

*Feature and Meta-Models in Clafer:  
Mixed, Specialized, and Coupled*

Kacper Bąk

Krzysztof Czarnecki

University of Waterloo

Andrzej Wąsowski

IT University of Copenhagen

October 12, 2010

# Outline

1. Feature vs. Meta-Modeling
2. *Clafer* (class, feature, reference)
3. Experimental Evaluation
4. Conclusion

# 1. Feature vs. Meta-Modeling

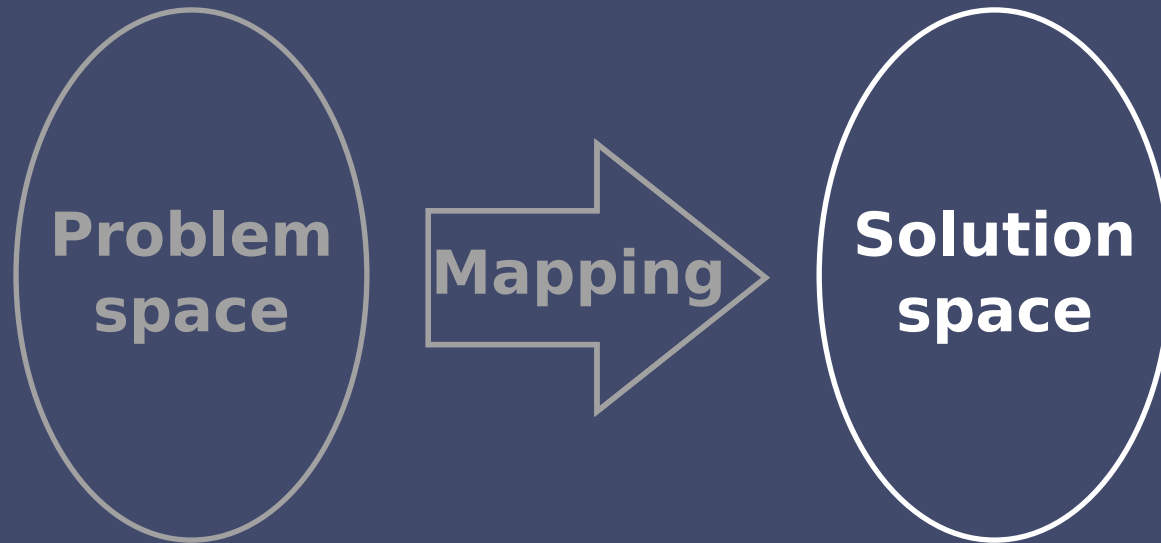
# Example: Telematics System



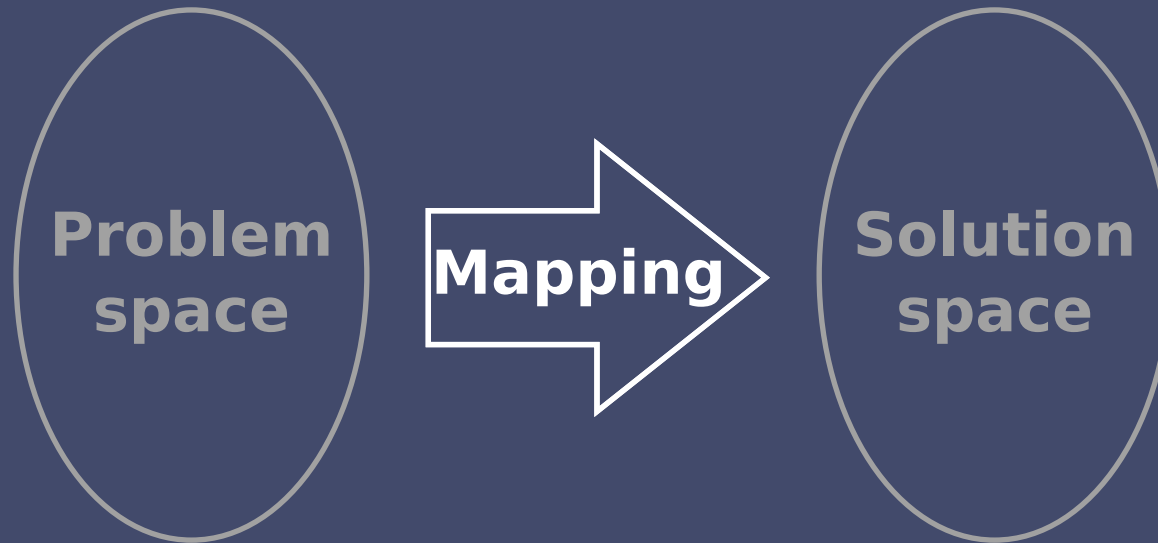
# Problem Space



# Solution Space

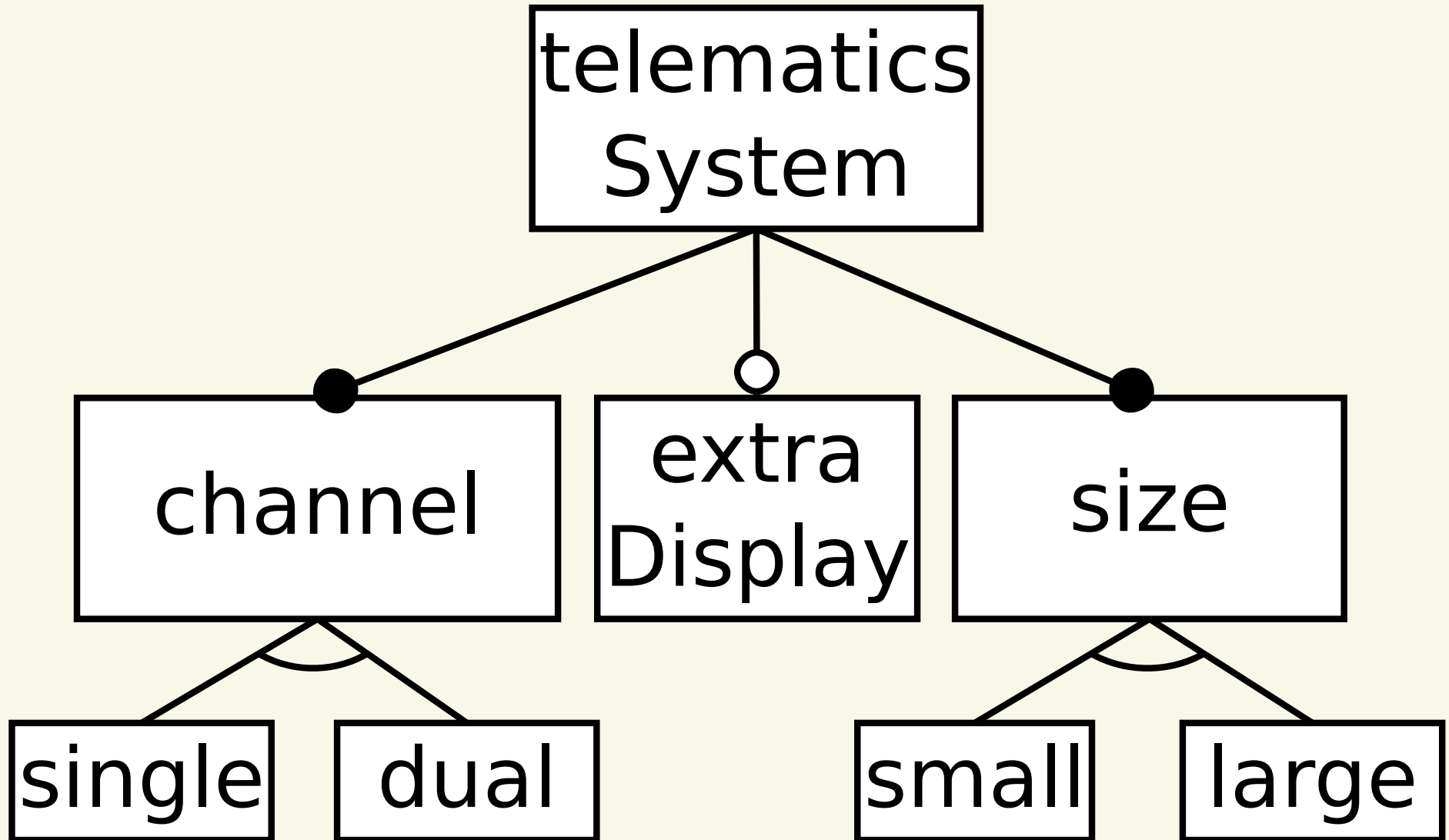


# Mapping





# Problem Space

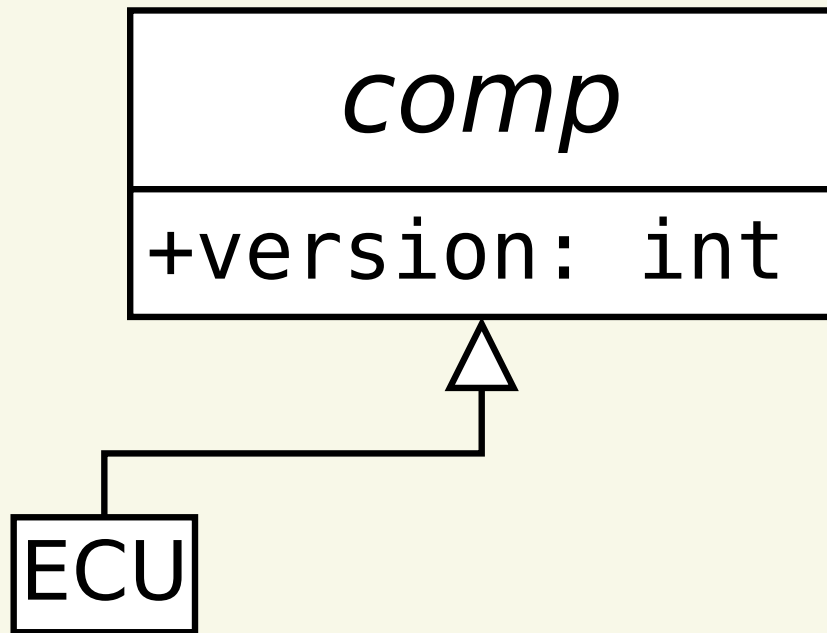


