

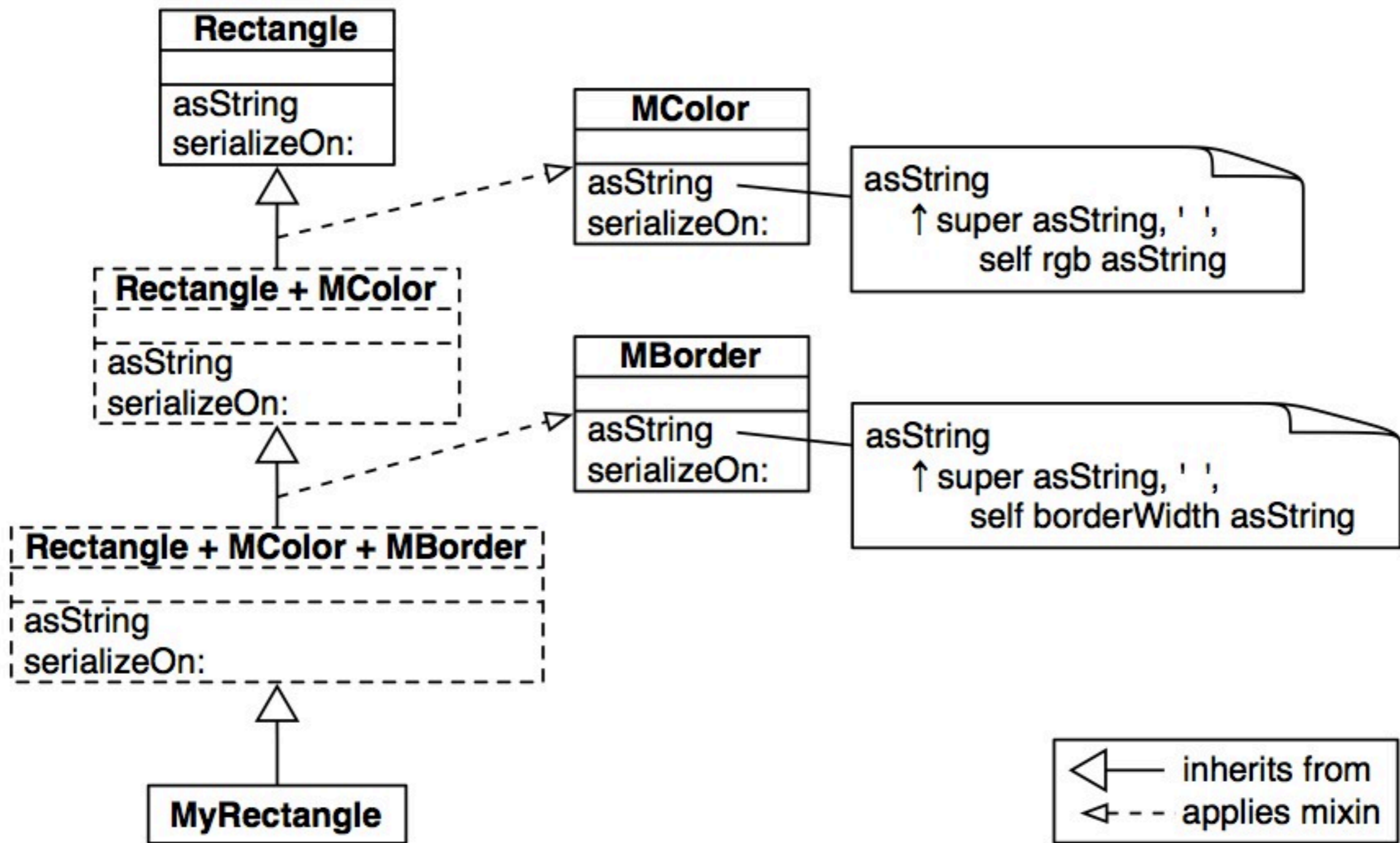
Talents: Dynamically Composable Units of Reuse

