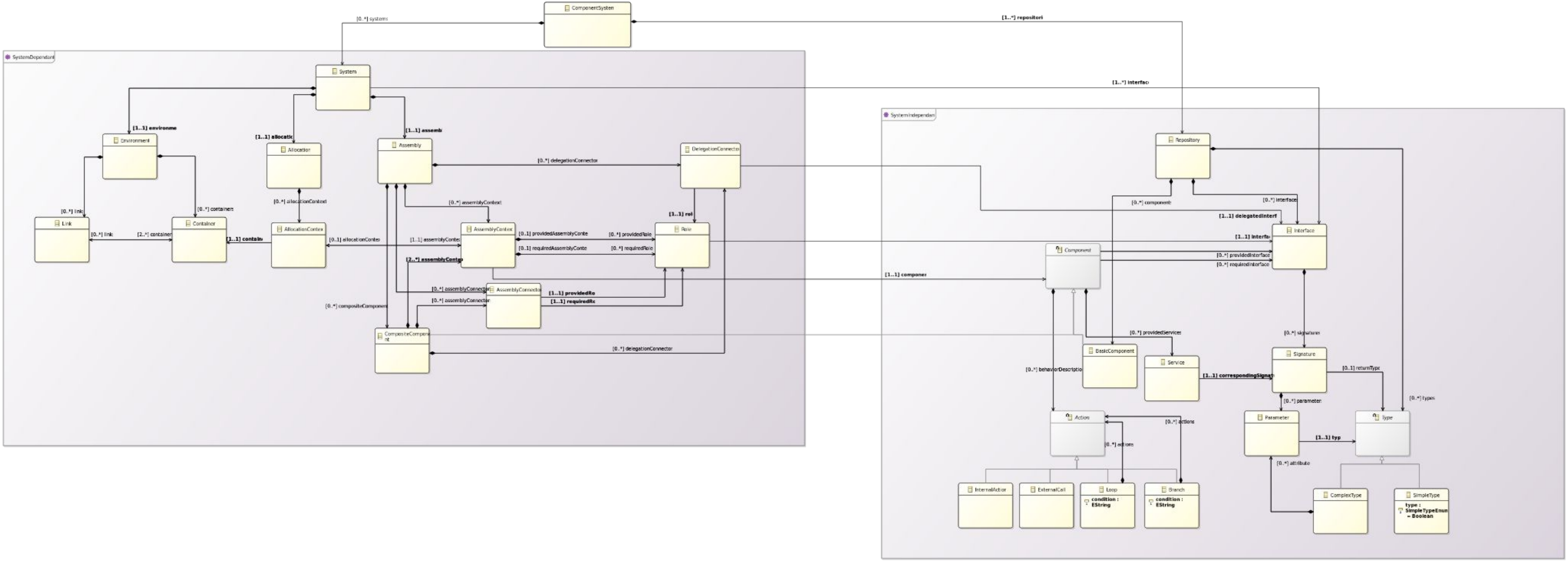
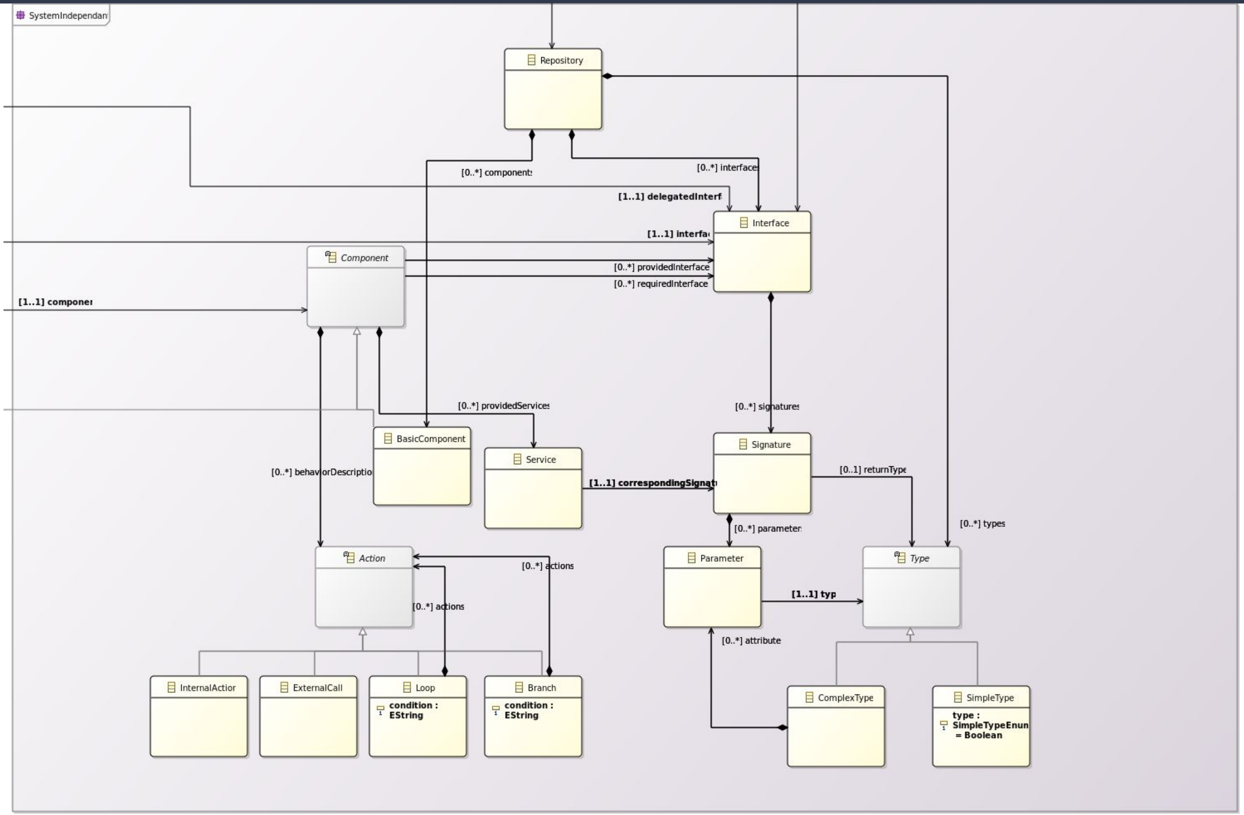
MDSD Abschlusspräsentation