# Solution Space

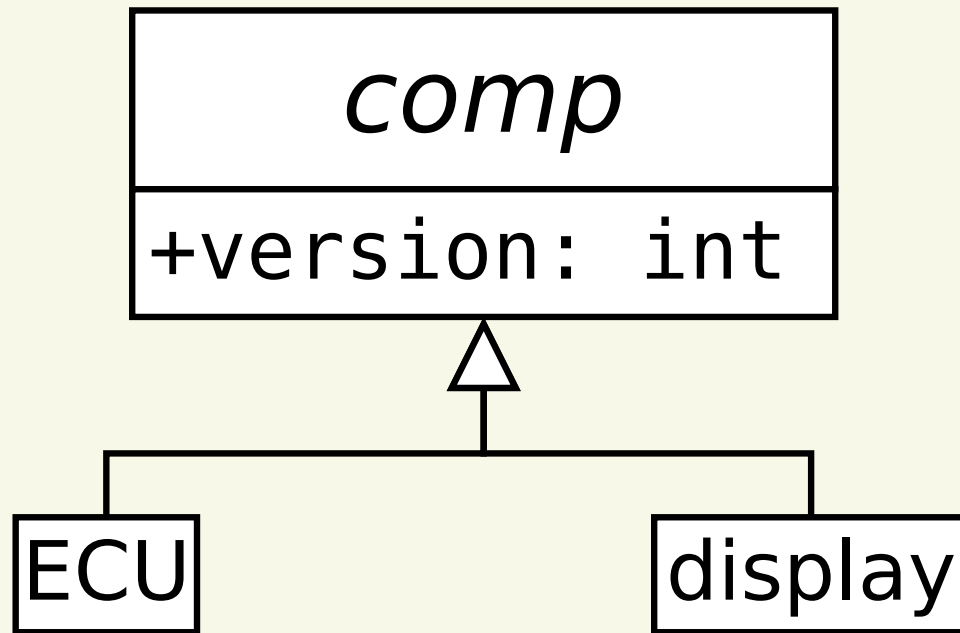
*comp*

+version: int

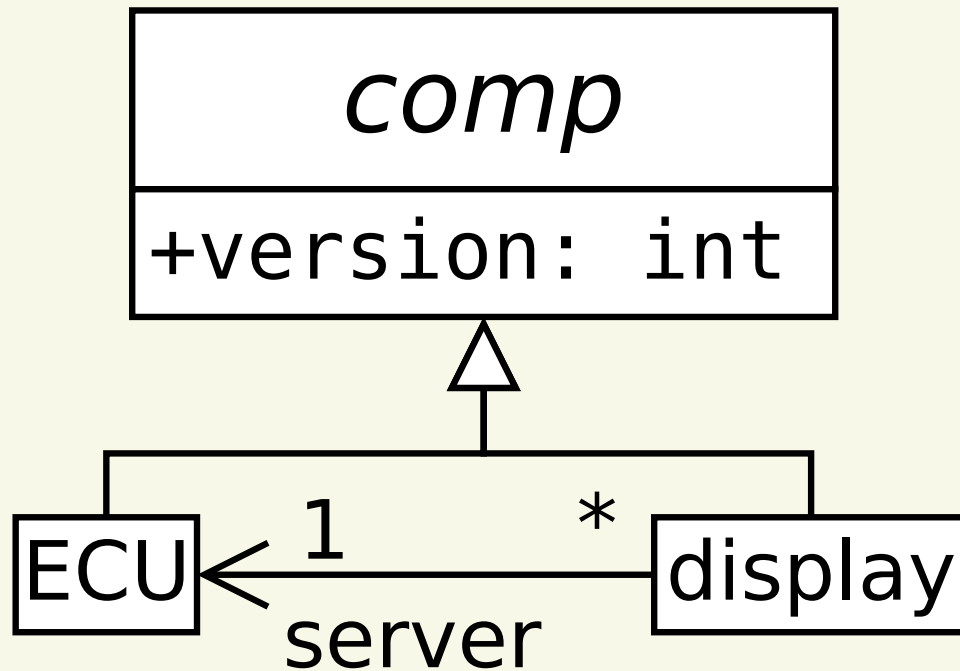
# Solution Space



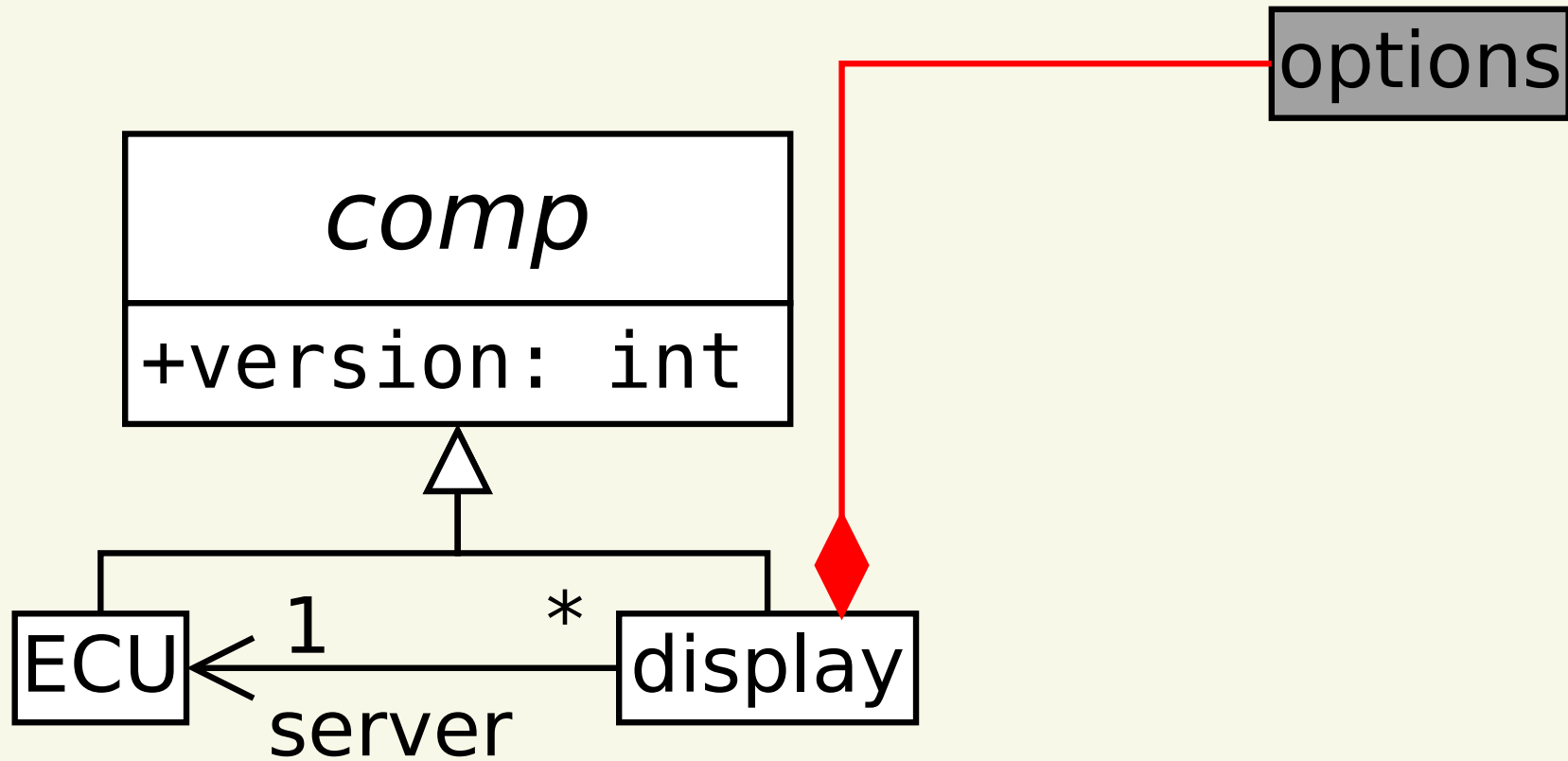
# Solution Space



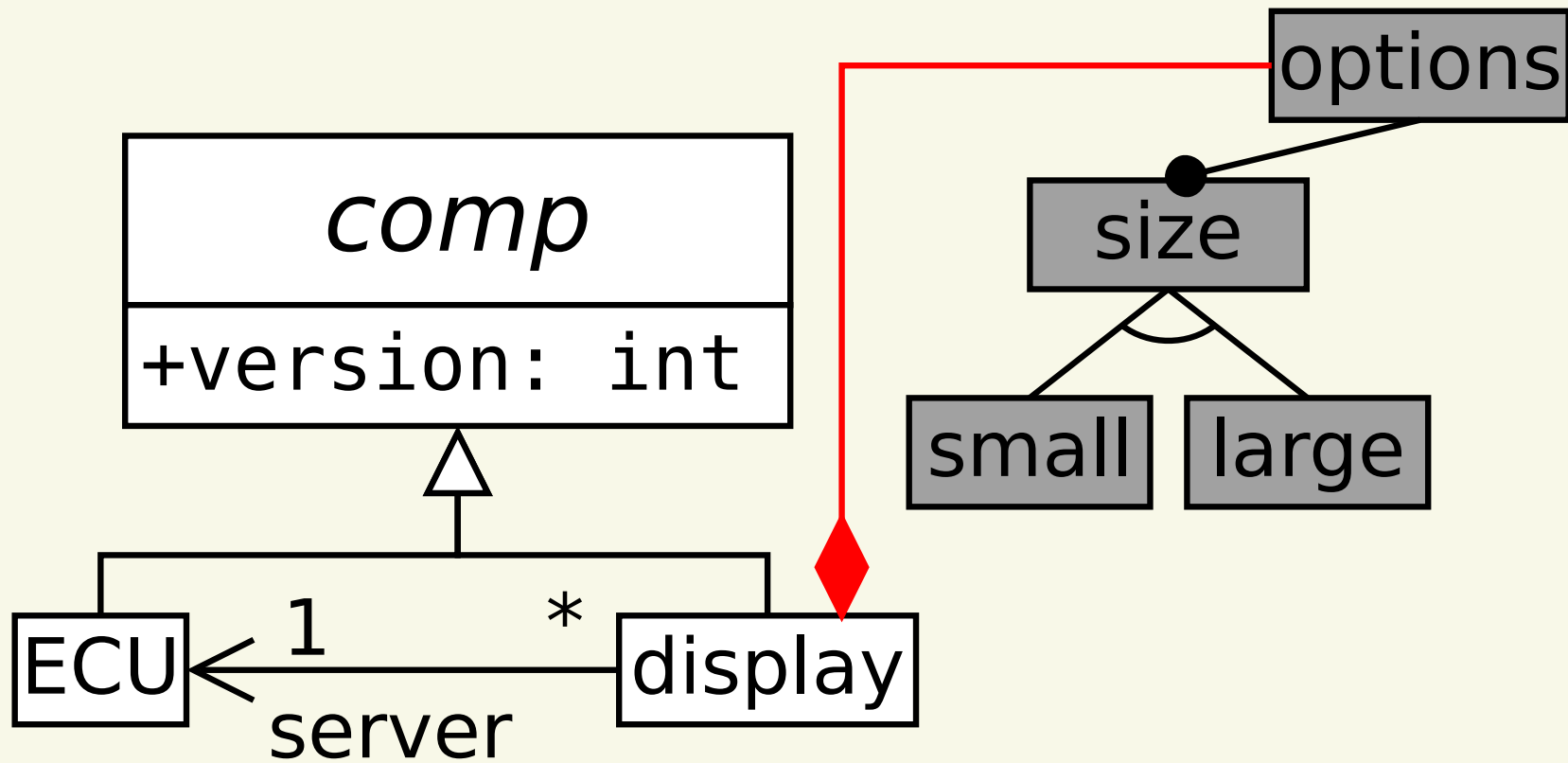
# Solution Space



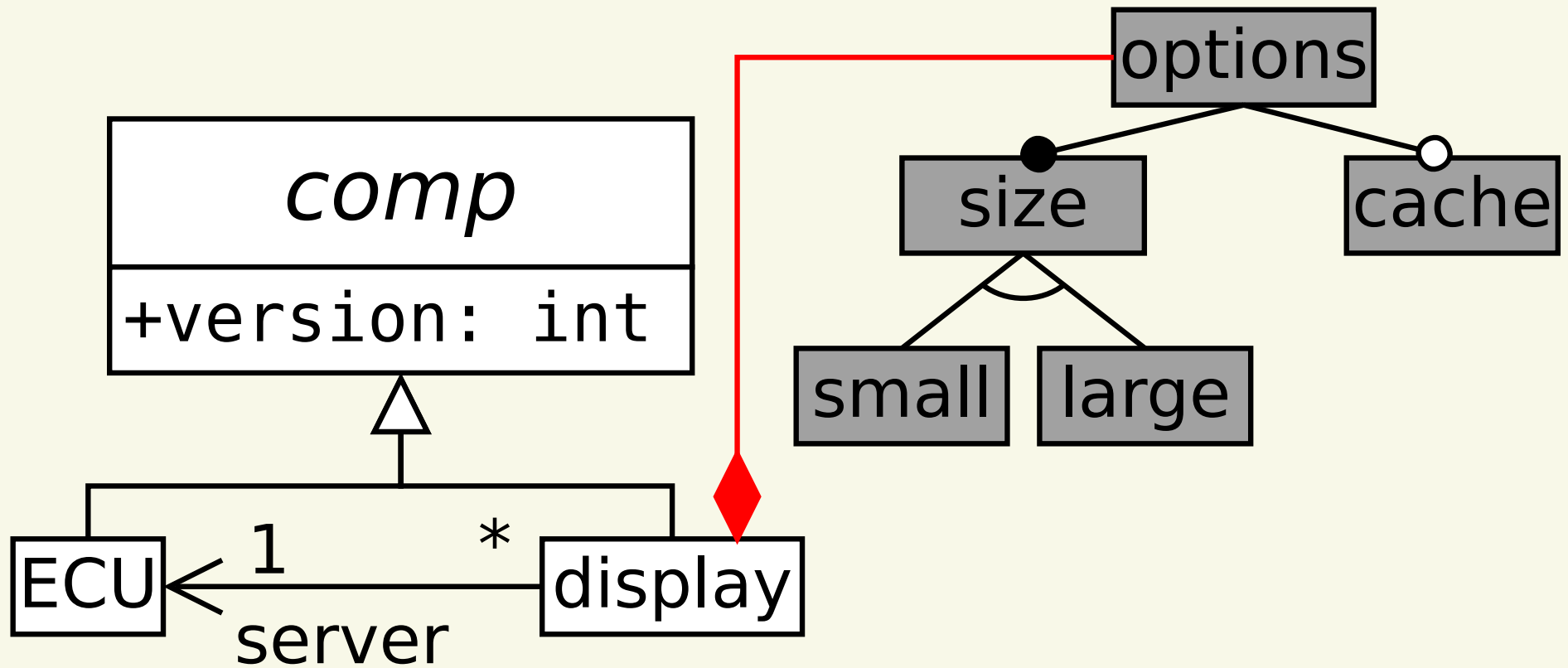
# Solution Space



# Solution Space

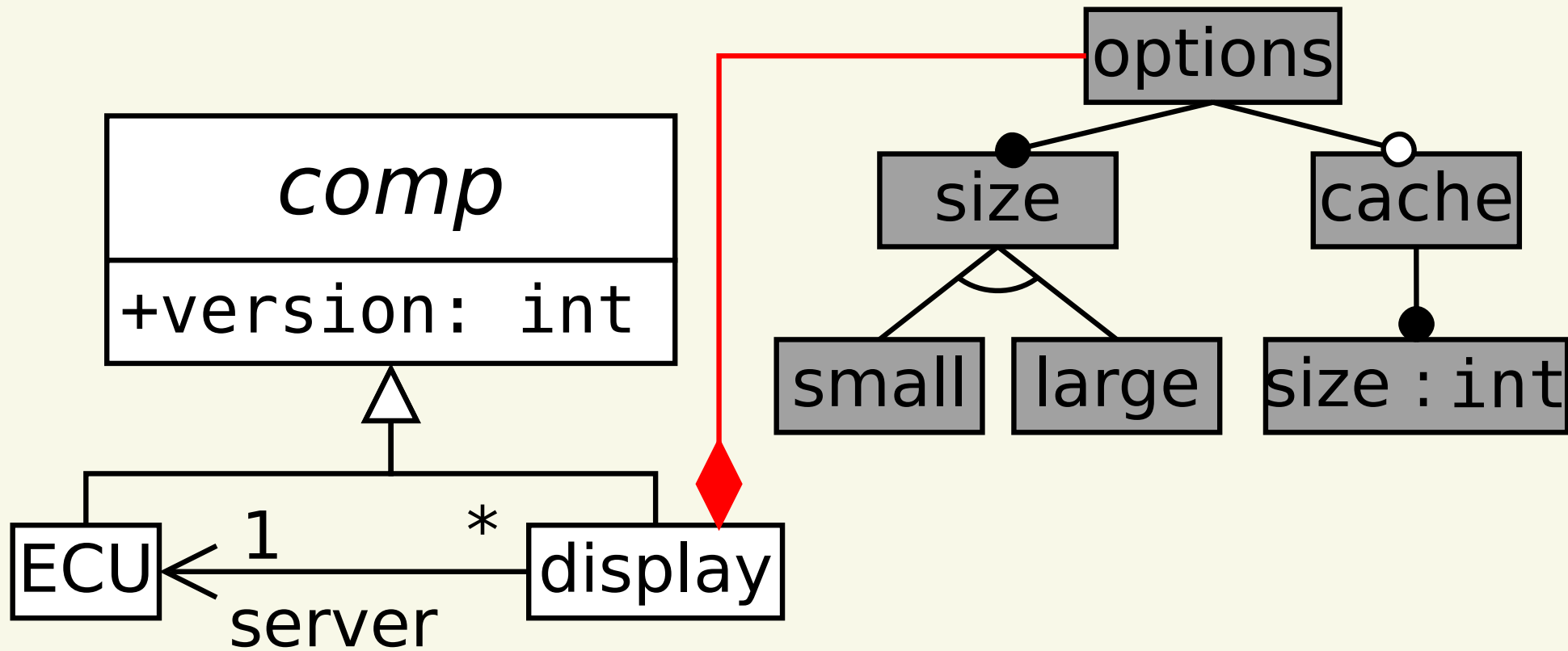


# Solution Space

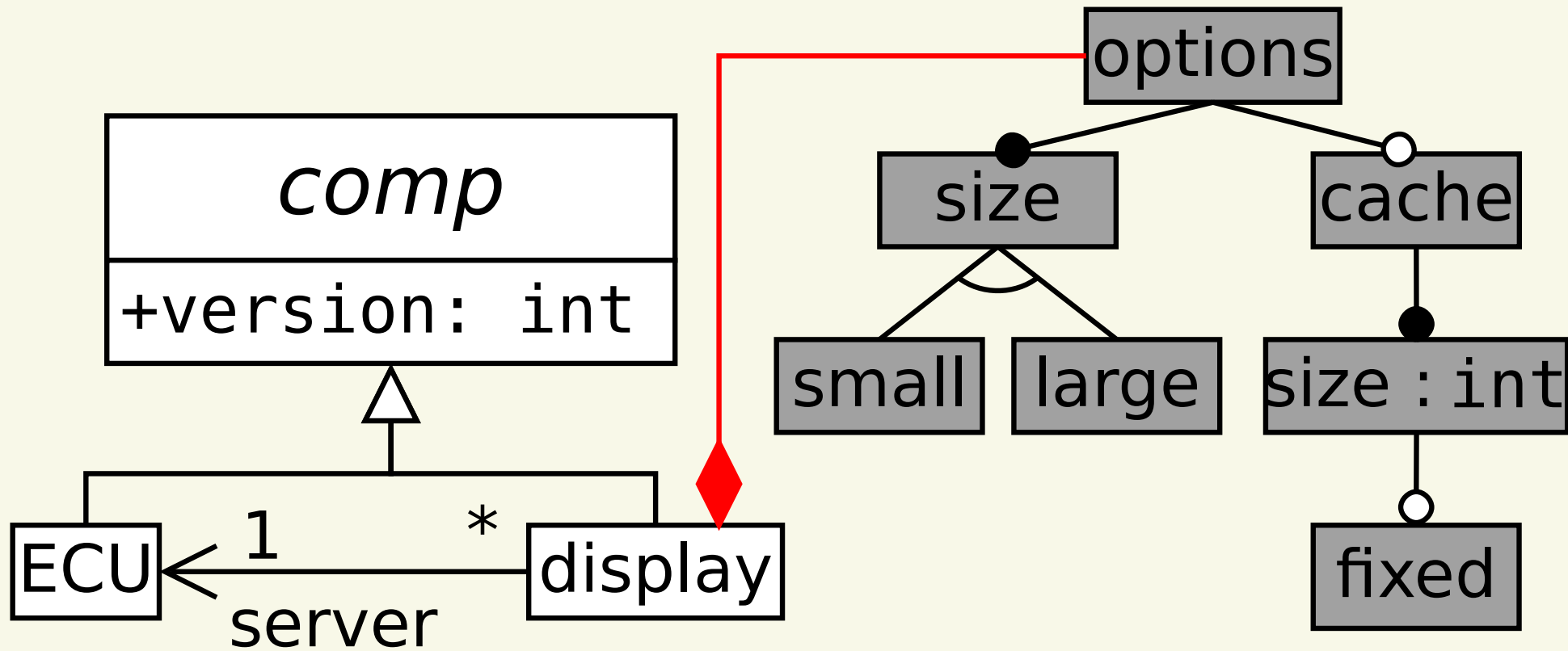




# Solution Space



# Solution Space



# Limitations (Motivations)

Single notation

# Motivation

Single notation

Components and options

# Motivation

Single notation

Components and options

