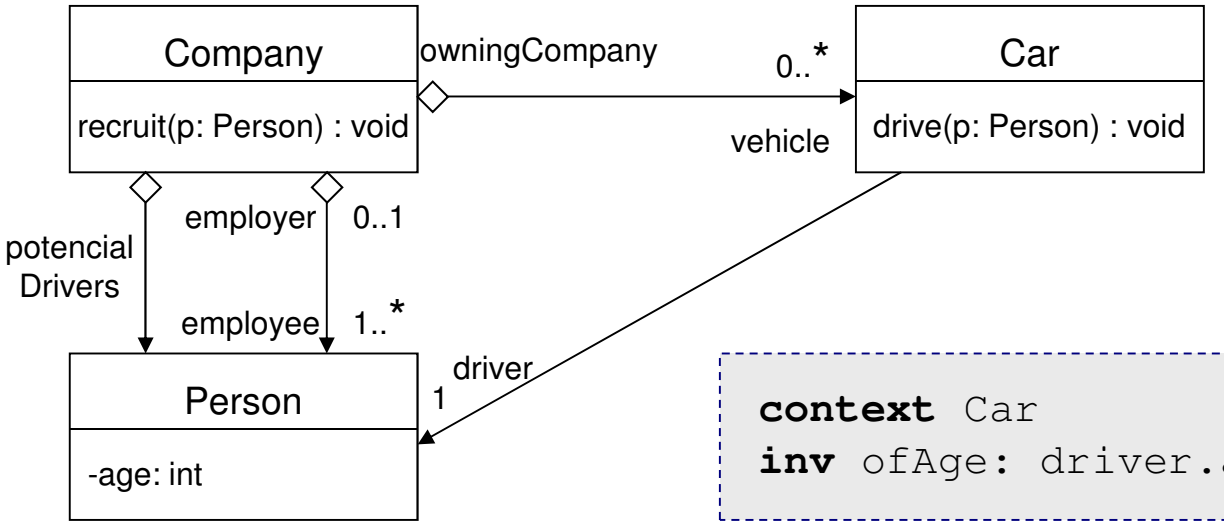
Metamodellek a szoftverfejlesztésben

Metamodellezés és architektúra

© BME IIT, László Zoltán

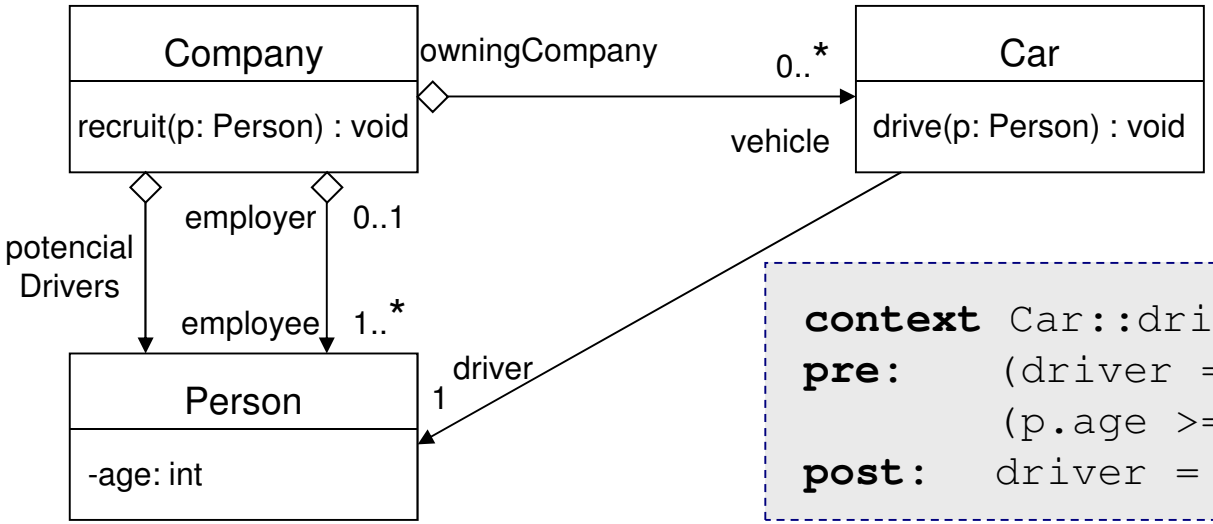
Metamodellezés és OCL




```
context Car
inv ofAge: driver.age >= 18
```

```
context Company
def potentialDrivers = employee->select(age >= 18)
```

