Metamodel

Global view
Some sources

- Object Modeling with UML: Advanced Modeling, Karin Palmkvist, Bran Selic, and Jos Warmer, March 2000

Instances: tokens, specific things
Classes: collections of instances
Instances: tokens, specific things
Classes: collections of instances
Models and Metamodules

Specific models: tokens, specific things, and relationships
Generic models: collections of instances, and relationships
Metamodules: ontology, …and epistemology, of generic models
Models and Metamodels

- **Specific models**: tokens, specific things, and relationships
- **Generic models**: collections of instances, and relationships
- **Metamodels**: ontology, …and epistemology, of generic models

Why?
Specific models: tokens, specific things, and relationships
Generic models: collections of instances, and relationships
Metamodels: ontology, and epistemology, of generic models
What is meta-modeling?

- Generally, the analysis, construction and development of the frames, rules, constraints, models and theories applicable and useful for the modeling in a predefined class of problems.

- In computer science and related disciplines, the construction of a collection of "concepts" (things, terms, etc.) within a certain domain.

- A model is an abstraction of phenomena in the real world, and a metamodel is yet another abstraction, highlighting properties of the model itself.

- A model should conform to its metamodel like a program conforms to the grammar of the programming language in which it is written.

“A valid metamodel is an ontology, but not all ontology are modeled explicitly as metamodels” [Söderström2002].
A meta-model defines concepts and their relationships thanks to a class diagram.

A meta-model only defines structure (no semantic).

A model is an instance of a meta-model if it respects the structure defined by the meta-model.

The UML meta-model defines the structure that all UML models must have.
An Example: UML metamodel in the 4 Layer

Figure 8 - An example of the four-layer metamodel hierarchy
Another Example:
UML metamodel in the 4 Layer
UML Meta Model

Metaclasses used in class, package, component and deployment diagrams

www.site.uottawa.ca/ftp/pub/courses/Winter/seg3310/coursenotes/Lecture7-8-Metamodelling.ppt
Points to Ponder

- Can we create a new version of UML, in consideration of the following?

  Meta-metamodeling
  - abstract language
  - specific paradigm (e.g., OO)
  - application independent

  Metamodeling
  - abstract language
  - specific paradigm (e.g., OO)
  - application specific
    - process or notation modeling, etc.

  Modeling
  - concrete language
  - specific paradigm (e.g., OO)
  - specific model
    - describes the data needed

  Information
  - applied in "real world"
  - specific paradigm (e.g., OO)
  - specific project

  MOF standard
  (OO paradigm)

  USPM (RUP) metamodel

  RUP model

  Project X Process

  Project X Artifacts

  UML standard

  UML metamodel

  Examples

- MetaClass, MetaAttribute, MetaOperation
- Class, Attribute, Operation, Component
- StockShare, owner, askPrice
- hacsSoftwareCoShare99, smartie, 11.18
Meta-meta-model

- MOF defines the language for defining meta-models
- MOF concepts are meta-class, meta-attribute, meta-association, etc.
- MOF concepts and their relationships can be defined by a class diagram. This diagram is also a meta-model (called the meta-meta-model)
- The meta-meta-model is self defined.
Meta-layers

- Mn+1 defines the structure of Mn
- Mn+1 is not an abstraction of Mn
- Meta-layer relationships are similar to grammar-layer relationships (BNF, or XML Schema)
Infrastructure: Consolidation of Concepts

Namespace

PackagableElement

RedefinableElement

Feature

Classifier

Namespace

PackagableElement

RedefinableElement

Feature
Infrastructure: Behavior Harmonization

- Common semantic base for all behaviors
  - Choice of behavioral formalism driven by application needs

```
Classifier 0..1  Behavior 0..*

Class UseCase Component ... Action Activity Interaction Statemachine
```
Common component paradigm

«enumeration»
ComponentKind
+ COMPOUND:
+ ELEMENTARY:
+ REPETITIVE:

Component
- kind: ComponentKind = COMPOUND

Port
+ owner + port

Connector
+ owner

Part
+ type

Encapsulation

Repetition of structural elements

Assembling

Composing
UML 2.0 Component
Object Constraint Language

- OCL defines the structure of models expressing constraints
  - Invariant, Pre-post conditions
- OCL is a meta-model instance of the MOF
- OCL is highly coupled with UML
- The OCL semantic is defined with models (operation without side effect)
- OCL defined a concrete syntax
Action Semantics

- AS defines the structure of models expressing sequences of actions
- AS was a meta-model and is now completely integrated in UML2.0
- AS has no concrete syntax (UML diagram)
- The semantic of AS is not formally defined (an RFP is published)
Model transformation is a template written in a dedicated language.

- UML
- PetStore
- UML2EJB Template
- Template Interpreter
- EJB
- PetStore EJB
Subsystems are used for system decomposition.
Collaboration
A model is an abstraction of a system, specifying the system from a certain viewpoint and at a certain level of abstraction.
Models and subsystems can be combined in a hierarchy:
Stereotype Notation

- Several choices

(a) with guillemets

(b) with icon

(c) iconified form