Common infrastructure

# Motivation

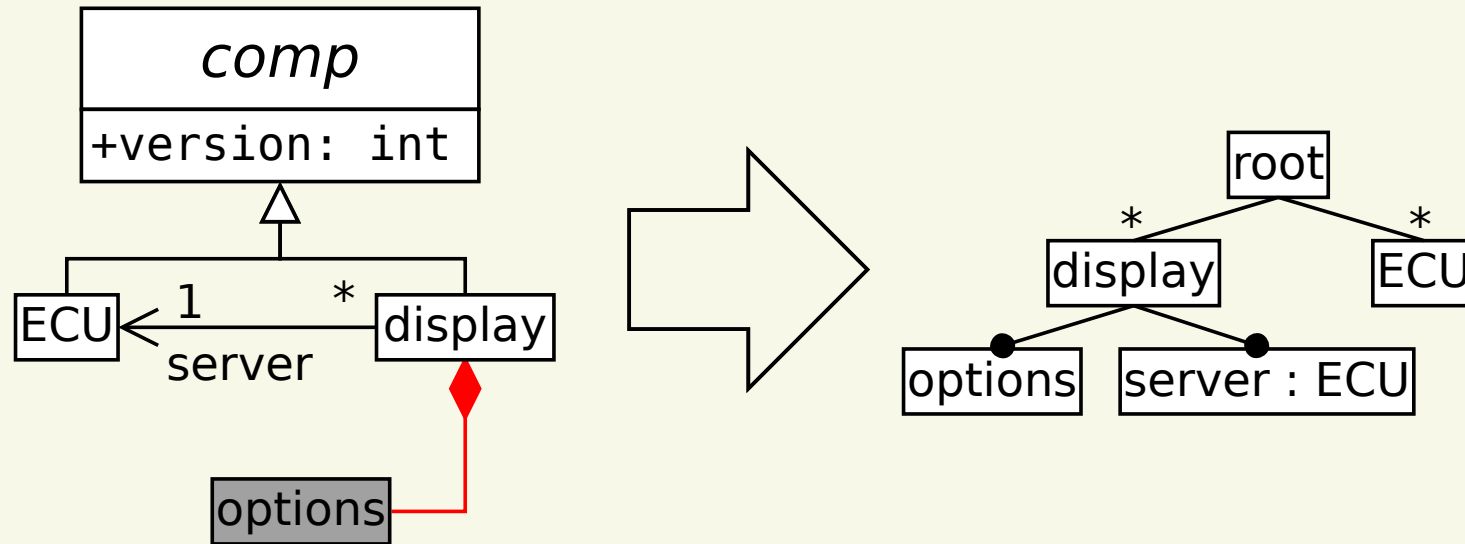
Single notation

Components and options

Common infrastructure

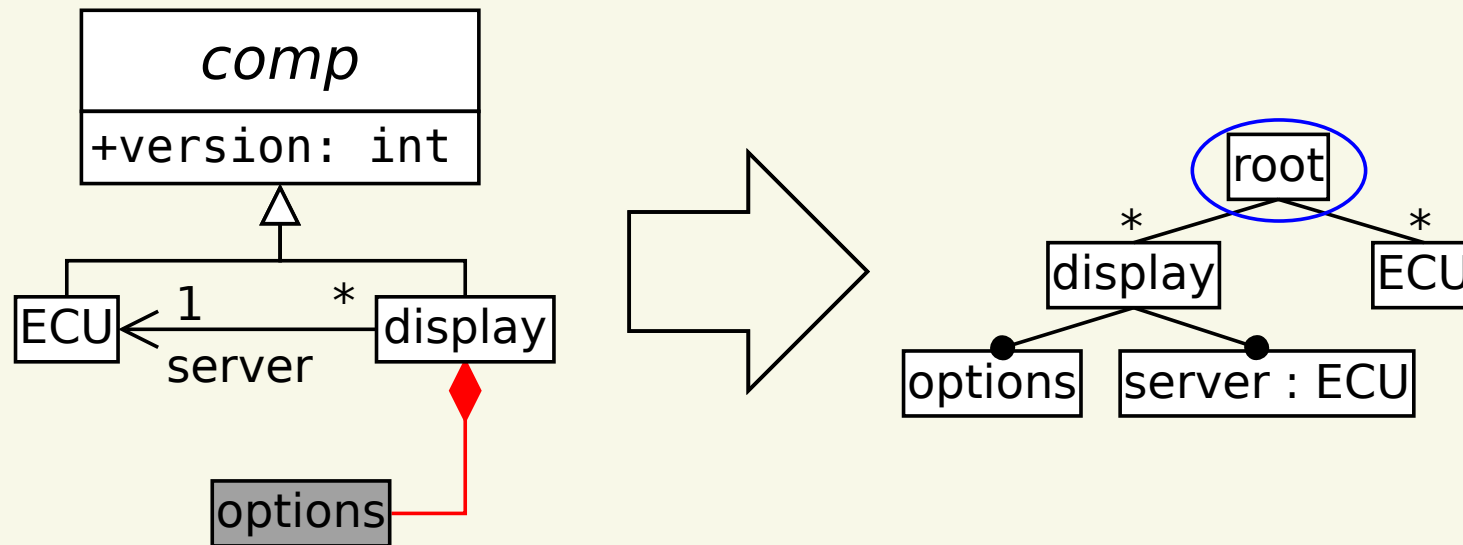
Model analysis

# Components as Cardinality-Based FM



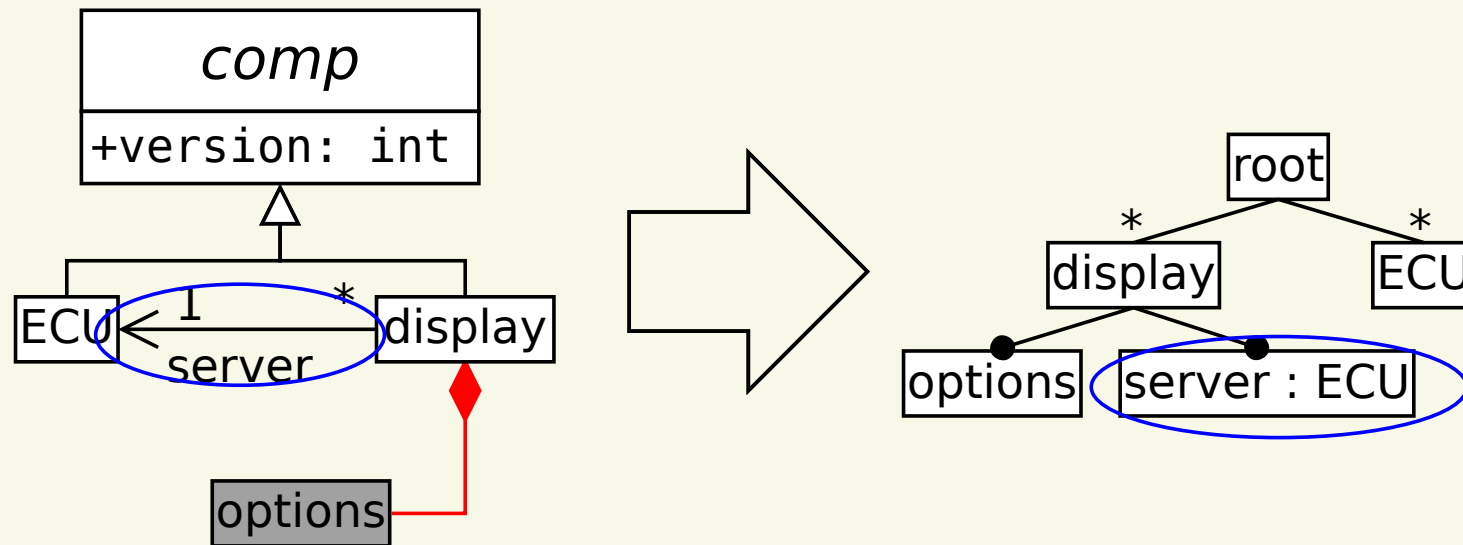


# Components as Cardinality-Based FM



Synthetic root

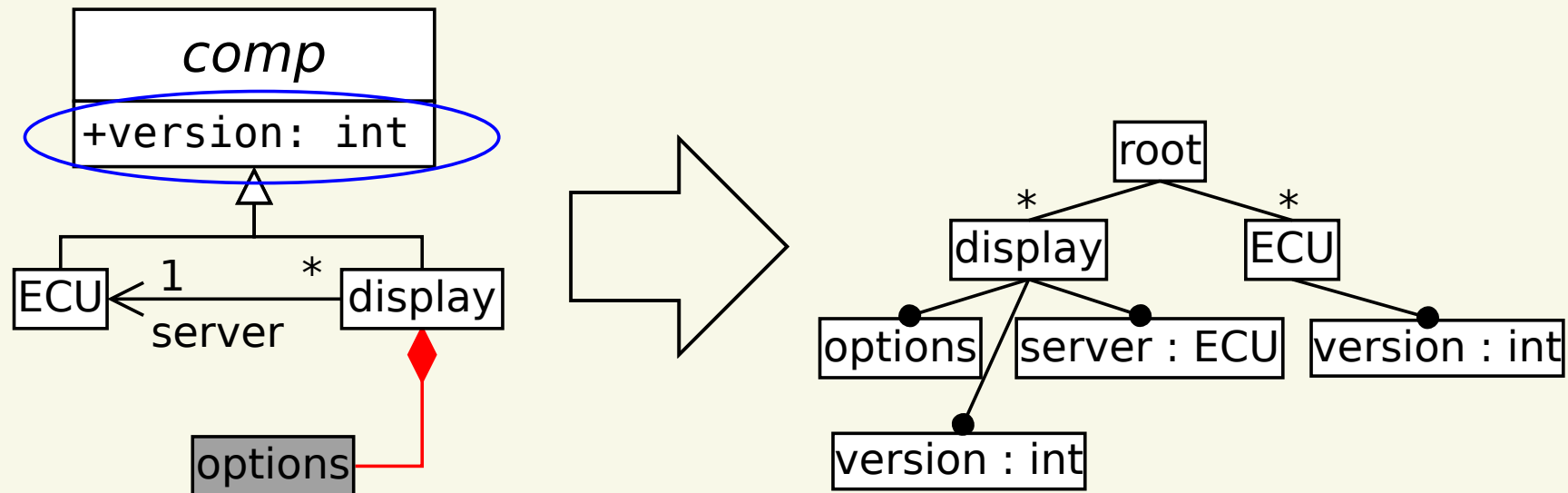
# Components as Cardinality-Based FM



Synthetic root

Reference to ECU

# Components as Cardinality-Based FM



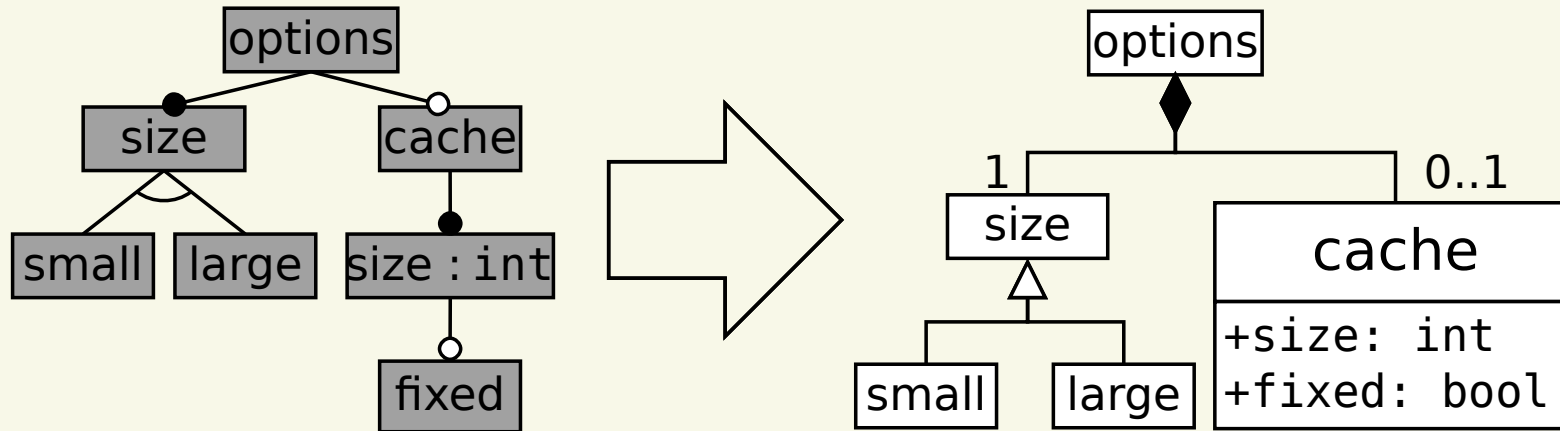
Synthetic root

Reference to ECU

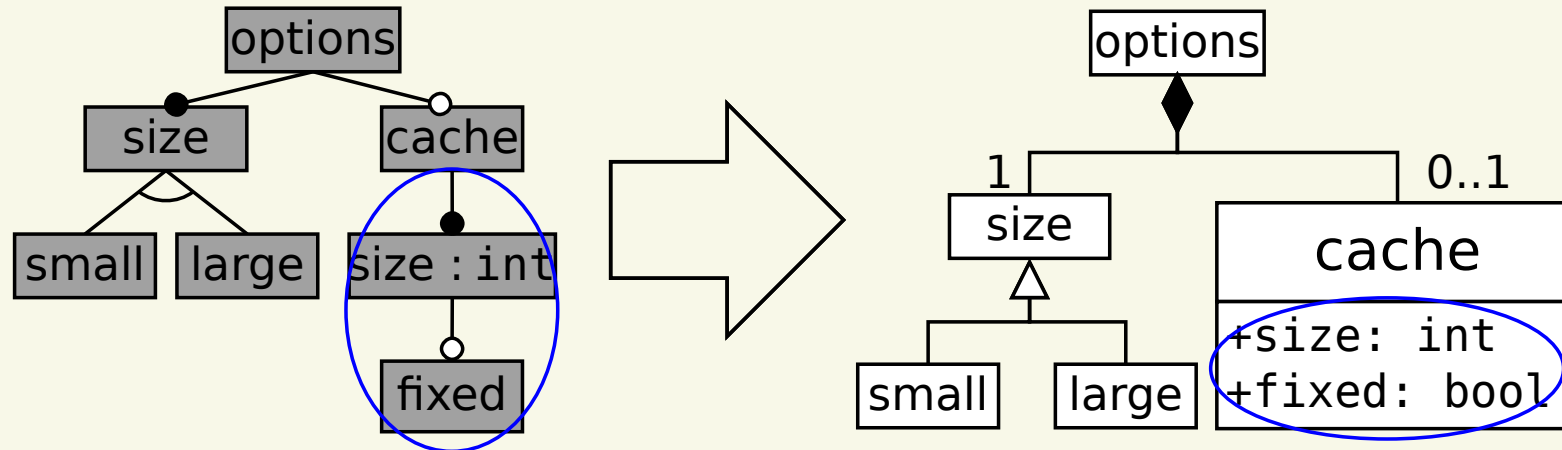
No reuse mechanism