Jorge Ressia, Tudor Gîrba, Oscar Nierstrasz,
Fabrizio Perin and Lukas Renggli

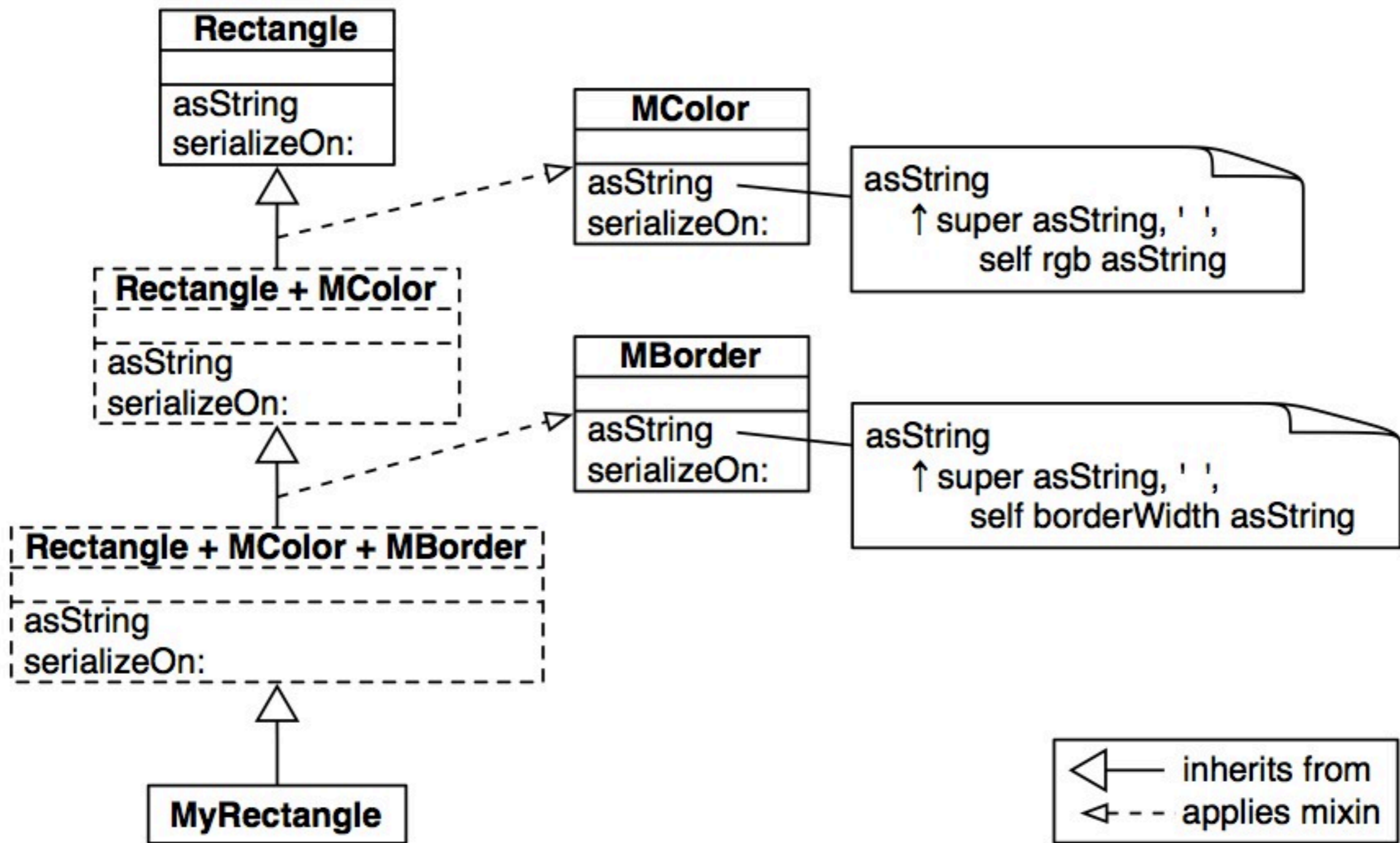
