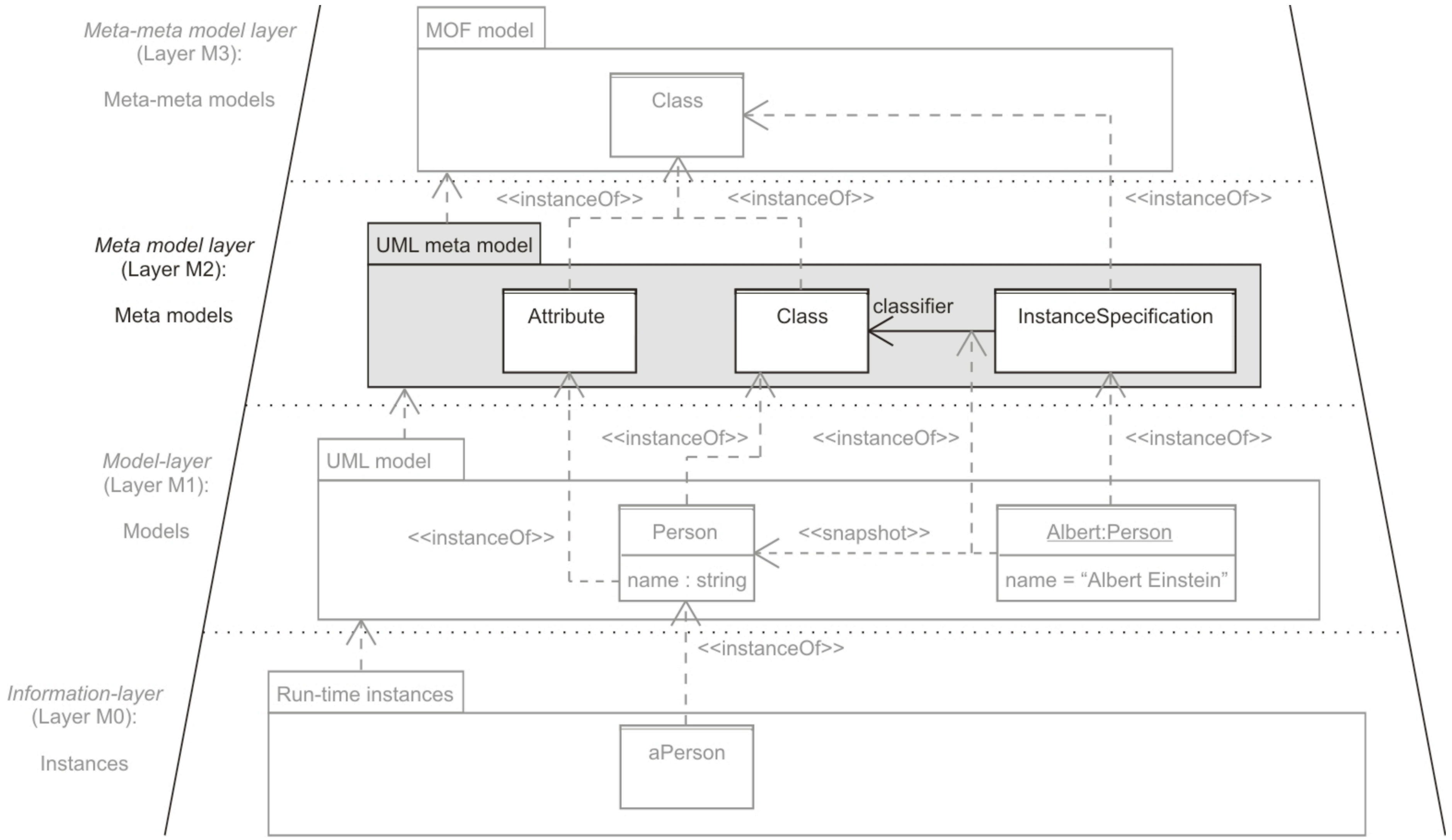
could be continued infinitely.

- MOF defines a four-layer meta model hierarchy
  - four layers suffice for most practical applications
- MOF and UML are aligned
  - UML infrastructure contains concepts for UML *and* MOF

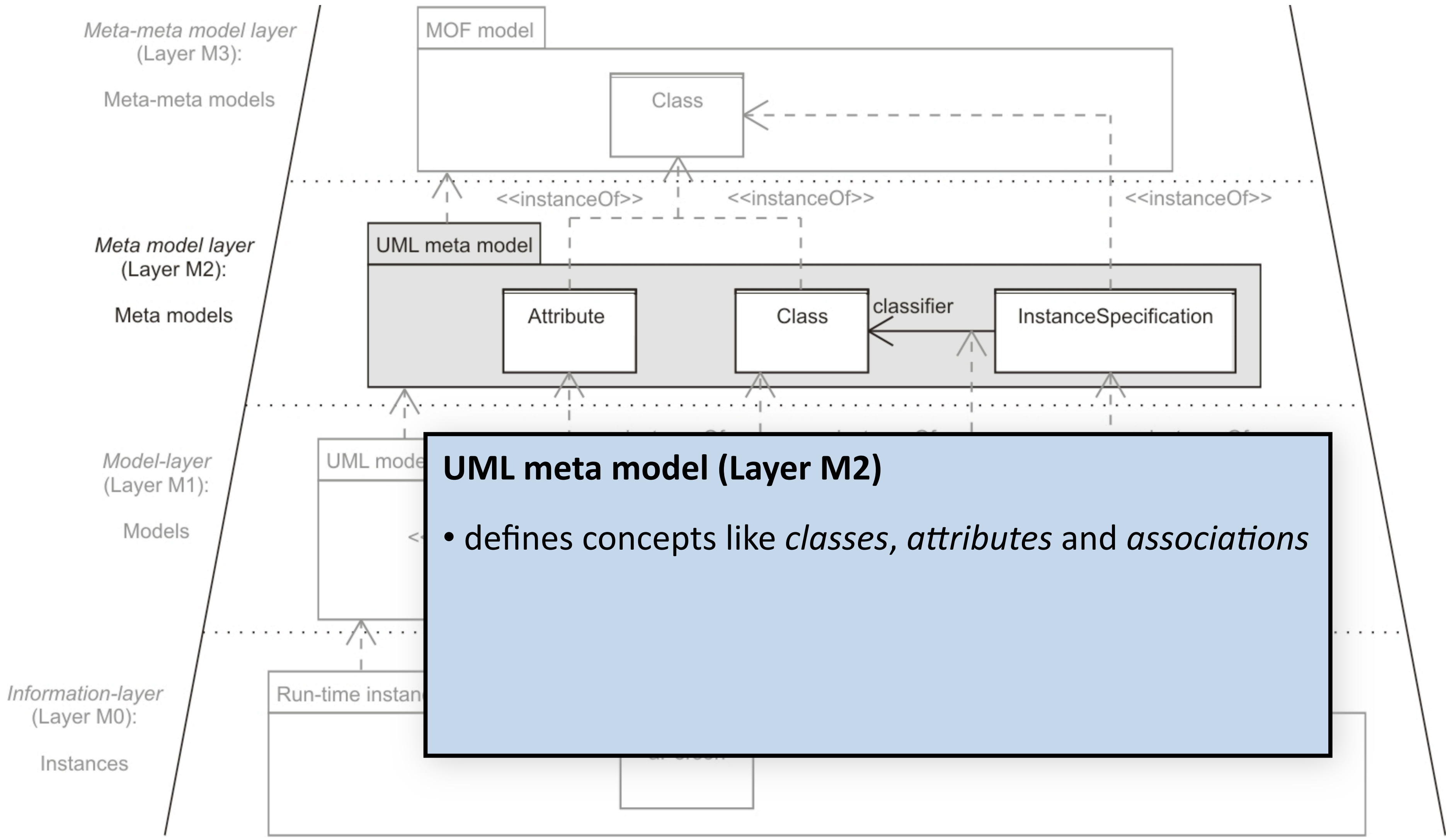
Meta model hierarchy of the MOF (UML-specific)



Meta model hierarchy of the MOF (UML-specific)



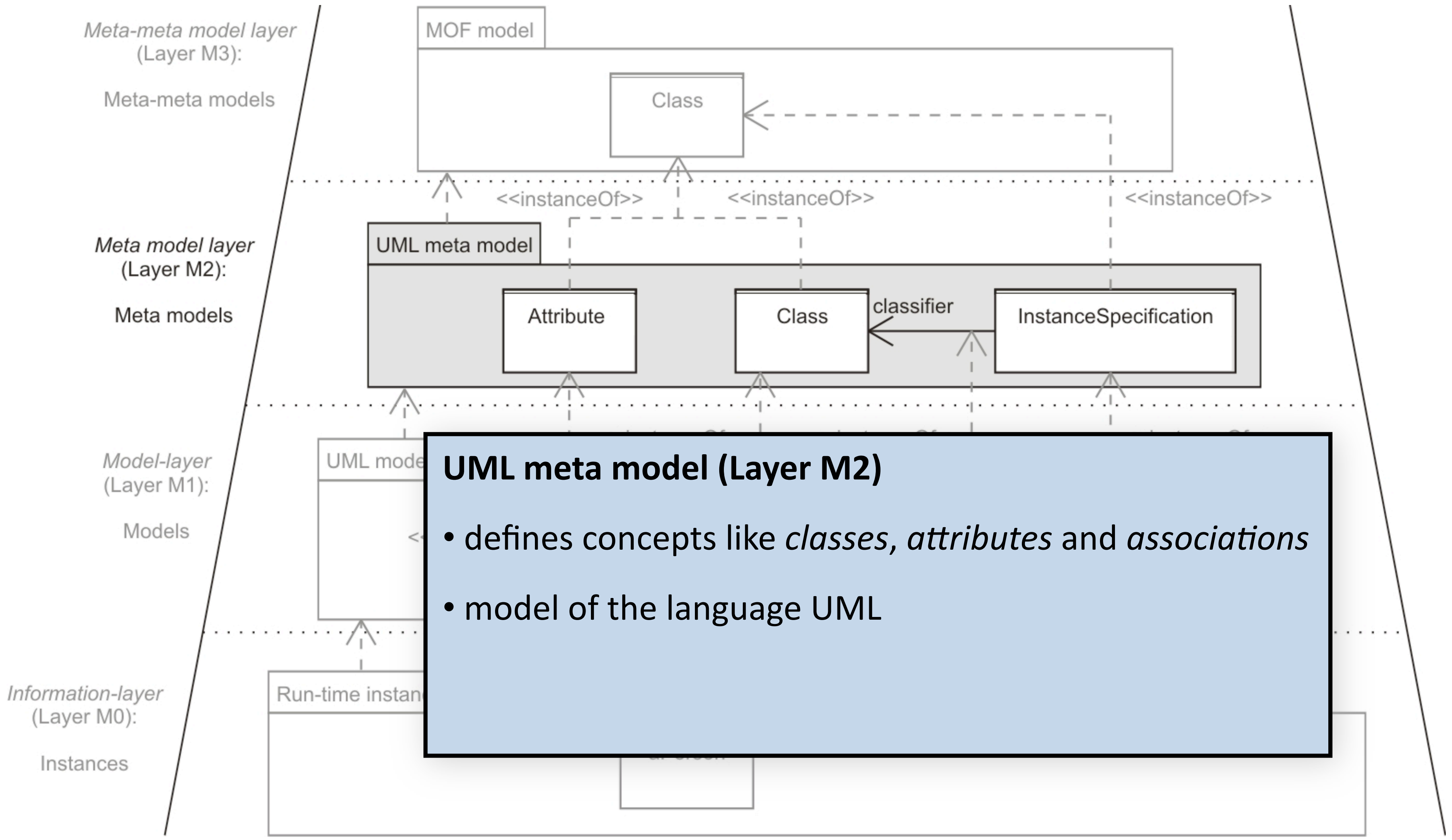
Meta model hierarchy of the MOF (UML-specific)



UML meta model (Layer M2)

• defines concepts like *classes*, *attributes* and *associations*

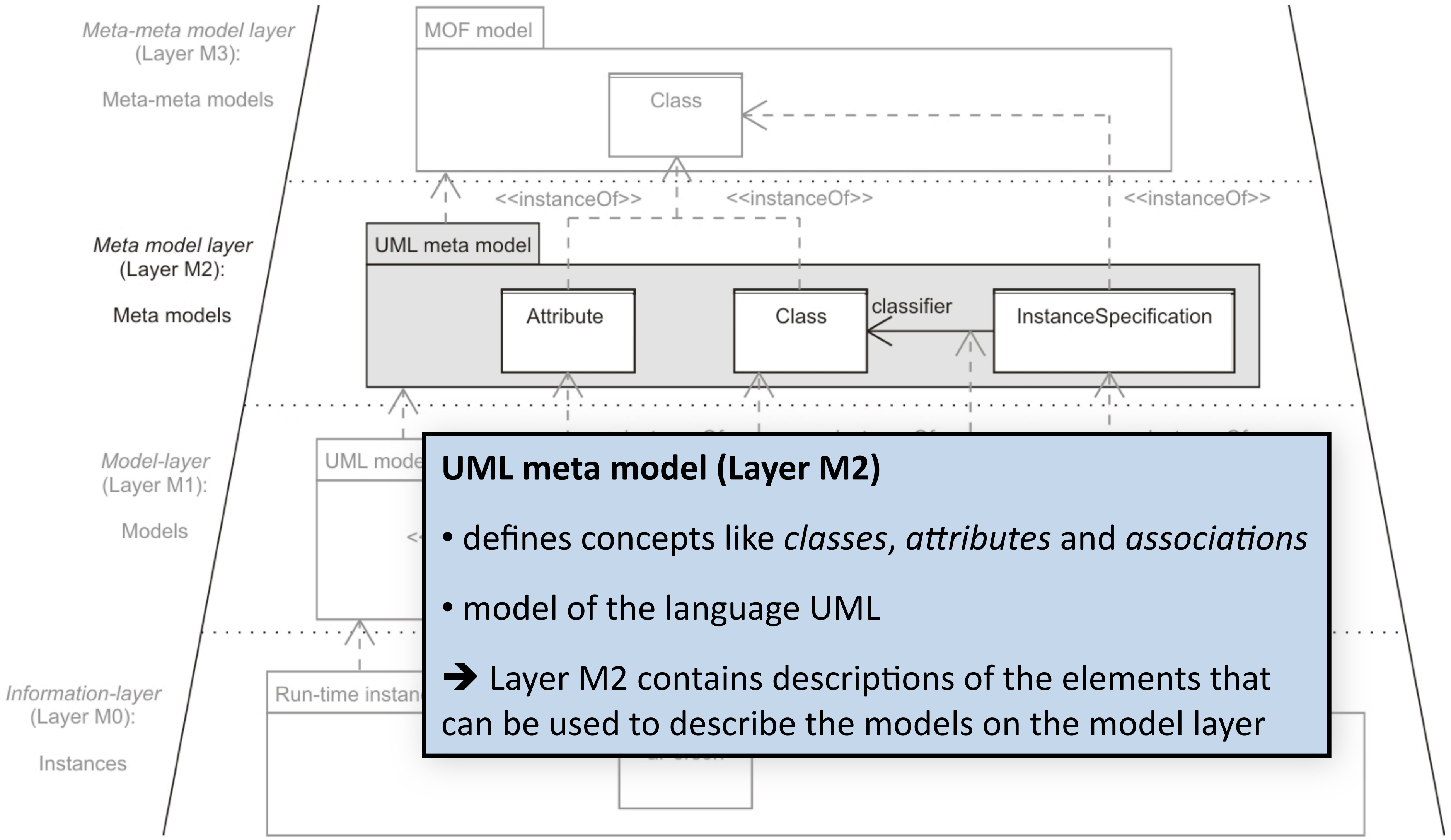
Meta model hierarchy of the MOF (UML-specific)



