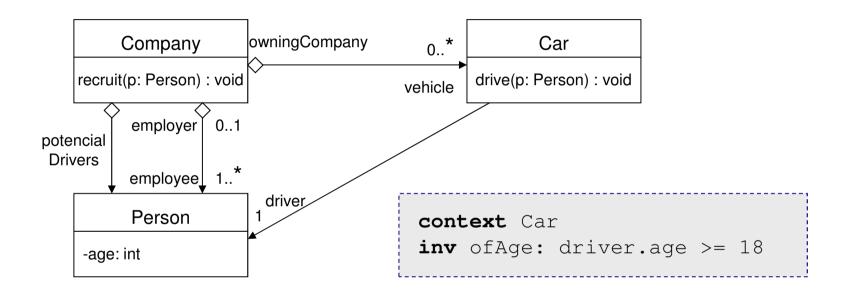
Metamodellek a szoftverfejlesztésben

Metamodellezés és architektúra



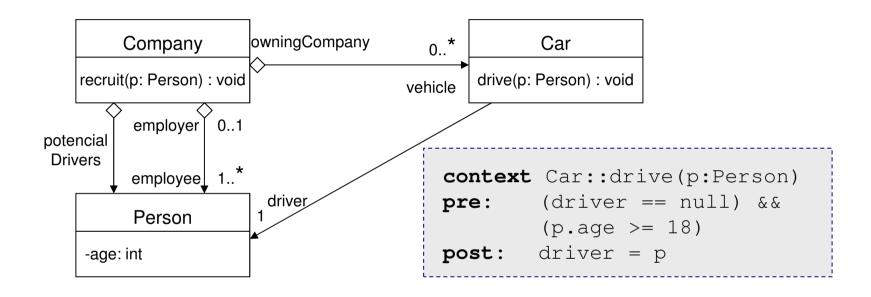
Metamodellezés és OCL



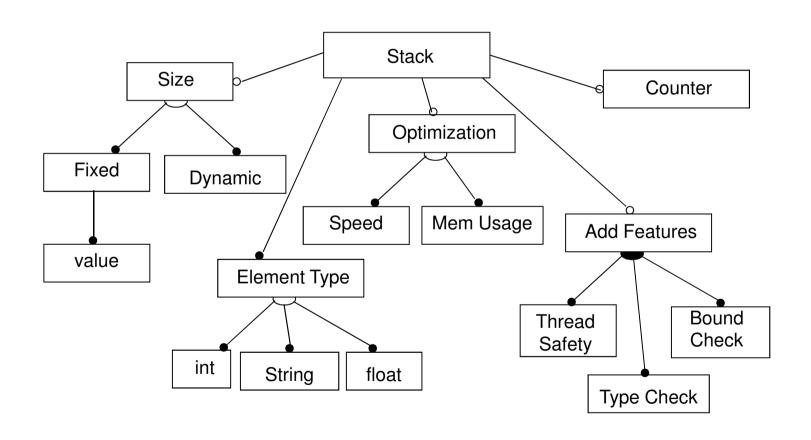
```
context Company
def potentialDrivers = employee->select(age >= 18)
```