Ekaterina Ehringhaus,
Annika Kienle,
Simon von Rönn

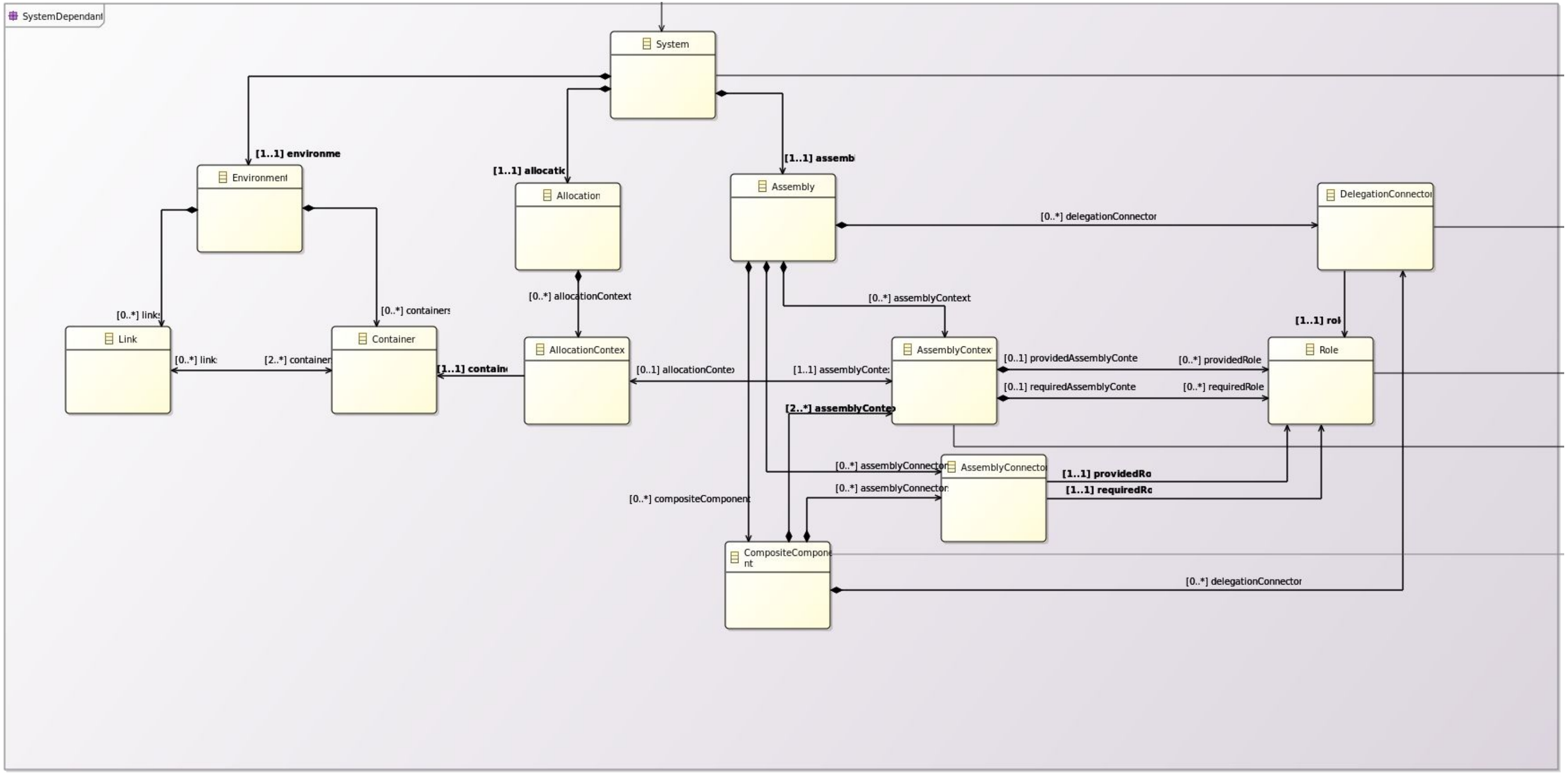