UML meta model (Layer M2)

- defines concepts like *classes*, *attributes* and *associations*
- model of the language UML

Meta model hierarchy of the MOF (UML-specific)



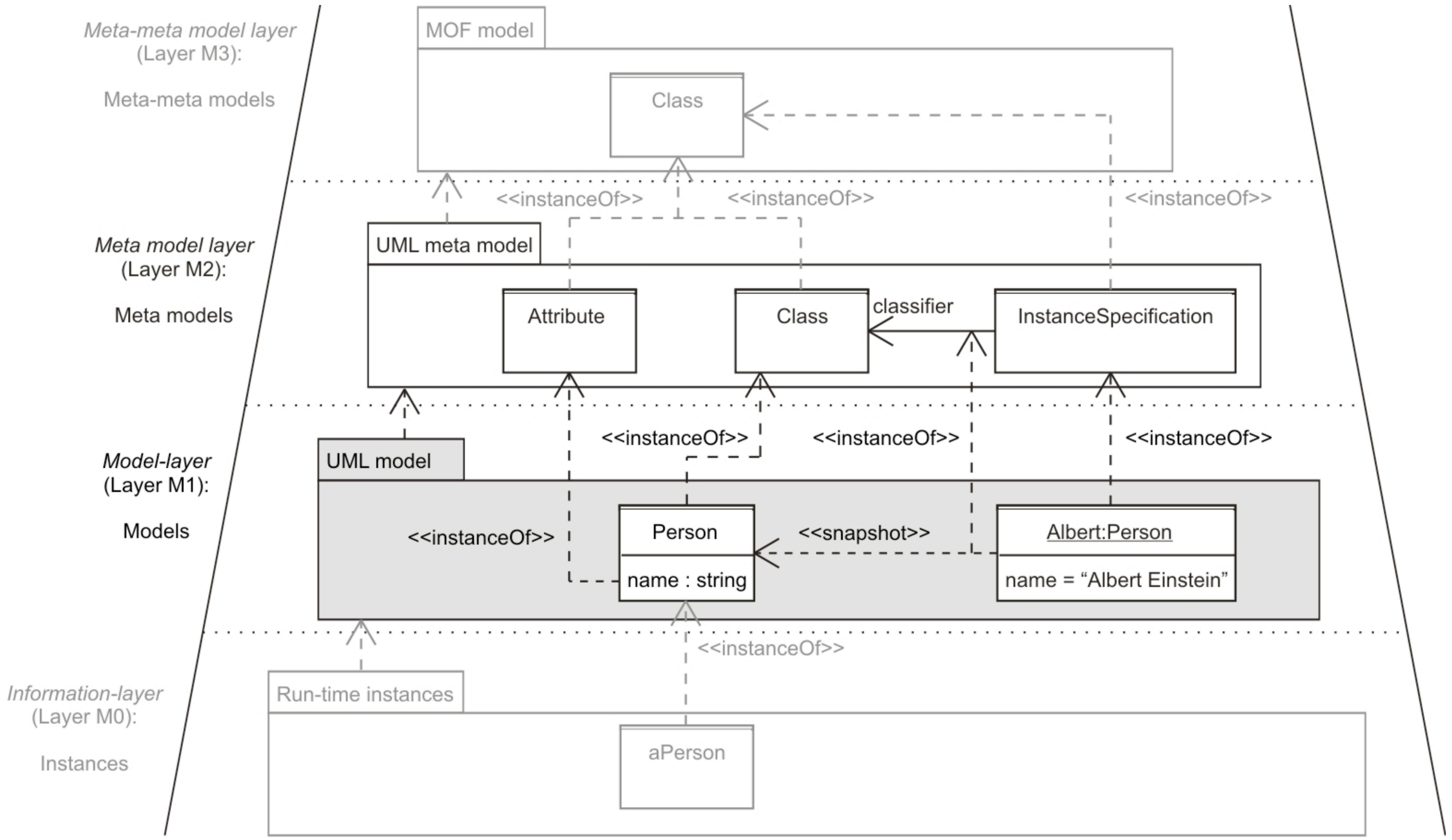
UML meta model (Layer M2)

• defines concepts like *classes*, *attributes* and *associations*

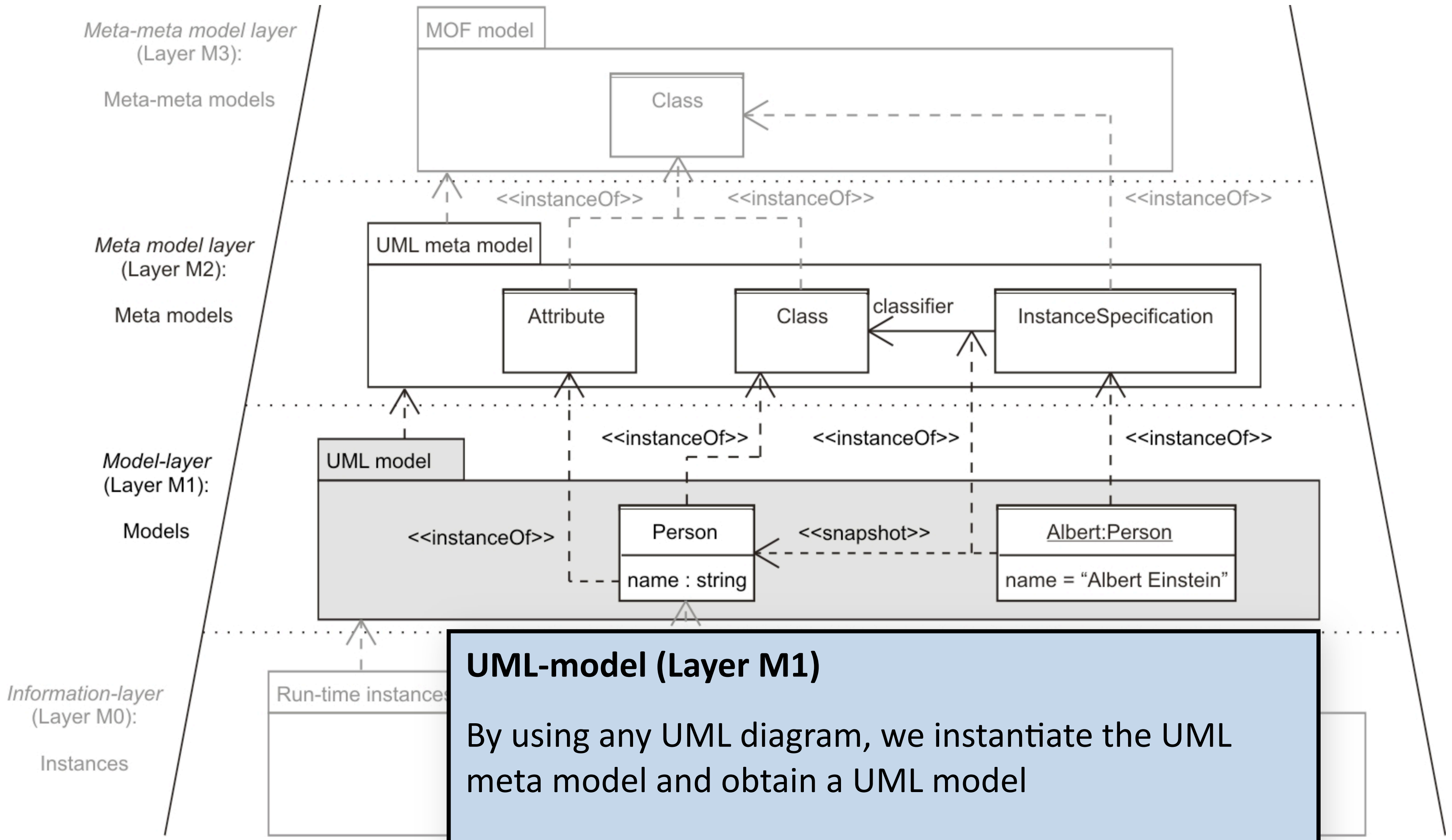
• model of the language UML

➔ Layer M2 contains descriptions of the elements that can be used to describe the models on the model layer

Meta model hierarchy of the MOF (UML-specific)



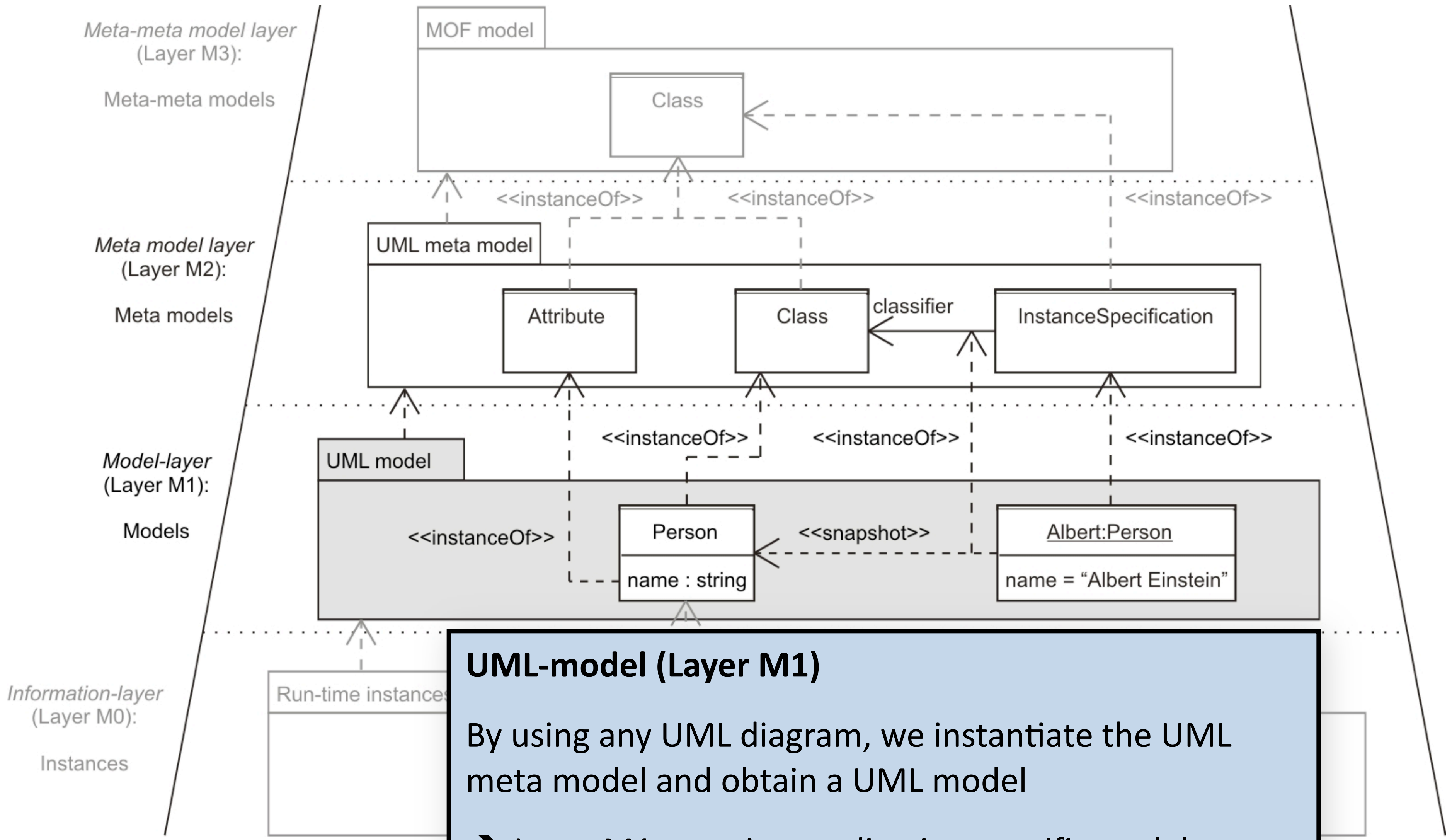
Meta model hierarchy of the MOF (UML-specific)



UML-model (Layer M1)

By using any UML diagram, we instantiate the UML meta model and obtain a UML model

Meta model hierarchy of the MOF (UML-specific)

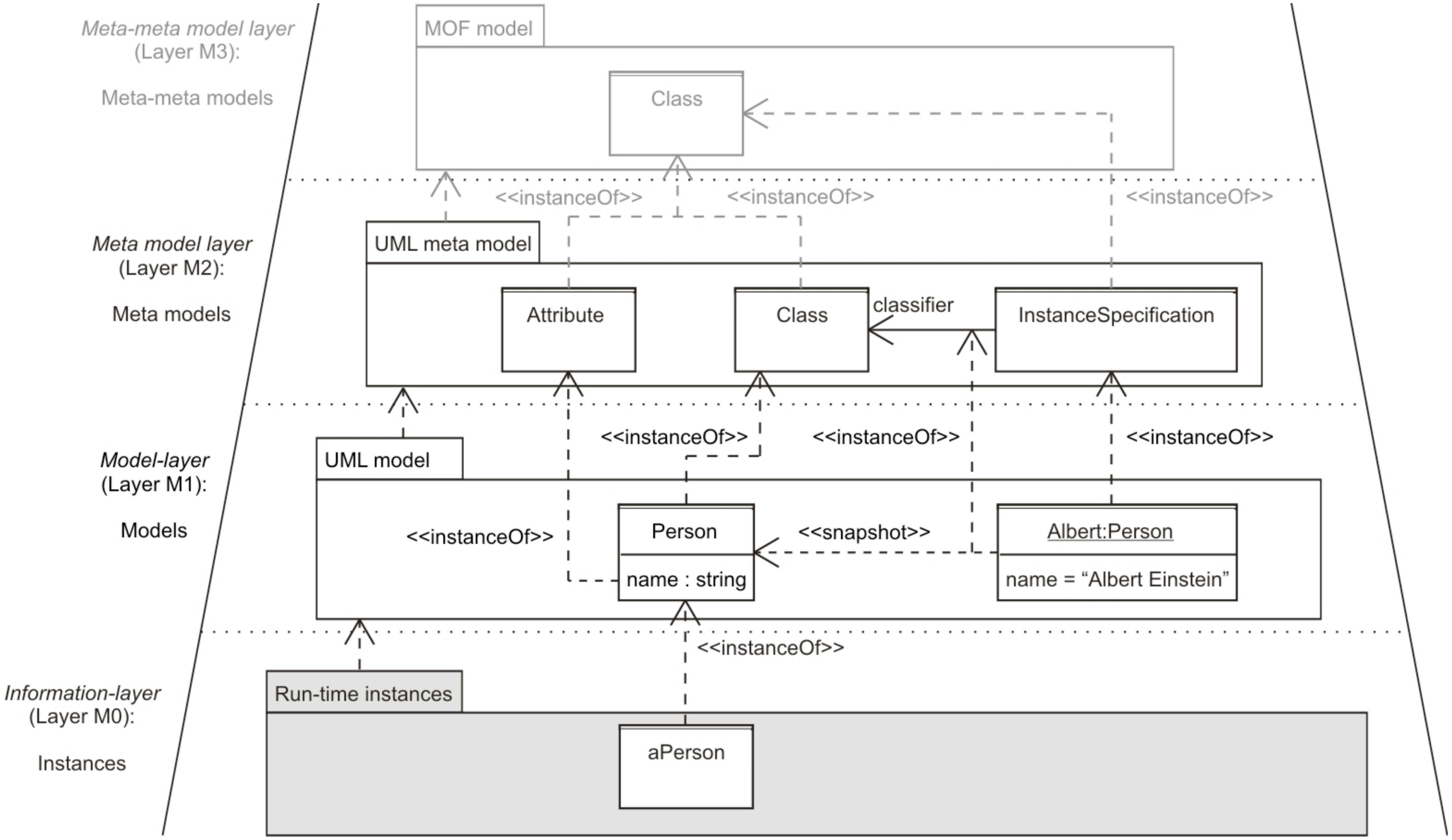


UML-model (Layer M1)