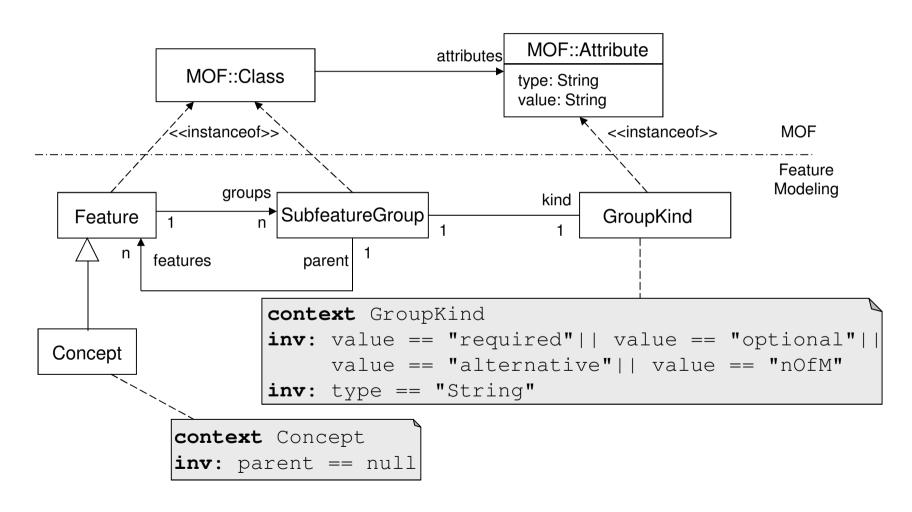
Metamodellezés és OCL



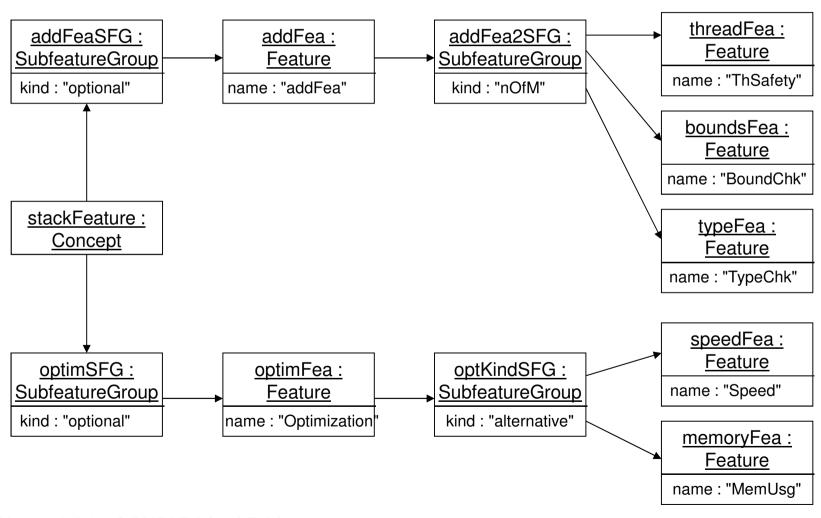




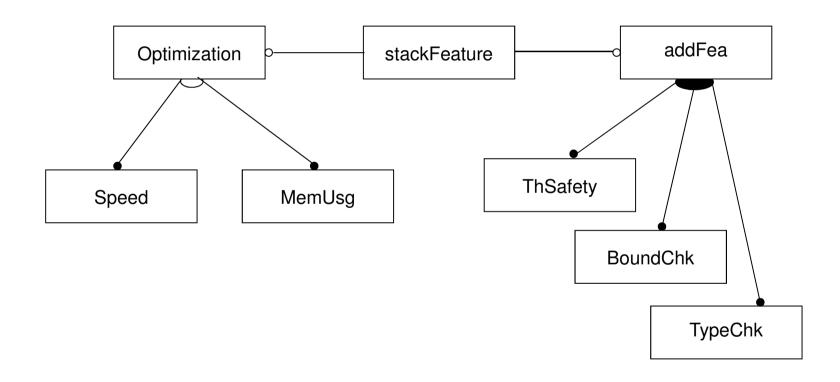




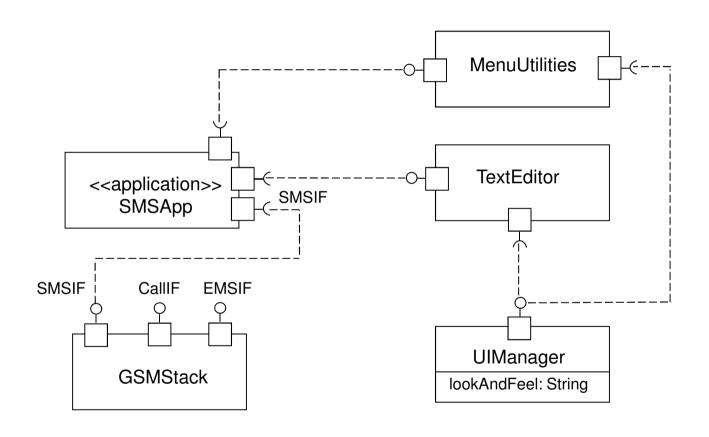




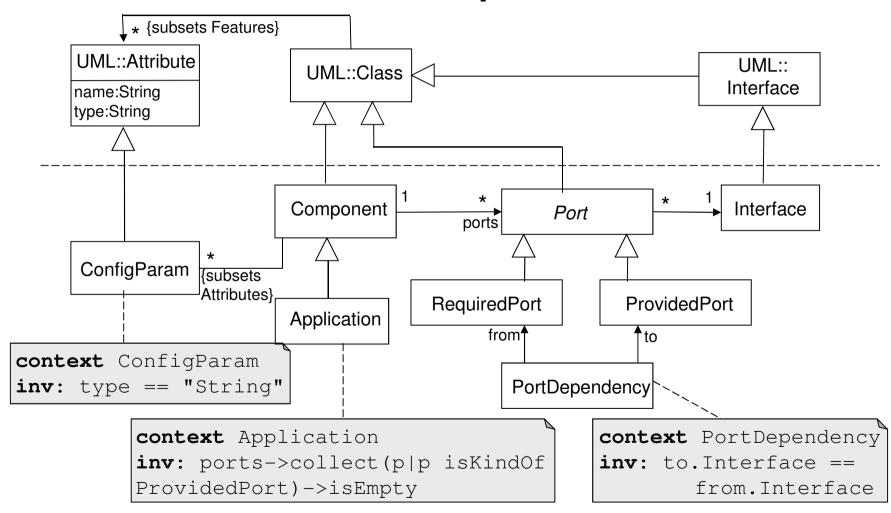