1 Unser Metamodell: Übersicht



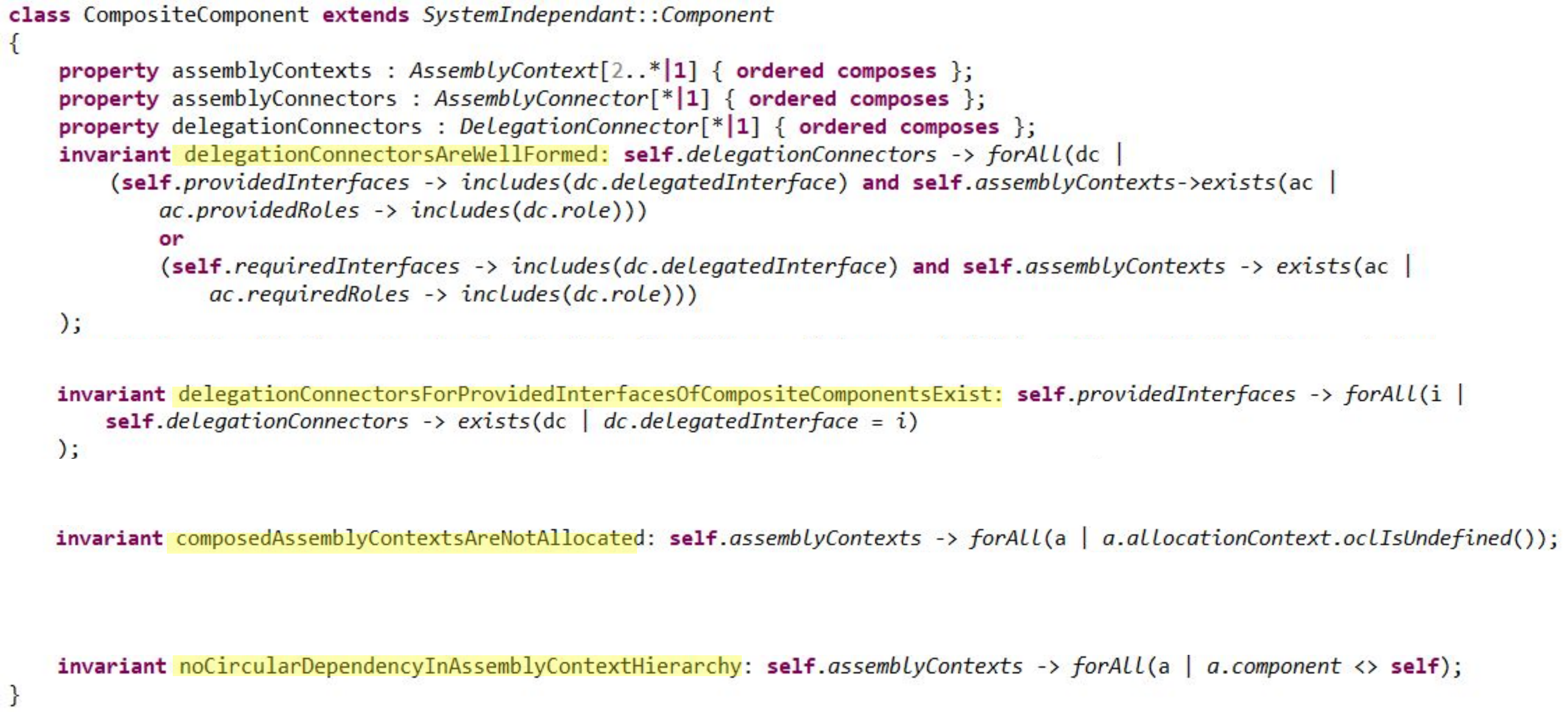
1 Unser Metamodell: ‘System Independant’ Teil



