

Advanced Topics in Software Engineering (02265)

Ekkart Kindler

DTU Informatics Department of Informatics and Mathematical Modeling

athematical Modeling $f(x+\Delta x) = \sum_{i=0}^{\infty} \frac{(\Delta x)^{i}}{i!} f^{(i)}(x)$



Meta-modelling is a core part of DSL design: Abstract syntax DTU

₩

1. Background / Motivation

Mid / end 90ties:

- CASE (Computer Aided Software Engineering) modelling tools become more popular
 - code generation and round-trip-engineering
 - "UML-like" notations (and others "Booch", "OMG")
 - many dialects, variations, extensions
- Though UML starts prevailing, many other notations are in use (today called Domain Specific Languages/DSLs)
- The technology supported by the Different ways in which code is generated tool was not used for its
- Tools programmed manually
- \Rightarrow Tools, models, generated code, ... incompatible
- \Rightarrow Hinder industrial success

A bit coarse and "rosy" look at history!



implementation.



- Standardisation of a single notation: UML
- Standardisation of a transfer format
- \Rightarrow Still many problems with exchanging models \Rightarrow Need for other modeling notations
- Observation: Basic infrastructure for any CASE tool is independent from the modeling notation
- CASE tools should be implemented using their technology



M2	Unified Modelling Notation	modelling notation
M1	Model	design-time
MO	User data	run-time

Re: Example (from L01)

DTU Informatics

Department of Informatics and Mathematical Modelling Ekkart Kindler







. . .



PetriNet

Object

Node

Transition

sourc

1 target

Place

Arc

Token



= conforms to / is instance of

Meta Object Facility (MOF)

DTU Informatics Department of Informatics and Mathematical Modelling Ekkart Kindler







- Is that any good?
 - There is one level that we did not have before!
 So, this seems to be more complicated!
 - If UML can be defined in terms of itself, why should we define it in terms of something else?

DTU Informatics Department of Informatics and Mathematical Modelling Ekkart Kindler





Meaning of "Meta-"



Meta (from Greek: $\mu\epsilon\tau\dot{\alpha}$ = "after", "beyond", "with", "adjacent", "self"), is a prefix used in English in order to indicate a concept which is an abstraction from another concept, used to complete or add to the latter.

In epistemology, the prefix **meta-** is used to mean *about (its own category)*. For example, metadata are data about data, something about something (who has produced them, when, what format the data are in and so on). Similarly, metamemory in psychology means an individual's knowledge about whether or not they would remember something if they concentrated on recalling it. Furthermore, metaemotion in psychology means an individual's emotion about his/her own basic emotion, or somebody else's basic emotion.

Another, slightly different interpretation of this term is "about" but not "on" (exactly its own category). For example, in linguistics a grammar is considered as being expressed in a metalanguage, or a sort of language for describing *another* language (and not itself). A **meta-answer** is not a real answer but a reply, such as: "*this is not a good question*", "*I suggest you ask your professor*". Here, we have such concepts as meta-reasoning and meta-knowledge.

From: http://en.wikipedia.org/wiki/Meta



Co-notations and meaning in Software Engineering:

- beyond, "one level higher"
- possibly self-referential (with all the problems of self-referentiality)









Additional constraints (e.g.):

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

DTU Informatics Department of Informatics and Mathematical Modelling Ekkart Kindler







DTU

₩

EMOF Primitive Types

DTU Informatics Department of Informatics and Mathematical Modelling Ekkart Kindler



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

EMOF Packages

DTU Informatics Department of Informatics and Mathematical Modelling Ekkart Kindler



DTU

₩

EMOF Discussion

DTU Informatics Department of Informatics and Mathematical Modelling Ekkart Kindler



- Can EMOF be defined with its own concepts?
- Is it expessible enough?

EMOF stands for Essential MOF; we will discuss more complete model, Complete MOF (CMOF), later.

- What is missing (as compared to UML diagrams)?
- How does EMOF relate to ECore (the model underlying EMF)?
- Can UML be expressed in it?

EMF / ECore might be the reason, EMOF was included in the MOF standard.

Any other problems?