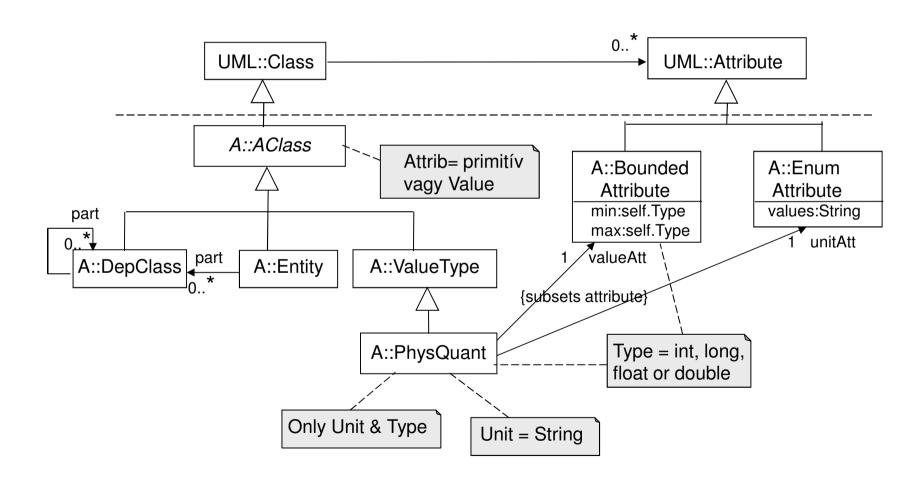






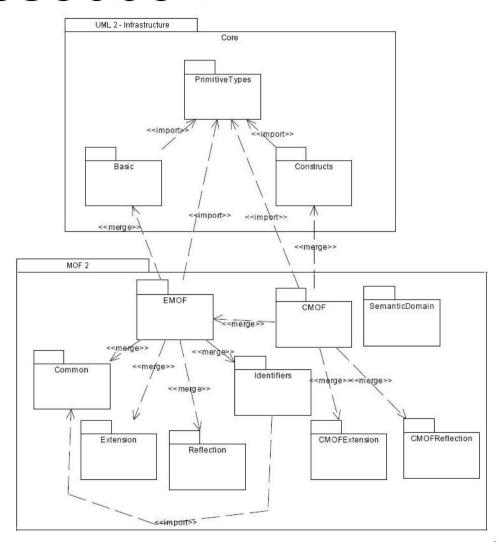






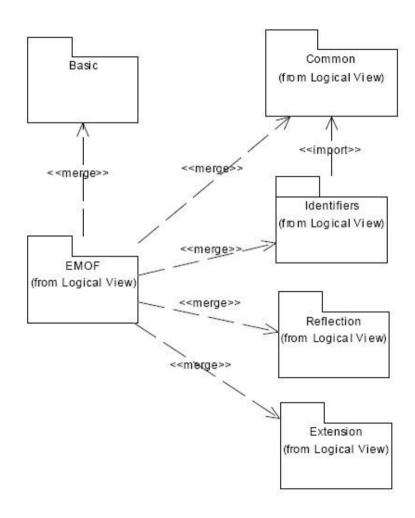


UML2-n alapul

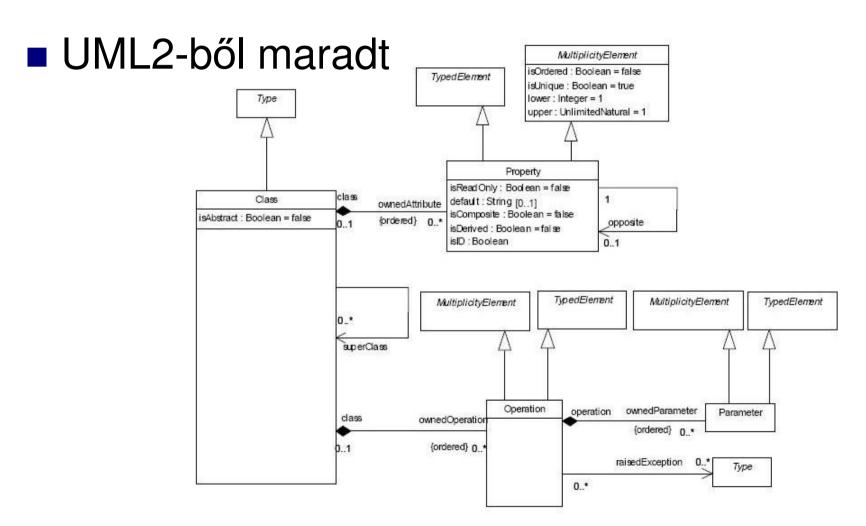




- EMOF essential
 - minimális
 - □assoc. helyett ref.