1 Unser Metamodell: ‘System Dependant’ Teil



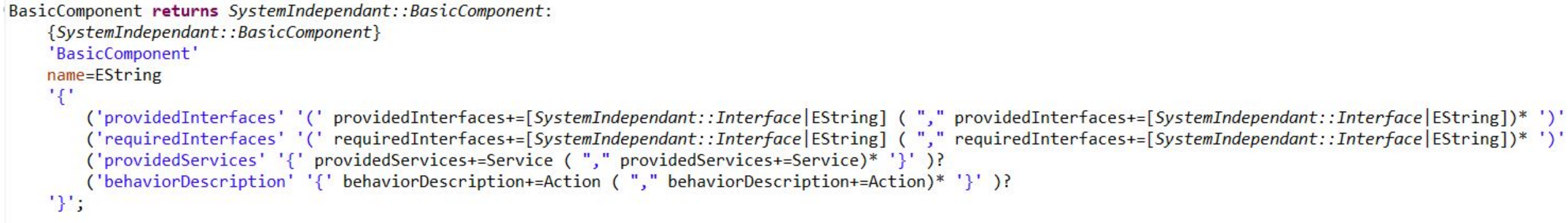
1 Unser Metamodell: OCL Constraints



```
class CompositeComponent extends SystemIndependant::Component
{
    property assemblyContexts : AssemblyContext[2..*|1] { ordered composes };
    property assemblyConnectors : AssemblyConnector[*|1] { ordered composes };
    property delegationConnectors : DelegationConnector[*|1] { ordered composes };
    invariant delegationConnectorsAreWellFormed: self.delegationConnectors -> forAll(dc |
        (self.providedInterfaces -> includes(dc.delegatedInterface) and self.assemblyContexts->exists(ac |
            ac.providedRoles -> includes(dc.role)))
            or
            (self.requiredInterfaces -> includes(dc.delegatedInterface) and self.assemblyContexts -> exists(ac |
                ac.requiredRoles -> includes(dc.role)))
    );


    invariant delegationConnectorsForProvidedInterfacesOfCompositeComponentsExist: self.providedInterfaces -> forAll(i |
        self.delegationConnectors -> exists(dc | dc.delegatedInterface = i)
    );


    invariant composedAssemblyContextsAreNotAllocated: self.assemblyContexts -> forAll(a | a.allocationContext.oclIsUndefined());



    invariant noCircularDependencyInAssemblyContextHierarchy: self.assemblyContexts -> forAll(a | a.component <> self);
}
```

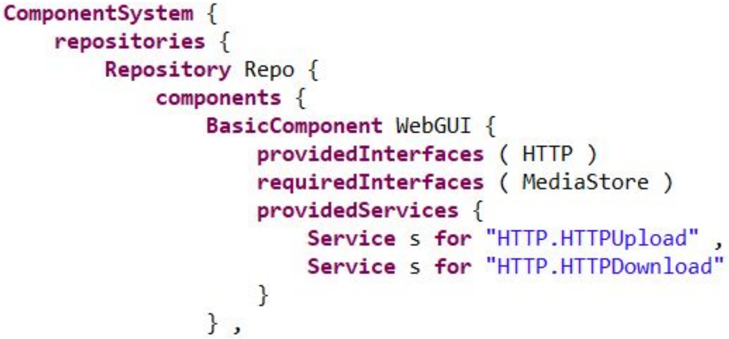
2 Unsere Xtext Grammatik: Basic Component

Code



```
BasicComponent returns SystemIndependant::BasicComponent:
    {SystemIndependant::BasicComponent}
    'BasicComponent'
    name=EString
    '{'
        ('providedInterfaces' '(' providedInterfaces+=[SystemIndependant::Interface|EString] ( "," providedInterfaces+=[SystemIndependant::Interface|EString])* ')'
        ('requiredInterfaces' '(' requiredInterfaces+=[SystemIndependant::Interface|EString] ( "," requiredInterfaces+=[SystemIndependant::Interface|EString])* ')'
        ('providedServices' '{' providedServices+=Service ( "," providedServices+=Service)* '}' )?
        ('behaviorDescription' '{' behaviorDescription+=Action ( "," behaviorDescription+=Action)* '}' )?
    '}';
```