# Cardinality-Based Feature Modeling Is a Misnomer

# Options as Meta-Model

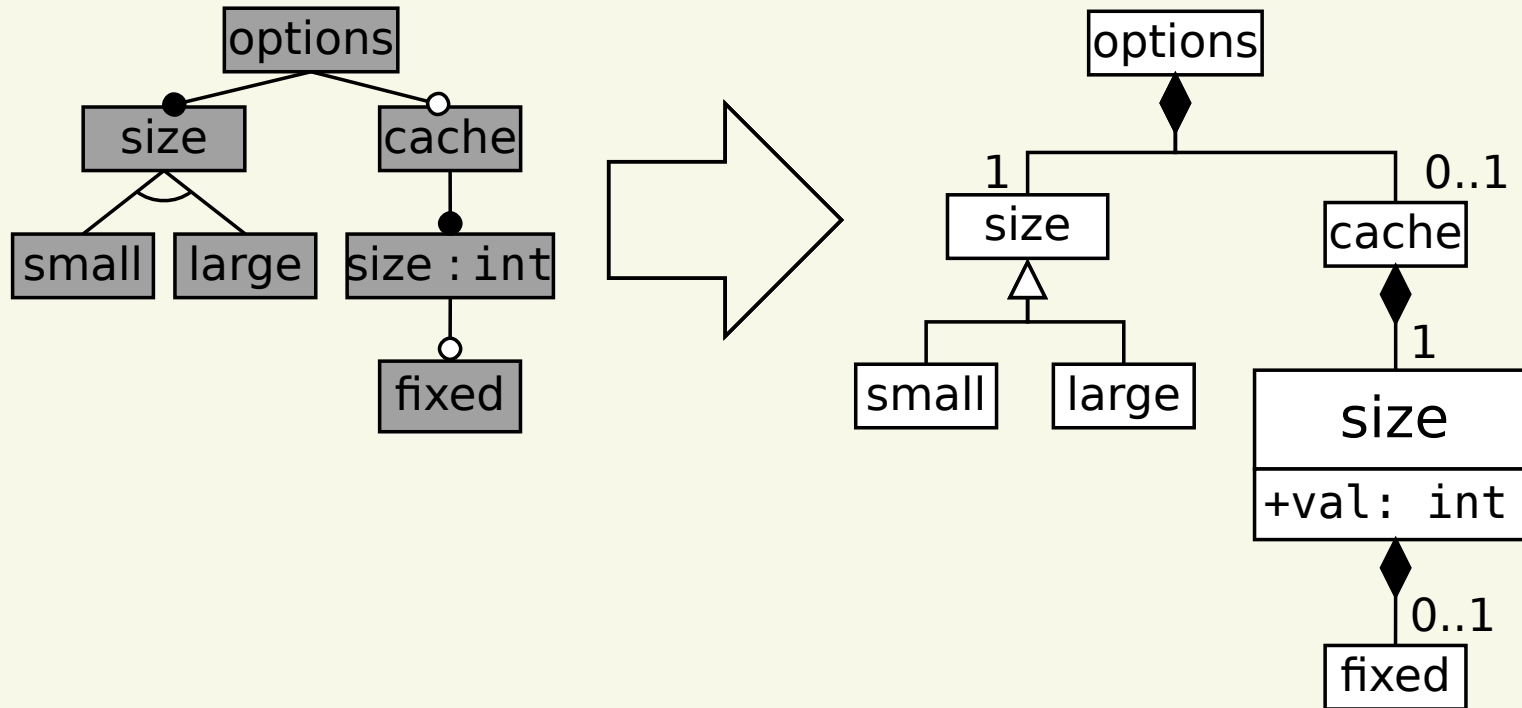


# Options as Meta-Model



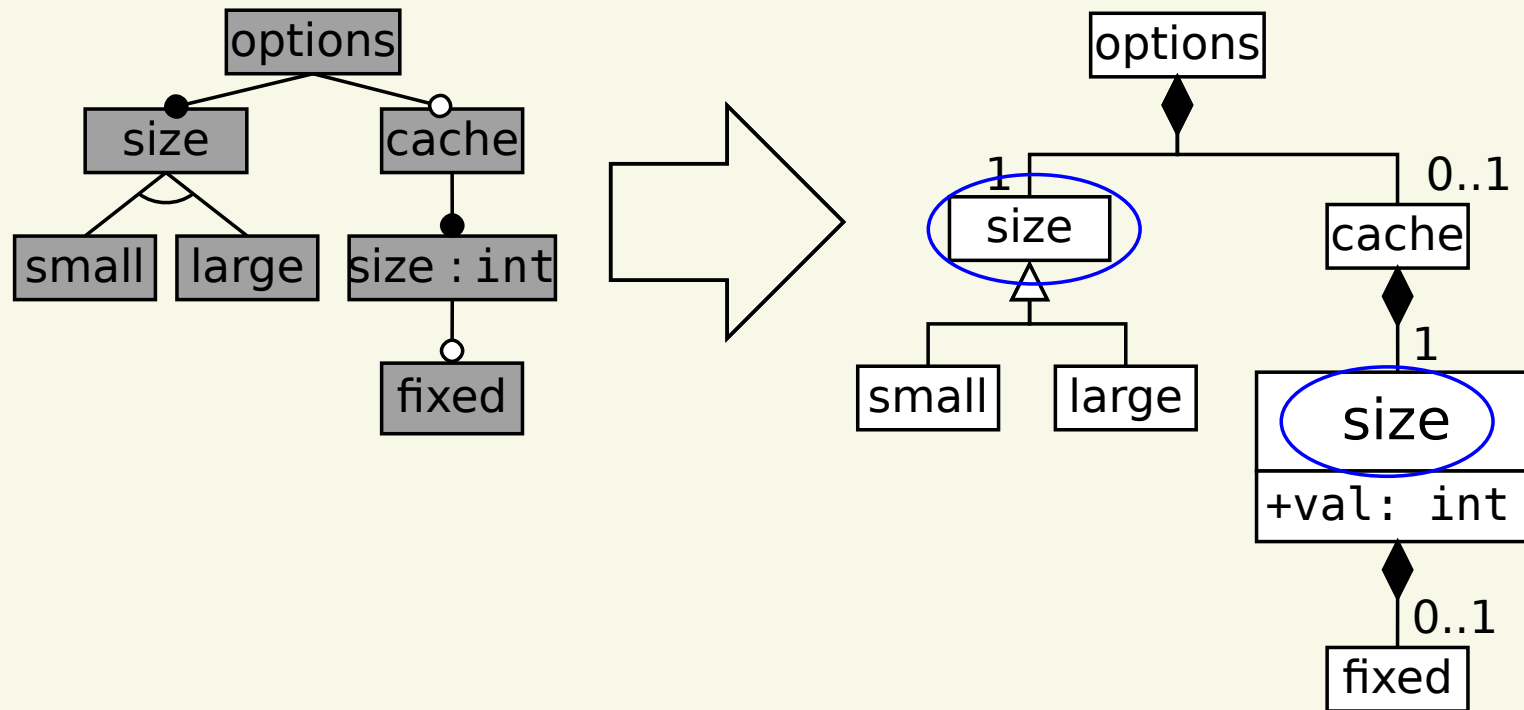
Lost feature nesting

# Options as Meta-Model



Lost feature nesting

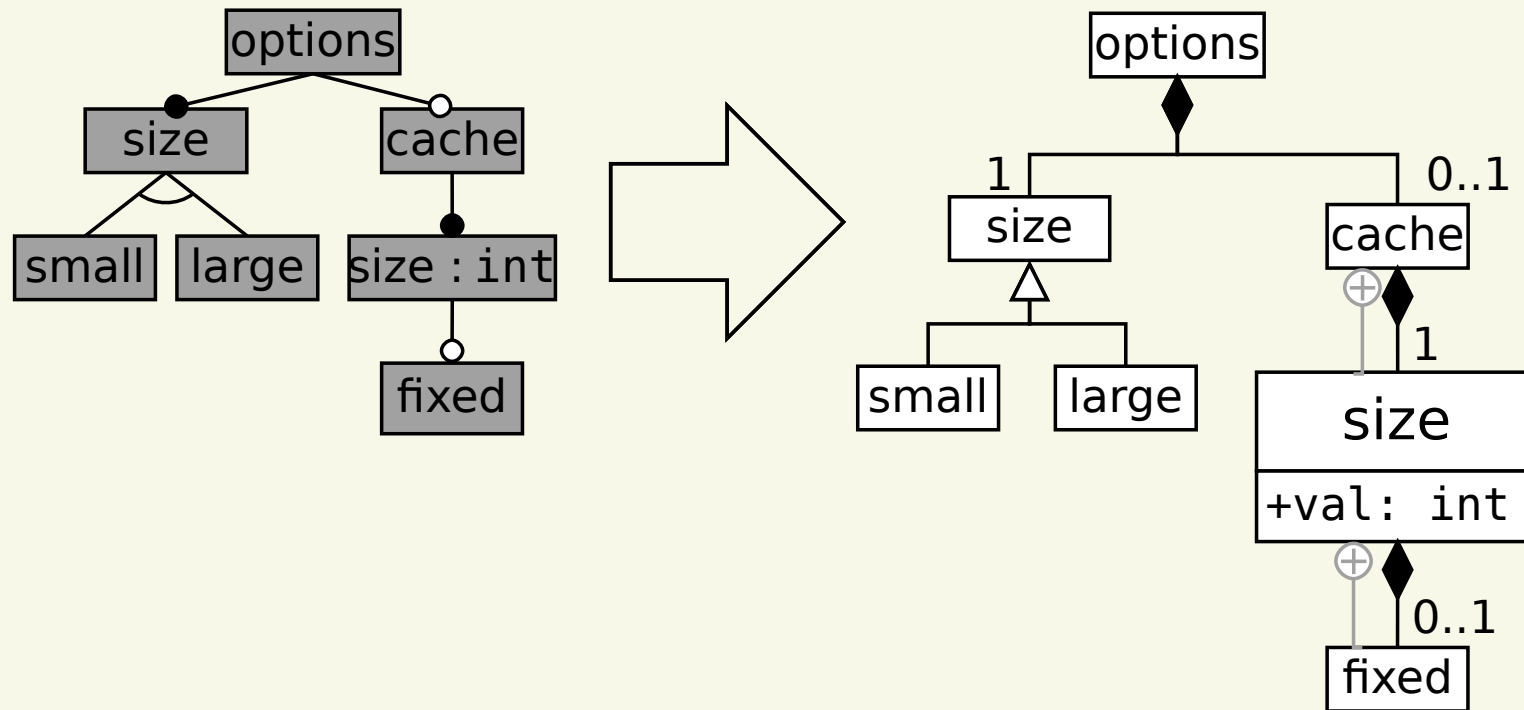
# Options as Meta-Model



Lost feature nesting, Name clash

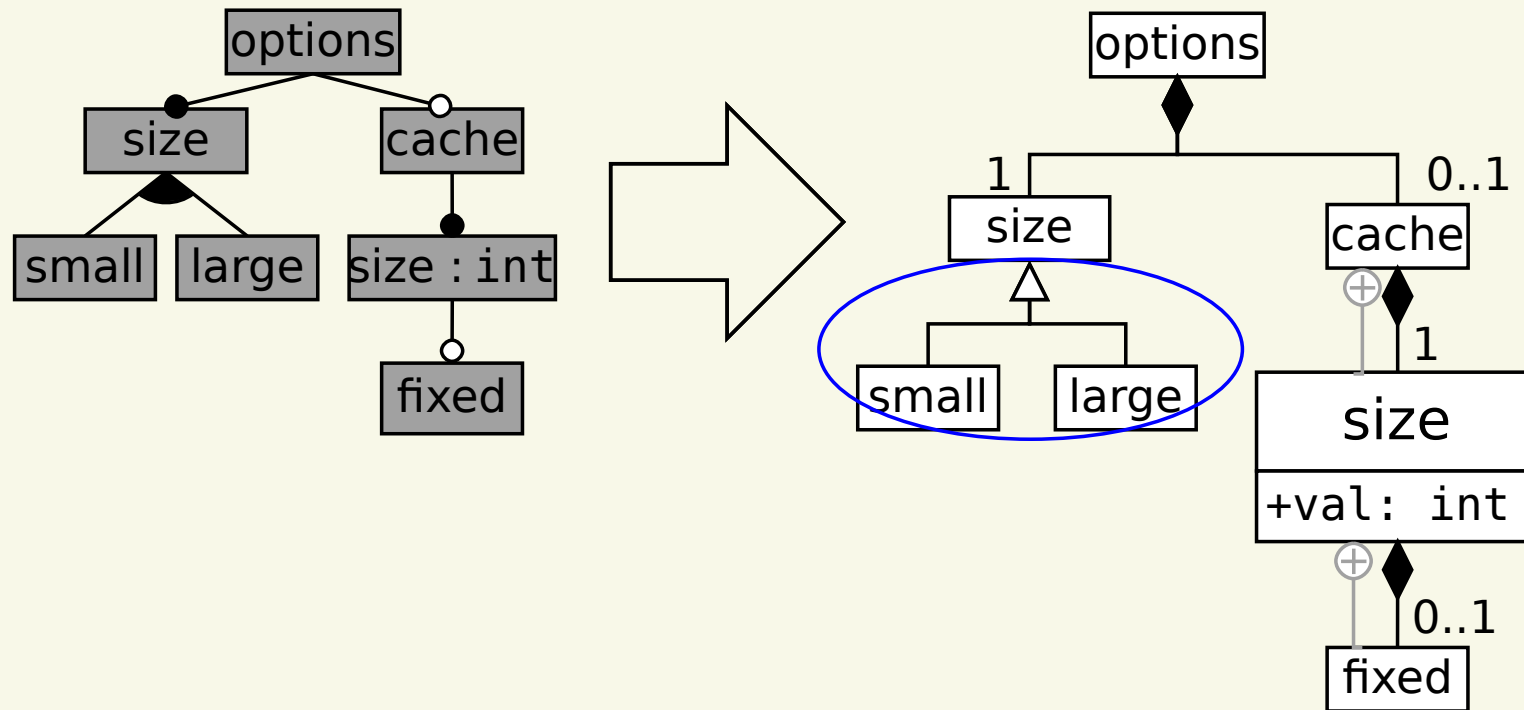


# Options as Meta-Model



Lost feature nesting, Name clash

# Options as Meta-Model



Lost feature nesting, Name clash

Group conversion problem

No First-Class Support  
for Feature Modeling

## 2. *Clafer*

(class, feature, reference)