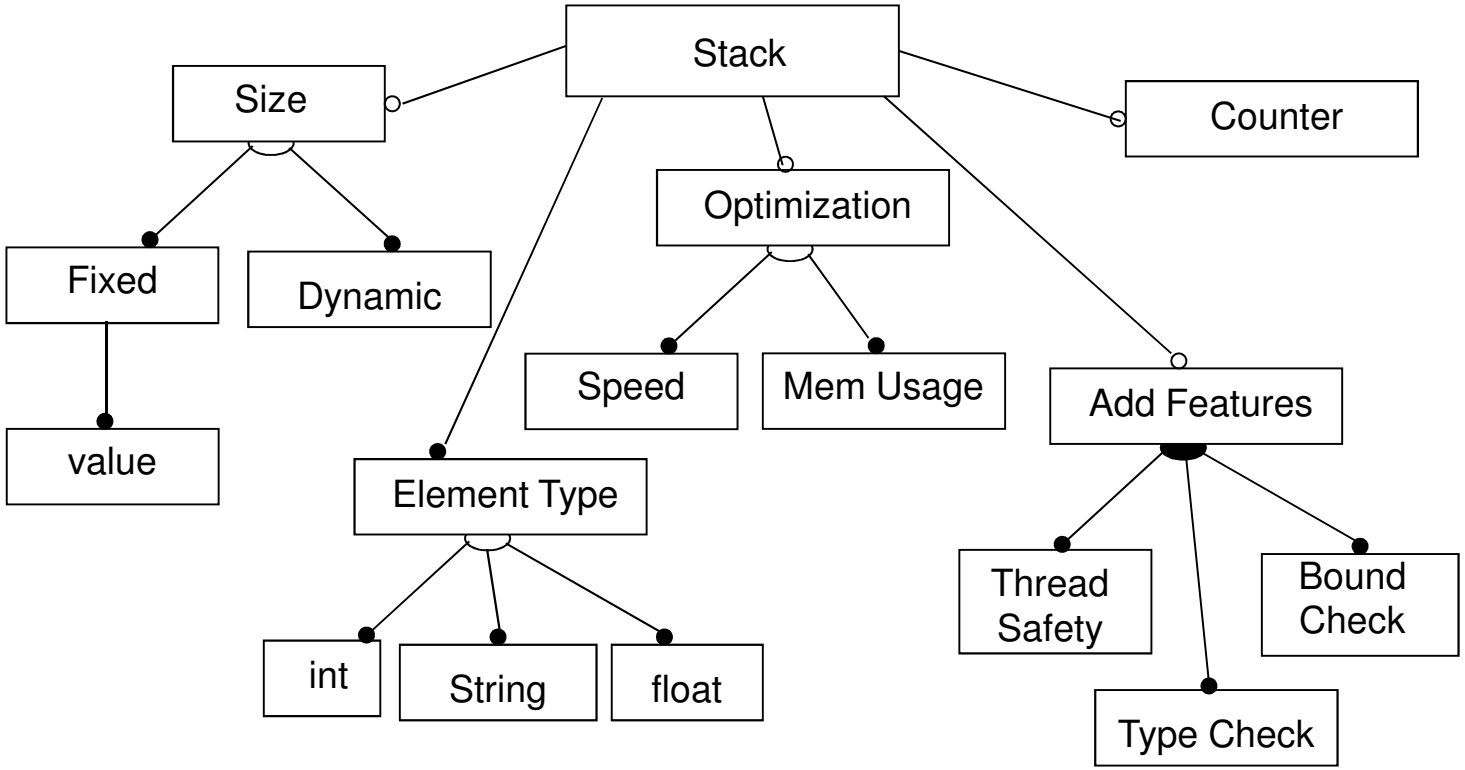
Metamodellezés és OCL



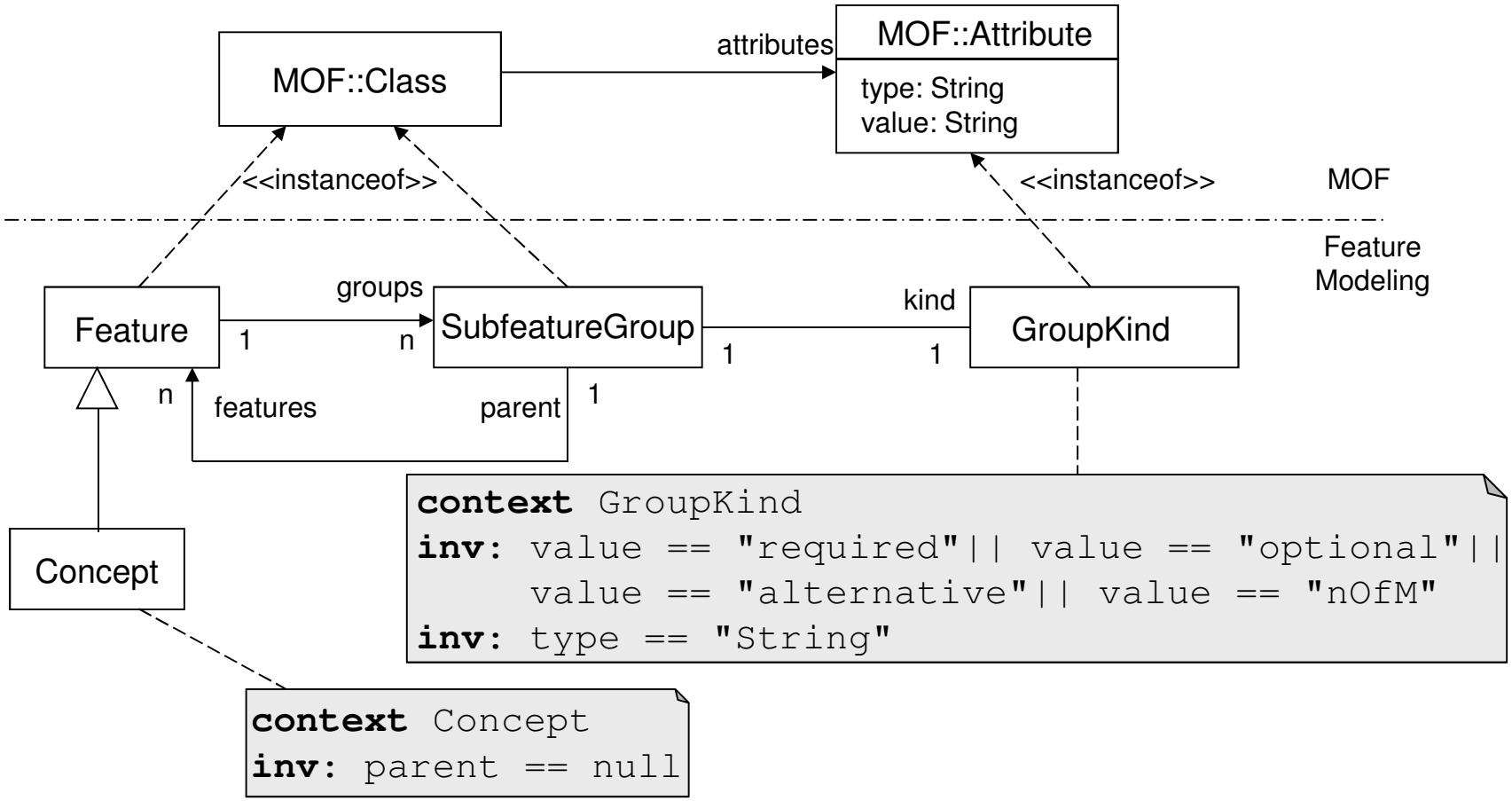

```
context Car::drive(p:Person)
pre:    (driver == null) &&
        (p.age >= 18)
post:   driver = p
```

```
context Company::recruit(p:Person)
pre:    -- none
post:   (employee.size = employee@pre.size + 1) &&
        (employee.includes(p))
```