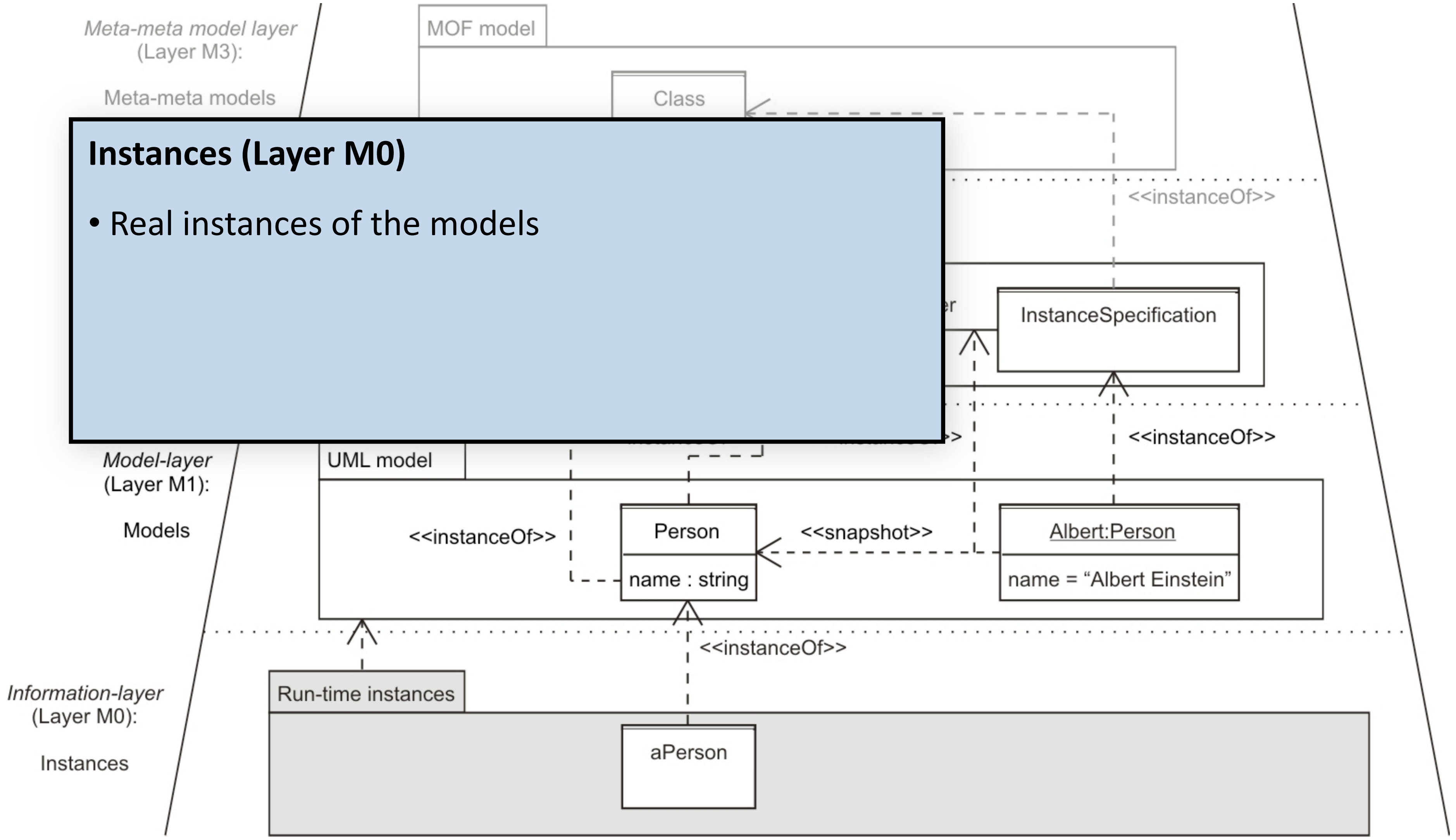
By using any UML diagram, we instantiate the UML meta model and obtain a UML model

➔ Layer M1 contains *application-specific* models

Meta model hierarchy of the MOF (UML-specific)



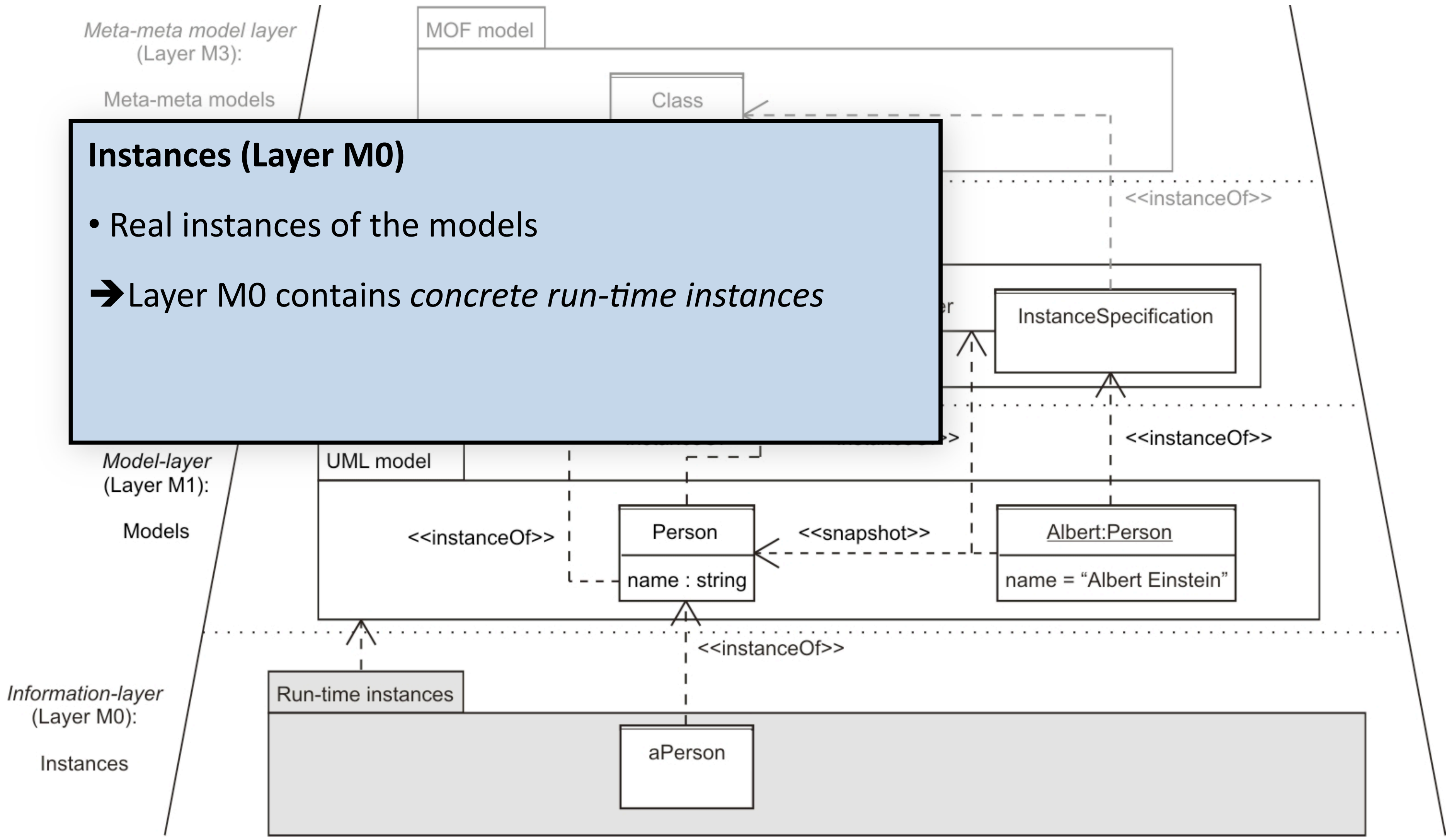
Meta model hierarchy of the MOF (UML-specific)



Instances (Layer M0)

- Real instances of the models

Meta model hierarchy of the MOF (UML-specific)

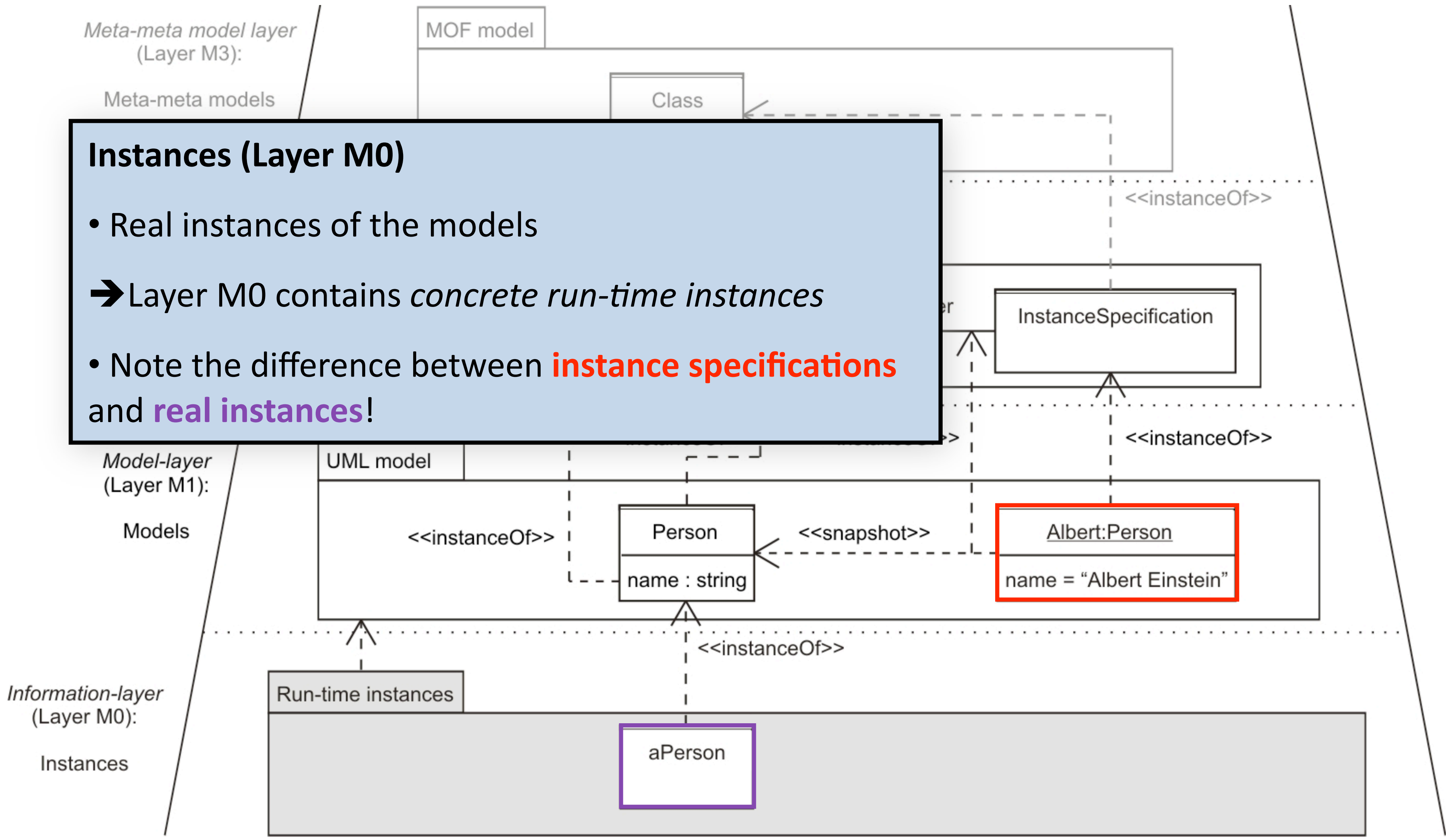


Instances (Layer M0)

- Real instances of the models

➔Layer M0 contains *concrete run-time instances*

Meta model hierarchy of the MOF (UML-specific)