Minimalistic modeling language

# Clafer Goals

Minimalistic modeling language

Concise notation for FM and MM

# Clafer Goals

Minimalistic modeling language

Concise notation for FM and MM

Mixing feature and meta-models

# Clafer Goals

Minimalistic modeling language

Concise notation for FM and MM

Mixing feature and meta-models

Uniform semantics



# Clafer Goals

Minimalistic modeling language

Concise notation for FM and MM

Mixing feature and meta-models

Uniform semantics

Constraints

# Clafer Goals

Minimalistic modeling language

Concise notation for FM and MM

Mixing feature and meta-models

Uniform semantics

Constraints

Reasoning

# Problem Space

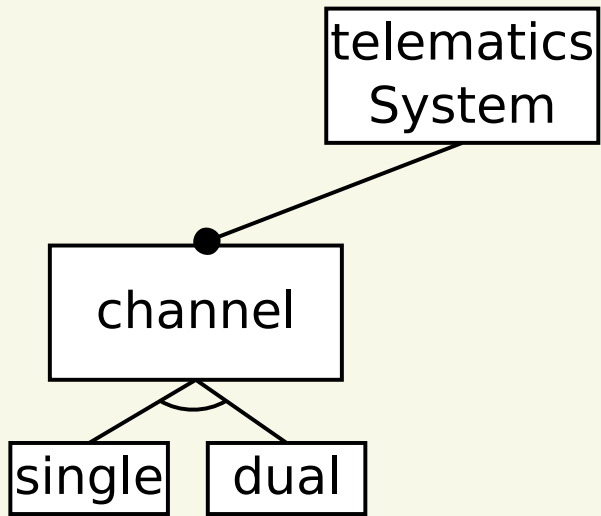


# Feature Model

telematics  
System

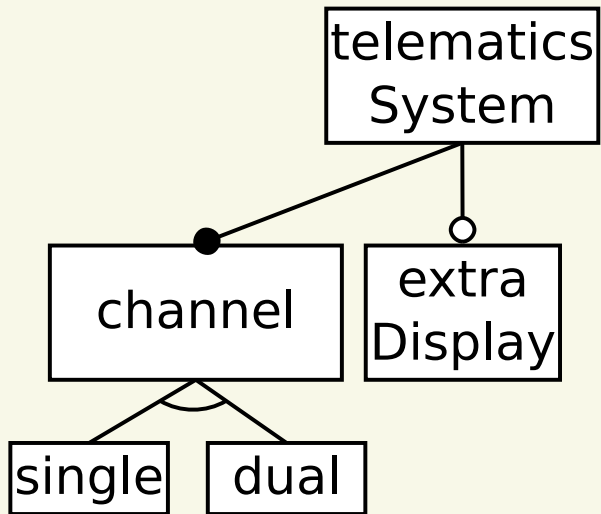
telematicsSystem

# Feature Model



telematicsSystem  
**xor** channel  
single  
dual

# Feature Model



telematicsSystem

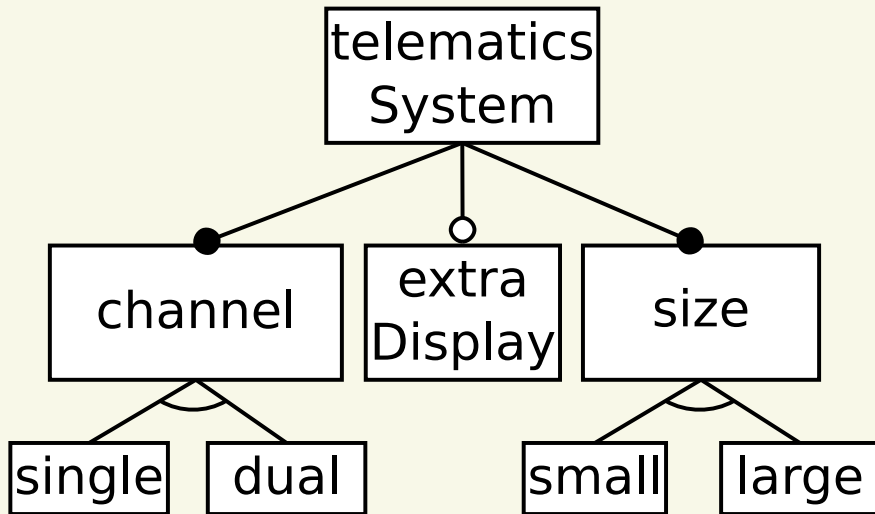
**xor** channel

single

dual

extraDisplay ?

# Feature Model



telematicsSystem

**xor** channel

single

dual

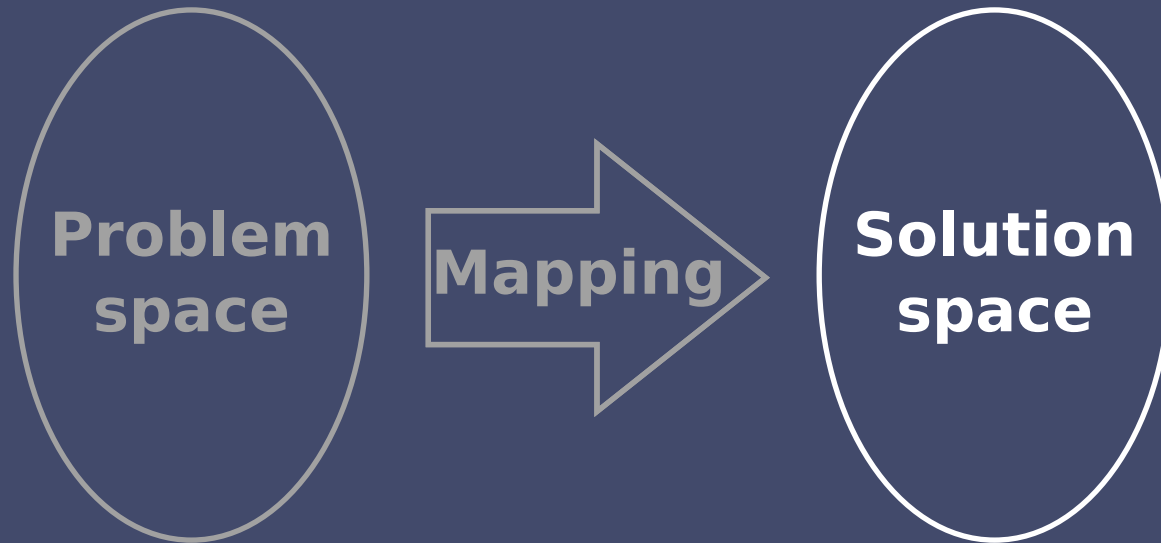
extraDisplay ?

**xor** size

small

large

# Solution Space



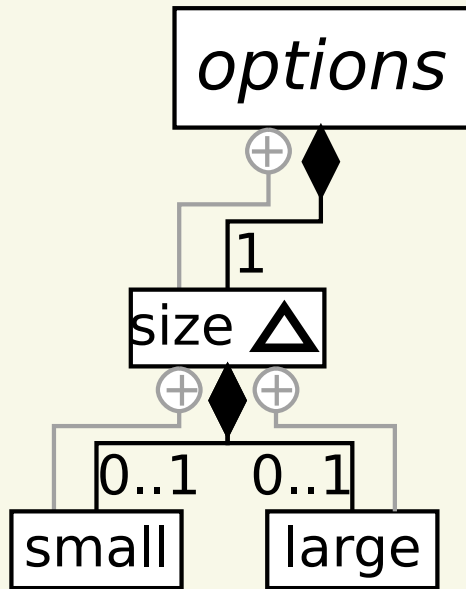


# Options

*options*

**abstract options**

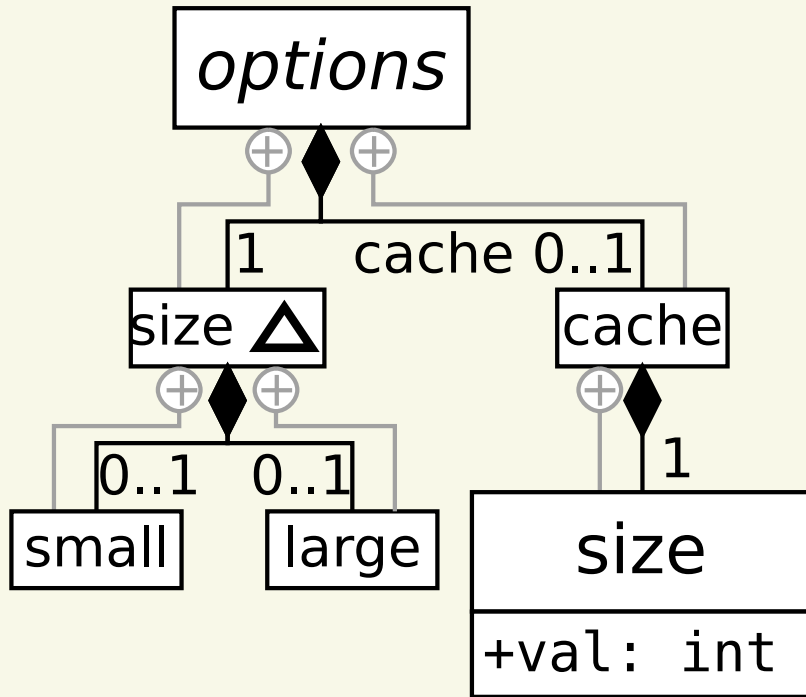
# Options



**abstract** options  
**xor** size  
small  
large



# Options



**abstract options**

**xor size**

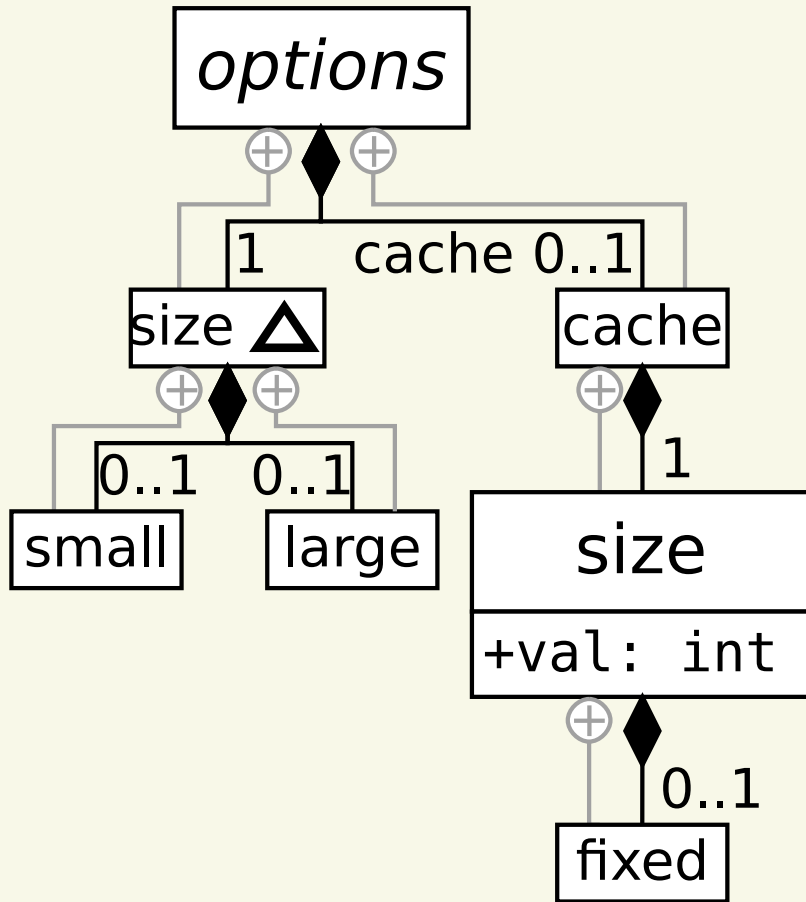
small

large

cache ?

size → integer

# Options



**abstract options**

**xor size**

small

large

cache ?

size  $\rightarrow$  integer

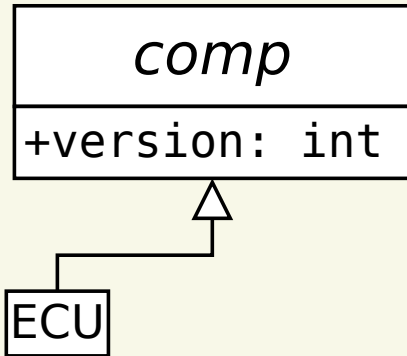
fixed ?