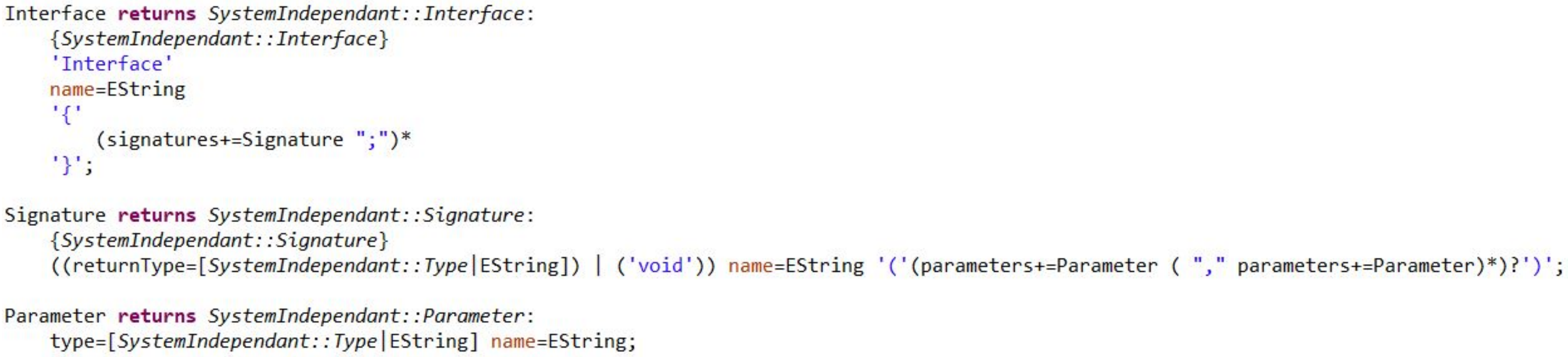
Instance



```
ComponentSystem {
    repositories {
        Repository Repo {
            components {
                BasicComponent WebGUI {
                    providedInterfaces ( HTTP )
                    requiredInterfaces ( MediaStore )
                    providedServices {
                        Service s for "HTTP.HTTPUpload" ,
                        Service s for "HTTP.HTTPDownload"
                    }
                } ,
```

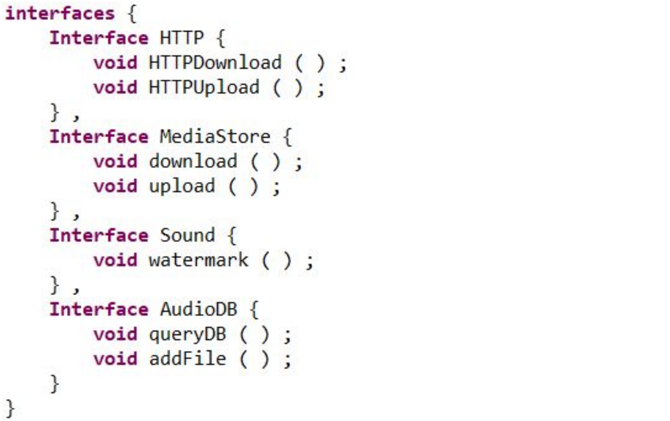
2 Unsere Xtext Grammatik: Interface

Code

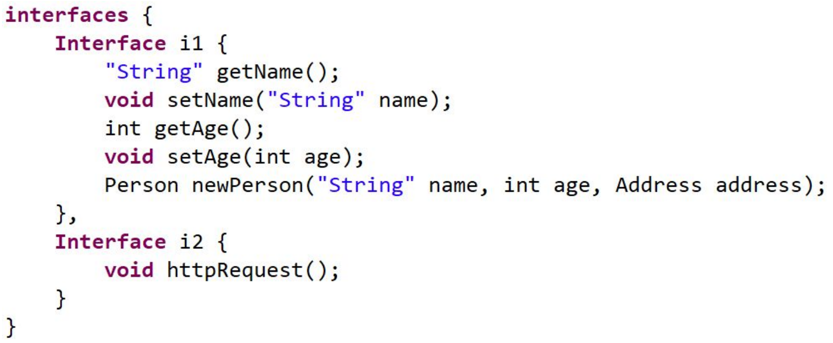


```
Interface returns SystemIndependant::Interface:
    {SystemIndependant::Interface}
    'Interface'
    name=EString
    '{'
        (signatures+=Signature ";")*
    '}';

Signature returns SystemIndependant::Signature:
    {SystemIndependant::Signature}
    ((returnType=[SystemIndependant::Type|EString]) | ('void')) name=EString '('(parameters+=Parameter ( "," parameters+=Parameter)*)?')';

Parameter returns SystemIndependant::Parameter:
    type=[SystemIndependant::Type|EString] name=EString;
```