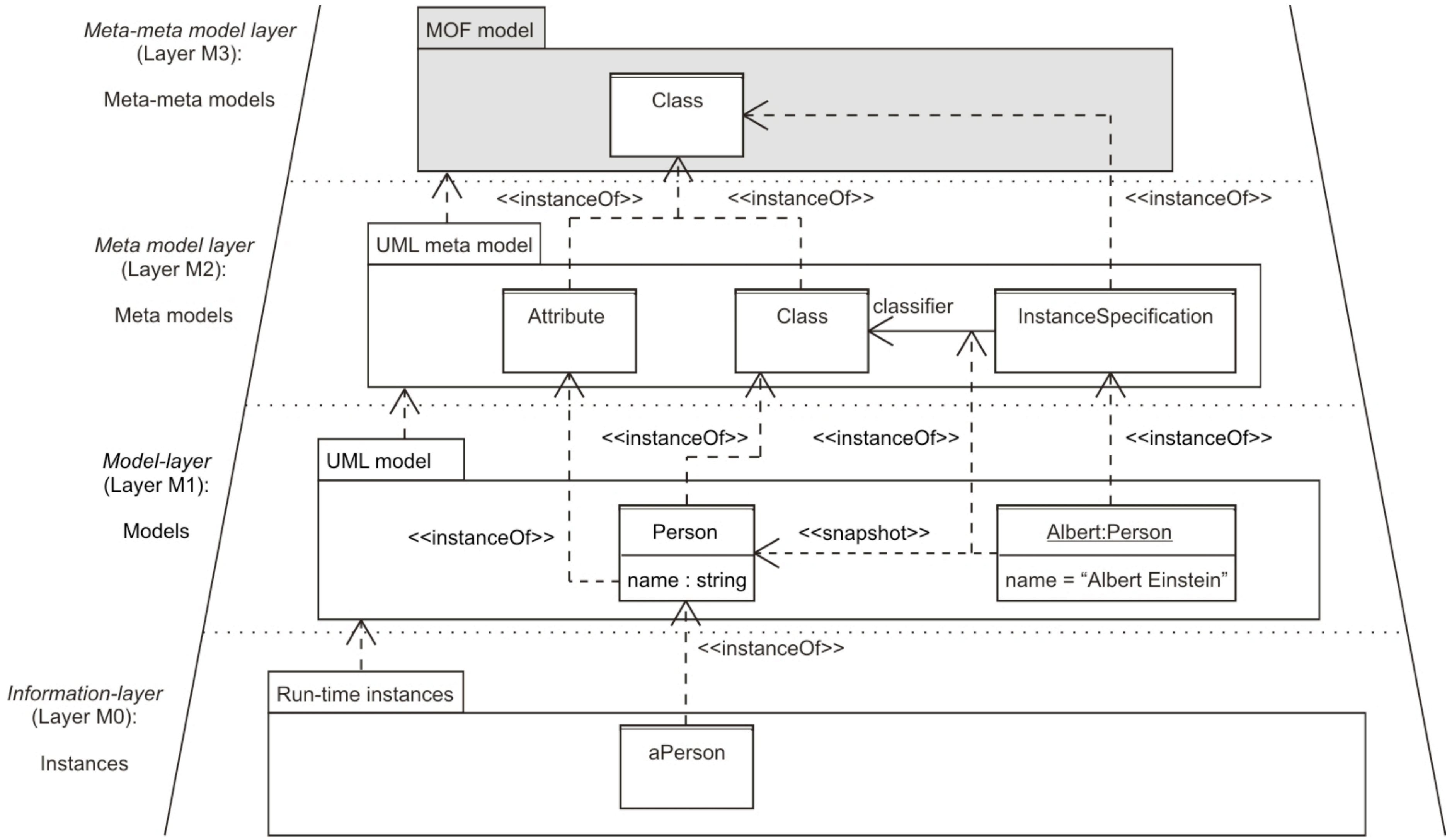
Instances (Layer M0)

- Real instances of the models

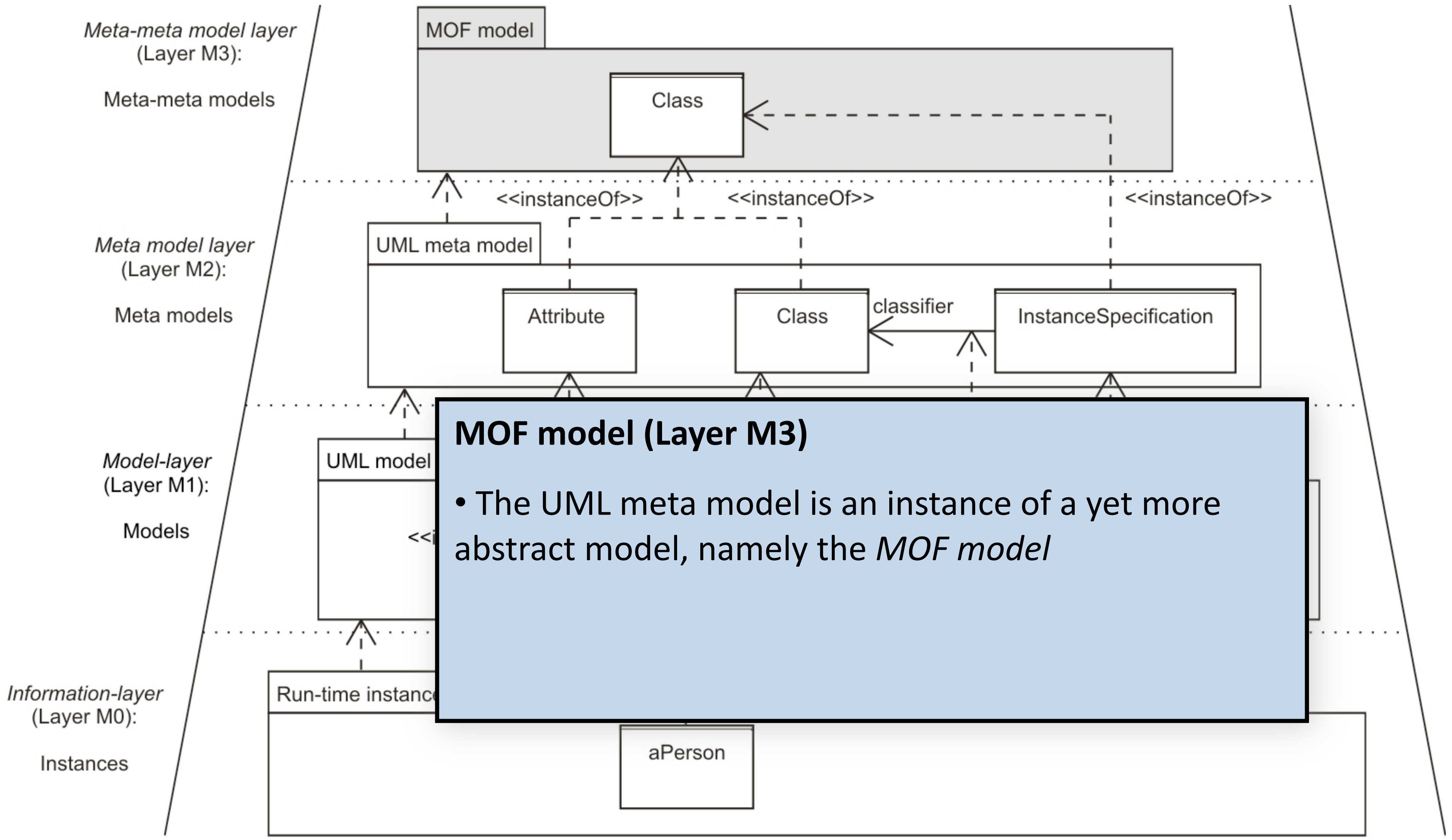
➔Layer M0 contains *concrete run-time instances*

- Note the difference between **instance specifications** and **real instances**!

Meta model hierarchy of the MOF (UML-specific)



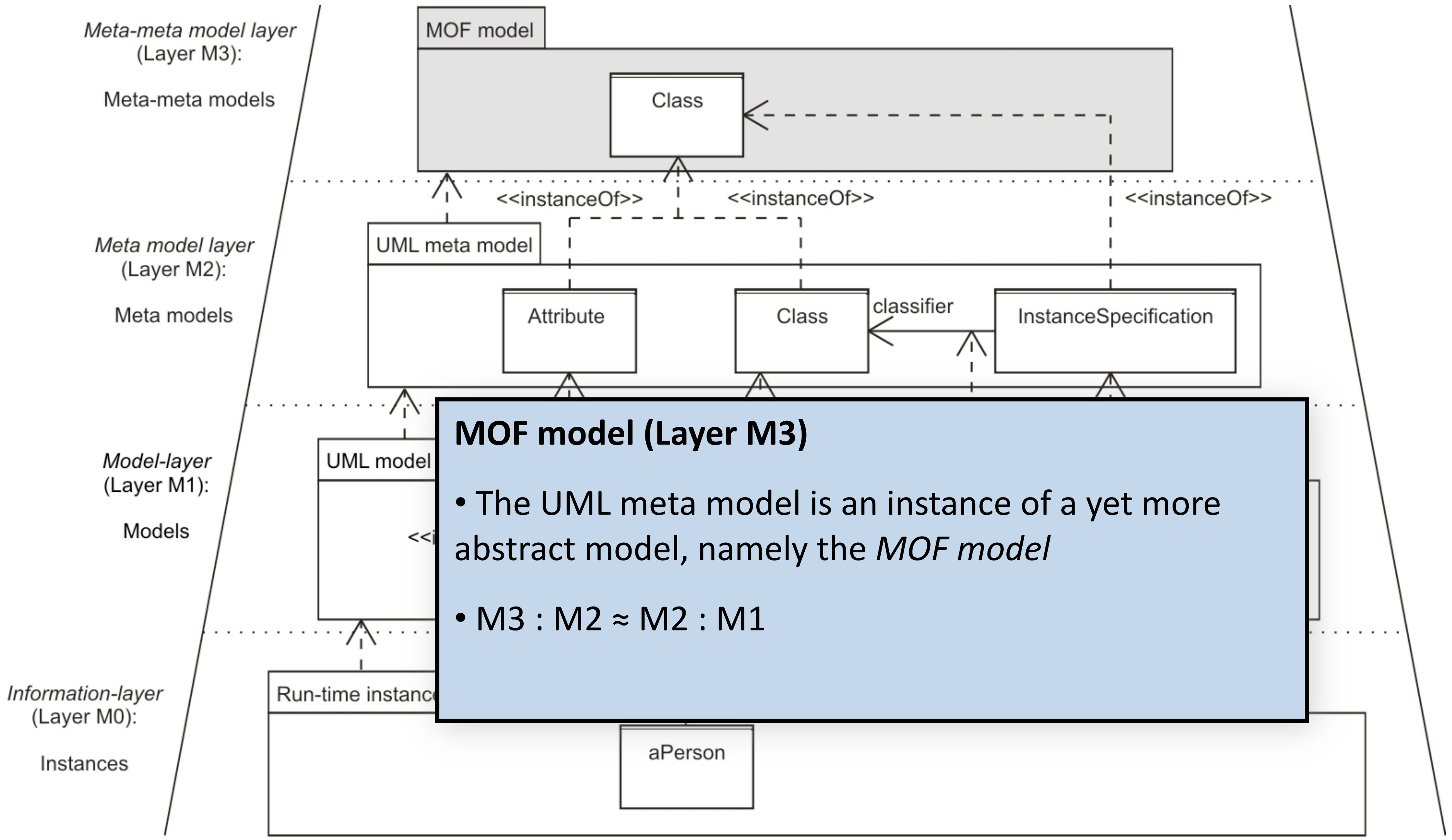
Meta model hierarchy of the MOF (UML-specific)



MOF model (Layer M3)

• The UML meta model is an instance of a yet more abstract model, namely the *MOF model*

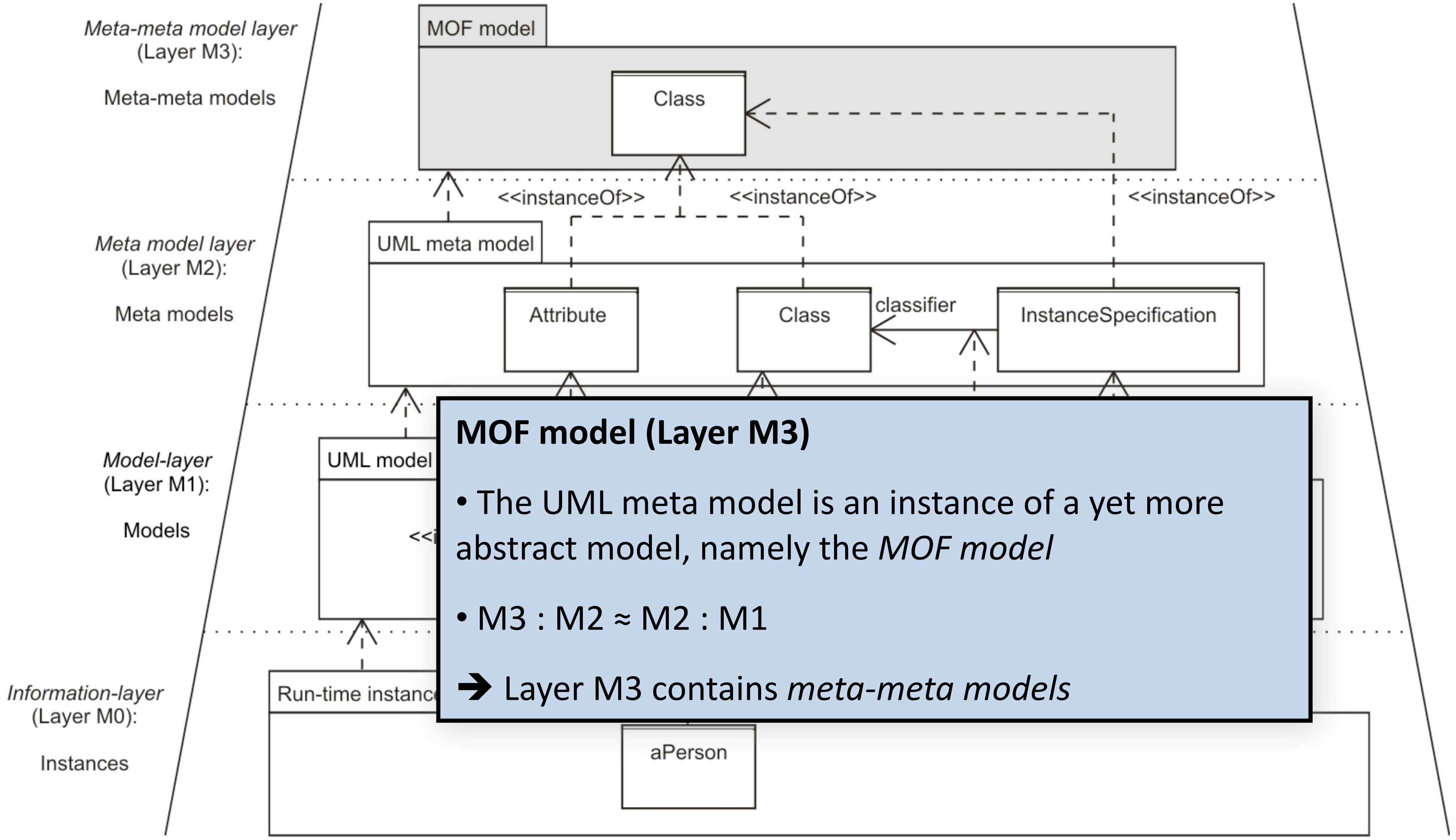
Meta model hierarchy of the MOF (UML-specific)



MOF model (Layer M3)

- The UML meta model is an instance of a yet more abstract model, namely the *MOF model*
- M3 : M2 ≈ M2 : M1

Meta model hierarchy of the MOF (UML-specific)