# Mixed: Meta-Model with Options

<i>comp</i>
+version: int

**abstract comp**  
version → integer

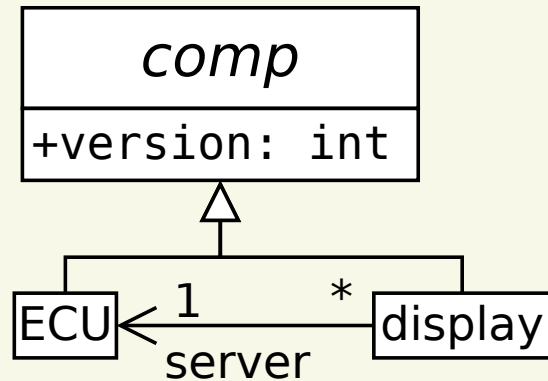
# Mixed: Meta-Model with Options



**abstract** *comp*  
version → integer

**abstract** `ECU` **extends** *comp*

# Mixed: Meta-Model with Options



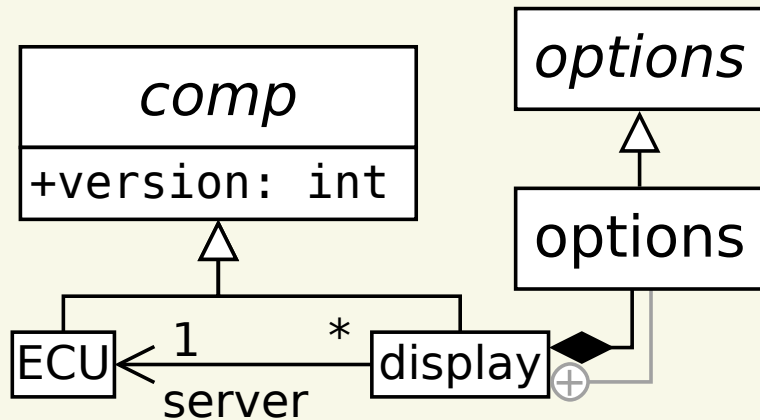
**abstract** comp  
version → integer

**abstract** ECU **extends** comp

**abstract** display **extends** comp  
server → ECU



# Mixed: Meta-Model with Options

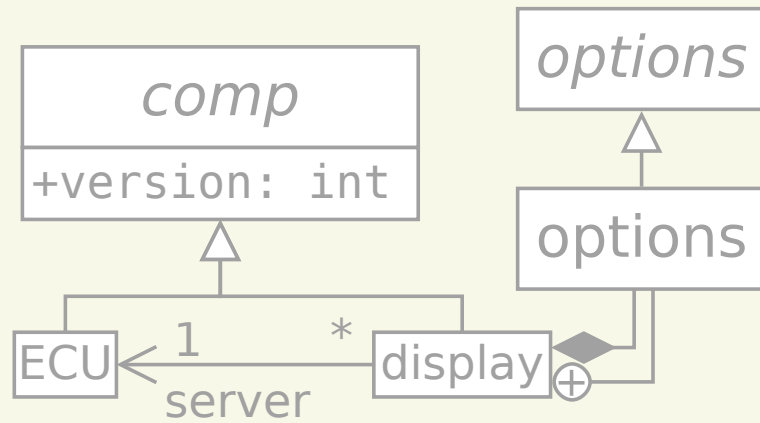


**abstract comp**  
version → integer

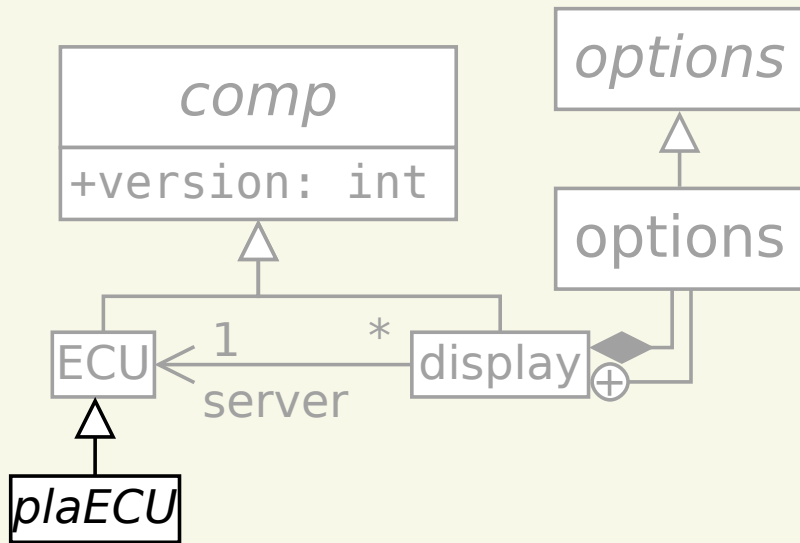
**abstract ECU extends comp**

**abstract display extends comp**  
server → ECU  
`options

# Specialized: Architectural Template

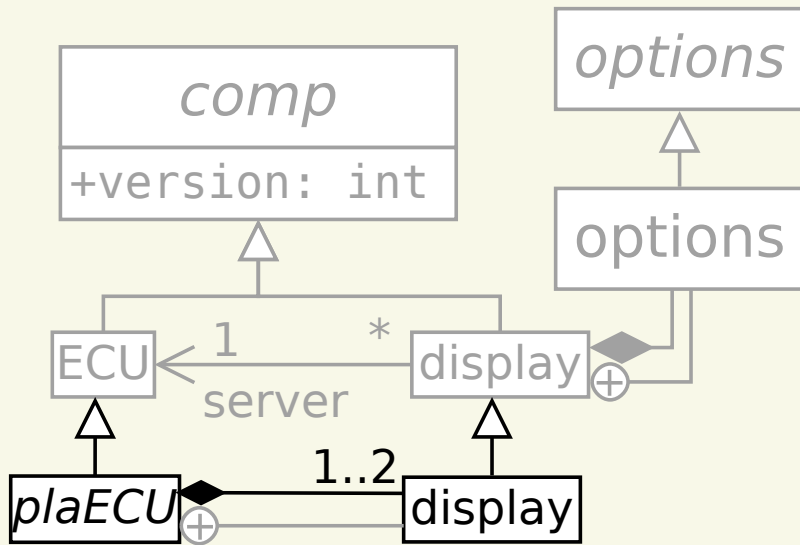


# Specialized: Architectural Template



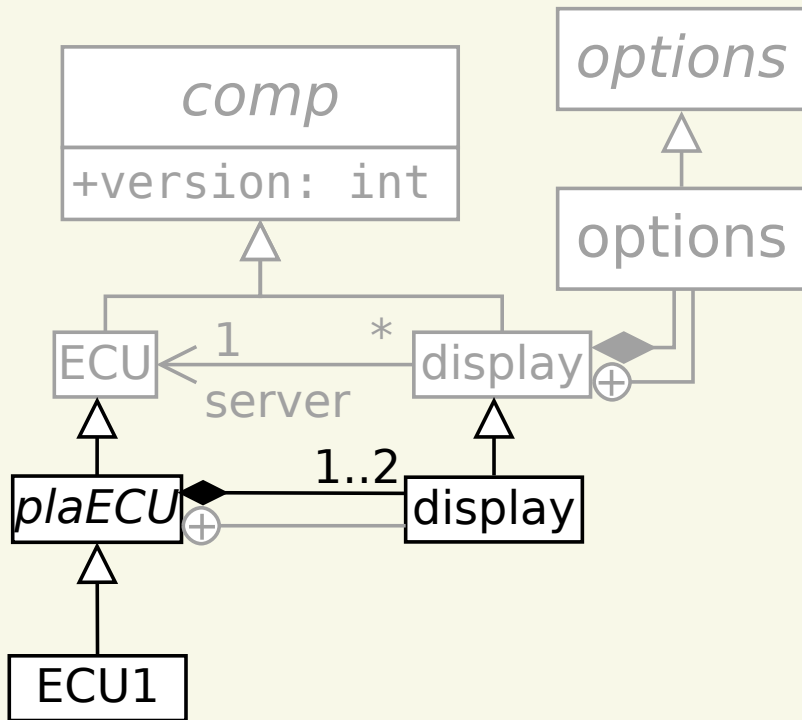
**abstract** plaECU extends ECU

# Specialized: Architectural Template



**abstract** plaECU **extends** ECU  
`display 1..2  
[server = parent  
¬options.cache]

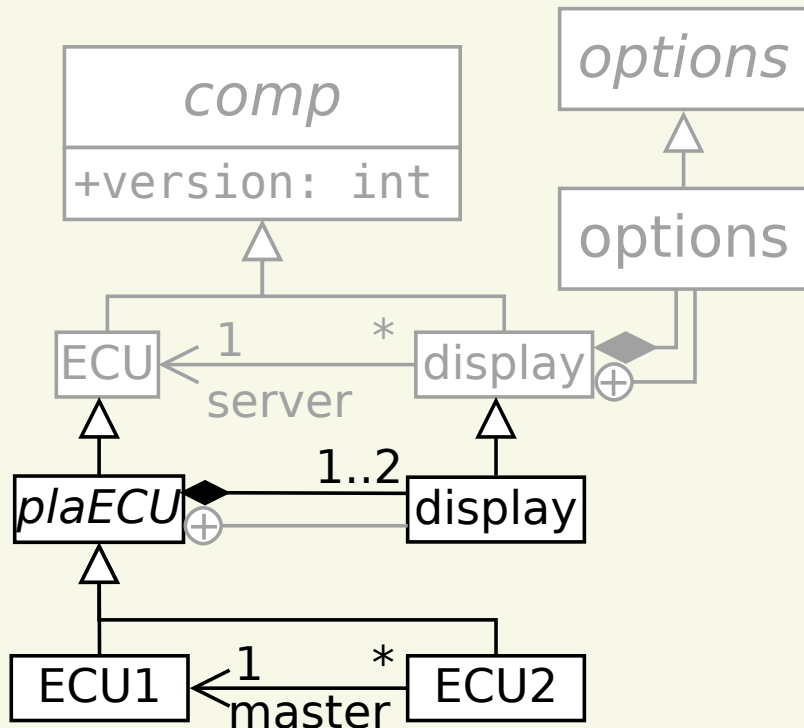
# Specialized: Architectural Template



**abstract** plaECU **extends** ECU  
`display 1..2  
[server = parent  
¬options.cache]

**ECU1 extends** plaECU

# Specialized: Architectural Template

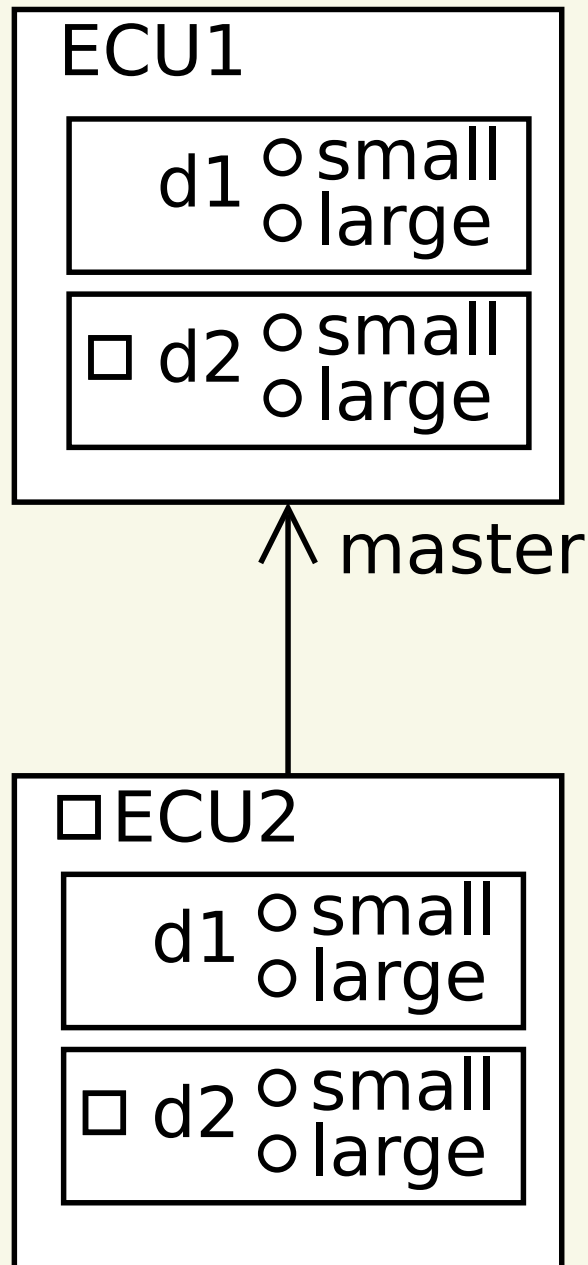


**abstract** plaECU **extends** ECU  
`display 1..2  
[server = parent  
¬options.cache]

**ECU1 extends** plaECU

**ECU2 extends** plaECU ?  
master → ECU1

# Template Visualization



**abstract** plaECU **extends** ECU

``display 1..2`

`[server = parent`

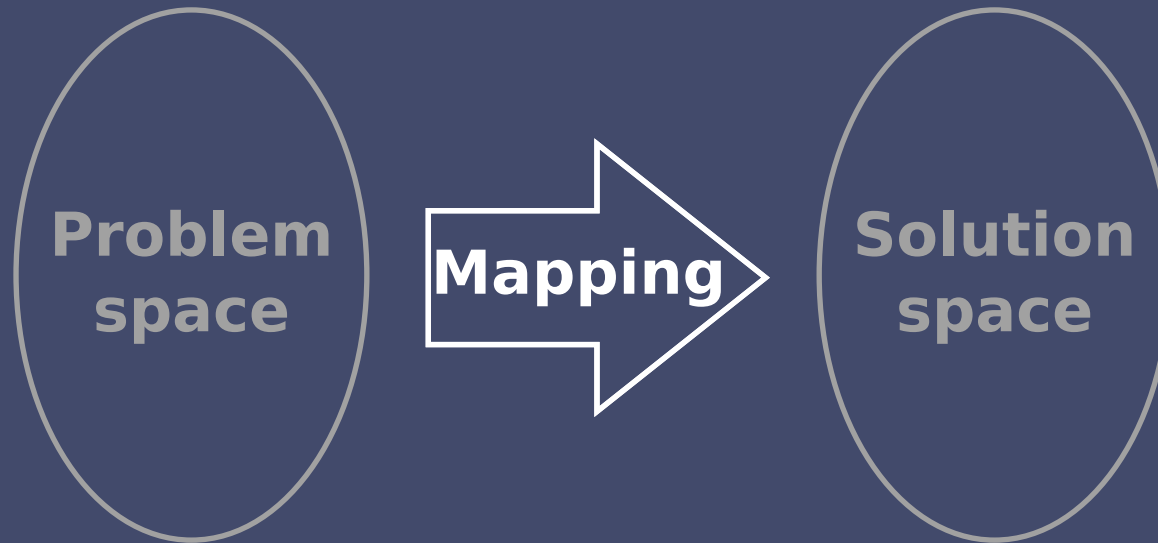
`¬options.cache]`

ECU1 **extends** plaECU

ECU2 **extends** plaECU ?

master → ECU1

# Mapping





# Coupled: Mapping Constraints

[dual  $\Leftrightarrow$  ECU2

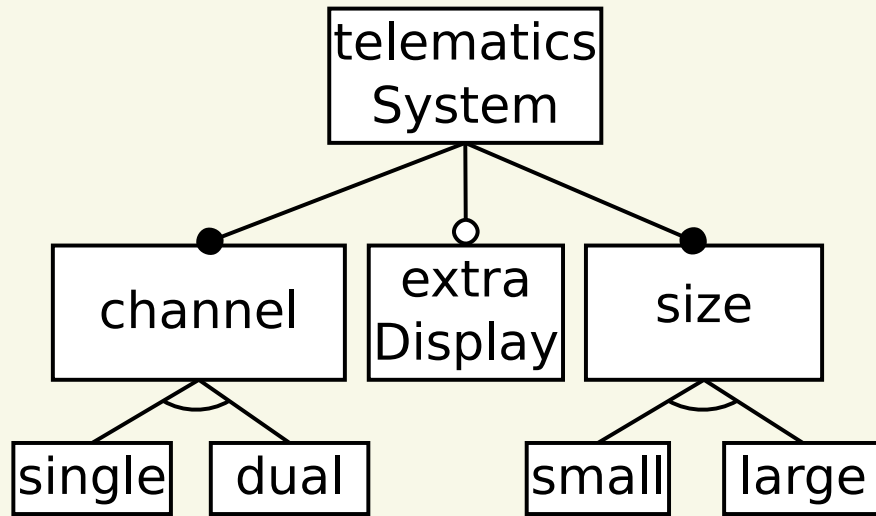
extraDisplay  $\Leftrightarrow$  #ECU1.display = 2

extraDisplay  $\Leftrightarrow$  (ECU2  $\Leftrightarrow$  #ECU2.display = 2)

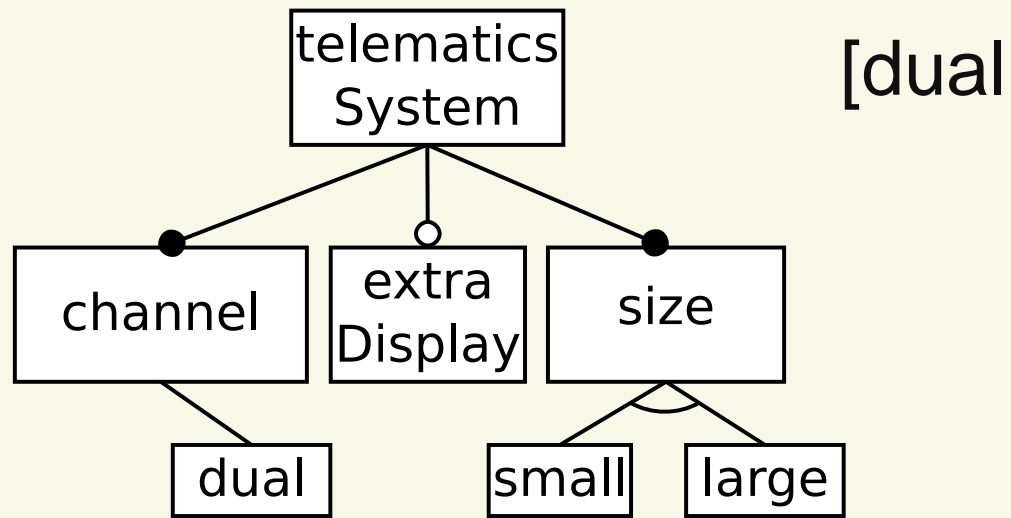
large  $\Leftrightarrow$   $\neg$ plaECU.display.options.size.small

small  $\Leftrightarrow$   $\neg$ plaECU.display.options.size.large]