www.scg.unibe.ch

Reuse

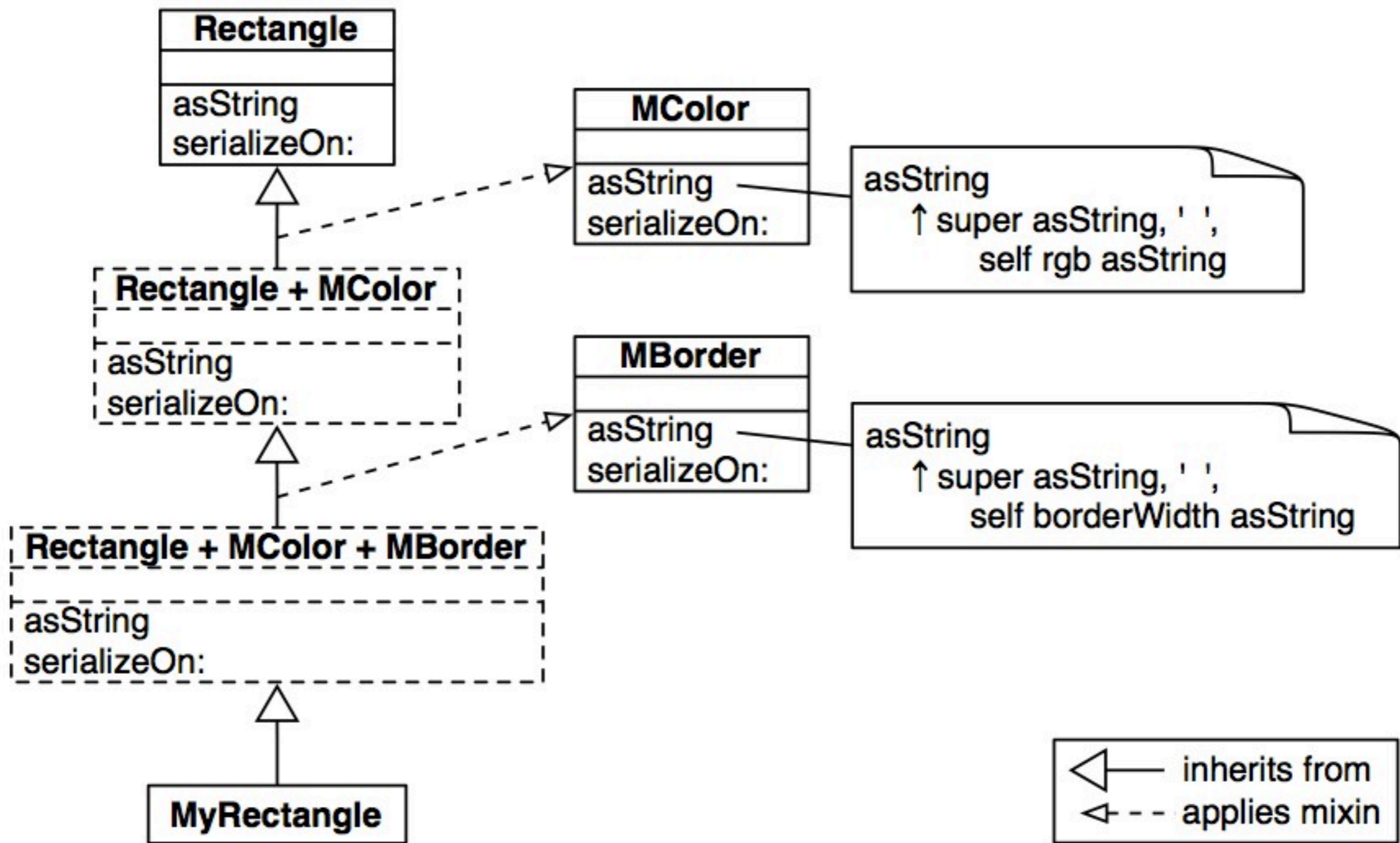
Mixins



Linear Composition