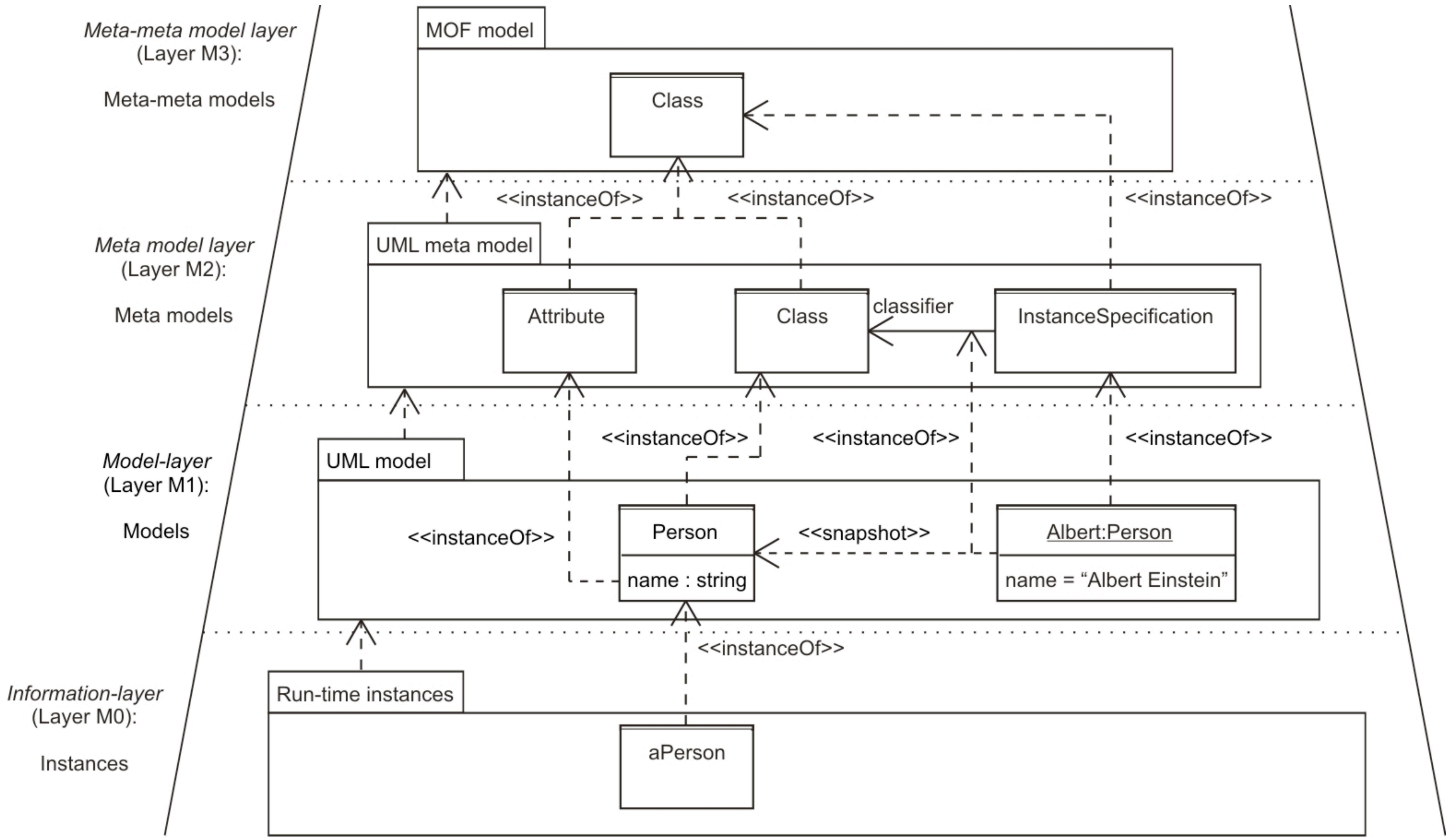
MOF model (Layer M3)

• The UML meta model is an instance of a yet more abstract model, namely the *MOF model*

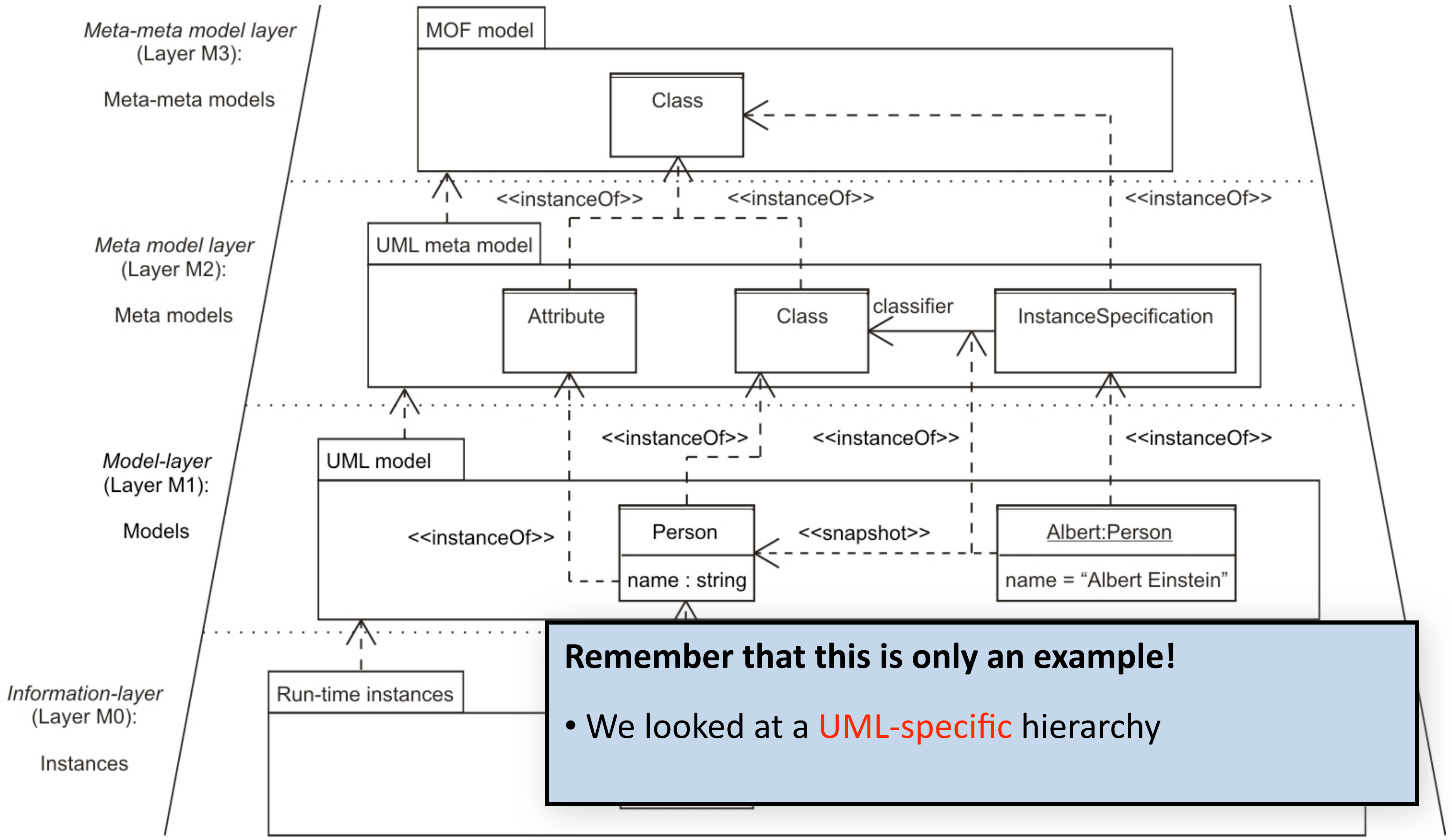
• M3 : M2 ≈ M2 : M1

➔ Layer M3 contains *meta-meta models*

Meta model hierarchy of the MOF (UML-specific)



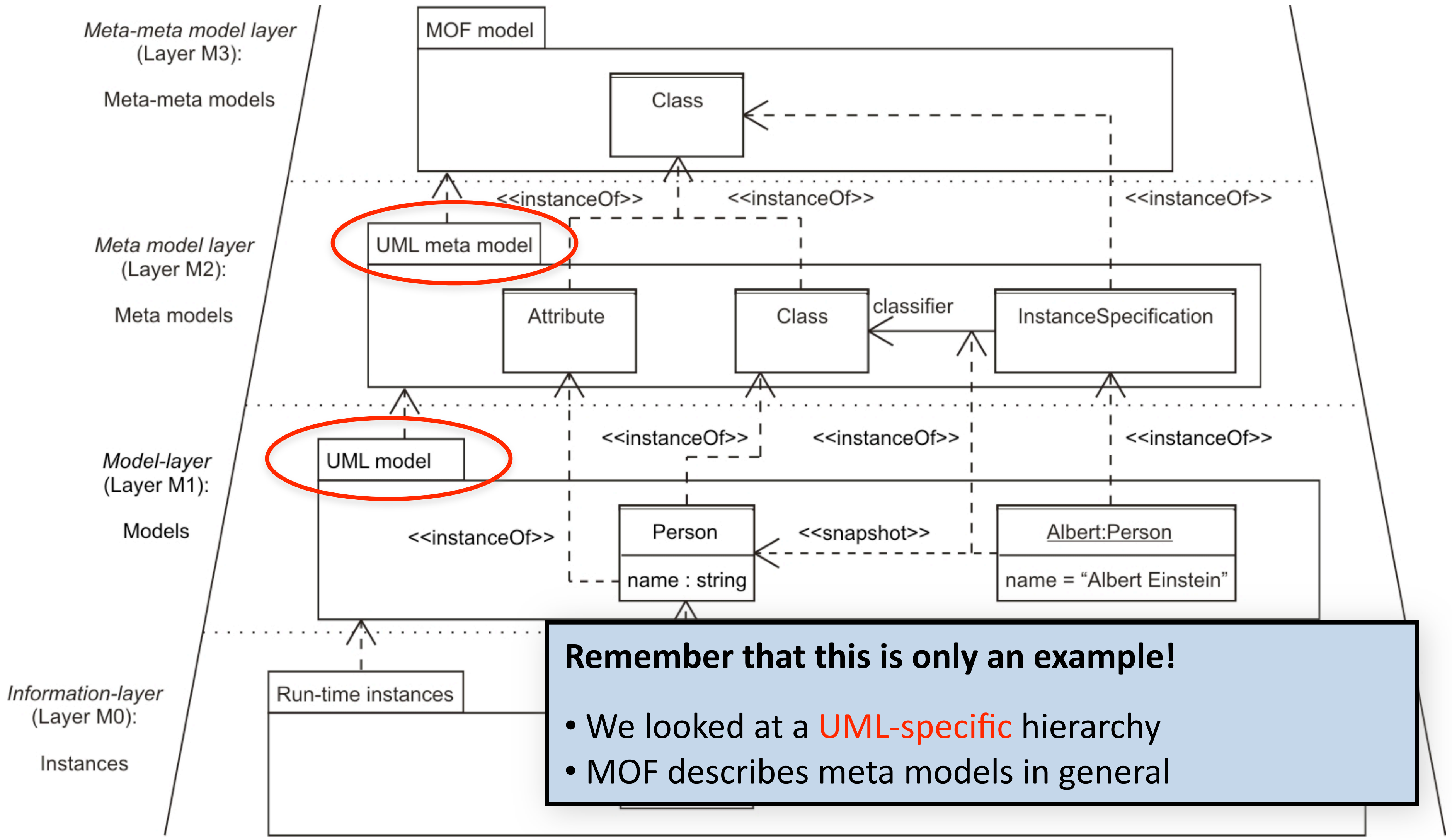
Meta model hierarchy of the MOF (UML-specific)



Remember that this is only an example!

- We looked at a UML-specific hierarchy

Meta model hierarchy of the MOF (UML-specific)



Remember that this is only an example!

- We looked at a UML-specific hierarchy
- MOF describes meta models in general

Where are we?

- ✓ From model instances to meta models
- ✓ MOF meta model hierarchy
- ➢ How UML relates to MOF
  - – Example: Use case diagram meta model
  - – Example: Class diagram meta model
- • Different notations for the UML meta model describe the same language
- • UML Profiles: Adding new members to the family

How UML relates to MOF

How UML relates to MOF

- UML is MOF-compliant:

  The UML meta model is an instance of the MOF model

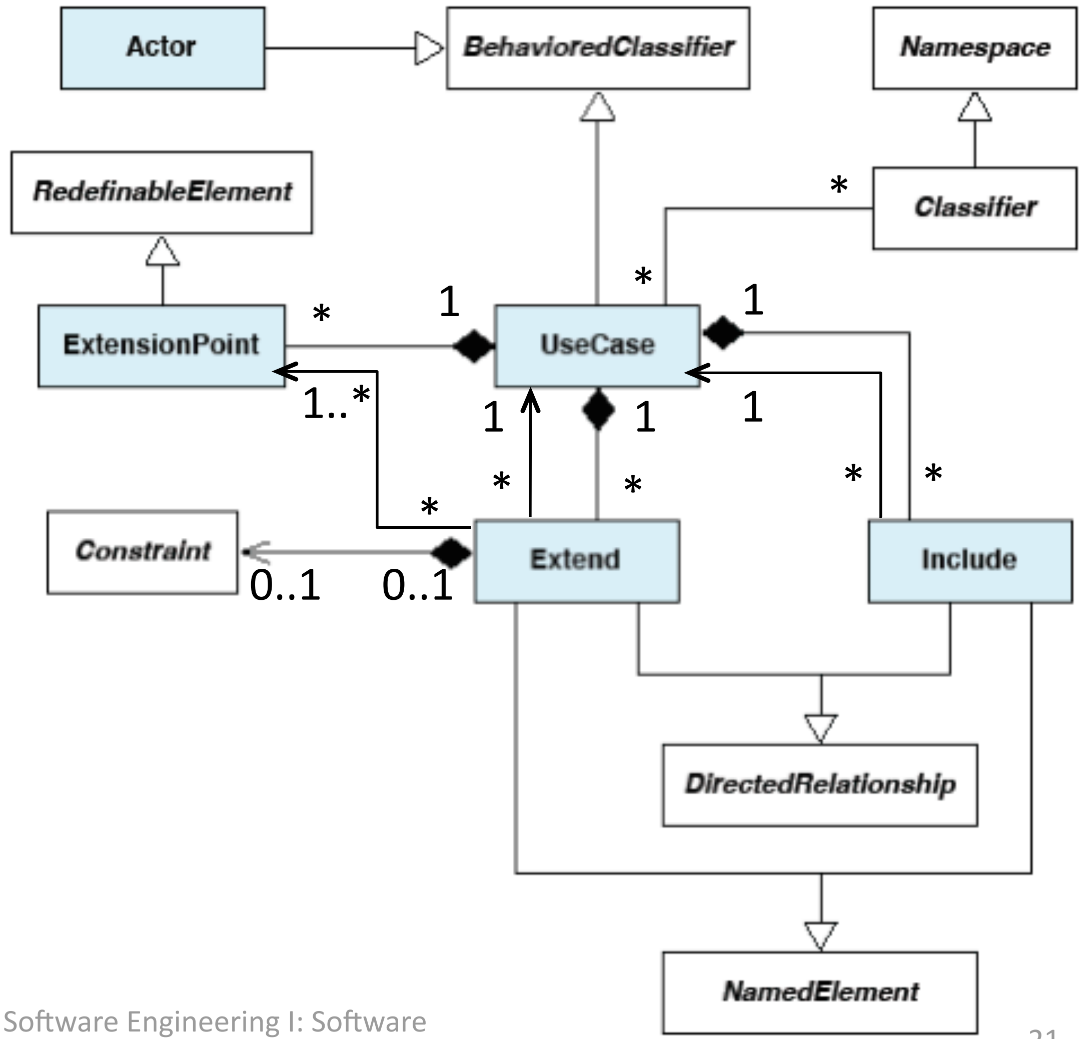
How UML relates to MOF

- UML is MOF-compliant:

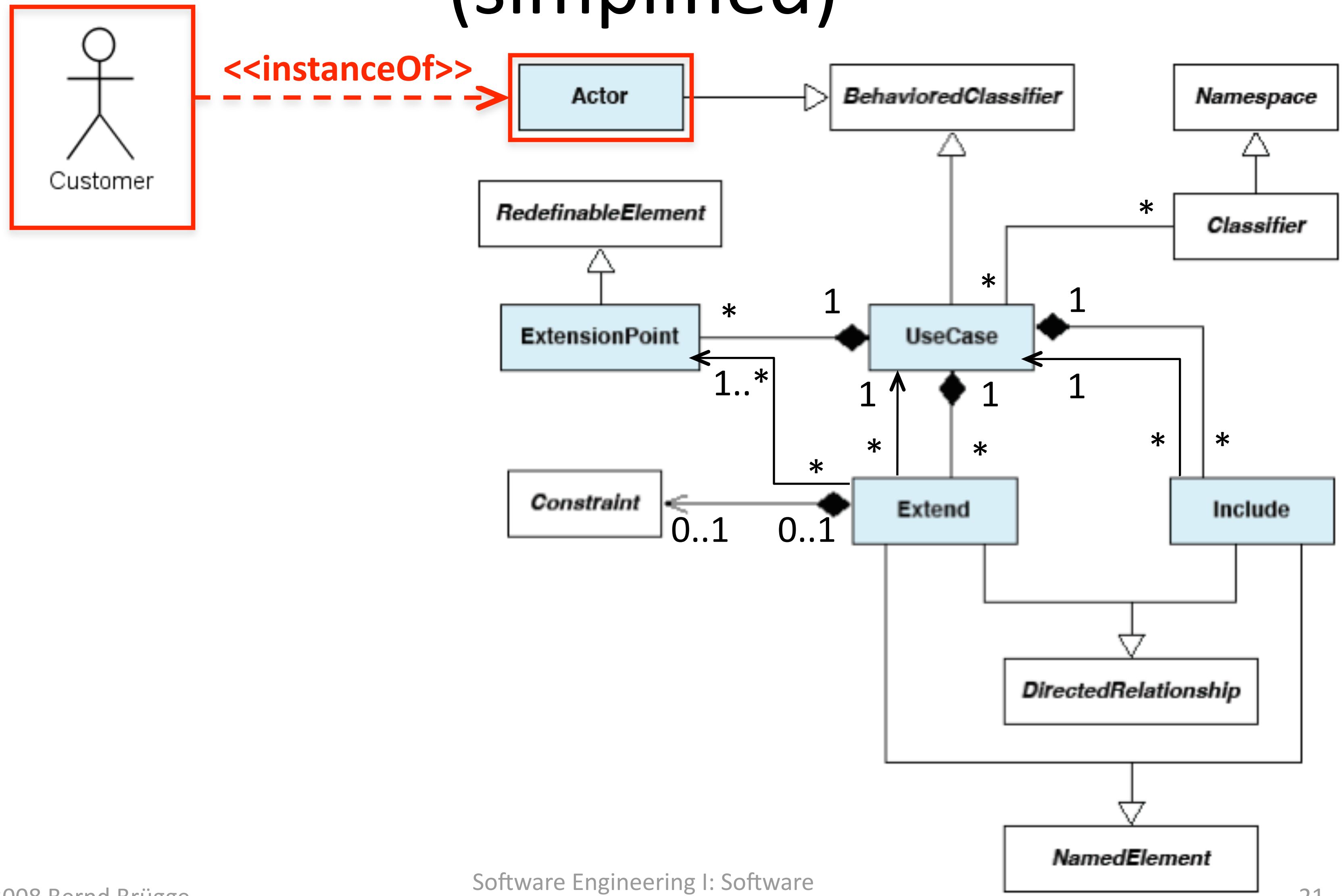
  The UML meta model is an instance of the MOF model

- Let’s see the UML meta model in action!

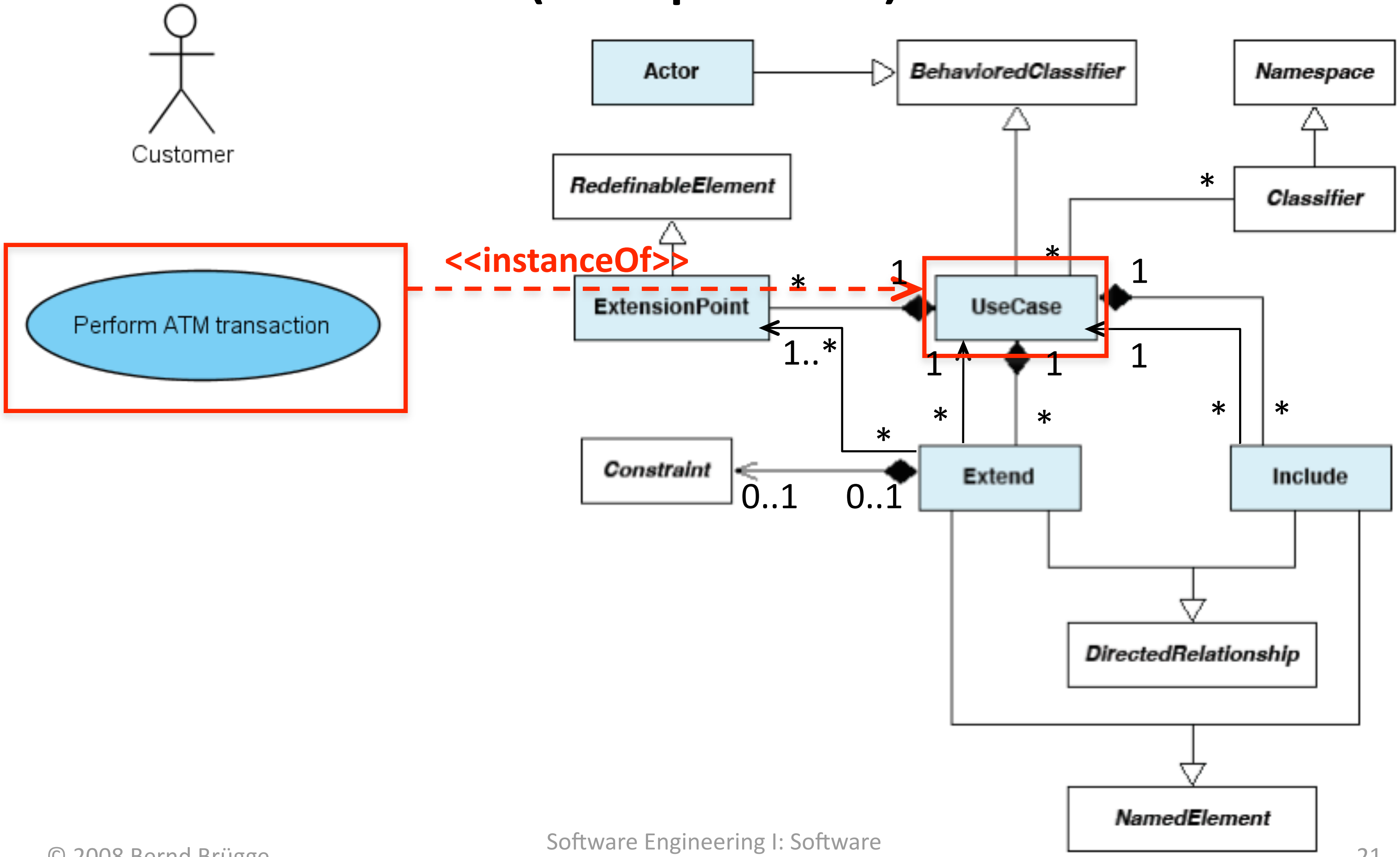
Use Case Diagram Meta Model (simplified)



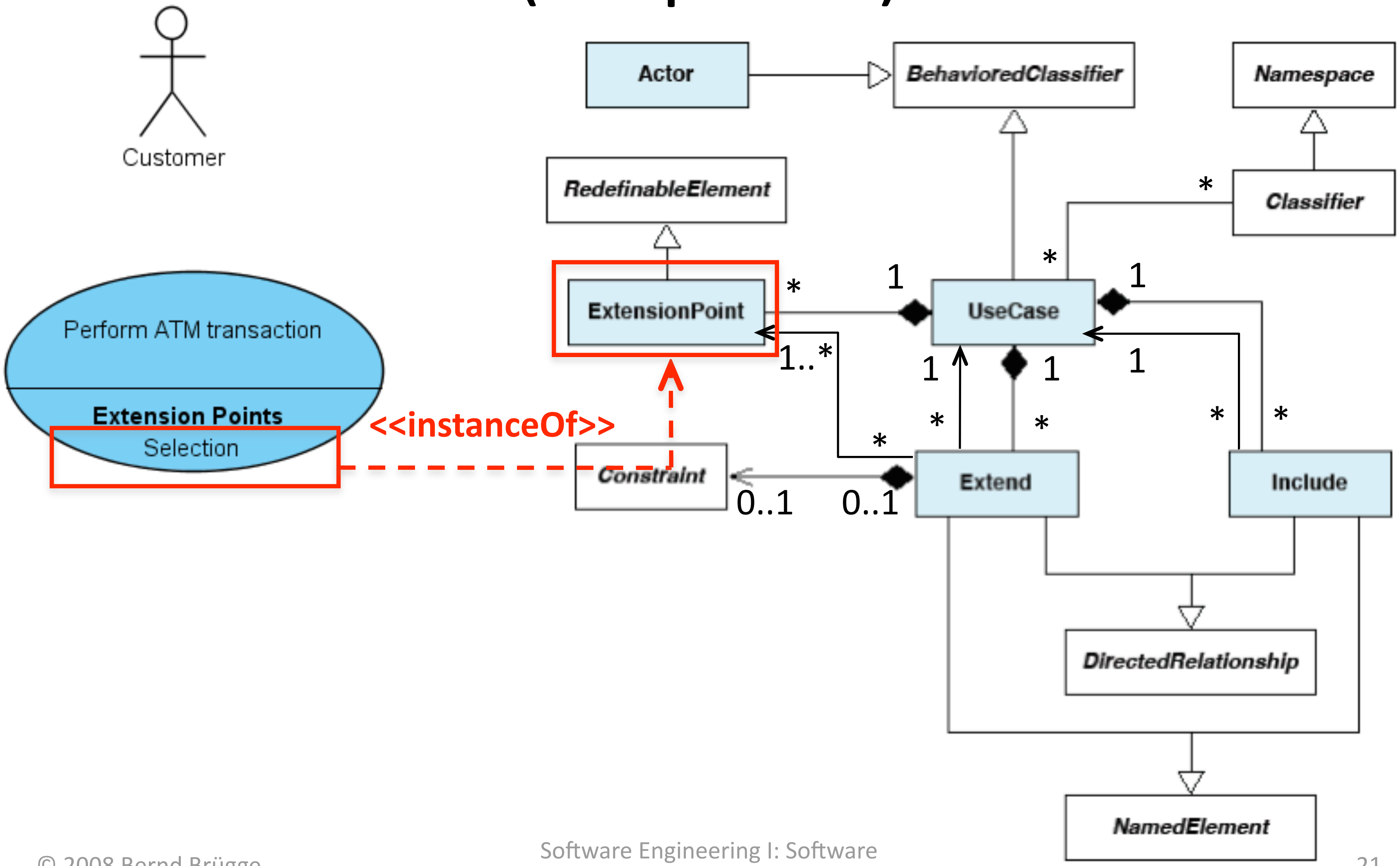
Use Case Diagram Meta Model (simplified)



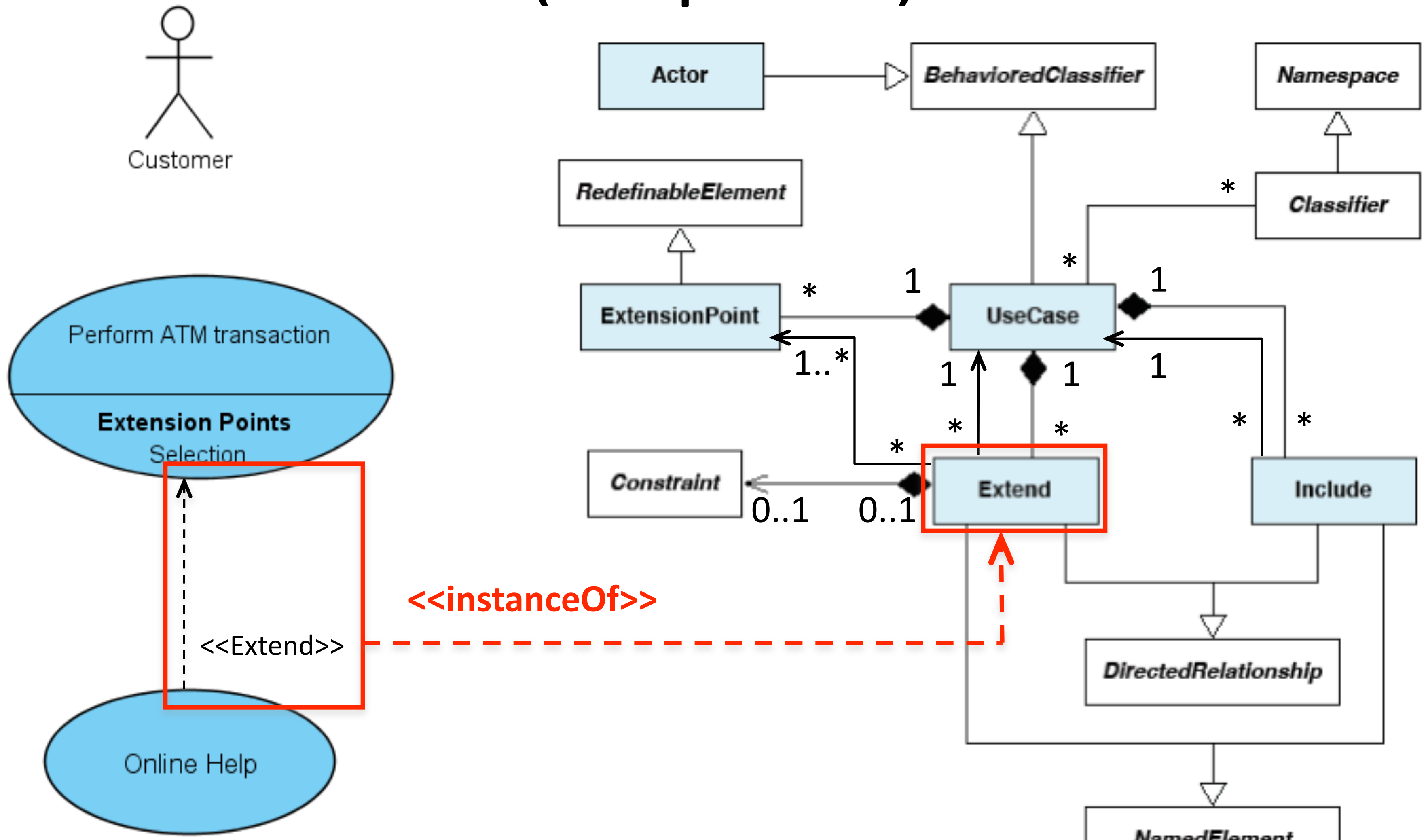
Use Case Diagram Meta Model (simplified)