 - CMOF konvert.

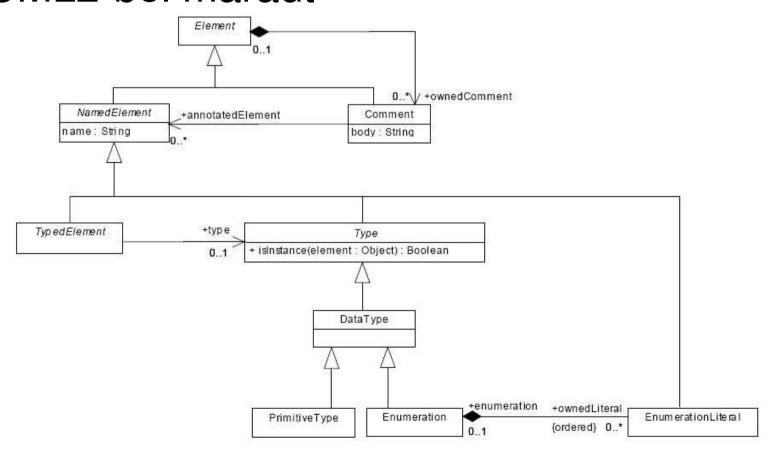








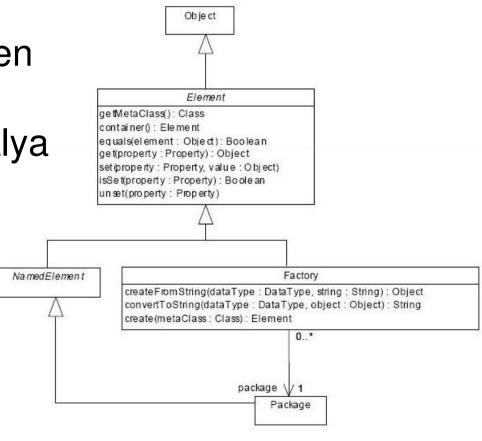
■ UML2-ből maradt





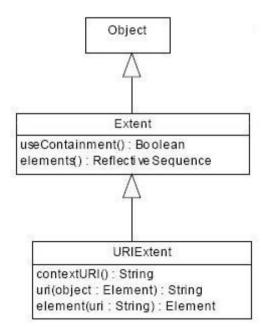
- Reflection
 - □ Element = minden modell elem és példány ősosztálya