Single Composition



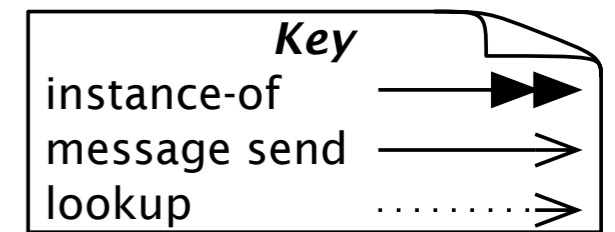
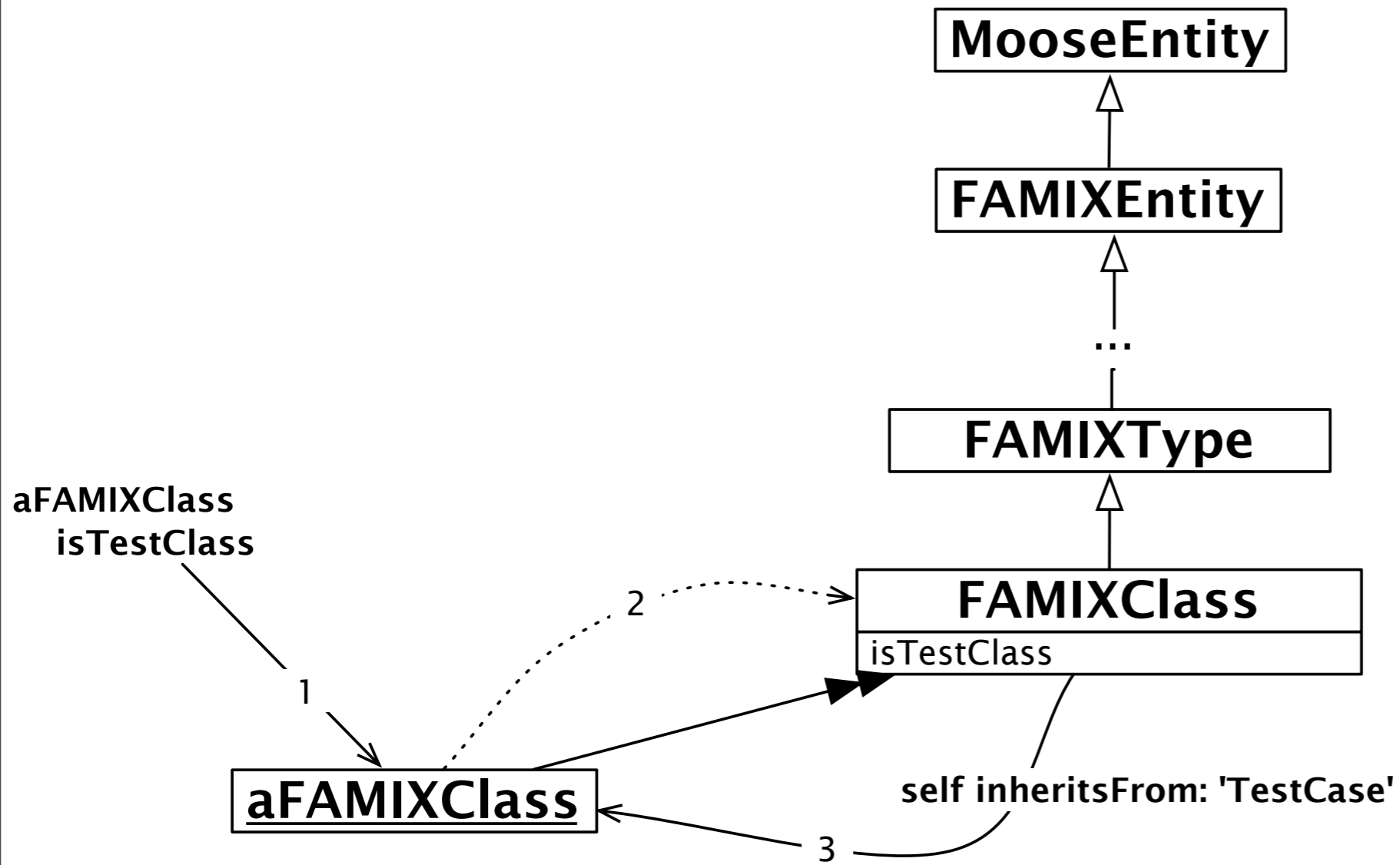
Traits