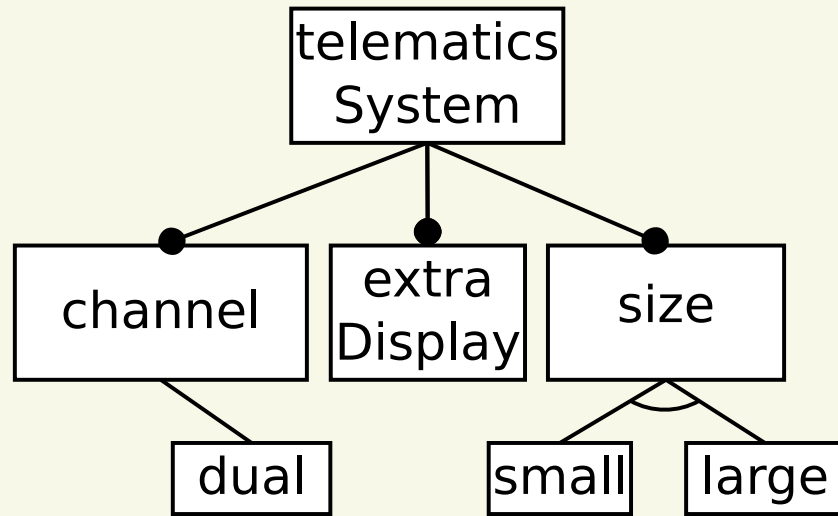
# Configured: Constraints



# Configured: Constraints

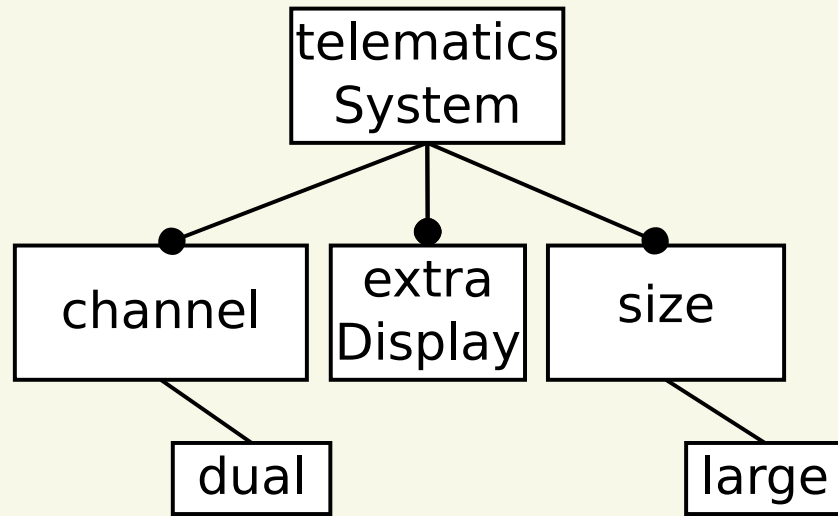


# Configured: Constraints



[dual  
extraDisplay

# Configured: Constraints



[dual  
extraDisplay  
telematicsSystem.size.large]

# Reasoning: Clafer to Alloy

Clafer to Alloy

# Reasoning: Clafer to Alloy

Clafer to Alloy

Structural modeling

# Reasoning: Clafer to Alloy

Clafer to Alloy

Structural modeling

Model checking



# Reasoning: Clafer to Alloy

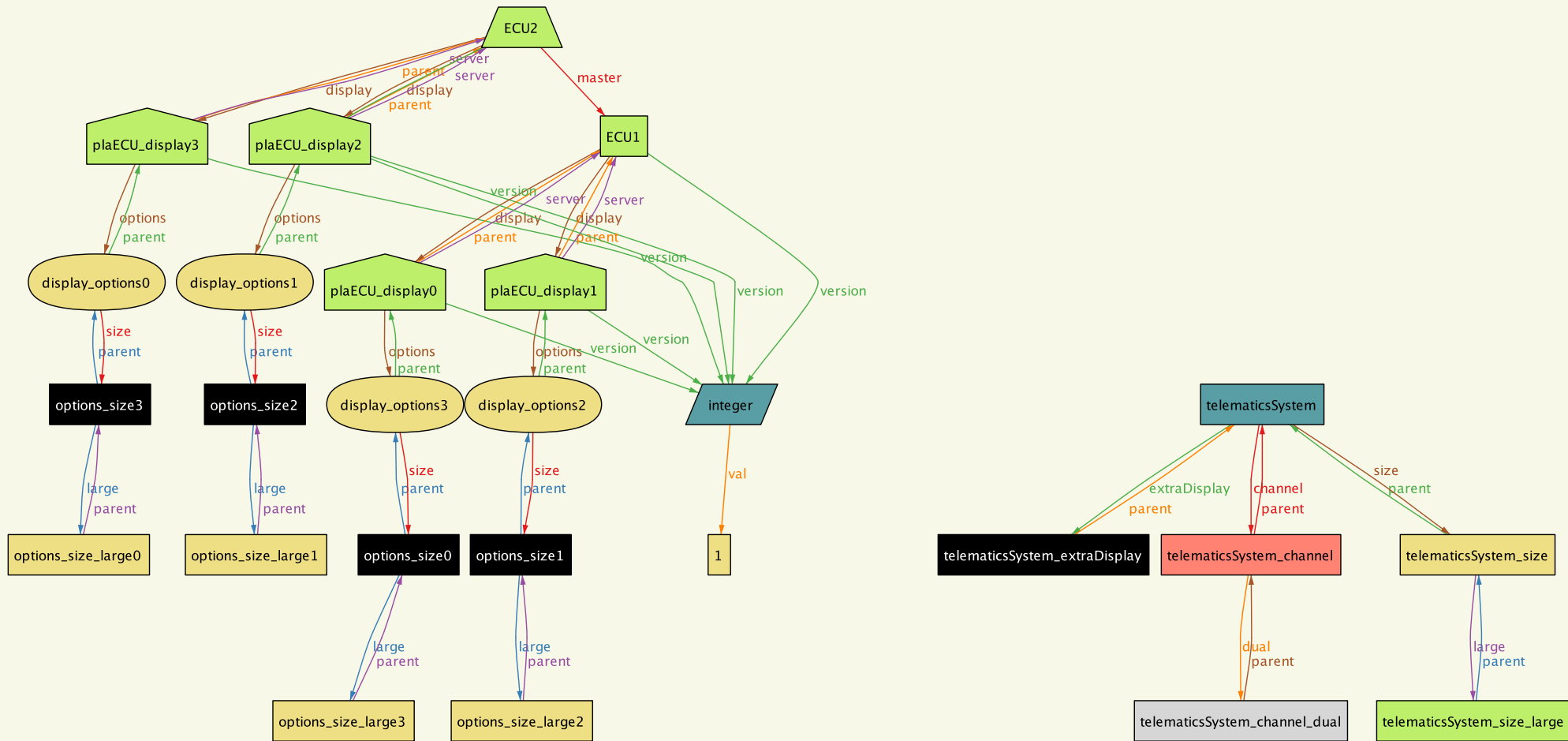
Clafer to Alloy

Structural modeling

Model checking

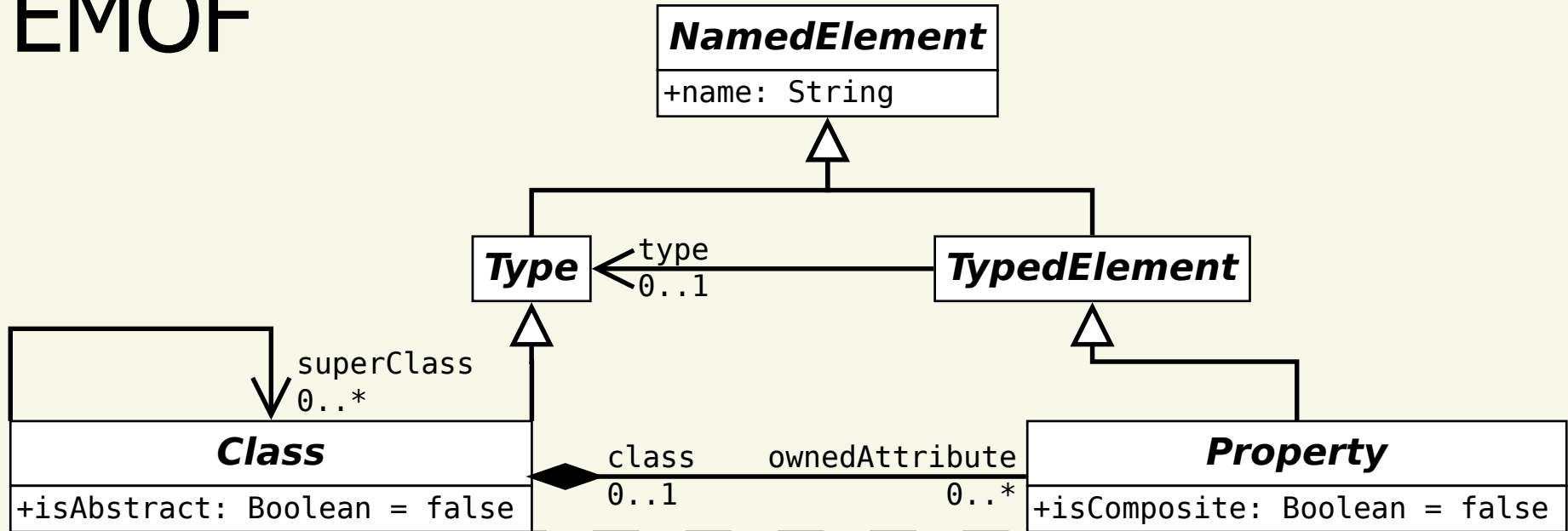
Generating instances

# Reasoning: Instance



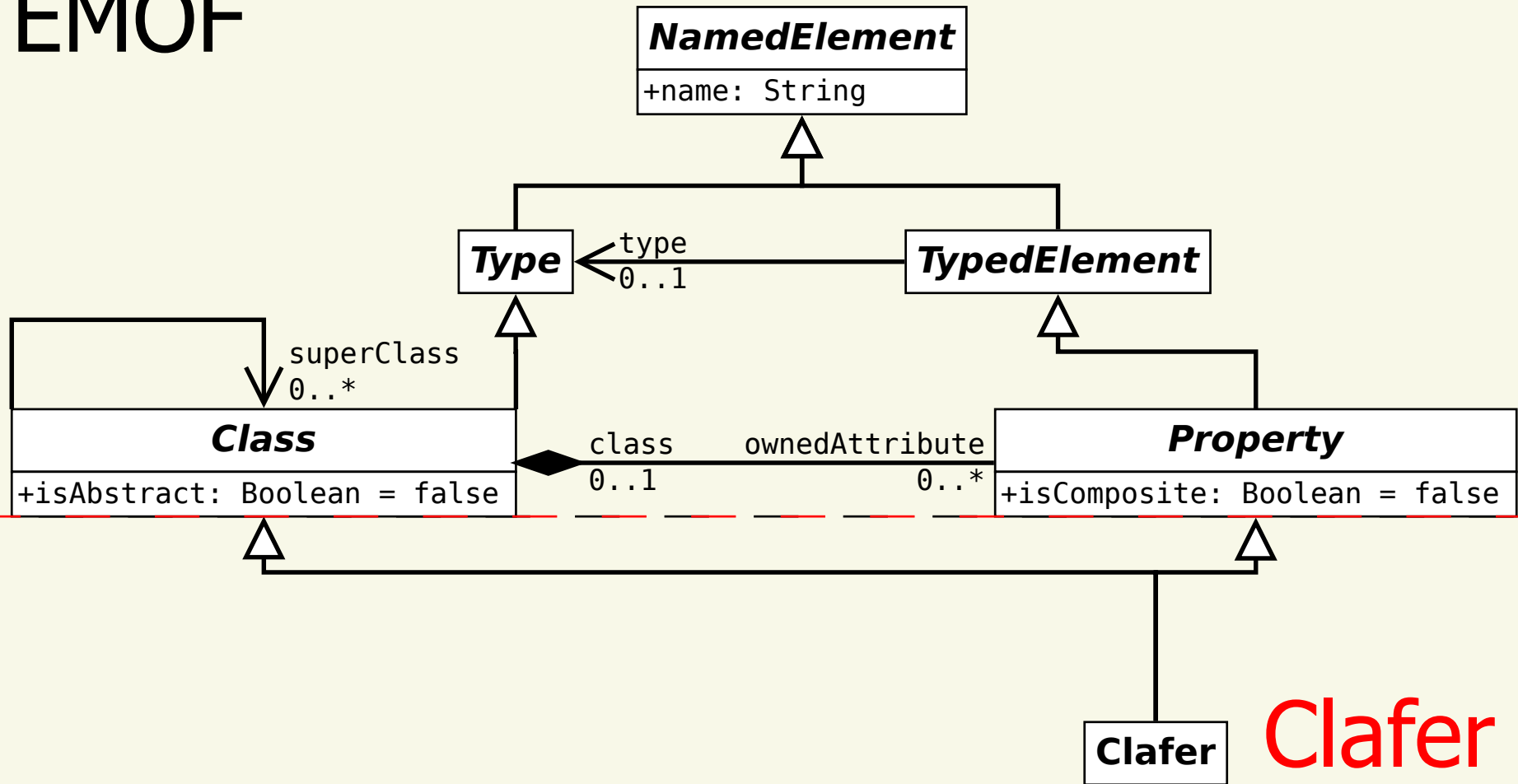
# What is Clafer?