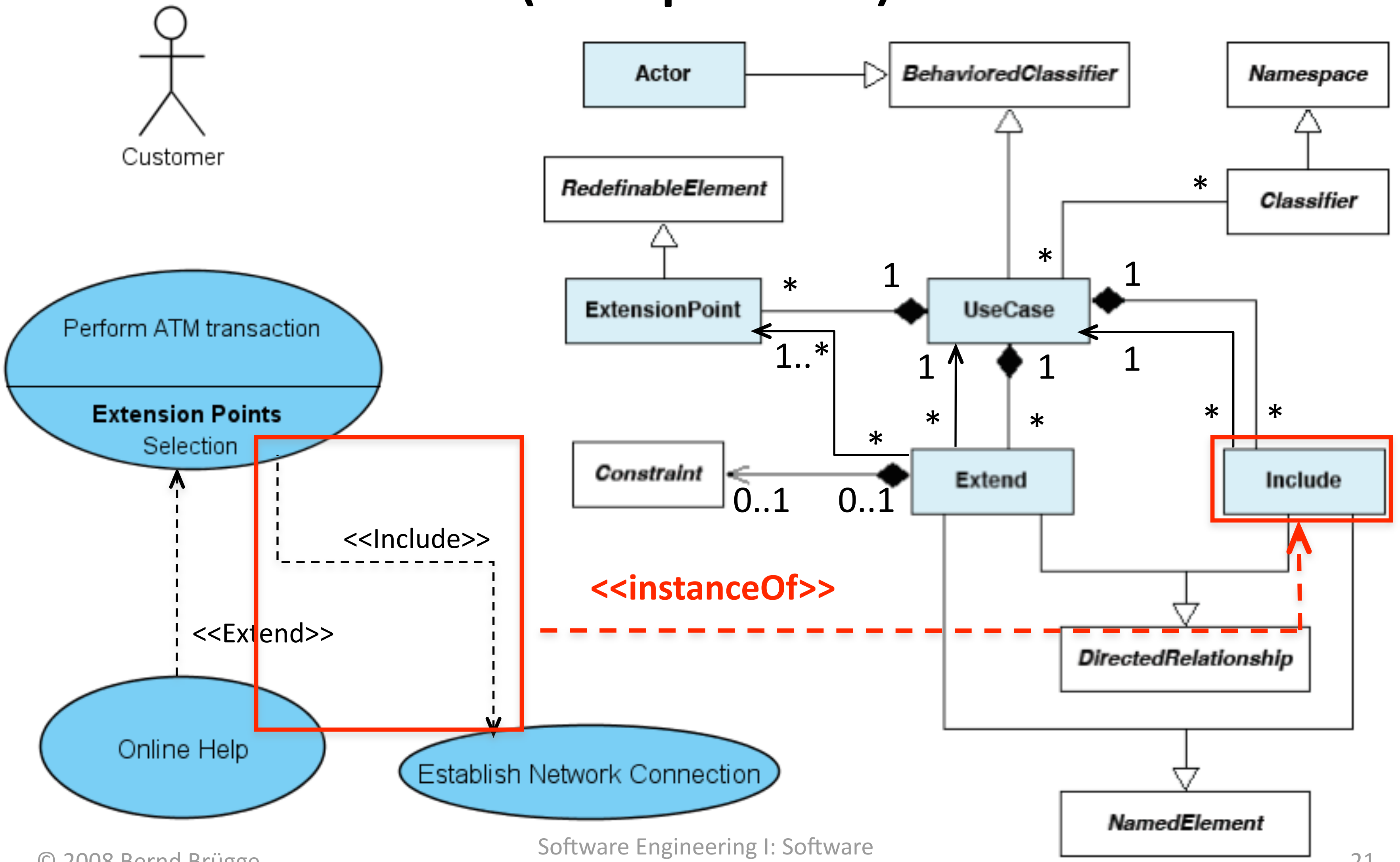
Use Case Diagram Meta Model (simplified)



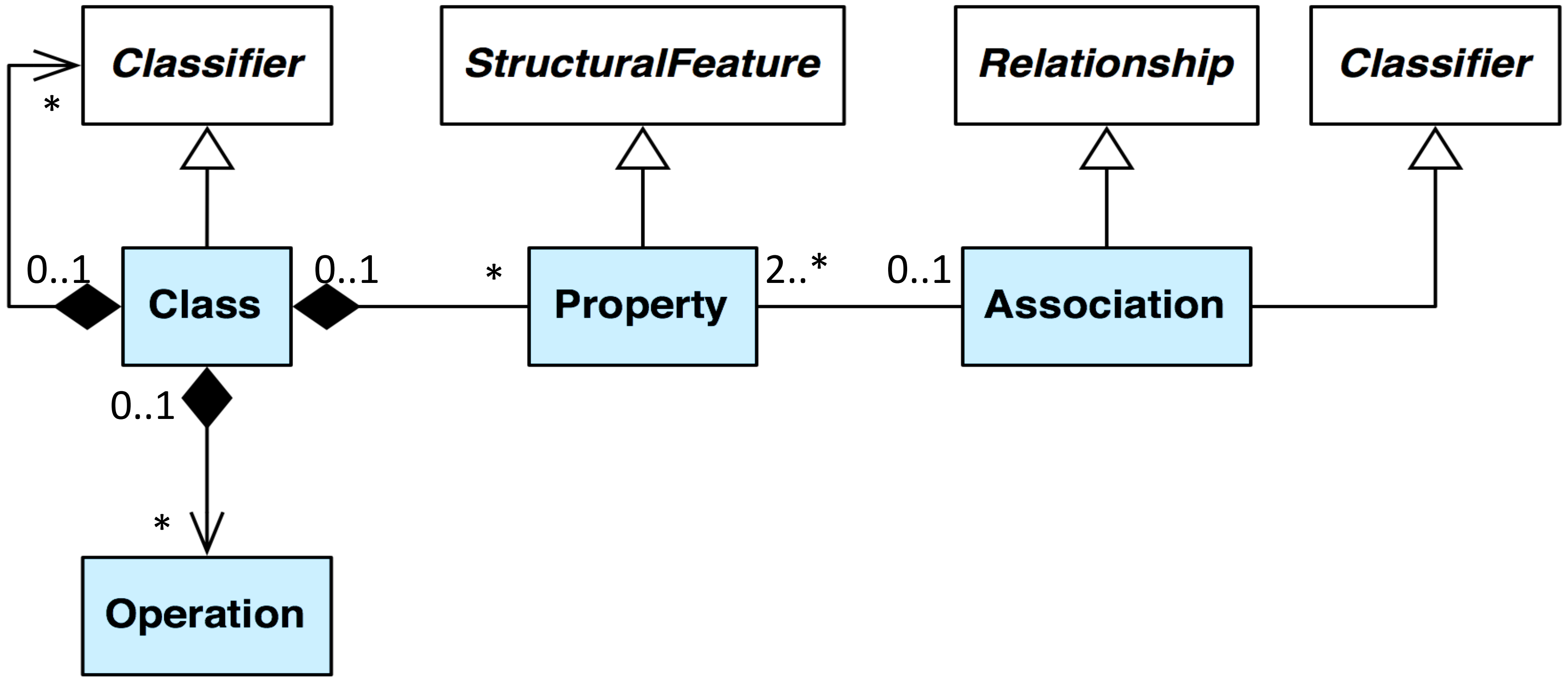
Use Case Diagram Meta Model (simplified)



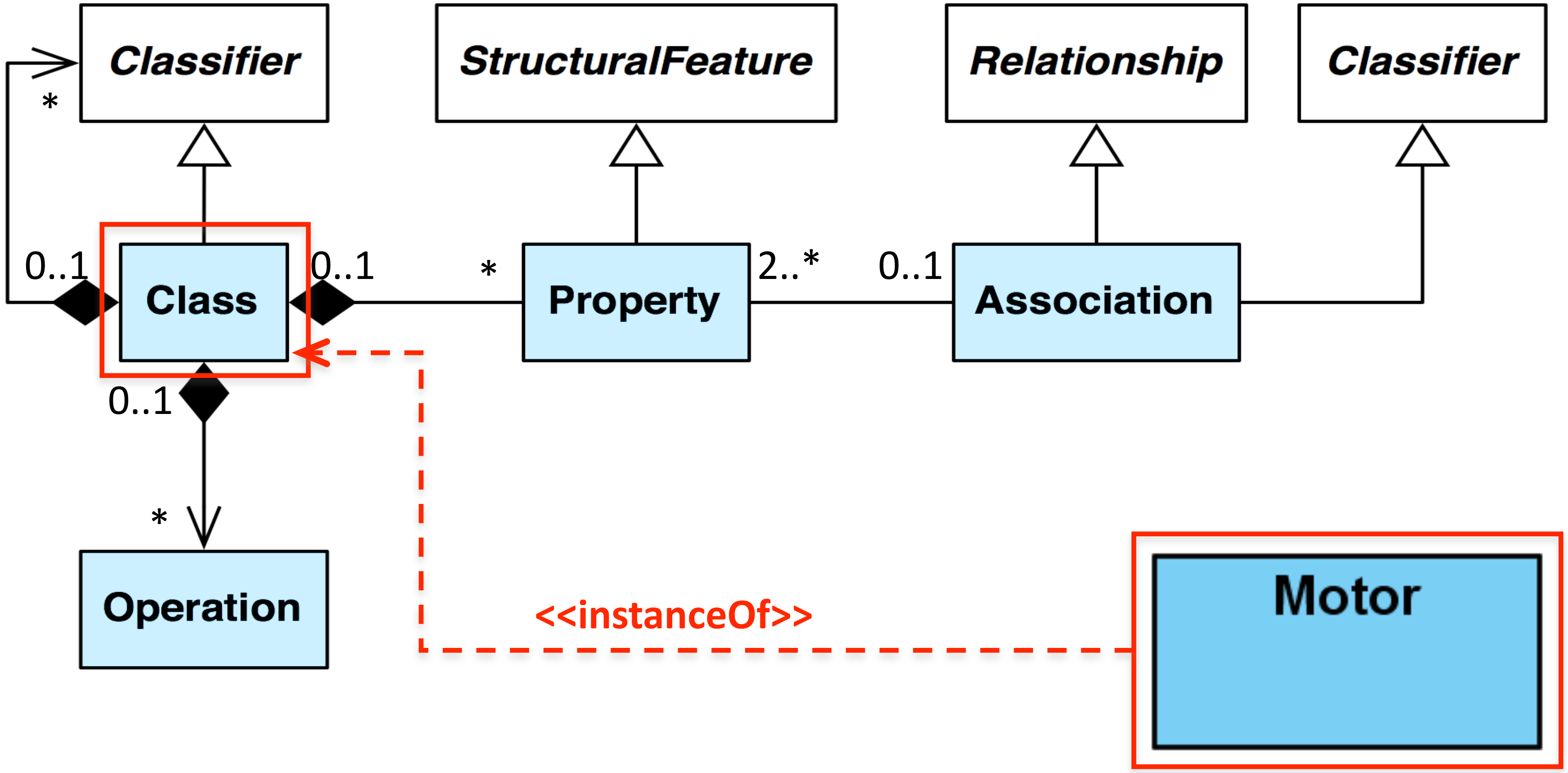
Use Case Diagram Meta Model (simplified)



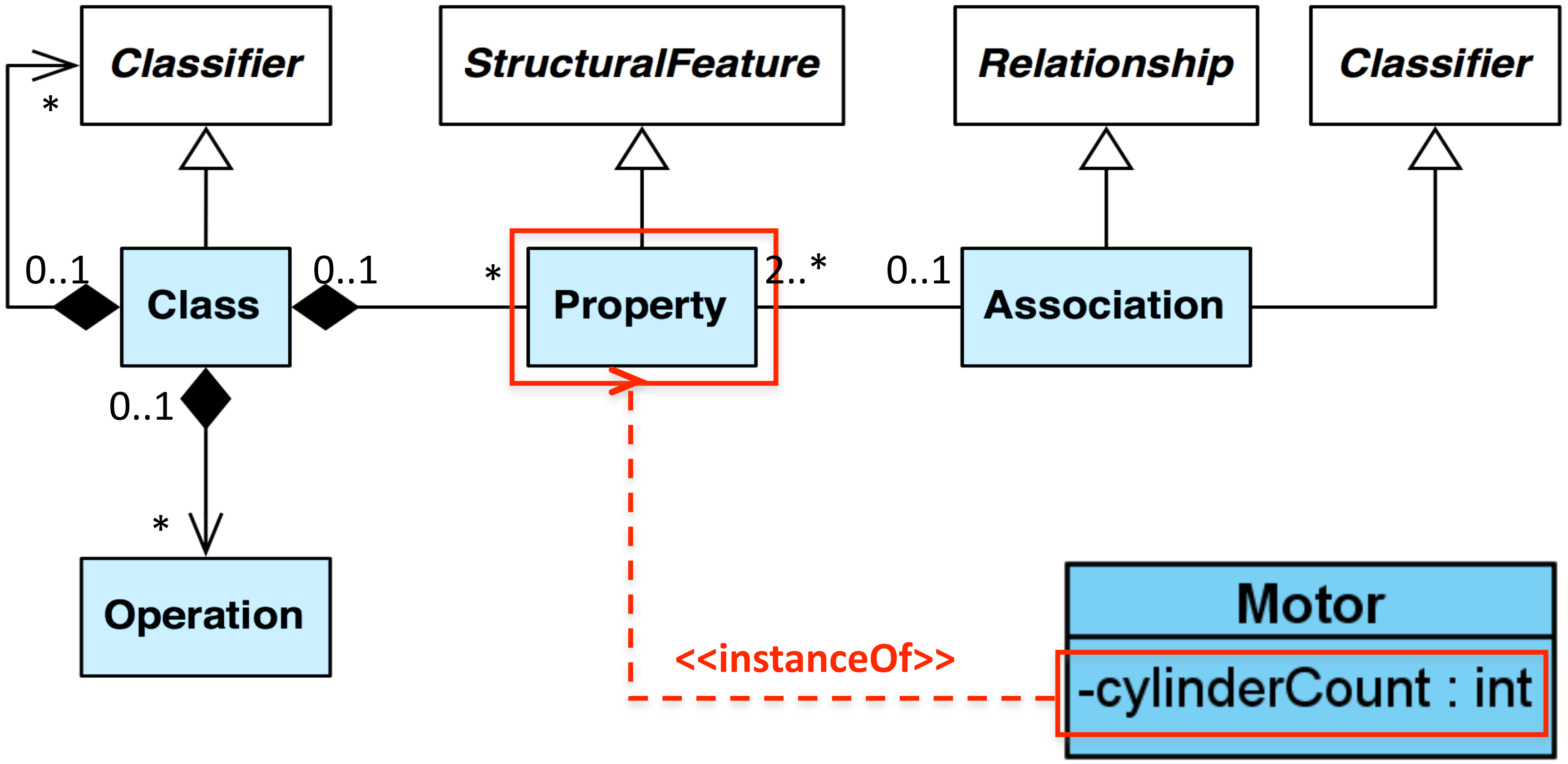
Class Diagram Meta Model (simplified)