Instance

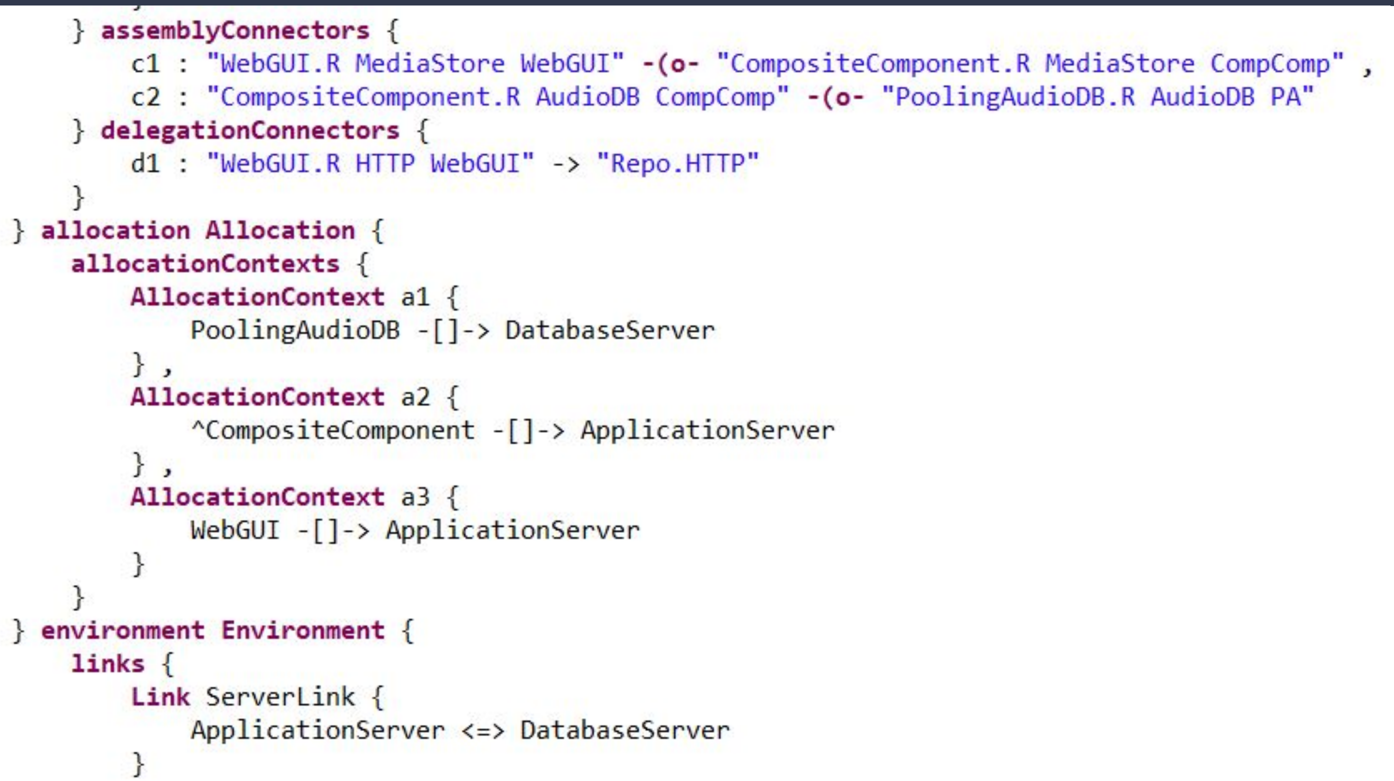


```
interfaces {
    Interface HTTP {
        void HTTPDownload ( ) ;
        void HTTPUpload ( ) ;
    } ,
    Interface MediaStore {
        void download ( ) ;
        void upload ( ) ;
    } ,
    Interface Sound {
        void watermark ( ) ;
    } ,
    Interface AudioDB {
        void queryDB ( ) ;
        void addFile ( ) ;
    }
}
```



```
interfaces {
    Interface i1 {
        "String" getName();
        void setName("String" name);
        int getAge();
        void setAge(int age);
        Person newPerson("String" name, int age, Address address);
    },
    Interface i2 {
        void httpRequest();
    }
}
```

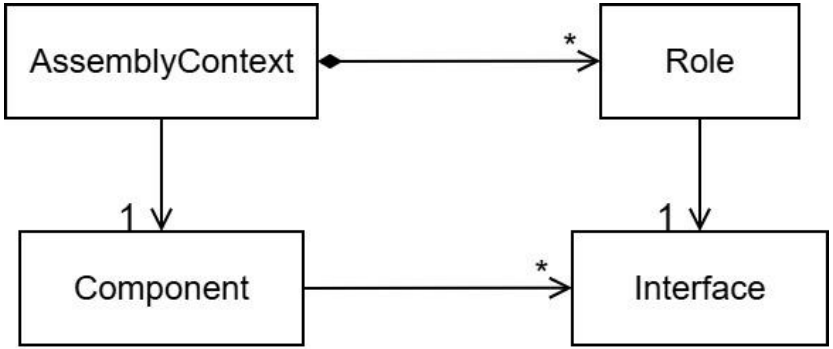
2 Unsere Xtext Grammatik: Custom composition arrows