## EMOF

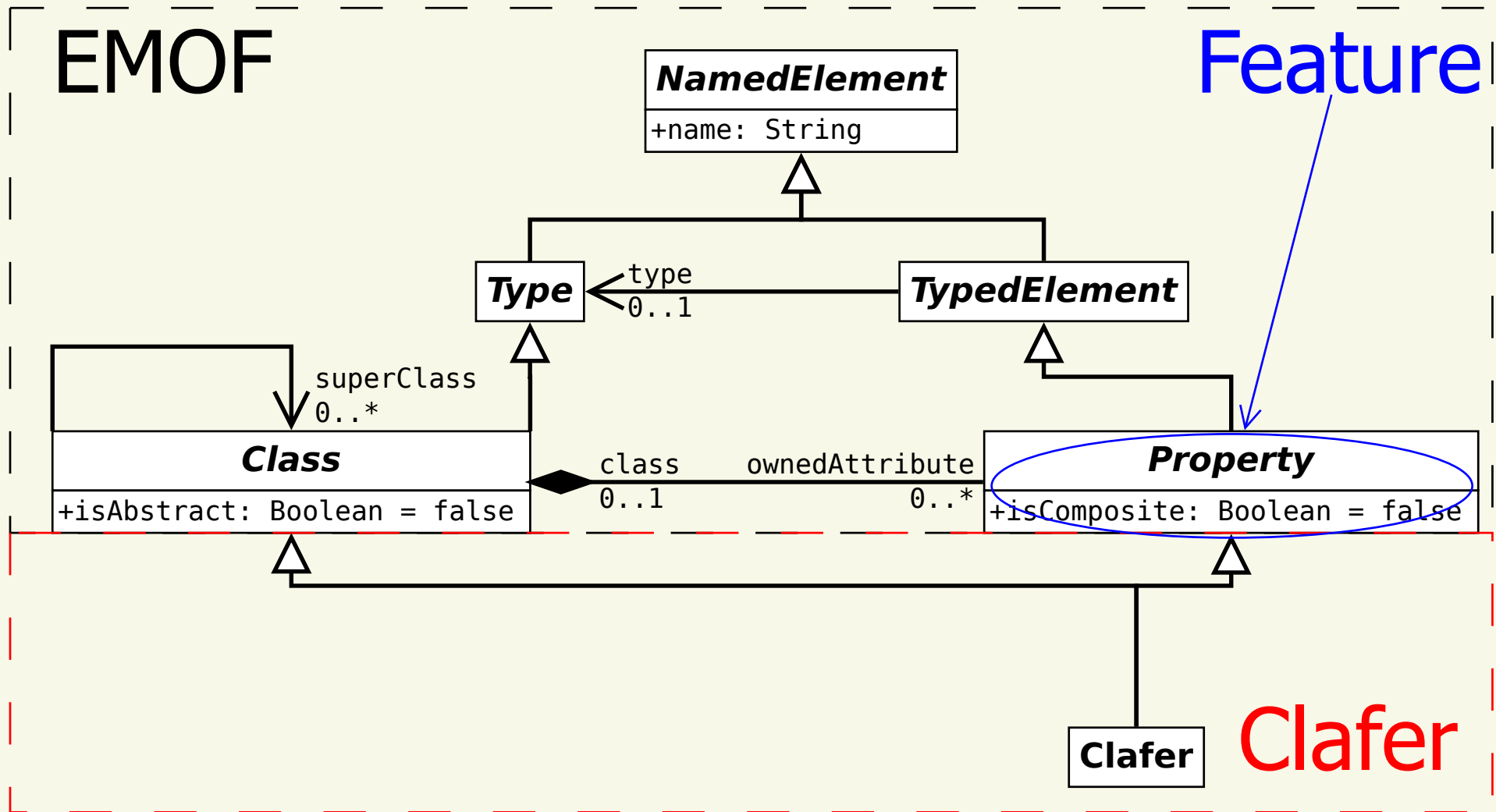


# What is Clafer?

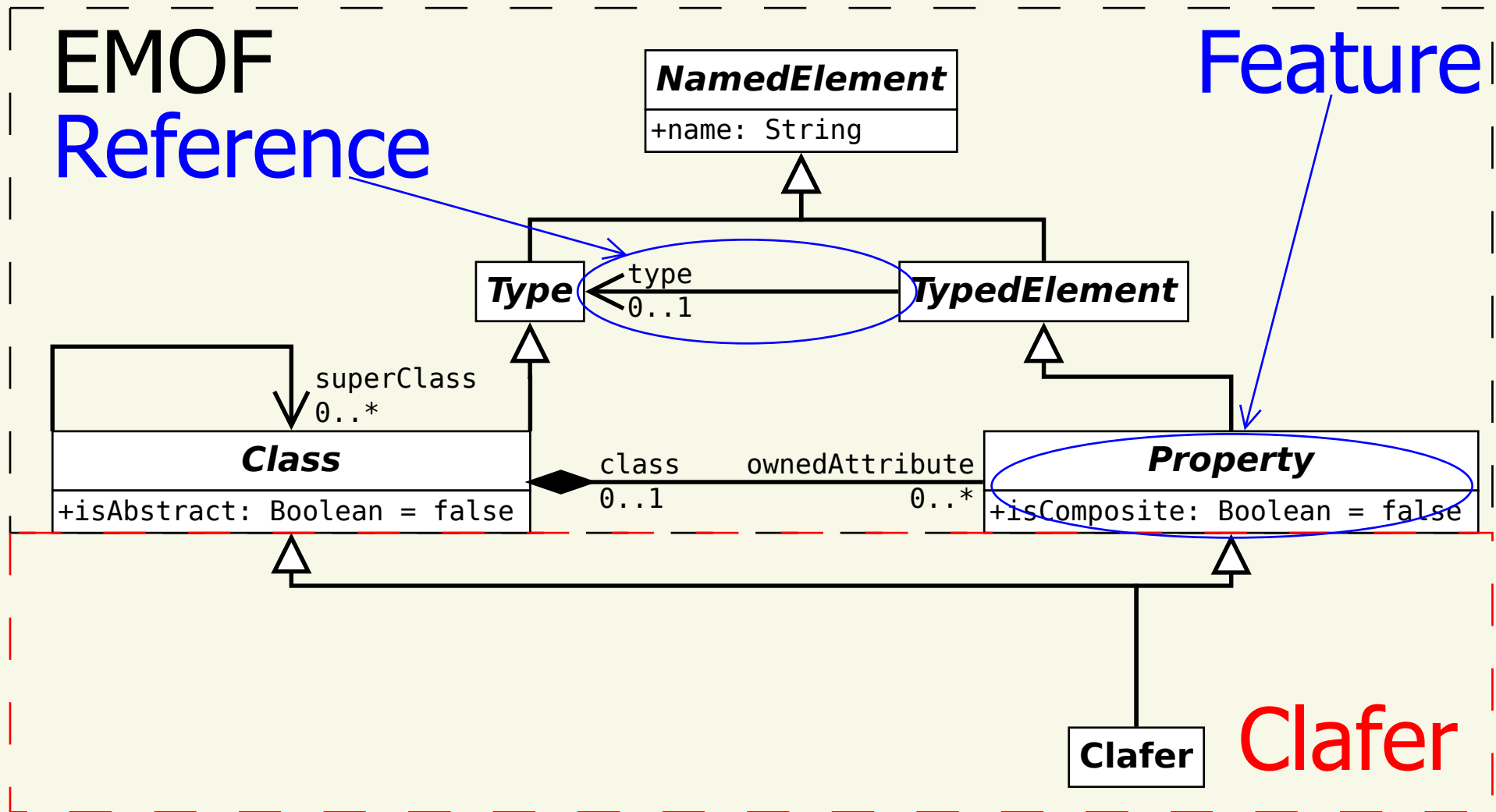
## EMOF



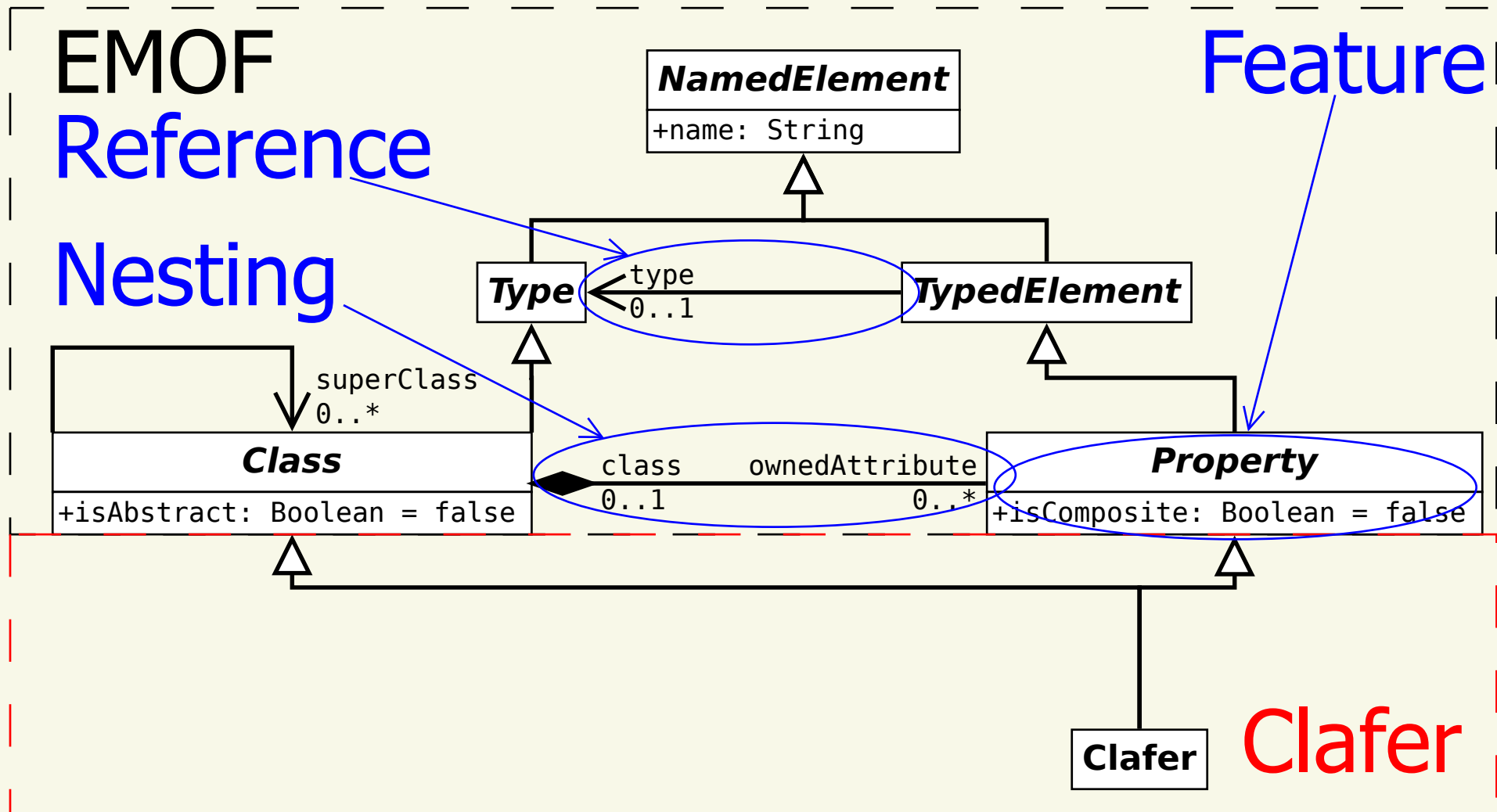
# What is Clafer?



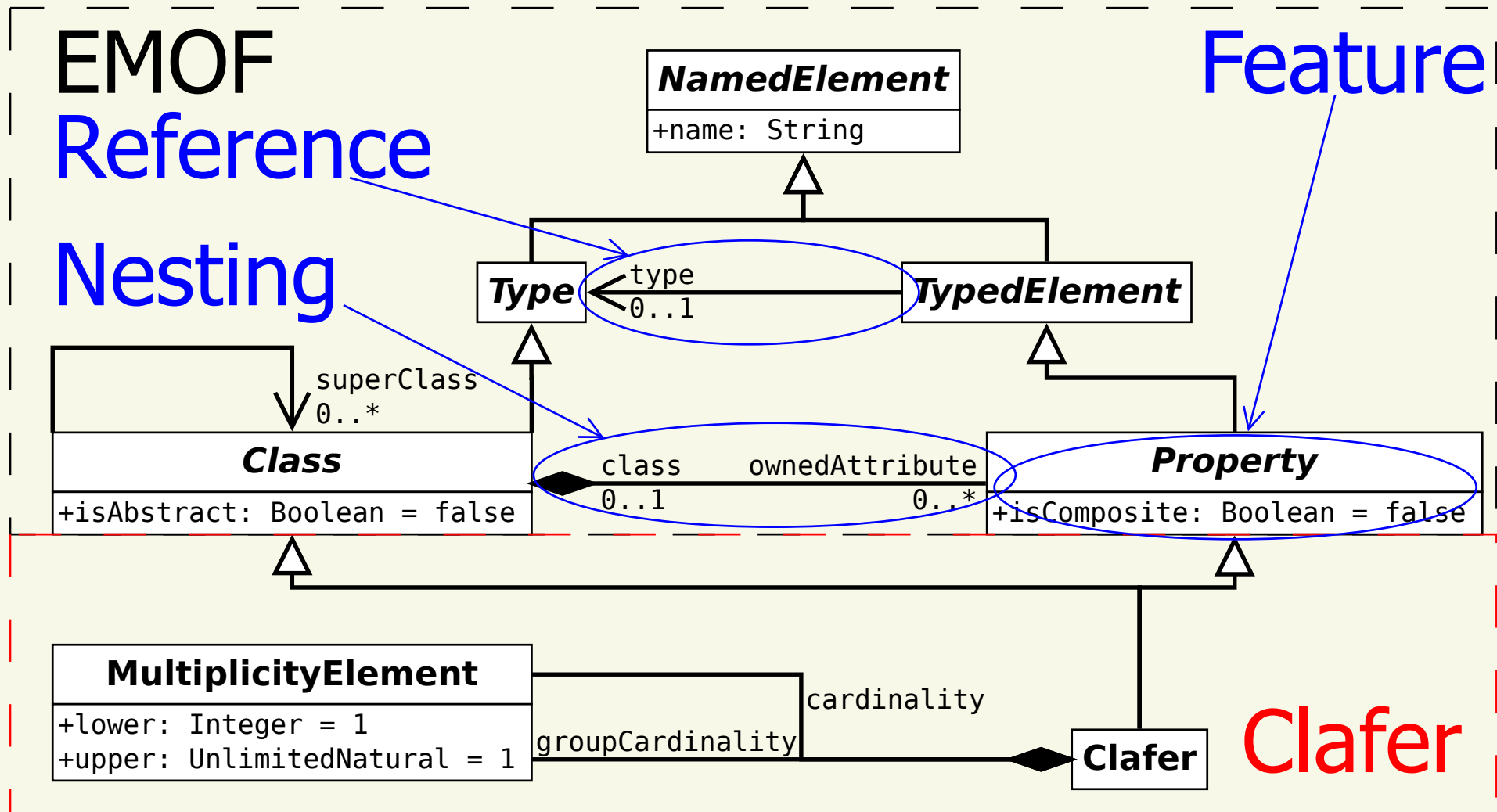
# What is Clafer?



# What is Clafer?



# What is Clafer?





# 3. Experimental Evaluation

# Analyzed Models

Feature models

Meta-models

Feature-based model templates

Instance finding

# Feature Models

Realistic and generated

Up to 5000 features, 500 constraints

Instantiation

Realistic: within hundreds of ms

Generated: within seconds

Bottleneck: Alloy's front-end

Tens of classes and constraints

UML2 the biggest

Liveness analysis of slices

Timings within hundreds of ms

# Feature-Based Model Templates

Up to 78 classes, 314 constraints

eCommerce the biggest

Liveness analysis of slices

Timings within seconds

# 4. Conclusion

Problem, solution spaces, mapping

# Conclusion

Problem, solution spaces, mapping  
Common infrastructure



# Conclusion

Problem, solution spaces, mapping  
Common infrastructure  
clafer2alloy translator

# Conclusion

Problem, solution spaces, mapping

Common infrastructure

clafer2alloy translator

Feature – class modeling relationship

# Conclusion

Problem, solution spaces, mapping

Common infrastructure

clafer2alloy translator

Feature – class modeling relationship

Promising experimental results

Thanks for listening!

# Questions?

[gsd.uwaterloo.ca/clafer](https://gsd.uwaterloo.ca/clafer)

# Meta-Model: EMF Text

```
abstract class display extends comp {  
    reference ECU server (1..1);  
    containment reference Options options (1..1);  
}
```

```
abstract class plaECU extends ECU {  
    containment reference Display display (1..2);  
}
```

```
class ECU1 extends plaECU {}
```

```
class ECU2 extends plaECU {  
    reference ECU1 master (1..1);  
}
```

```
class Root {  
    containment reference ECU1 ECU1 (1..1);  
    containment reference ECU2 ECU2 (0..1);  
}
```

# Meta-Model: Alloy

```
sig display {  
    parent : one plaECU  
}  
parent = plaECU.this  
this.@server = parent  
not (some this.@options.@cache)  
}
```

```
abstract sig plaECU extends ECU {  
    display : set display  
}  
{ 1 ≤ #display and #display ≤ 2 }
```

```
one sig ECU1 extends plaECU {}
```

```
lone sig ECU2 extends plaECU {  
    master : one ECU1  
}
```