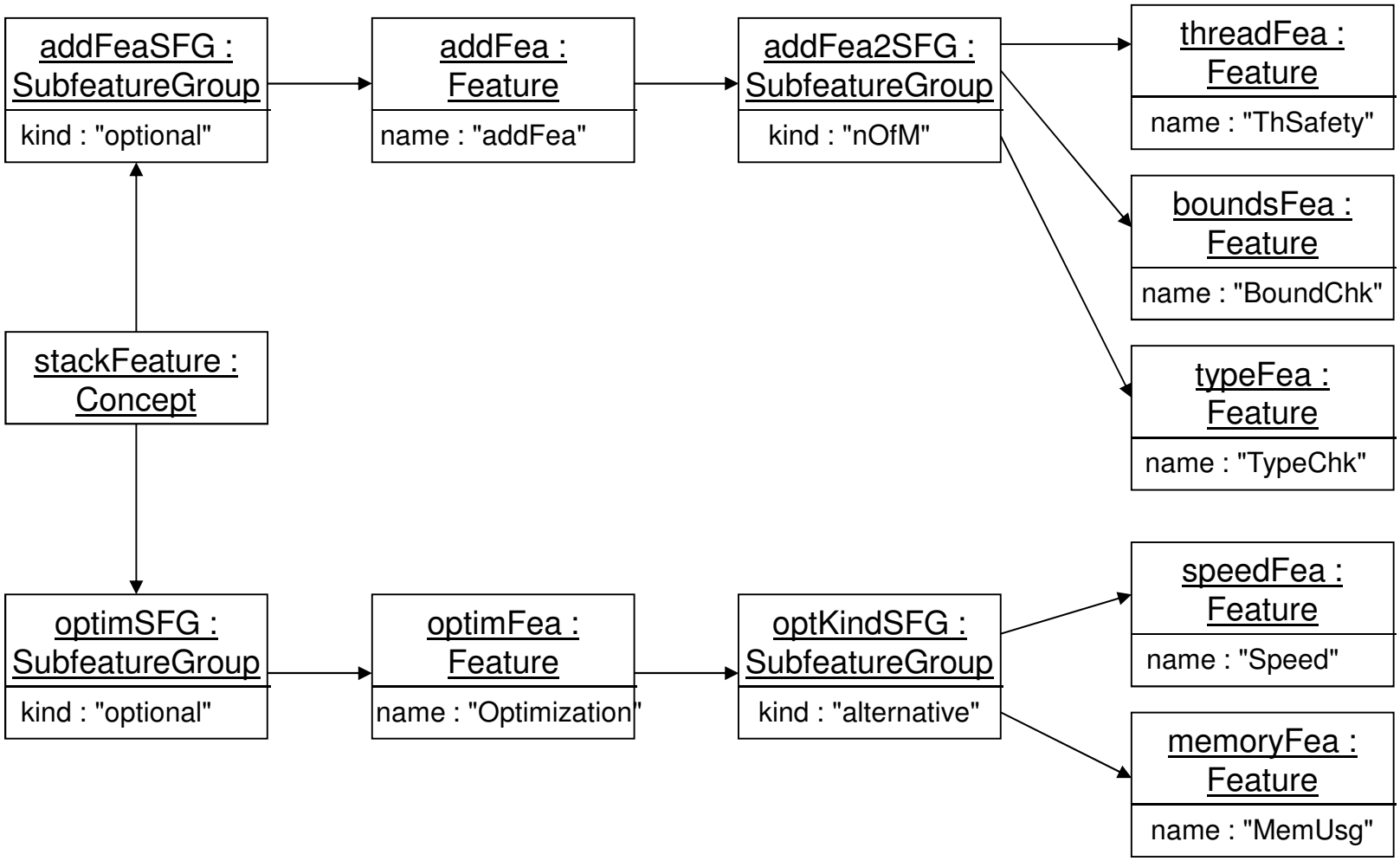
Metamodellezés példa 1



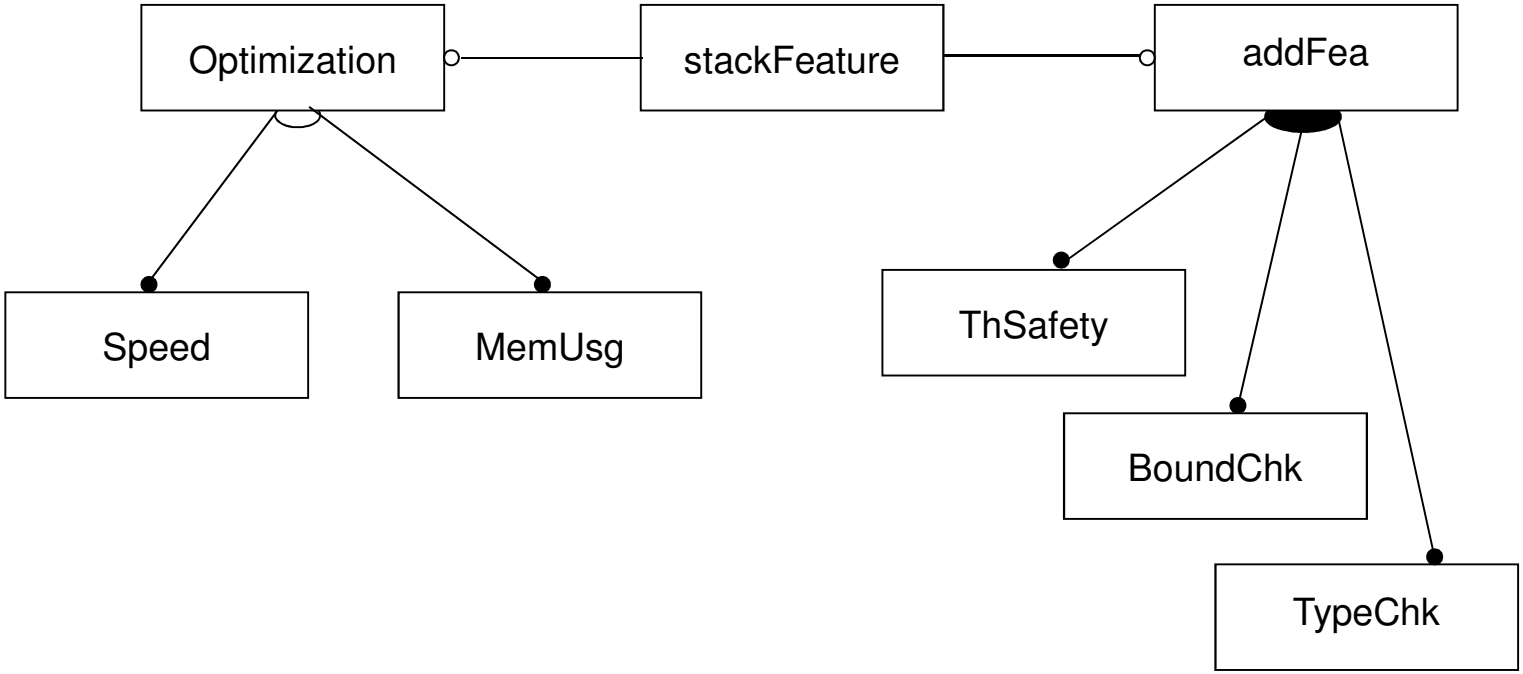
Metamodellezés példa 1



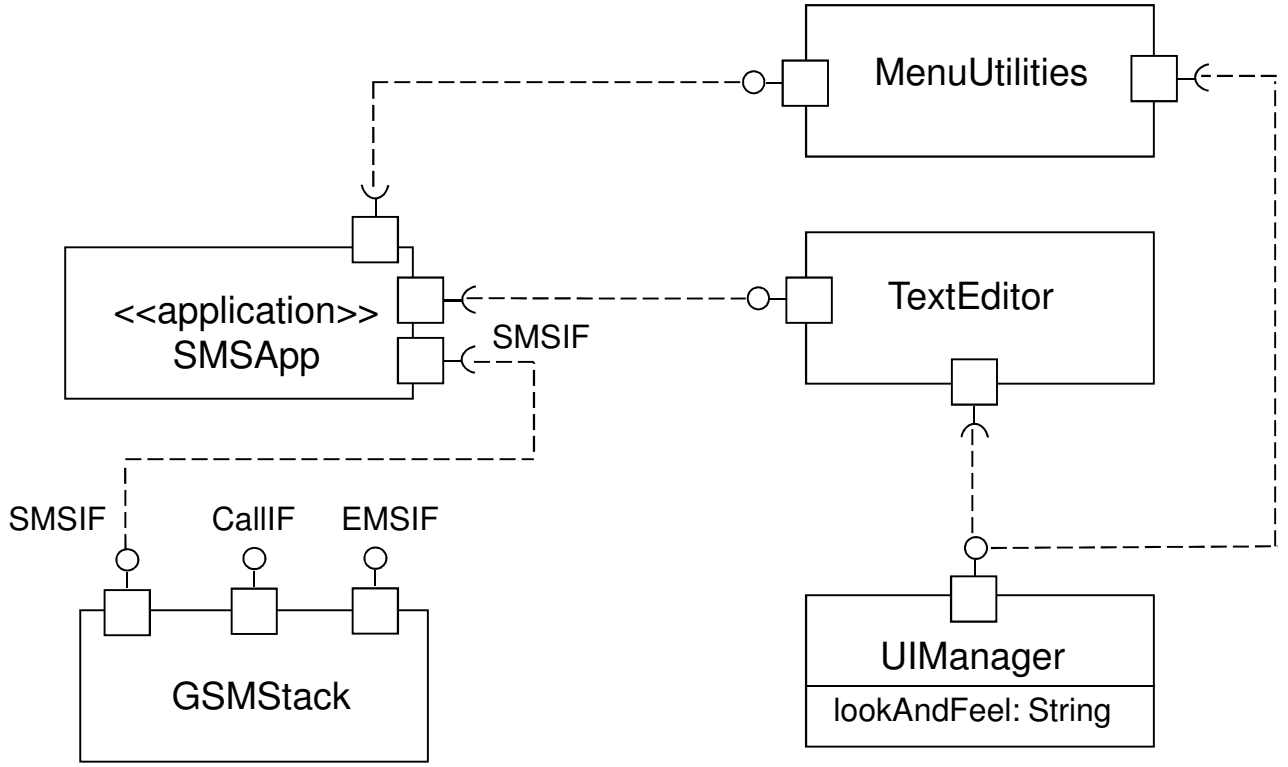
Metamodellezés példa 1



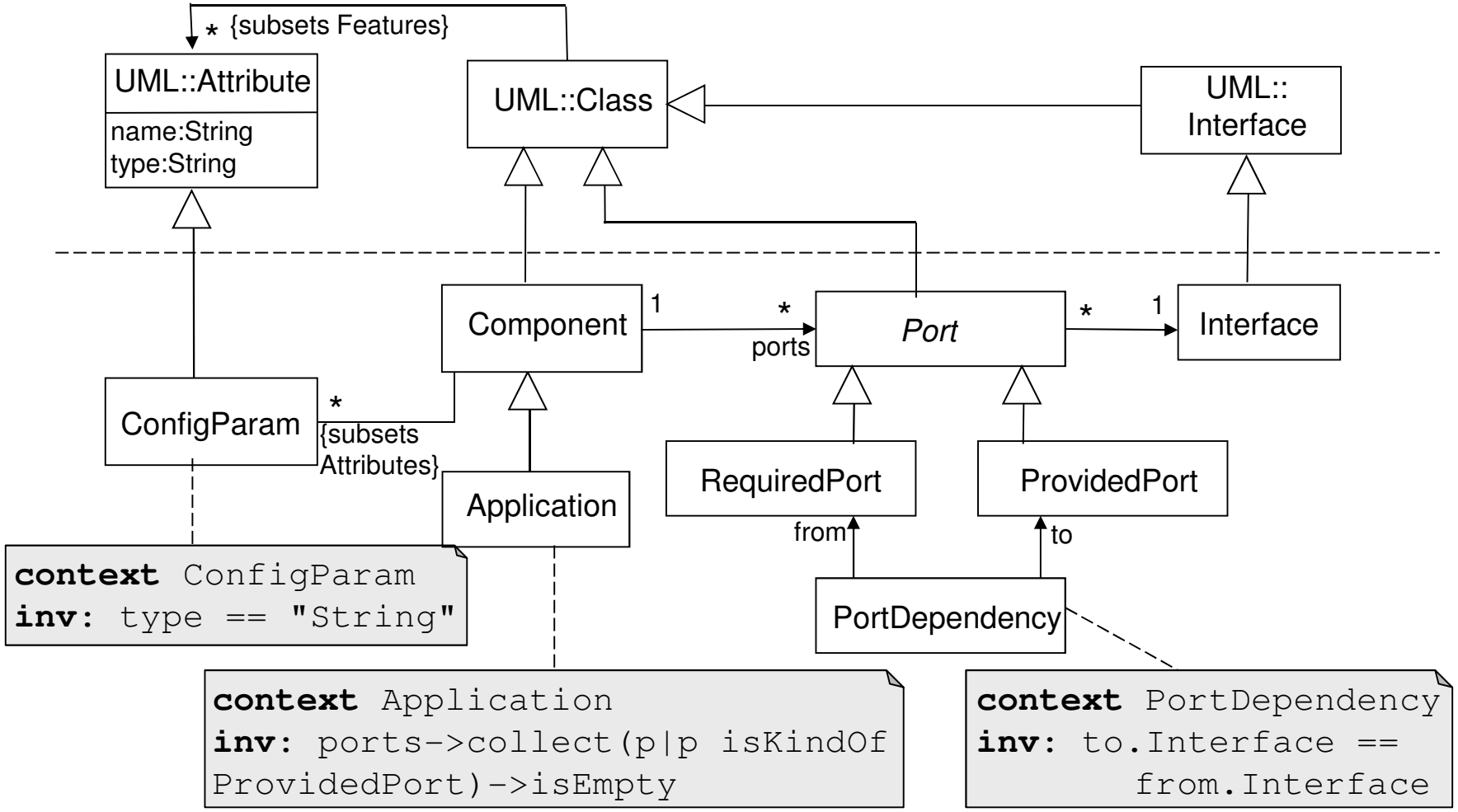
Metamodellezés példa 1



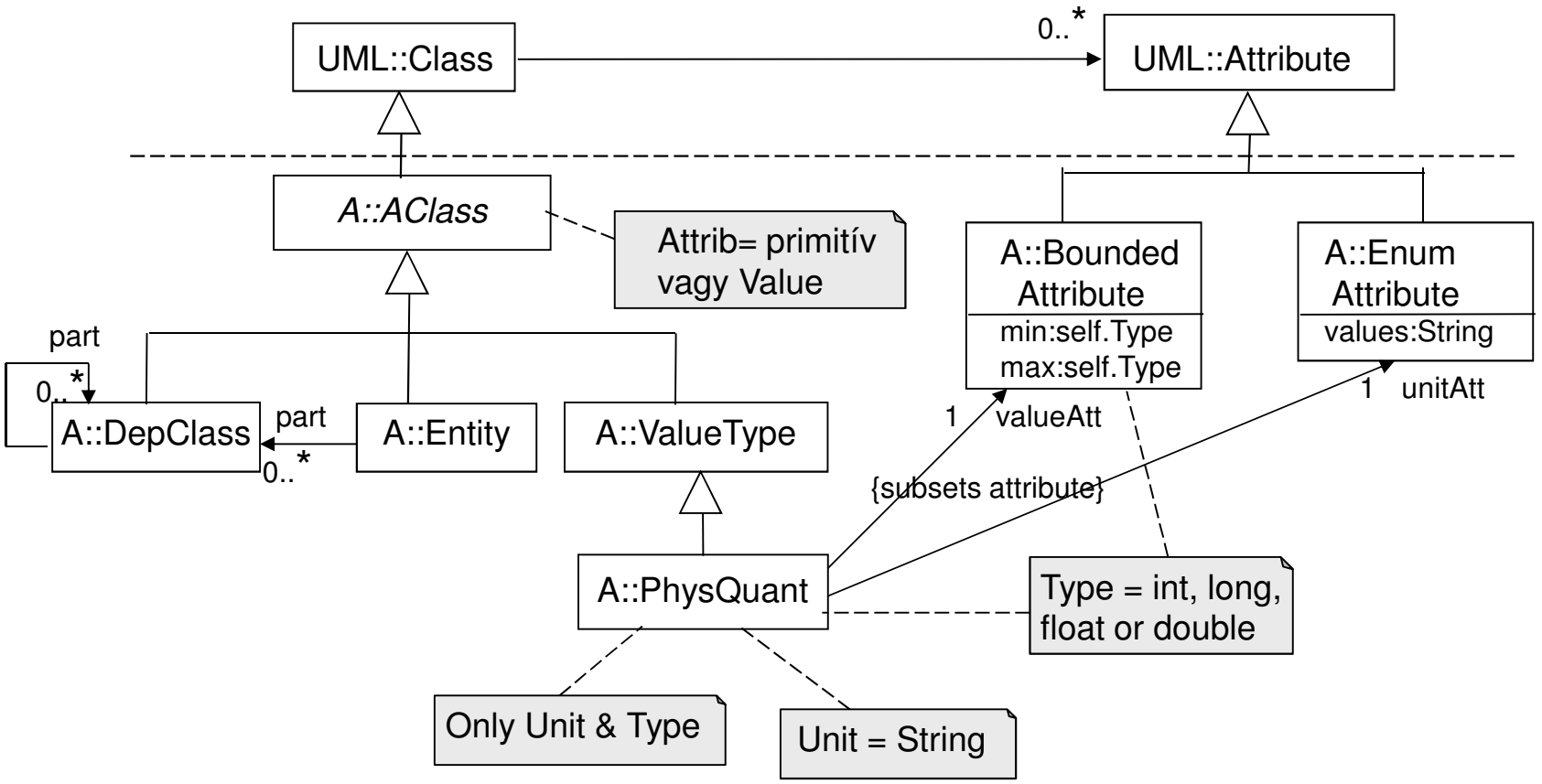
