Going Meta

Adrian Kuhn

Meta = Description

Example: SQL-Database

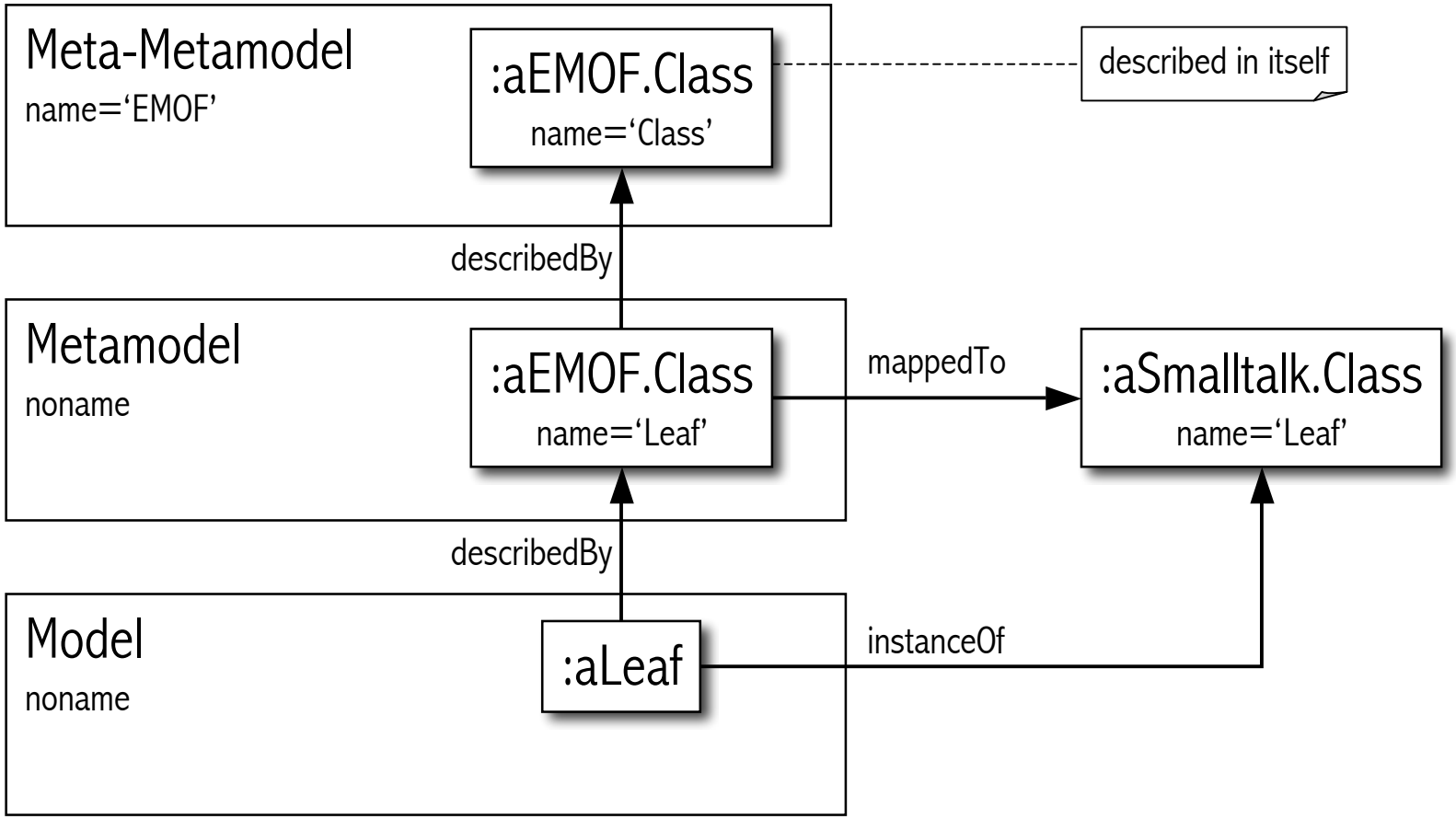
Real World

Model = Data about Real World

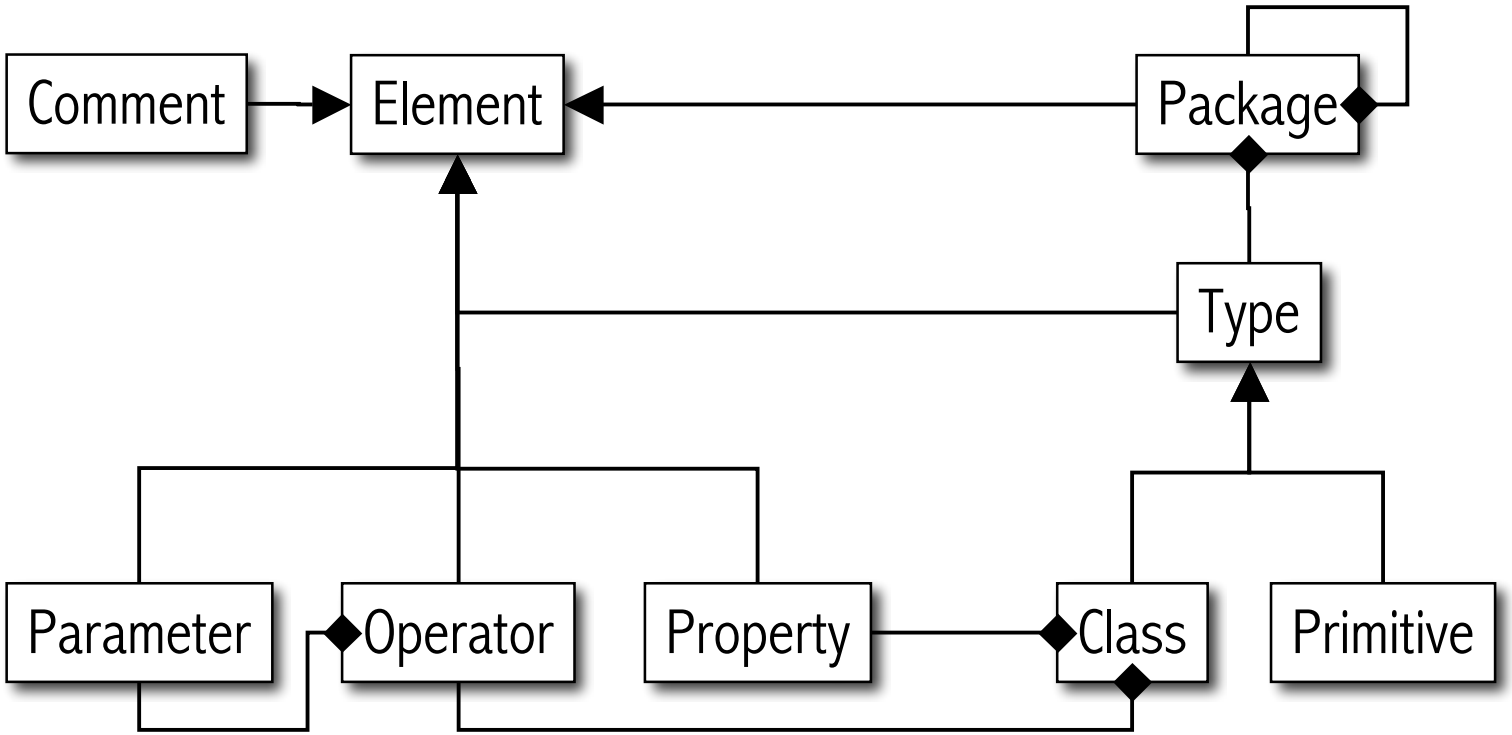
Metamodel = Schema of Data

Metametamodel = Schema of Schema

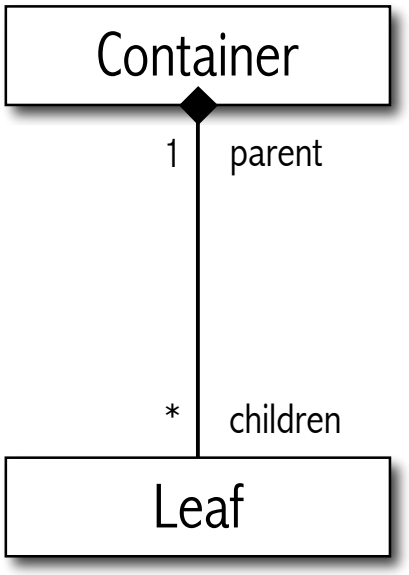
There are Three Layers



EMOF


Comment
Element
Package
Type
Parameter
Operator
Property
Class
Primitive

Lets implement this



Write Accessors

```
Leaf>>parent

Leaf>>parent:

Container>>children

Container>>children:
```

Describe Property

```
Leaf class>>metamodelParent
  ^EMOF.Property
     name: #parent
     type: Container
```

Multiplicity

```
Container class>>metamodelChildren
  ^EMOF.Property
     name: #child
     type: Leaf
     multiplicity: #many
```

!! The property's name is singular, but the names of the accessors use plural.

Opposite

```
Container class>>metamodelChildren
  ^EMOF.Property new
     name: #child;
     type: Leaf;
     opposite: #parent;
     multiplicity: #many;
     yourself
```

Containment

```
Leaf class>>metamodelParent
  ^EMOF.Property new
     name: #parent;
     type: Container;
     opposite: #child;
     isContainer: true;
     yourself
```

Apply Changes

```
anObject updateMetaDescription

aNameSpace updateMetaDescriptions
```

Accessing

```
aLeaf mGet:#parent

aLeaf mSet:#parent value:aContainer

aLeaf metaDescription

aLeaf metaDescription allAttributes
```

Moose Uses Pragmas

```
<navigation:>

<property:longName:description:>

<property:>

<menuItem:>

<menuItem:category:>
```

Methods vs. Pragmas

- Defined with “metamodel…”
  - Attributes for import
  - Attributes for export
- Moose pragmas
  - Properties for UI
  - Navigation for UI
  - UI menu operations

Import and Export

```
Meta.Importer new
    importFile: aFileName;
    elements
```

```
Meta.Exporter new
    addAll: collection;
    contentString
```

MSE File Format

```
(Container (id: 1)
  (child (idref: 2) (idref: 3))

(Leaf (id: 2)
  (parent (idref: 1))

(Leaf (id: 3)
  (parent: (idref: 1))
```

Meta = Description

More? akuhn@iam.unibe.ch