3.2 Reflection capability

- Creating models and their instances (resp. metamodels and their conforming models) dynamically
- Navigating between model elements and instance
- ⇒By navigation between different meta-levels in an arbitrary way, MOF is not restricted to a fixed number of levels.

"Reflection": Knowing something (and reasoning) about oneself. DTU

Reflection package

DTU Informatics Department of Informatics and Mathematical Modelling Ekkart Kindler



3.3 Complete MOF

DTU Informatics Department of Informatics and Mathematical Modelling Ekkart Kindler



- As discussed, EMOF lacks some features:
 - Associations (only by paired properties, and only binary)
 - Visibilities
 - Subsetting and redefing properties

Like EMOF, in CMOF these definitions refer to UML infrastructure.

- In the following, we indicate how some of these features are captured in CMOF
- Like EMOF, CMOF can be defined in terms of its own concepts (or in terms of EMOF)

UML Core::Constructs

DTU Informatics

Department of Informatics and Mathematical Modelling Ekkart Kindler





UML Core::Constructs

DTU Informatics Department of Informatics and Mathematical Modelling Ekkart Kindler





CMOF



- Reflection similar to EMOF (a bit more complex)
 - explicit creation of Links (see next slides)
 - invocation of operations
 - instanceOf can check class hierarchy (instance of subclasses)



- "Semantic domain model" introduces the concepts for the instances of a model and
- how the reflection capabilities behave on the model

CMOF instance model

DTU Informatics Department of Informatics and Mathematical Modelling Ekkart Kindler



Advanced Topics in Software Engineering (02265), L03: Meta-modelling

DTU

Ħ

4. XMI

Warning: If you change the meta

older versions! That is why XML

model, you often can no longer read



- Mapping MOF-models and its instances to XML in a standard way
- A MOF model is mapped to an XMLSchema for its instances
- XMI is a standard associated with MOF

=>You can easily exchange MOF models
=>Once you agree on the MOF-model, you can exchange instances of that model

Advanced Topics in Software Engineering (02265), L03: I ((later in this course).

Example: EMOF model

DTU Informatics Department of Informatics and Mathematical Modelling Ekkart Kindler



```
<?xml version="1.0" encoding="UTF-8"?>
<emof:Package xmi:version="2.0"</pre>
    xmlns:xmi="http://www.omg.org/XMI"
    xmlns:emof="http://schema.omg.org/spec/MOF/2.0/emof.xml" xmi:id="PetriNets"
    name="PetriNets" uri="APetriNetEditorIn15Minutes">
  <ownedType xmi:type="emof:Class" xmi:id="PetriNets.PetriNet" name="PetriNet">
    <ownedAttribute xmi:id="PetriNets.PetriNet.object" name="object"</pre>
        isOrdered="true"
        lower="0" upper="*" type="PetriNets.Object" isComposite="true"/>
  </ownedType>
  <ownedType xmi:type="emof:Class" xmi:id="PetriNets.Object" name="Object"</pre>
      isAbstract="true"/>
  <ownedType xmi:type="emof:Class" xmi:id="PetriNets.Node" name="Node"</pre>
      isAbstract="true, superClass="PetriNets.Object">
    <ownedAttribute xmi:id="PetriNets.Node.name" name="name" isOrdered="true"</pre>
        lower="0">
      <type xmi:type="emof:PrimitiveType"
          href="http://schema.omg.org/spec/MOF/2.0/emof.xml#String"/>
    </ownedAttribute>
    <ownedAttribute xmi:id="PetriNets.Node.in" name="in" isOrdered="true"</pre>
        lower="0" upper="*" type="PetriNets.Arc"
        opposite="PetriNets.Arc.target"/>
    <ownedAttribute xmi:id="PetriNets.Node.out" name="out" isOrdered="true"</pre>
        lower="0" upper="*" type="PetriNets.Arc"
        opposite="PetriNets.Arc.source"/>
  </ownedType>
```