*Class = Superclass + State + Traits +
Glue methods*

Flatten Composition

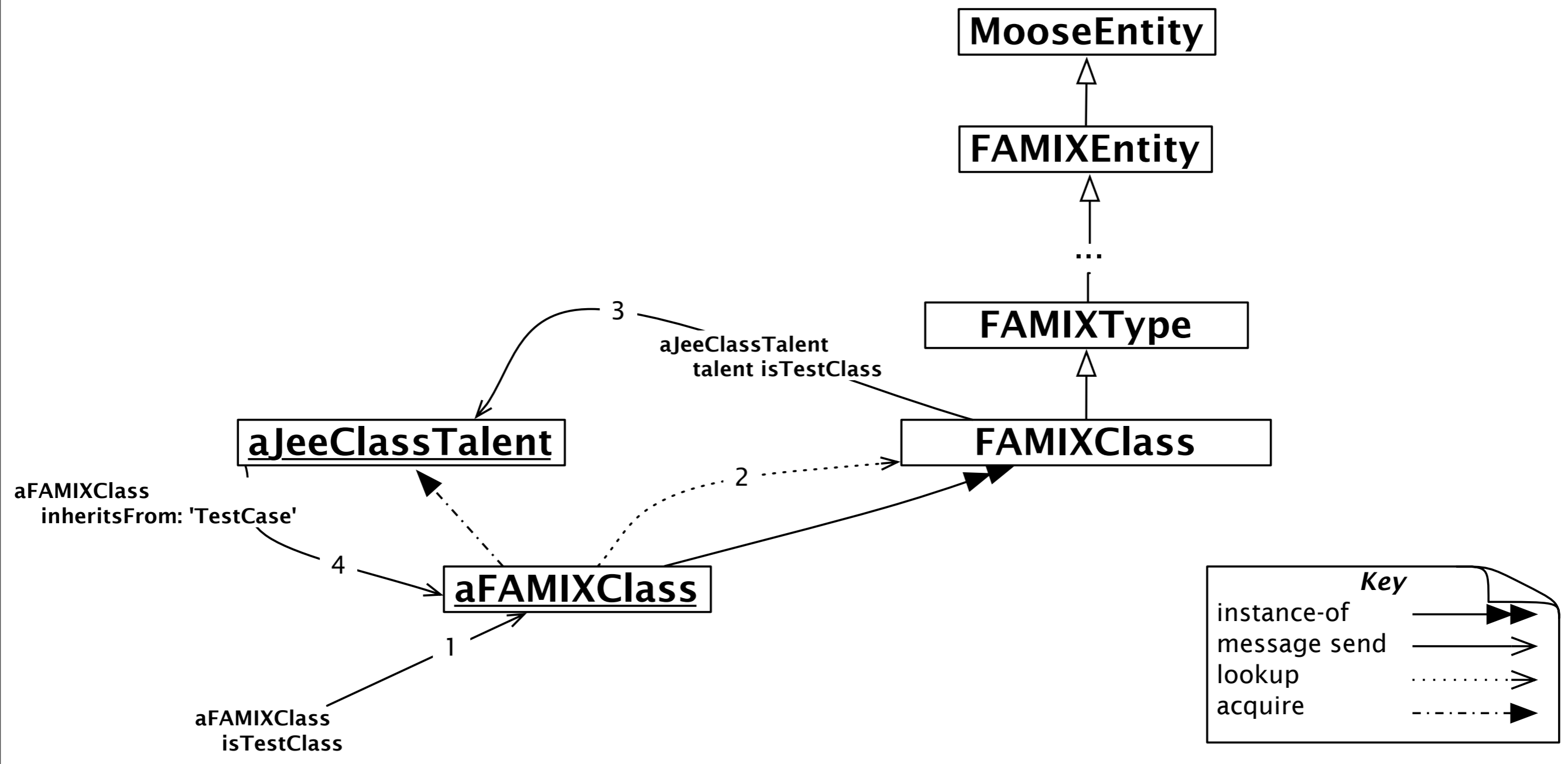
Multiple Composition

Problem



What about objects?

Talents



Operators

Composition

aClass := FAMIXClass new.

aClass acquire: (javaClassTalent , jeeClassTalent).

Alias

aClass := FAMIXClass new.

aClass acquire: javaClassTalent , (jeeClassTalent
@ {#isTestClass -> #isJEETestClass})

Exclusion

```
aClass := FAMIXClass new.
```

```
aClass acquire: javaClassTalent ,  
    (jeeClassTalent - #isTestClass).
```


Implementation

Reflection

Bifrost

scg.unibe.ch/research/bifrost

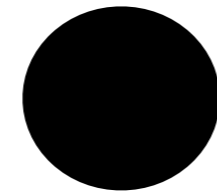
Organize the Meta-level

Explicit Meta-objects

Class



Talent



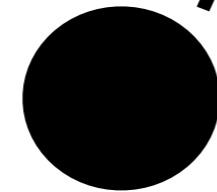
Object



Class



Talent



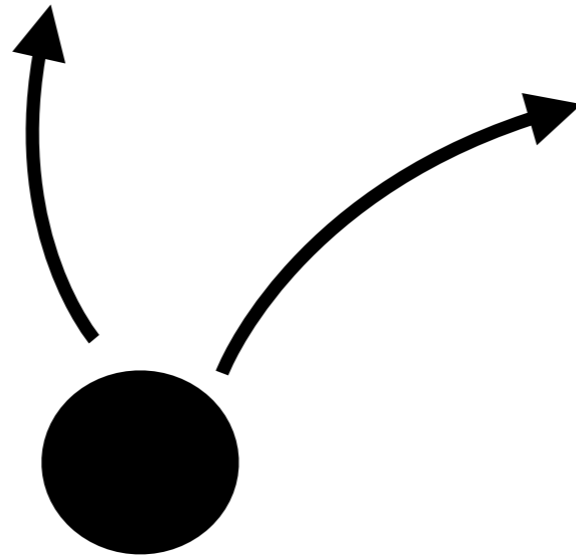
Object



Class



Talent



Talented Object

Flattening

Scoping

Talents

www.scg.unibe.ch/research/talents