 - MDA-hoz





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

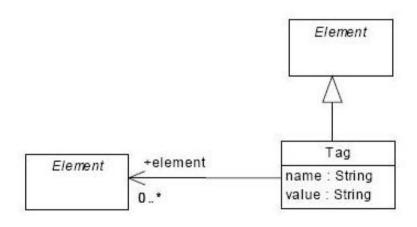




Common

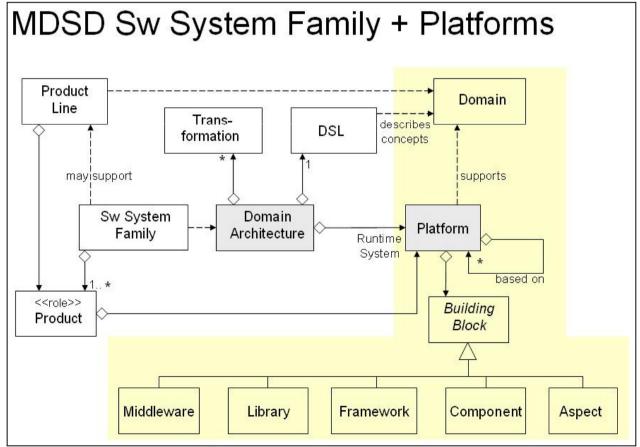
ReflectiveCollection add(object : Object) : Boolean addAll(objects : ReflectiveSequence) : Boolean clear() remove(object : Object) : Boolean size() : Integer ReflectiveSequence add(index : Integer, object : Object) get(index : Integer) : Object remove(index : Integer) : Object set(index : Integer, object : Object) : Object

Extension





Mi a "Target" architektúra ?

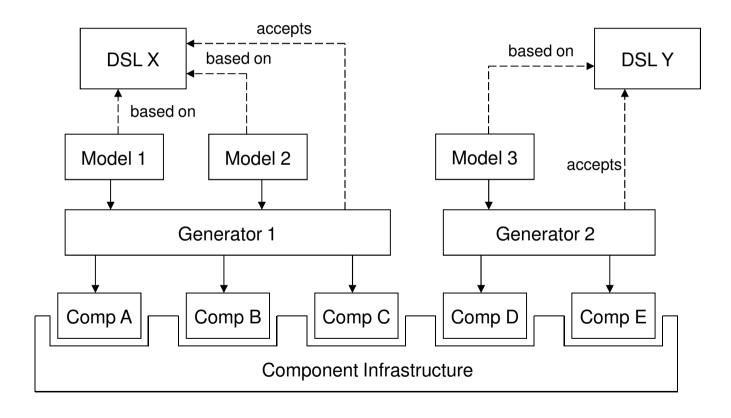




- Elemei:
 - □ Domain architecture transzformáció, eszköz
 - □ Platform
 - framework konfig + kiterjesztés
 - middleware speciális keretrendszer
 - komponens önhordó interfész körny. függ



Komponens rendszer





Komponens - Architektúra referencia modell

Application		
Business Platform	Main EntitiesMain ValuetypesBusiness RulesBusiness Services	
Technical Platform/ Middleware	PersistenceTransactionsDistributionSchedulingHardware Access	
Prog. Lang & Libraries		
Operating System		



Komponens - Függőségek

```
public class SMSAppImpl {
  public void doSomething() {
    TextEditor editor = (TextEditor)
         Factory.getComponent("TextEditor");
                                                                          MenuUtilities
                                                                          TextEditor
                                            <<application>>
                                                           SMSIF
                                              SMSApp
                                        SMSIF
                                                CalliF EMSIF
                                                                          UIManager
                                              GSMStack
                                                                        lookAndFeel: String
```