Example (cntd.)



```
<ownedType xmi:type="emof:Class" xmi:id="PetriNets.Arc" name="Arc"</pre>
        superClass="PetriNets.Object">
    <ownedAttribute xmi:id="PetriNets.Arc.source" name="source"</pre>
   isOrdered="true"
        type="PetriNets.Node" opposite="PetriNets.Node.out"/>
    <ownedAttribute xmi:id="PetriNets.Arc.target" name="target"</pre>
   isOrdered="true"
        type="PetriNets.Node" opposite="PetriNets.Node.in"/>
  </ownedType>
  <ownedType xmi:type="emof:Class" xmi:id="PetriNets.Transition"</pre>
      name="Transition" superClass="PetriNets.Node"/>
  <ownedType xmi:type="emof:Class" xmi:id="PetriNets.Place"</pre>
      name="Place" superClass="PetriNets.Node">
    <ownedAttribute xmi:id="PetriNets.Place.token" name="token"</pre>
        isOrdered="true" lower="0" upper="*"
        type="PetriNets.Token" isComposite="true"/>
  </ownedType>
  <ownedType xmi:type="emof:Class" xmi:id="PetriNets.Token"</pre>
      name="Token"/>
  <xmi:Extension extender="http://www.eclipse.org/emf/2002/Ecore">
    <nsPrefix>APetriNetEditorIn15Minutes</nsPrefix>
  </xmi:Extension>
</emof:Package>
```

XMI instance	DTU Informatics Department of Informatics	s nd Mathematical Modelling
xml version="1.0" encoding="UTF-8"? <apetrineteditorin15minutes:petrinet xmi<br="">xmlns:xmi="http://www.omg.org/XMI"</apetrineteditorin15minutes:petrinet>	Ekkart Kindler	The serialisation of instances can be customized in tools like EMF.
<pre>xmlns:xsi="http://www.w3.org/2001/XMLS xmlns:APetriNetEditorIn15Minutes="APet <object @object.7"="" out="//@object.4" xsi:type="APetriNetEditorIn15M:
in="></object></pre>	Schema-instance criNetEditorIn inutes:Transit	e" 15Minutes"> ion" name="t1"
<pre><object @object.5"="" out="//@object.6" xsi:type="APetriNetEditorIn15M:
in="></object></pre>	inutes:Transit >	ion" name="t2"
<pre><object @object.6"="" out="//@object.7" xsi:type="APetriNetEditorIn15M:
in="> <token></token></object></pre>	inutes:Place"	name="p1"
<pre><object @object.4"="" out="//@object.5" xsi:type="APetriNetEditorIn15M:
in="></object></pre>	inutes:Place"	name="p2"
<pre><object <="" @object.0"="" pre="" target="//@object.0" xsi:type="APetriNetEditorIn15M:
source="></object></pre>	inutes:Arc" ect.3"/>	ere, references are
<pre><object <="" @object.3"="" pre="" target="//@object.3" xsi:type="APetriNetEditorIn15M:
source="></object></pre>	inutes:Arc" vi ect.1"/>	a paths (XPath).
<pre><object <="" @object.1"="" pre="" target="//@object.1" xsi:type="APetriNetEditorIn15M:
source="></object></pre>	inutes:Arc" ect.2"/> If	model elements
<pre><object source="//@object.2" target="//@object.0" xsi:type="APetriNetEditorIn15Minutes:Arc"></object> have ids, the</pre>		
		he ids!



- MOF models can also be mapped to a programming language with an API
- The Java Metadata Interface (JMI) is a standard mapping to Java
- The mapping of EMF models to Java is another one (not compatible with JMI) In our tutorials, we

use EMF – a de facto standard.

- EMF has a reflective interfaces: see methods XXXPackage;
- Packages can be registered and used without code generation http://www.eclipse.org/gmt/epsilon/doc/articles/reflective-emf-tutorial/
- http://www.devx.com/Java/Article/29093/1954

6. Summary







- Why EMOF and CMOF?
- Relation between UML infrastructure and MOF
- Relation between EMOF and ECore (EMF)





Overview

DTU Informatics Department of Informatics and Mathematical Modelling Ekkart Kindler





http://www2.imm.dtu.dk/~pk/images/meta-levels.pdf (thanks to Patrick)