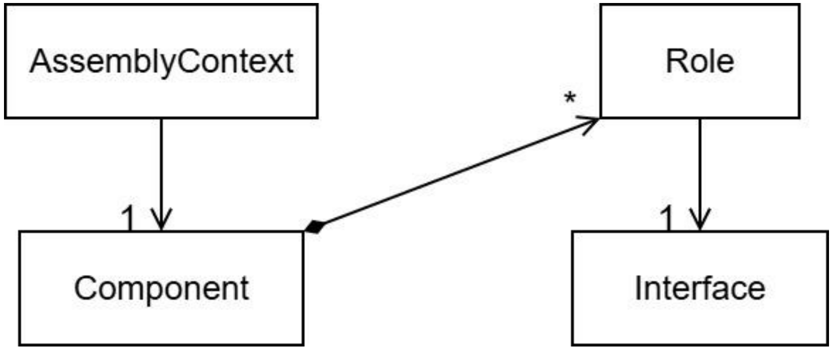
```
    } assemblyConnectors {
        c1 : "WebGUI.R MediaStore WebGUI" -(o- "CompositeComponent.R MediaStore CompComp" ,
        c2 : "CompositeComponent.R AudioDB CompComp" -(o- "PoolingAudioDB.R AudioDB PA"
    } delegationConnectors {
        d1 : "WebGUI.R HTTP WebGUI" -> "Repo.HTTP"
    }
} allocation Allocation {
    allocationContexts {
        AllocationContext a1 {
            PoolingAudioDB -[]-> DatabaseServer
        } ,
        AllocationContext a2 {
            ^CompositeComponent -[]-> ApplicationServer
        } ,
        AllocationContext a3 {
            WebGUI -[]-> ApplicationServer
        }
    }
} environment Environment {
    links {
        Link ServerLink {
            ApplicationServer <=> DatabaseServer
        }
```