Metamodellezés példa 2



Metamodellezés példa 2

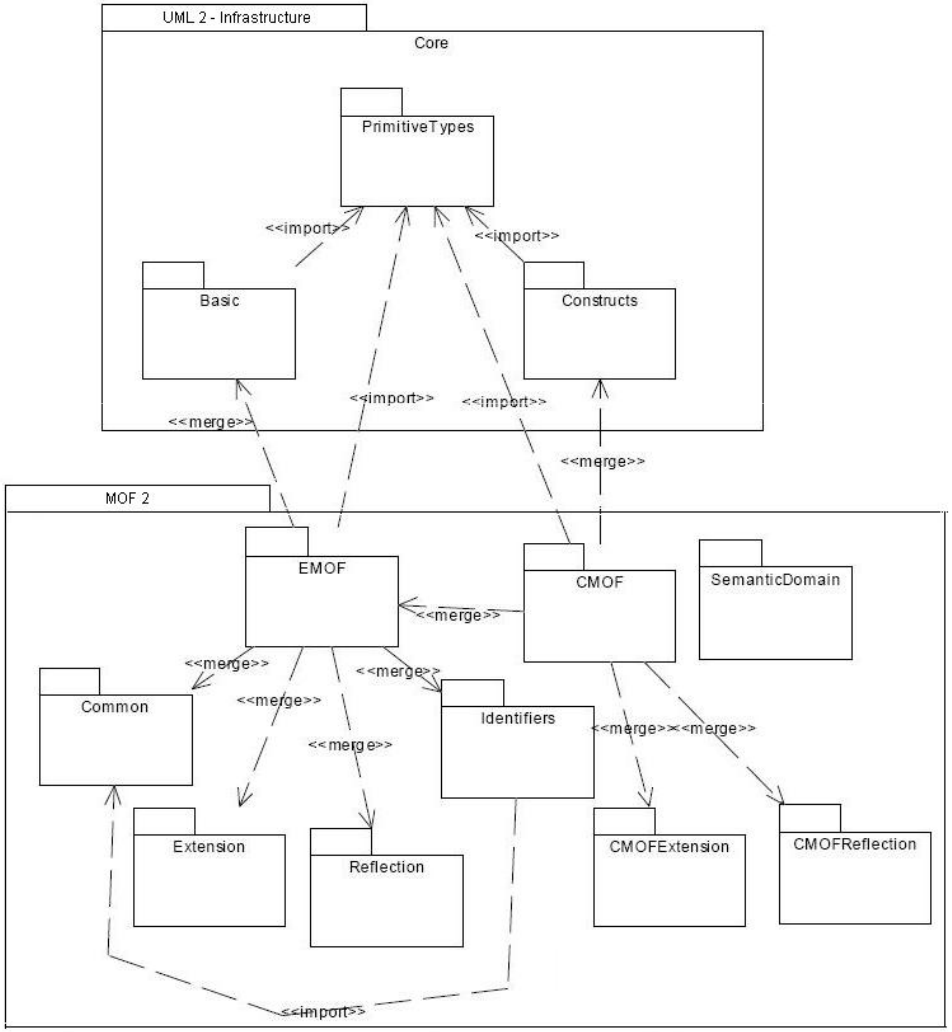


Metamodellezés példa 3



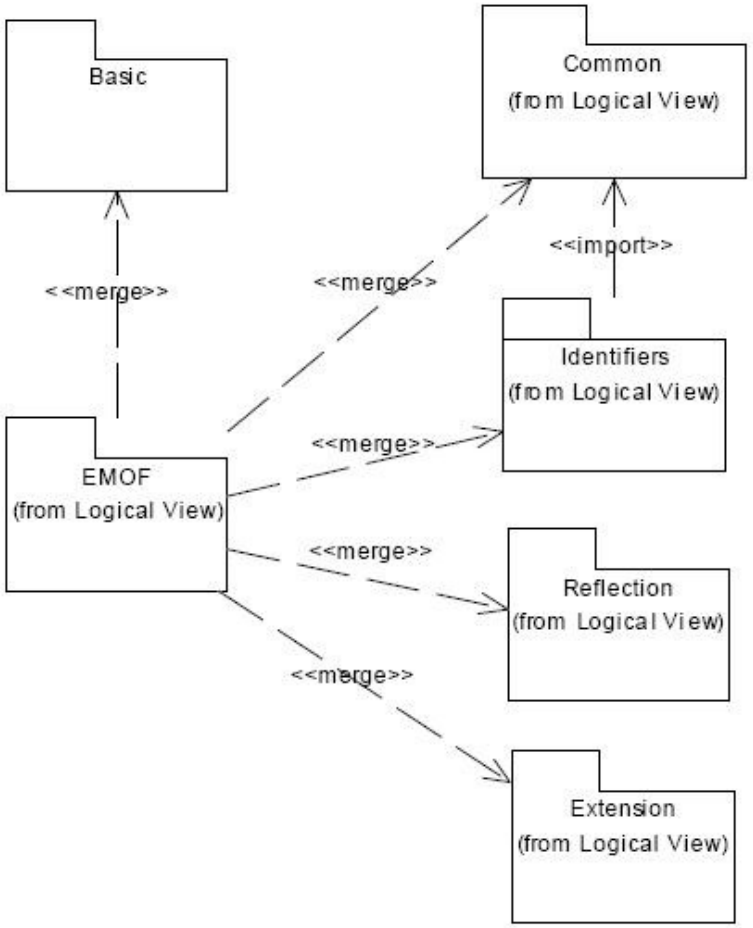
MOF részletesebben

- UML2-n alapul



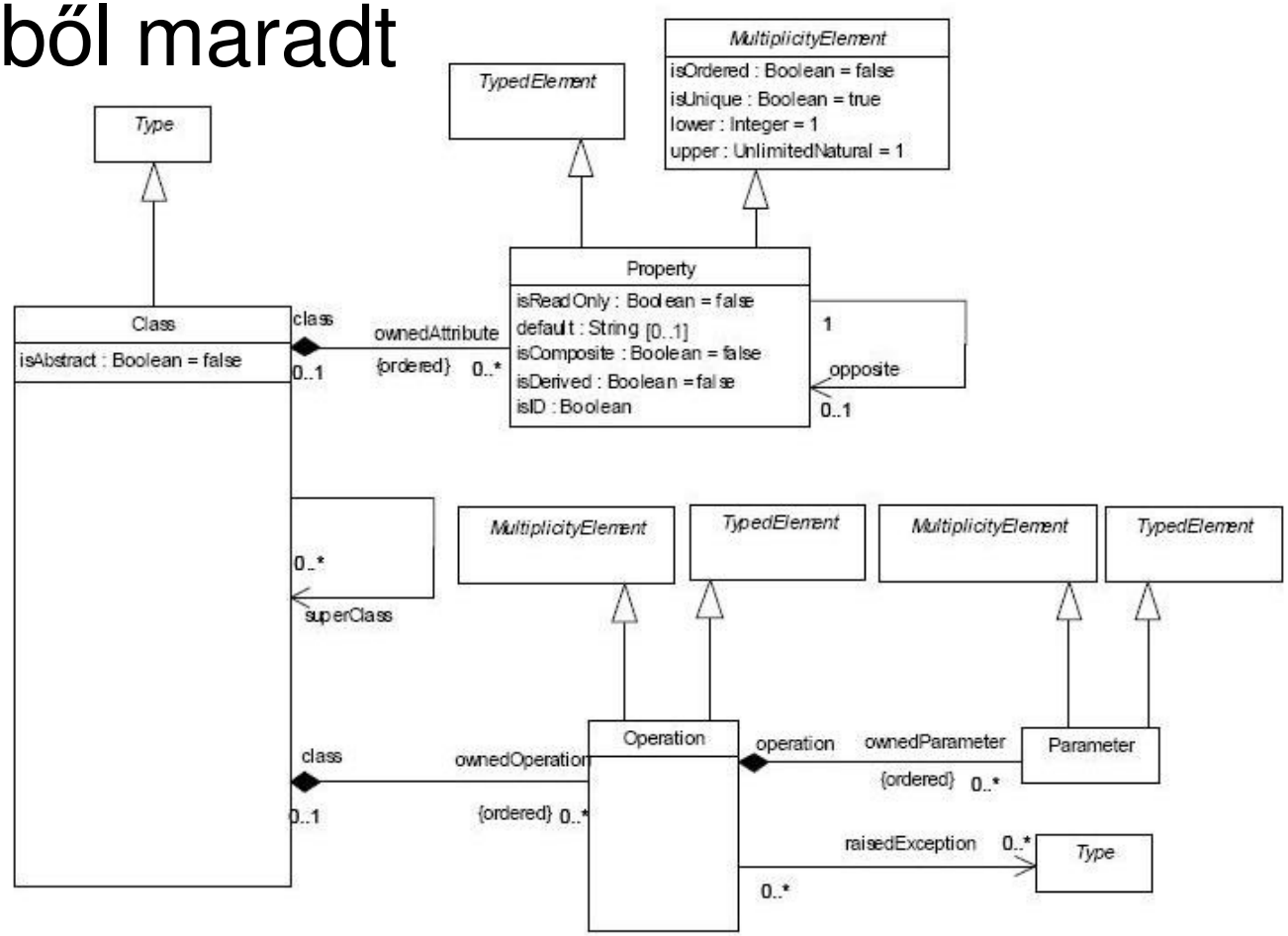
MOF részletesebben

- EMOF - essential
  - minimális
  - assoc. helyett ref.
  - CMOF konvert.