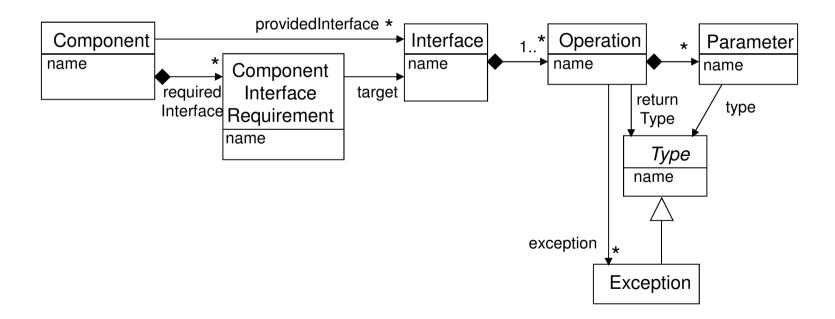


Komponens - Függőségek

```
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();
```

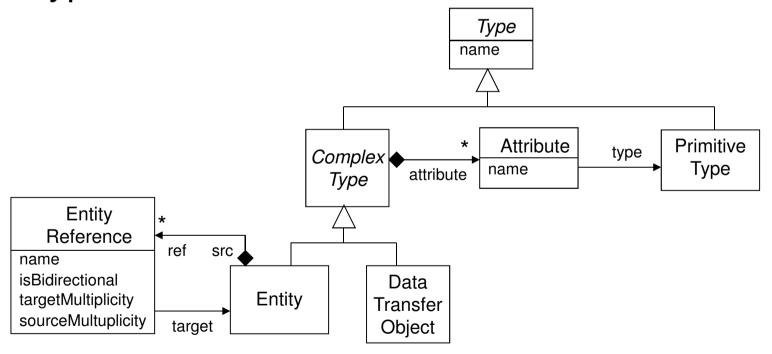


- Komponens nézetek
 - □ type view



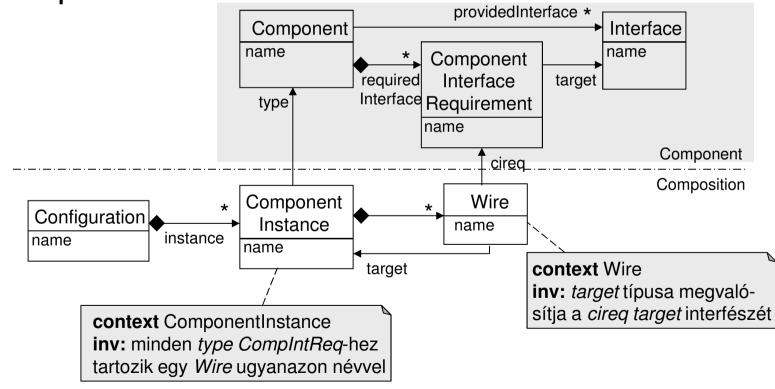


- Komponens nézetek
 - □ type view



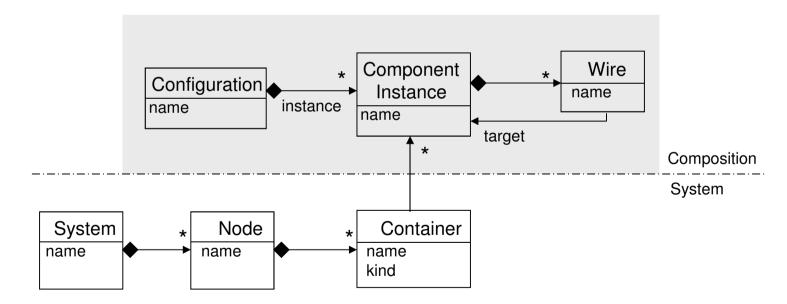


- Komponens nézetek
 - □ composition view





- Komponens nézetek
 - □ system view





Туре name Aspektus - példa Attribute Primiti∨e type Complex Type name attribute Type Entity Reference ref src name Component isBidirectional Data name **Entity** targetMultiplicity Transfer sourceMultuplicity target Object ref entity 0..1 $\mathsf{column}_{\pmb{\star}}$ EntityTable **DBPrimitive** RefTable Column name name Type name pk index Index Query Query {ordered} name name Component expression