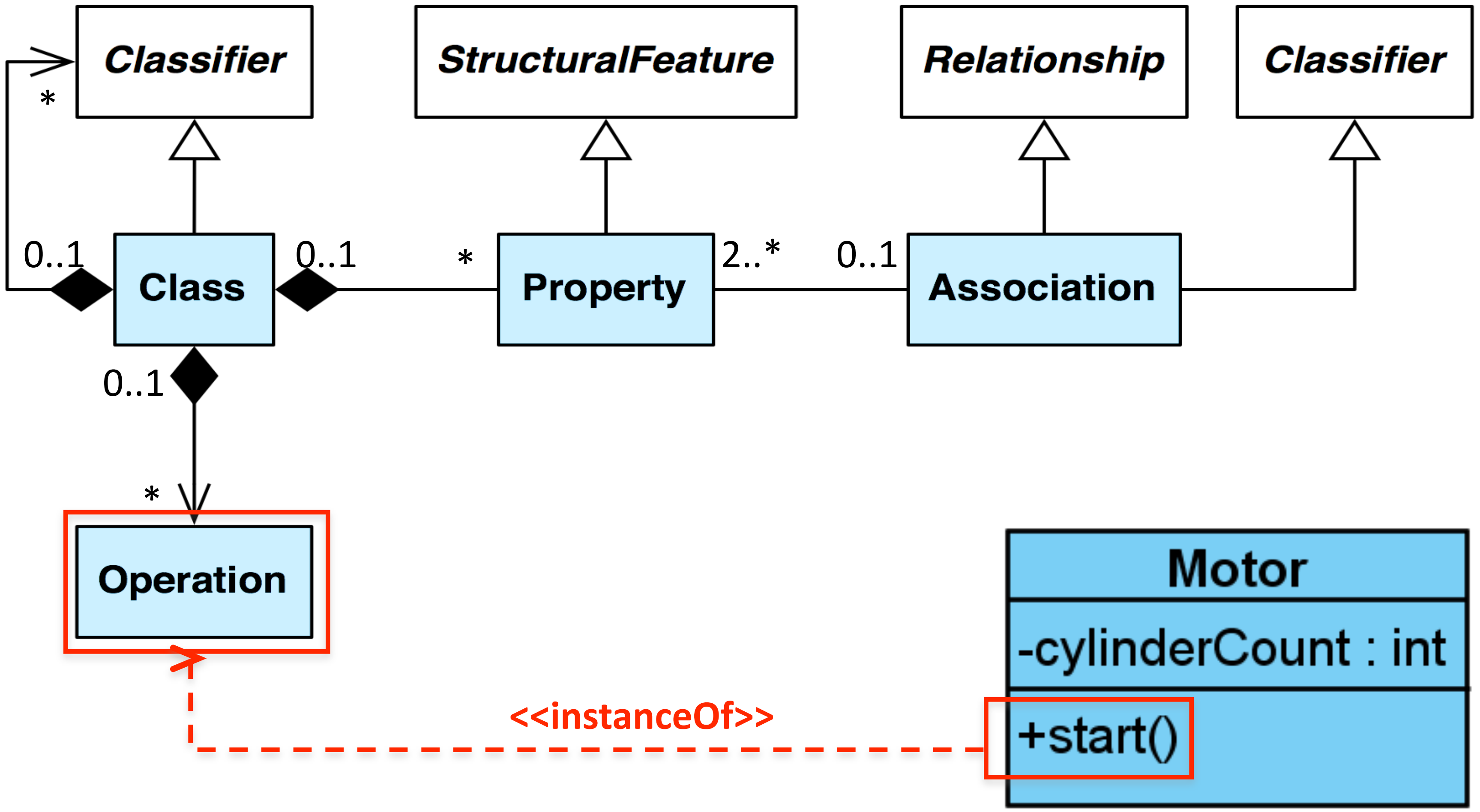
Class Diagram Meta Model (simplified)



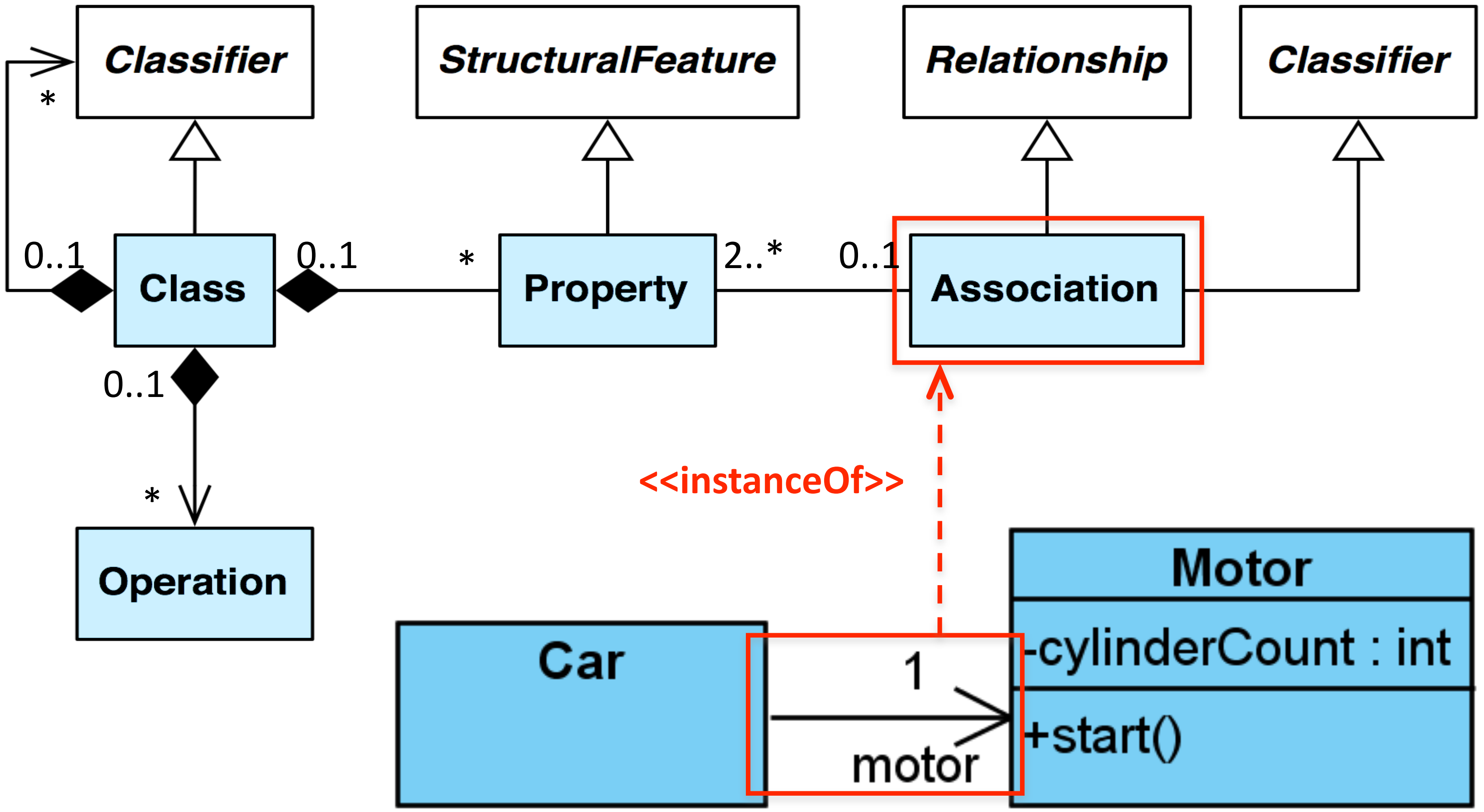
Class Diagram Meta Model (simplified)



Class Diagram Meta Model (simplified)



Class Diagram Meta Model (simplified)



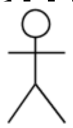
Where are we?

- ✓ From model instances to meta models
- ✓ MOF meta model hierarchy
- ✓ How UML relates to MOF
  - ✓ Example: Use case diagram meta model
  - ✓ Example: Class diagram meta model
- ➢ Different notations for the UML meta model describe the same language
- UML Profiles: Adding new members to the family

Notations for the UML meta model

- The UML meta model defines a language for specifying UML models
- The notation used to *depict* UML models provides graphical constructs representing instances of meta model elements

  (Sticky figure represents an Actor)
- The notation is a function from meta model elements to model elements

  (“*uml-notation*(Actor) =  “)

Where are we?

- ✓ From model instances to meta models
- ✓ MOF meta model hierarchy
- ✓ How UML relates to MOF
  - ✓ Example: Use case diagram meta model
  - ✓ Example: Class diagram meta model
- ✓ Different notations for the UML meta model describe the same language
- ➢ UML Profiles: Adding new members to the family

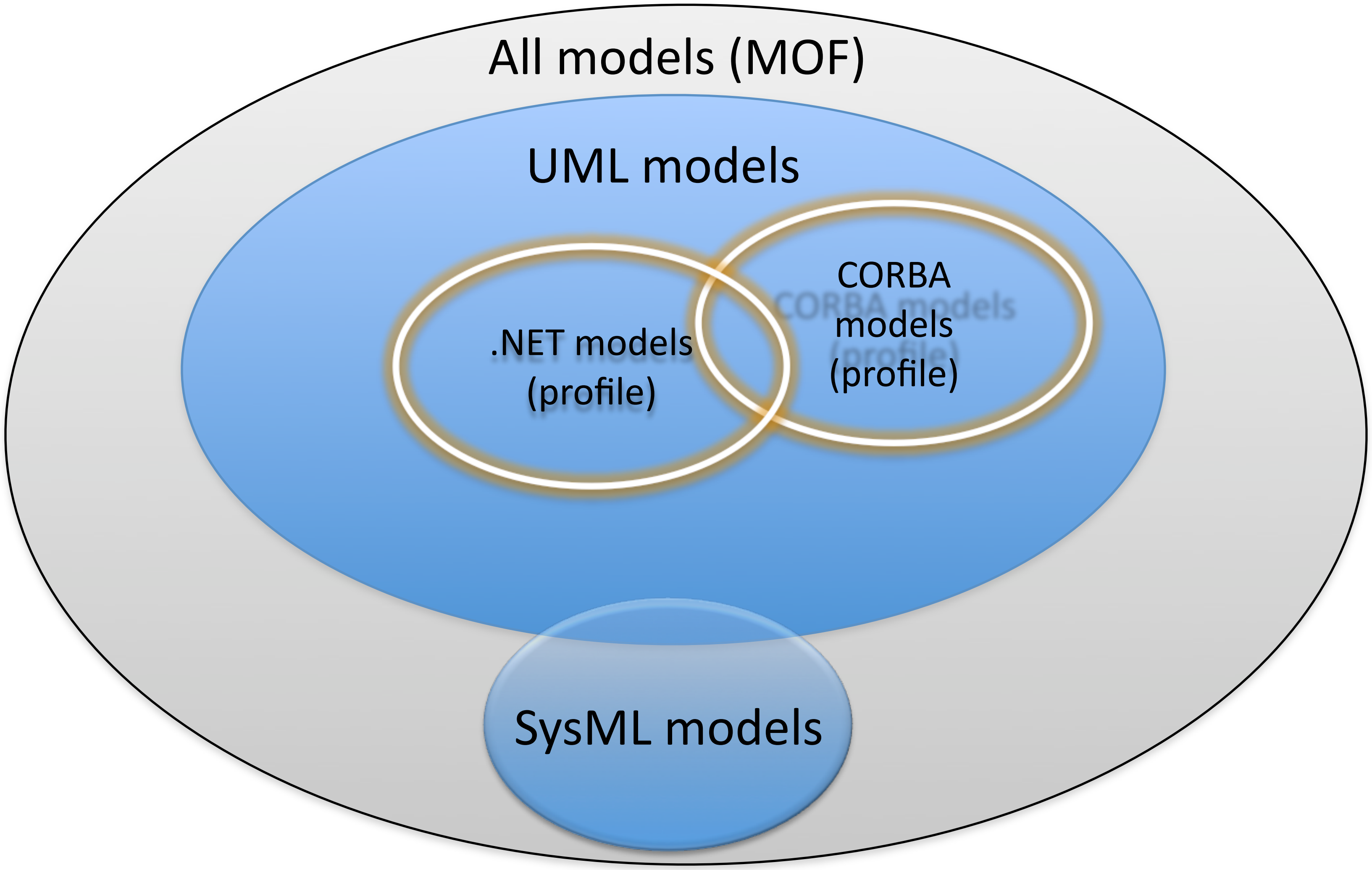
UML Profiles

- consist of stereotypes, tagged values and constraints
- customize UML models for particular domains or platforms
- are applied to elements of the UML meta model (M2)!
- are developed by manufacturers or standardization organizations (CORBA, .NET)

Applying UML Profiles

- By applying a UML profile,
  - you apply stereotypes to meta classes
  - you provide a deeper meaning for the model
  - the model gains integrity
  - you narrow the amount of valid models, as you can see on the following slide…

UML Profiles



Further reading

- MOF specification
  - URL to be delivered through the exercise portal
- Again, the UML 2 specification
  - See UML 2 Slides

Organizational Matters 1

- Mid term exam
  - Thursday, December 18th 17:30
  - Room to be announced
  - Registration procedure to be announced

Organizational Matters 2"

- For those of you interested in doing the homework, please read through the exercise sheet
  - Questions?
  - Please deliver the solution on Thursday, October 30th
    - Paper based: before the exercise
    - E-Mail based: send e-mail to Florian Schneider, same deadline