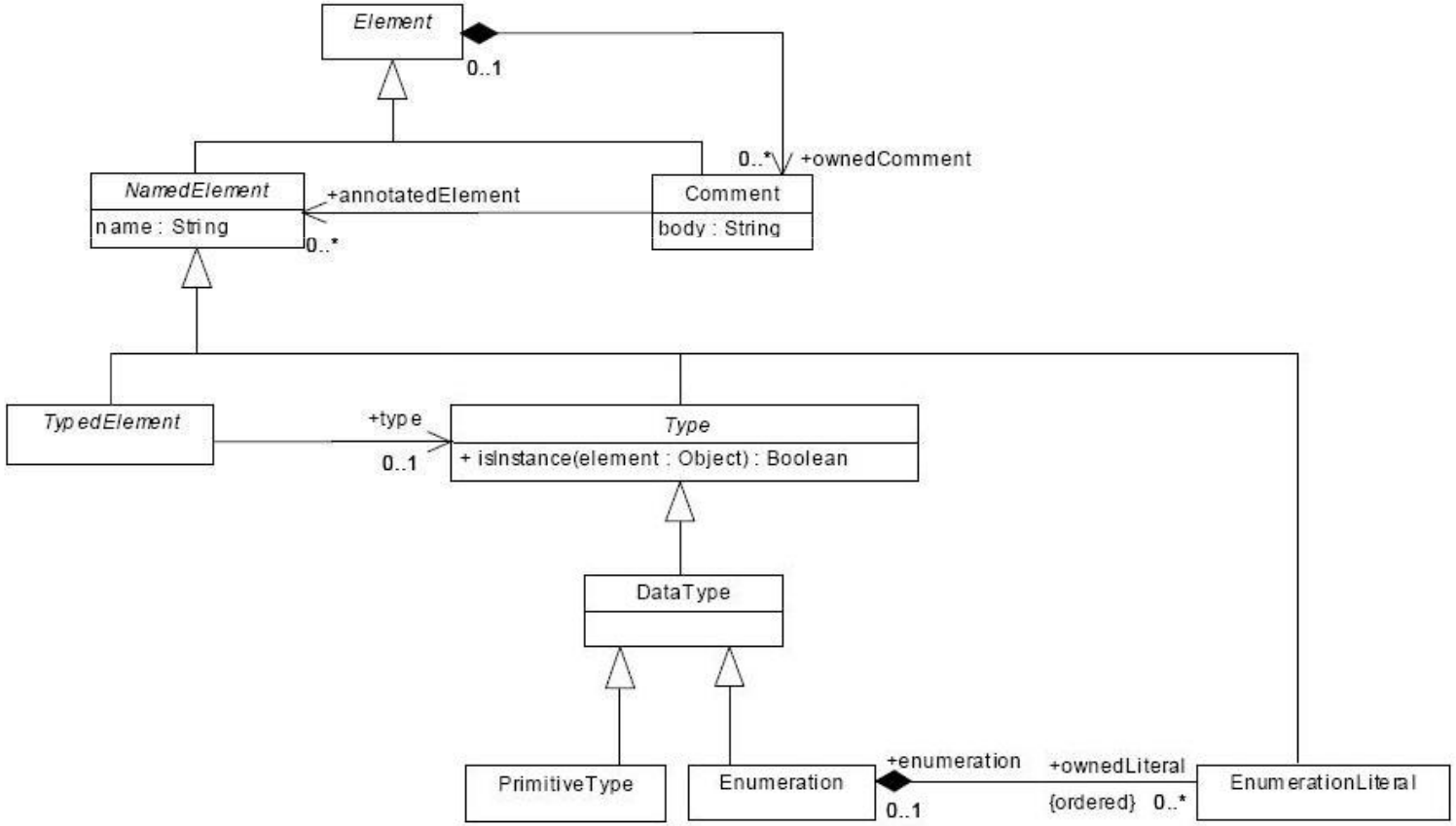
MOF részletesebben

- UML2-ből maradt



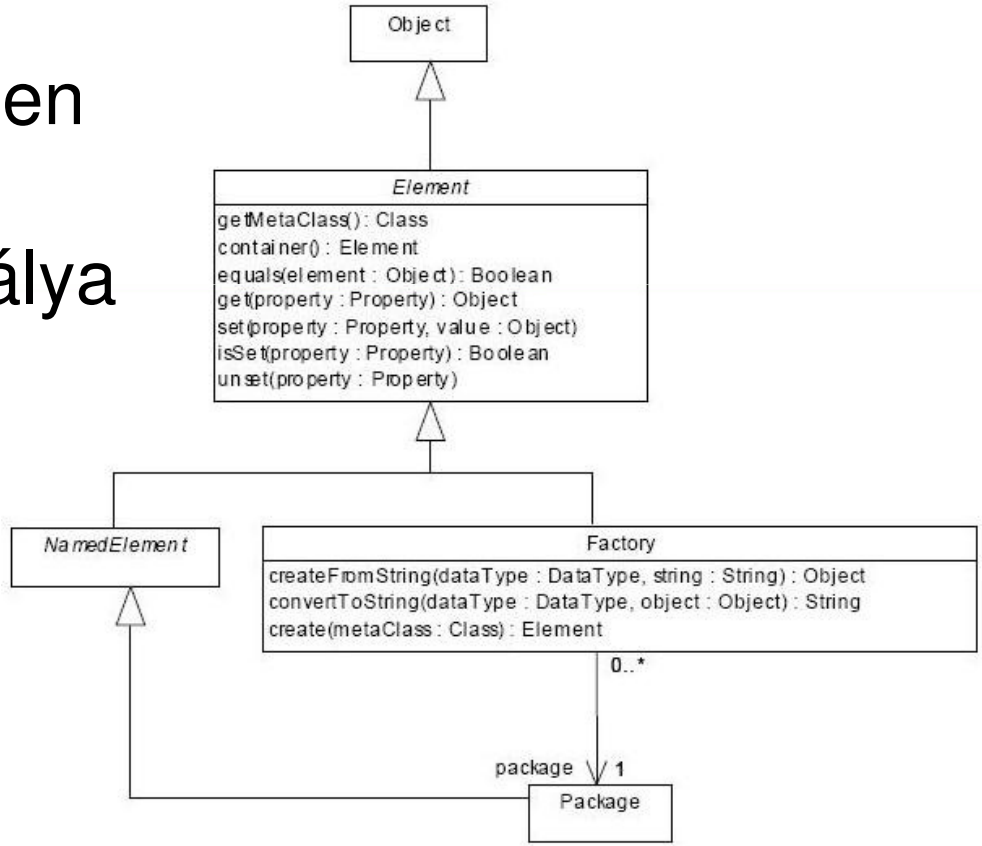
MOF részletesebben

- UML2-ből maradt



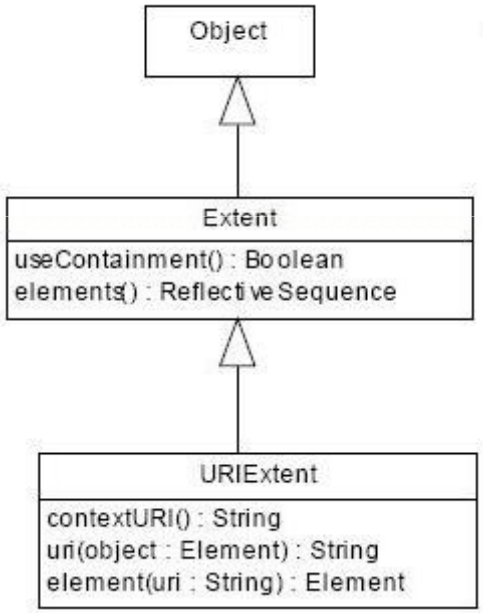
MOF részletesebben

- Reflection
  - Element = minden modell elem és példány ősosztálya
  - API
  - MDA-hoz



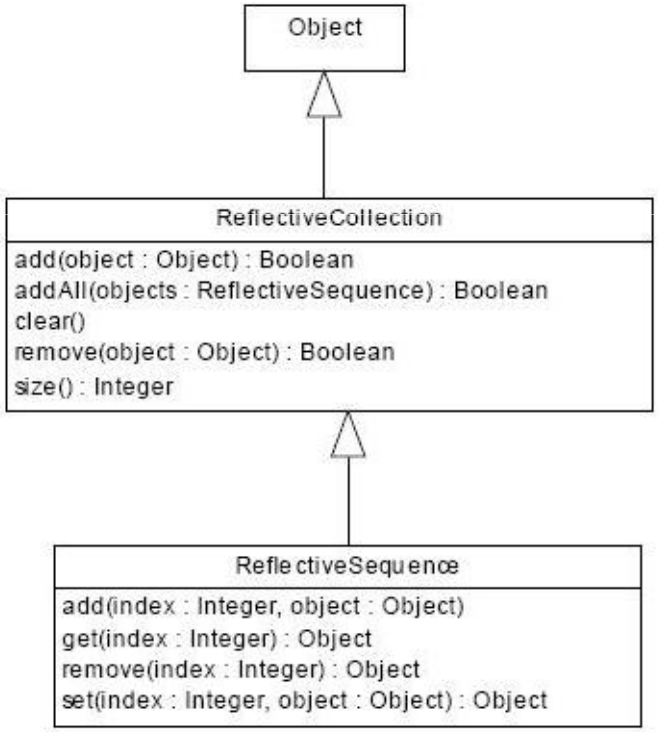
MOF részletesebben

- Identifiers
  - OID: property *isID* true
  - Extent: OID készlet
  - URIExtent: OID = URI

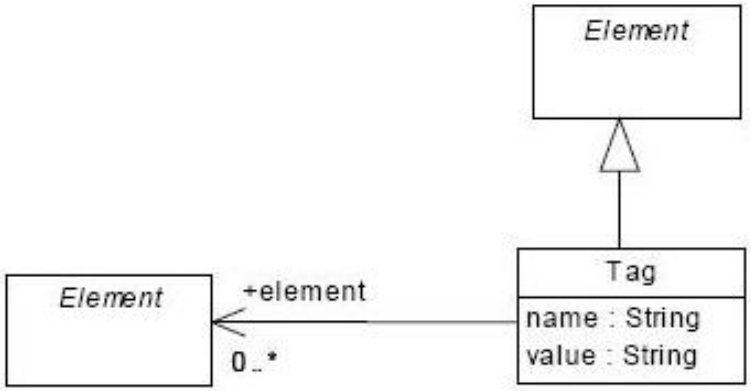


MOF részletesebben

- Common

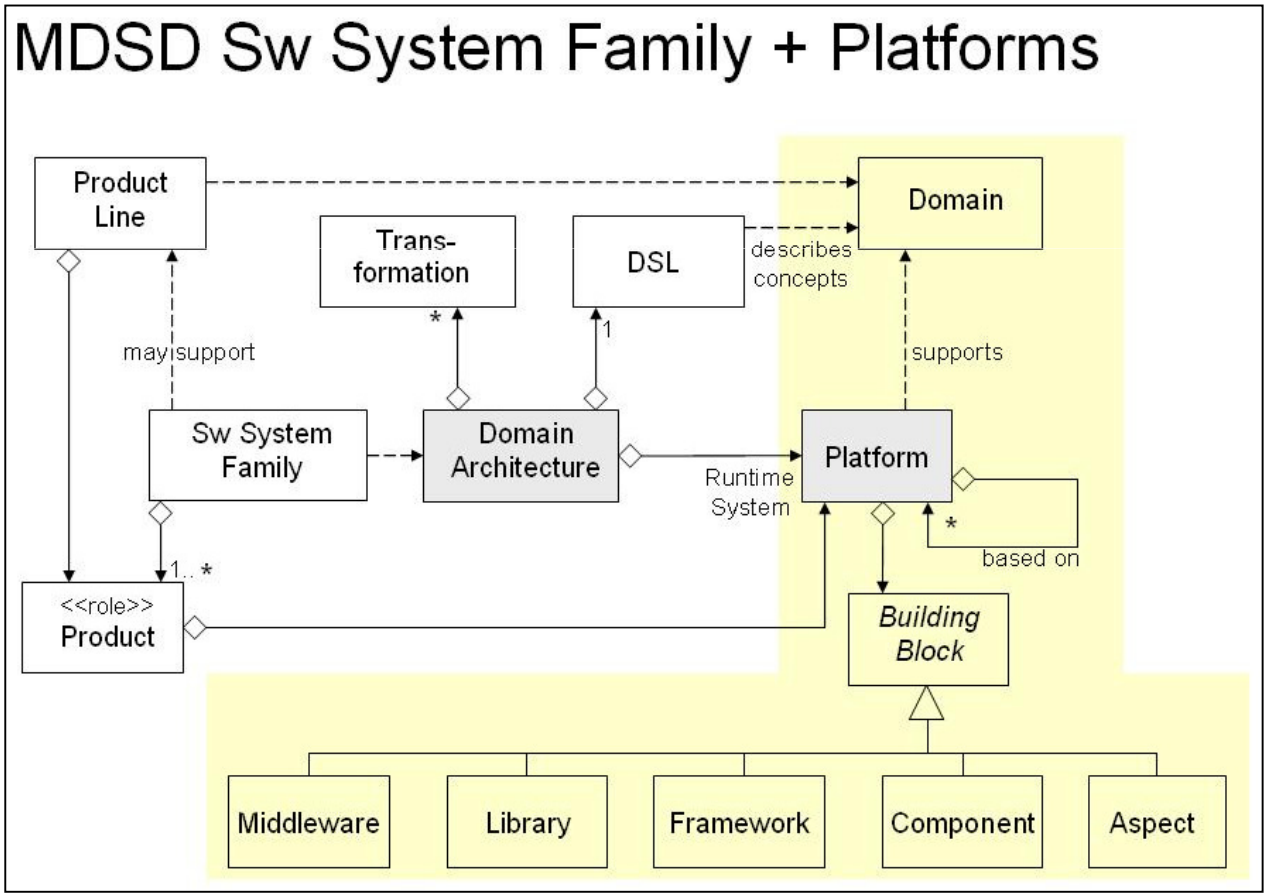


- Extension



MDSD Target Architectures

- Mi a "Target" architektúra ?

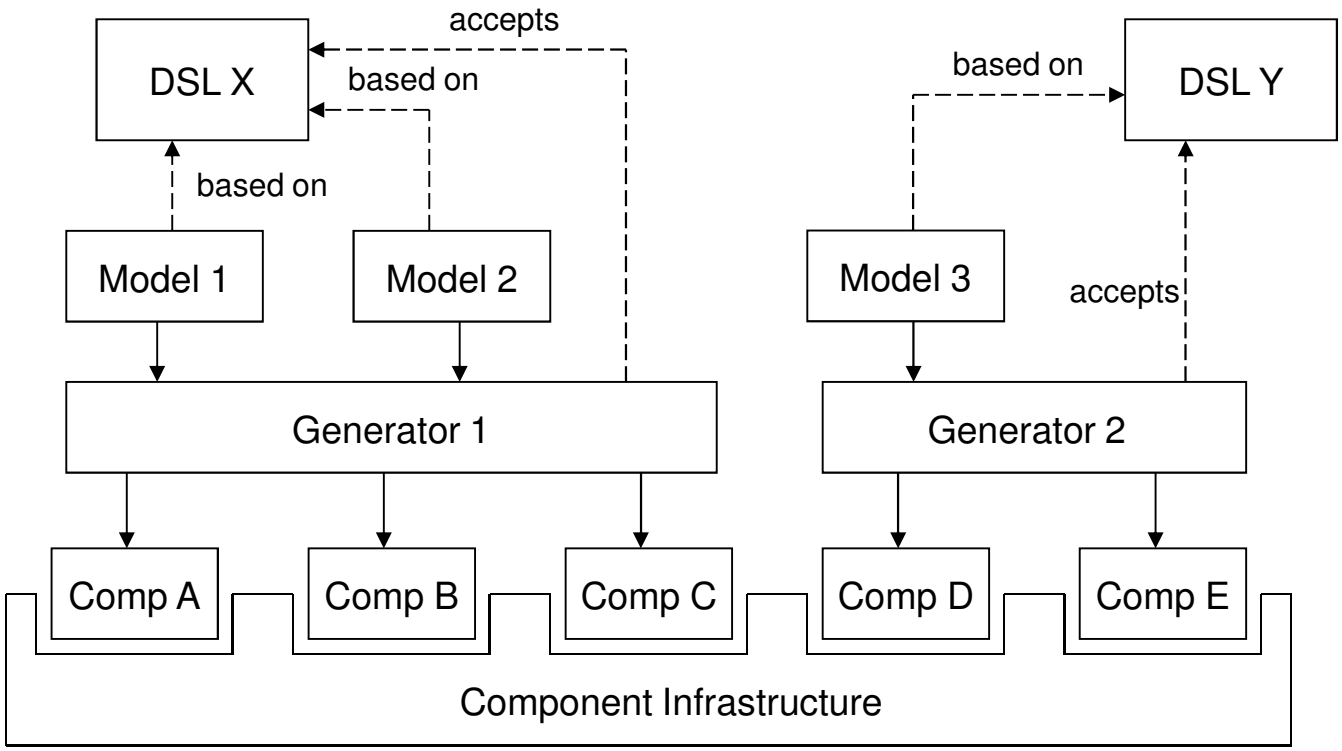


MDSD Target Architectures

- Elemei:
  - Domain architecture – transzformáció, eszköz
  - Platform
    - framework — konfig + kiterjesztés
    - middleware — speciális keretrendszer
    - komponens — önhordó – interfész – körny. függ

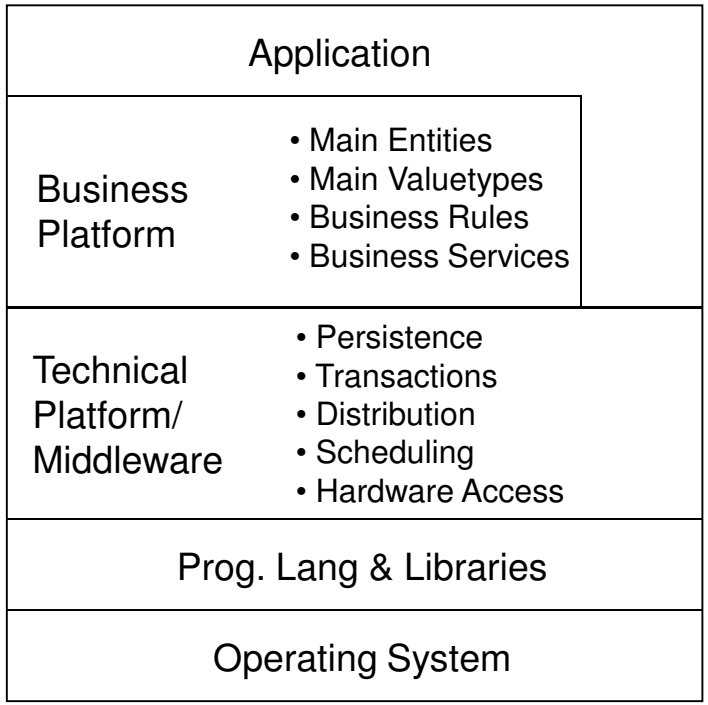
MDSD Target Architectures

- Komponens rendszer



MDSD Target Architectures

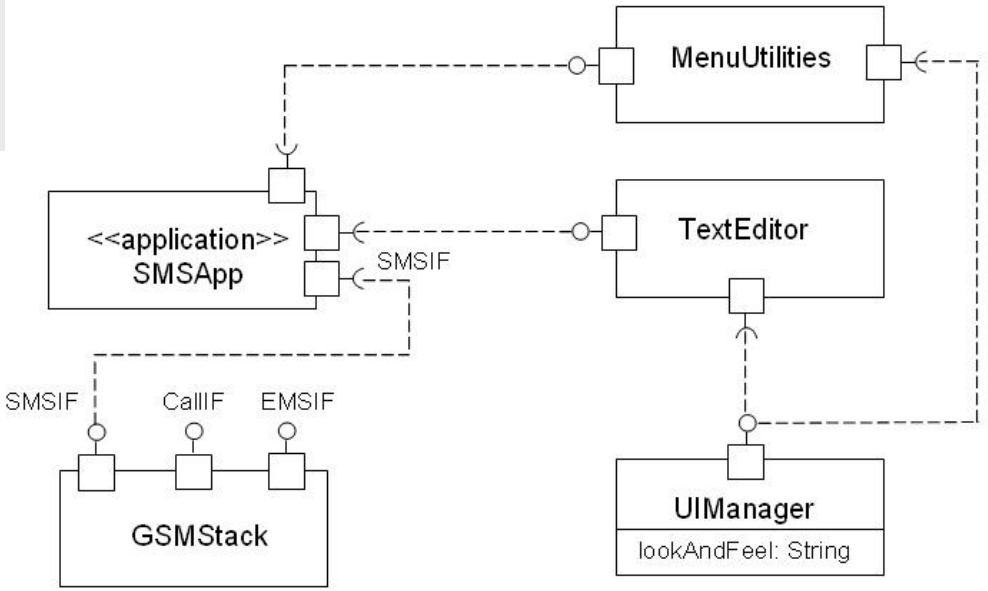
- Komponens - Architektúra referencia modell



MDSD Target Architectures

- Komponens - Függőségek

```
public class SMSAppImpl {
  public void doSomething() {
   TextEditor editor = (TextEditor)
       Factory.getComponent("TextEditor");
   ......
  }
}
```



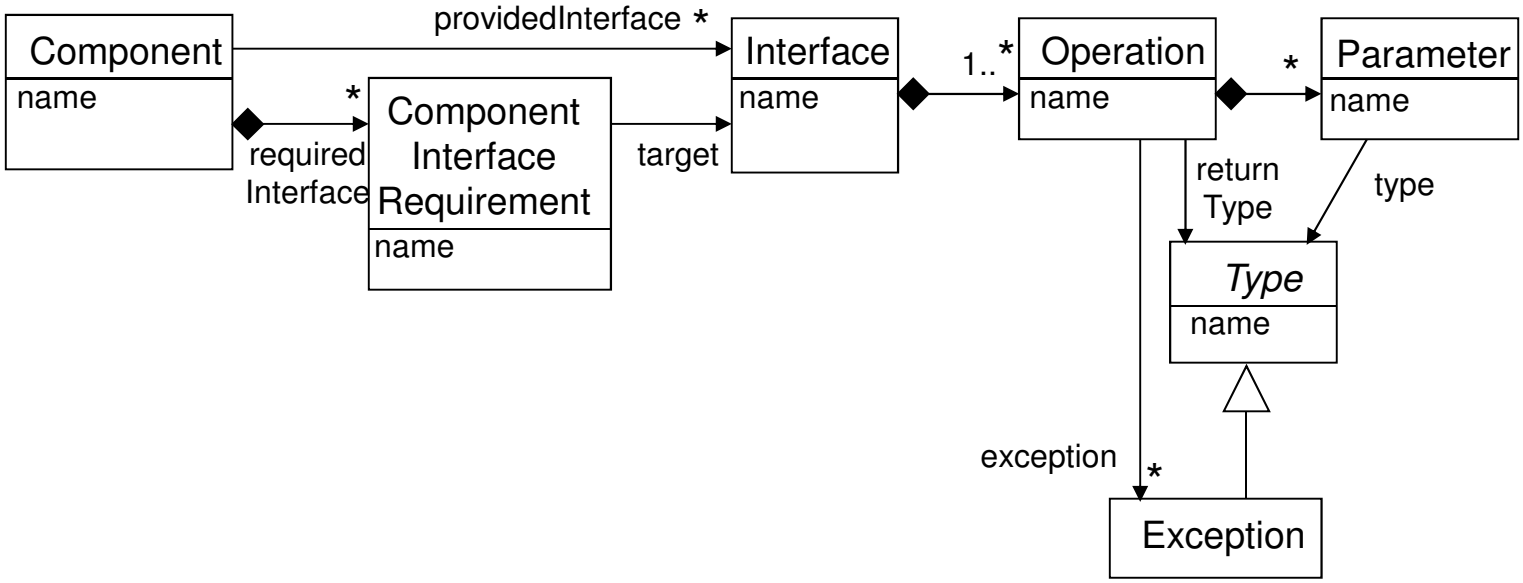
MDSD Target Architectures

- Komponens - Függőségek

```
public interface SMSAppContext
                 extends ComponentContext {
  public TextEditorIF getTextEditorIF();
  public SMSIF getSMSIF();
  public MenuIF getMenuIF();
}
public class SMSAppImpl implements Component {
  private SMSAppContext context = null;
  public void init(ComponentContext ctx) {
     context = (SMSAppContext)ctx;
  }
  public void doSomething() {
   TextEditor editor =
       context.getTextEditorIF();
    ......
  }
}
```

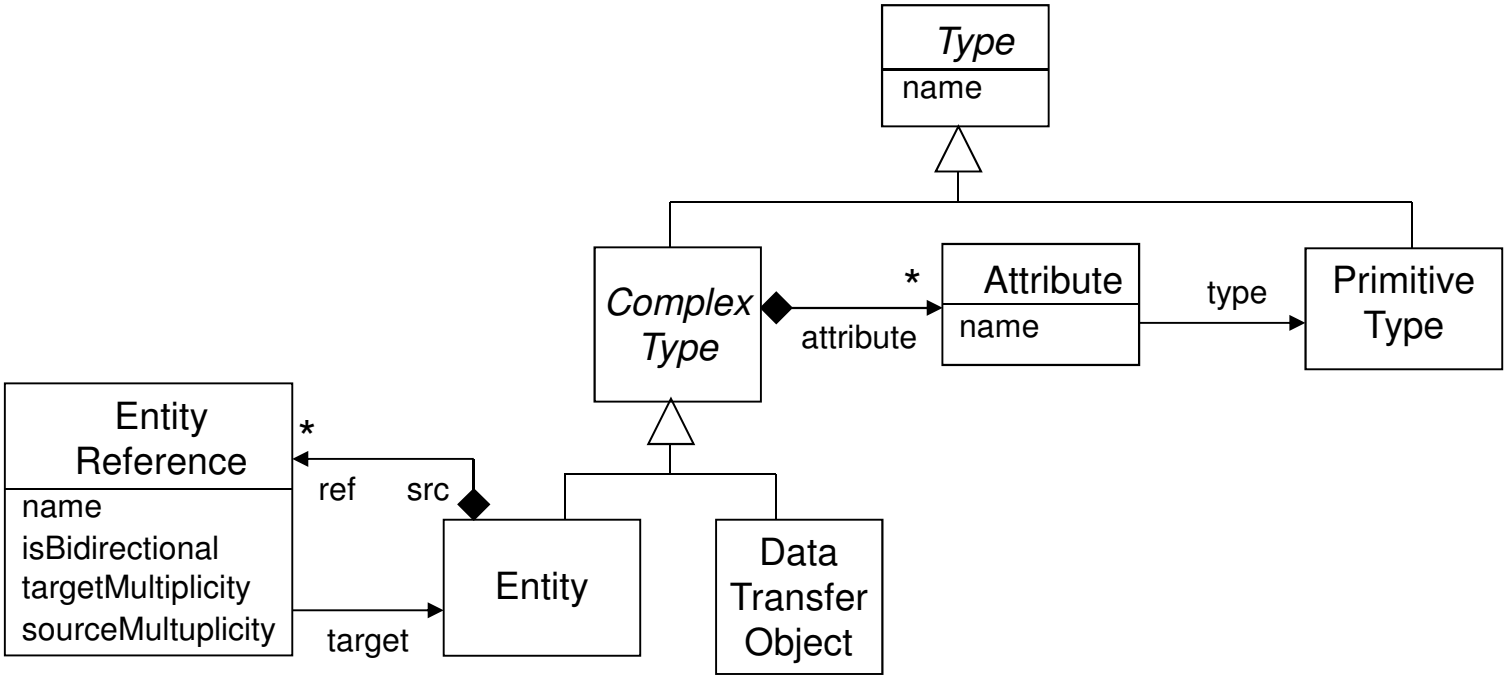
MDSD Target Architectures

- Komponens – nézetek
  - type view



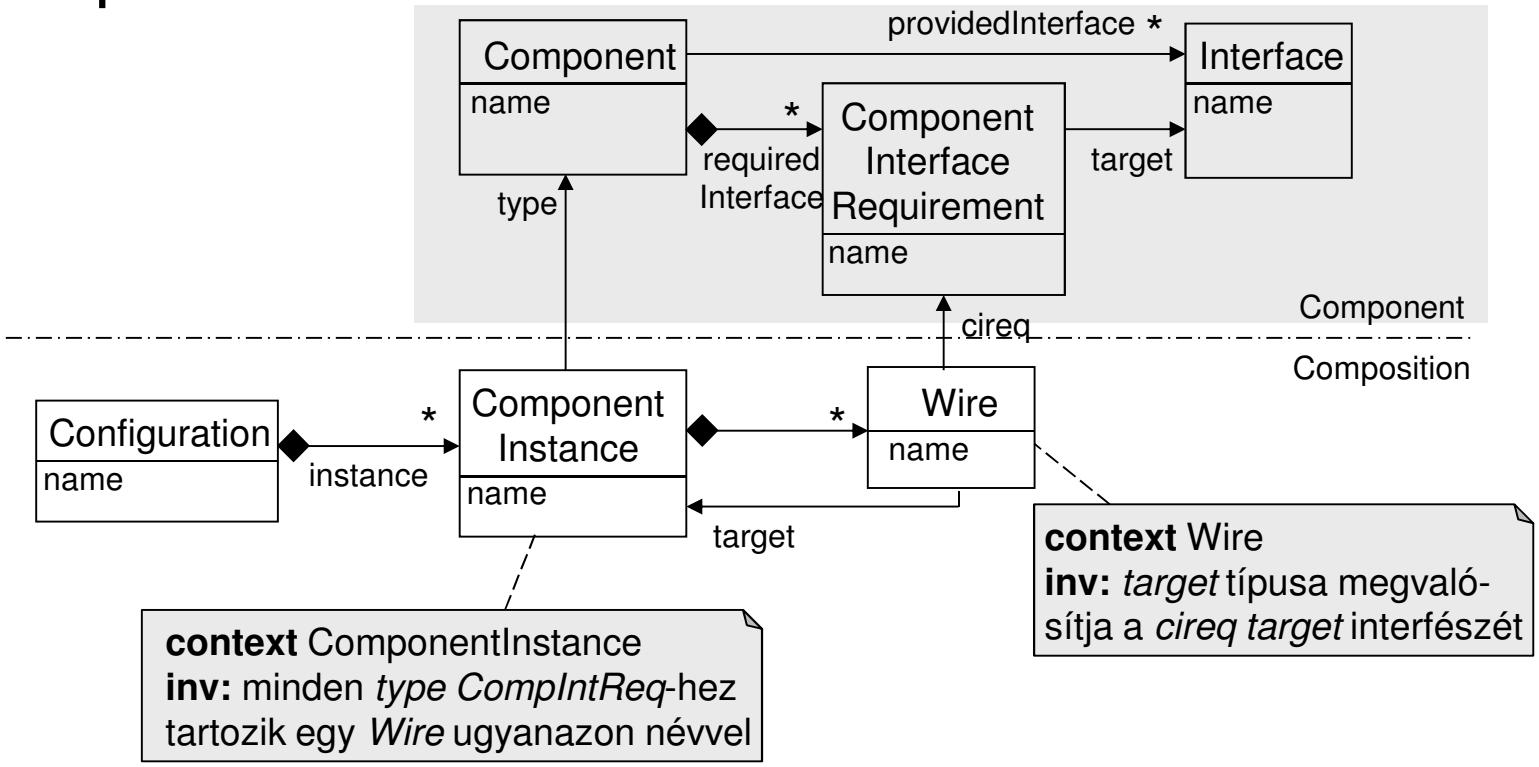
MDSD Target Architectures

- Komponens – nézetek
  - type view



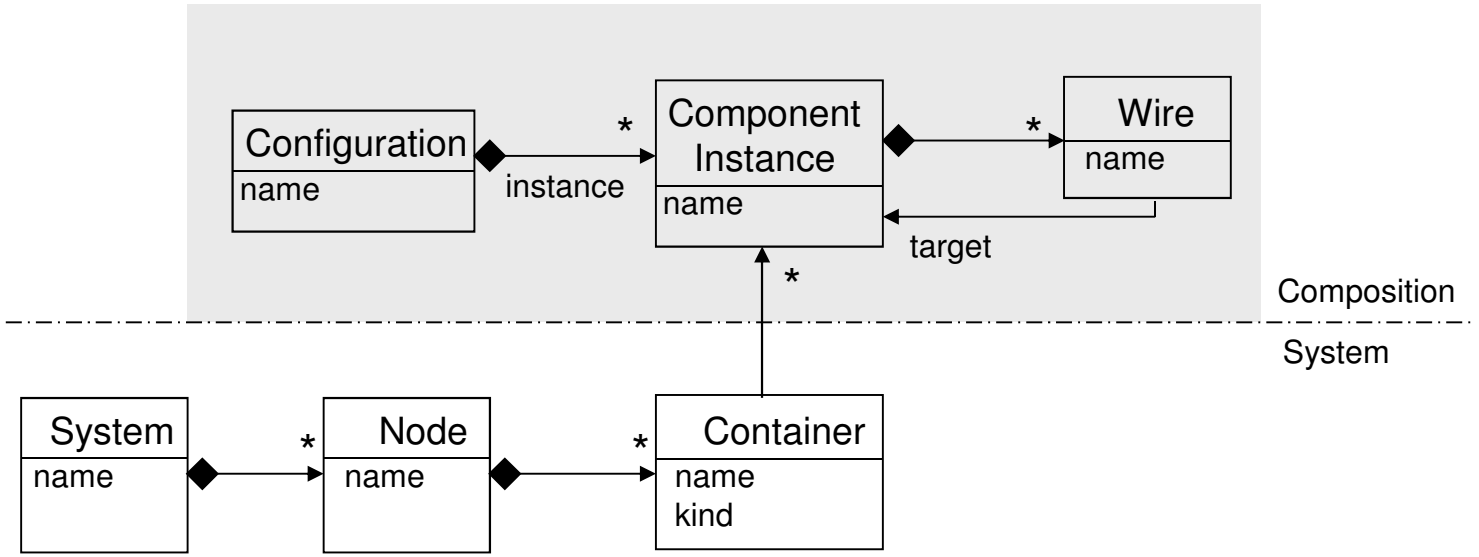
MDSD Target Architectures

- Komponens – nézetek
  - composition view



MDSD Target Architectures

- Komponens – nézetek
  - system view



MDSD Target Architectures

- Aspektus - példa