3 QVTo-Transformation: Unterschiede in Metamodellen

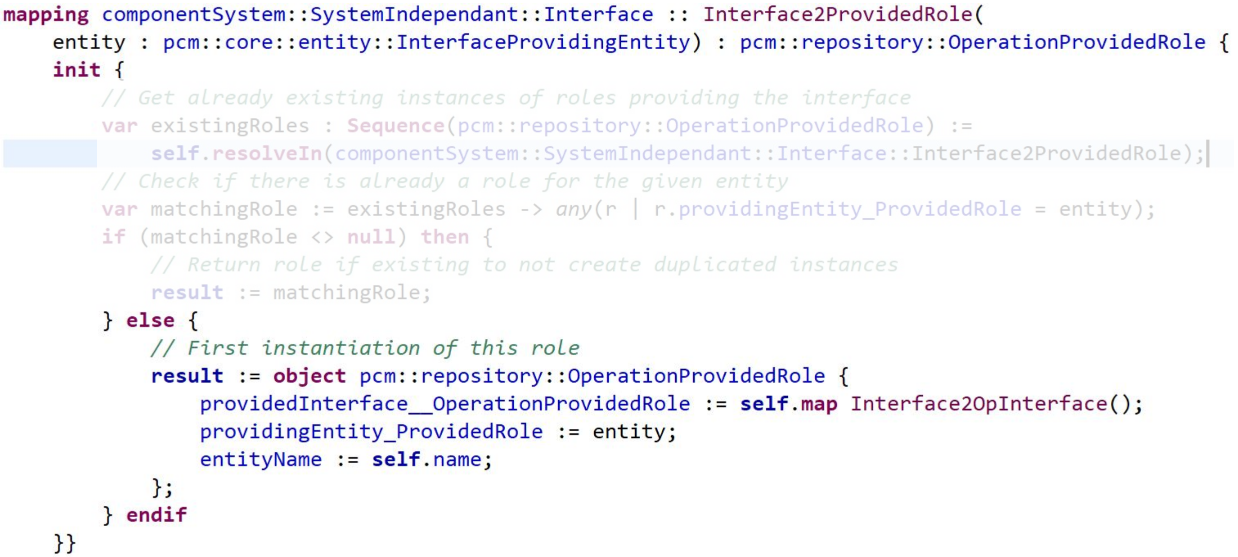
ComponentSystem



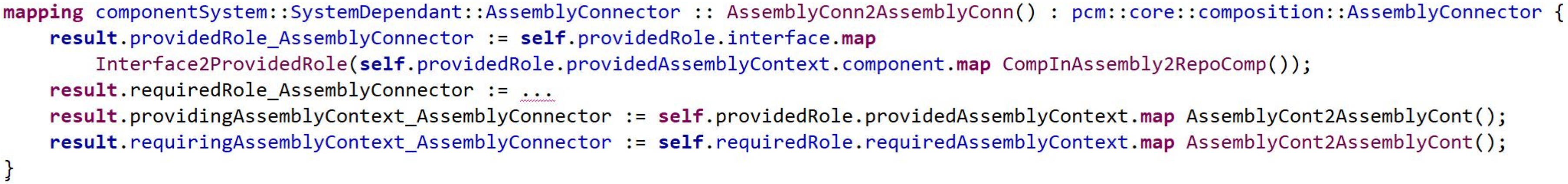
PCM



3 QVTo-Transformation: Unterschiede in Metamodellen

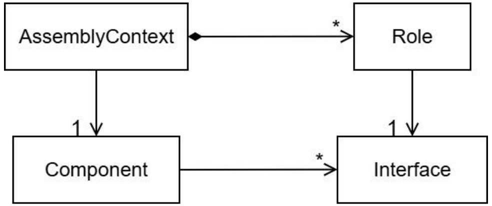


```
mapping componentSystem::SystemIndependant::Interface :: Interface2ProvidedRole(
    entity : pcm::core::entity::InterfaceProvidingEntity) : pcm::repository::OperationProvidedRole {
    init {
        // Get already existing instances of roles providing the interface
        var existingRoles : Sequence(pcm::repository::OperationProvidedRole) :=
            self.resolveIn(componentSystem::SystemIndependant::Interface::Interface2ProvidedRole);
        // Check if there is already a role for the given entity
        var matchingRole := existingRoles -> any(r | r.providingEntity_ProvidedRole = entity);
        if (matchingRole <> null) then {
            // Return role if existing to not create duplicated instances
            result := matchingRole;
        } else {
            // First instantiation of this role
            result := object pcm::repository::OperationProvidedRole {
                providedInterface__OperationProvidedRole := self.map Interface2OpInterface();
                providingEntity_ProvidedRole := entity;
                entityName := self.name;
            };
        } endif
    }}
```

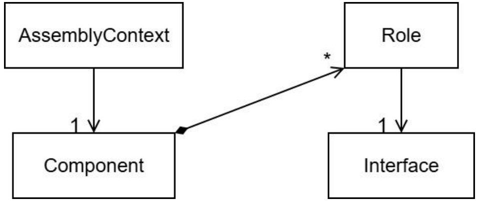


```
mapping componentSystem::SystemDependant::AssemblyConnector :: AssemblyConn2AssemblyConn() : pcm::core::composition::AssemblyConnector {
    result.providedRole_AssemblyConnector := self.providedRole.interface.map
        Interface2ProvidedRole(self.providedRole.providedAssemblyContext.component.map CompInAssembly2RepoComp());
    result.requiredRole_AssemblyConnector := ...
    result.providingAssemblyContext_AssemblyConnector := self.providedRole.providedAssemblyContext.map AssemblyCont2AssemblyCont();
    result.requiringAssemblyContext_AssemblyConnector := self.requiredRole.requiredAssemblyContext.map AssemblyCont2AssemblyCont();
}
```

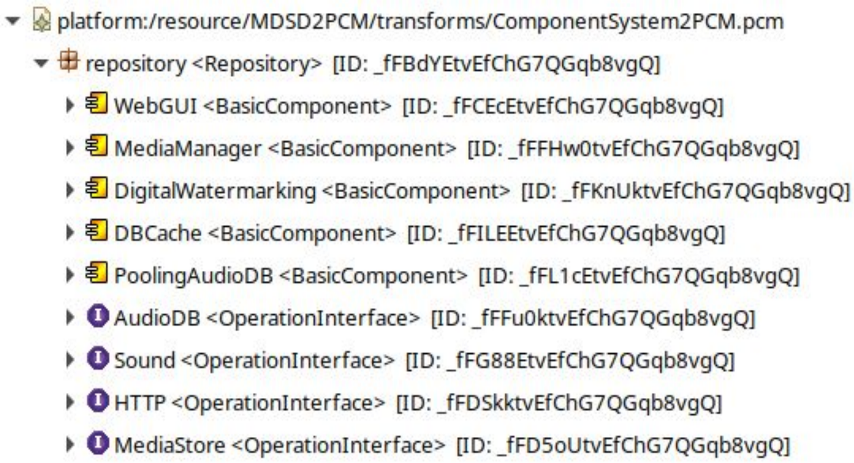
ComponentSystem



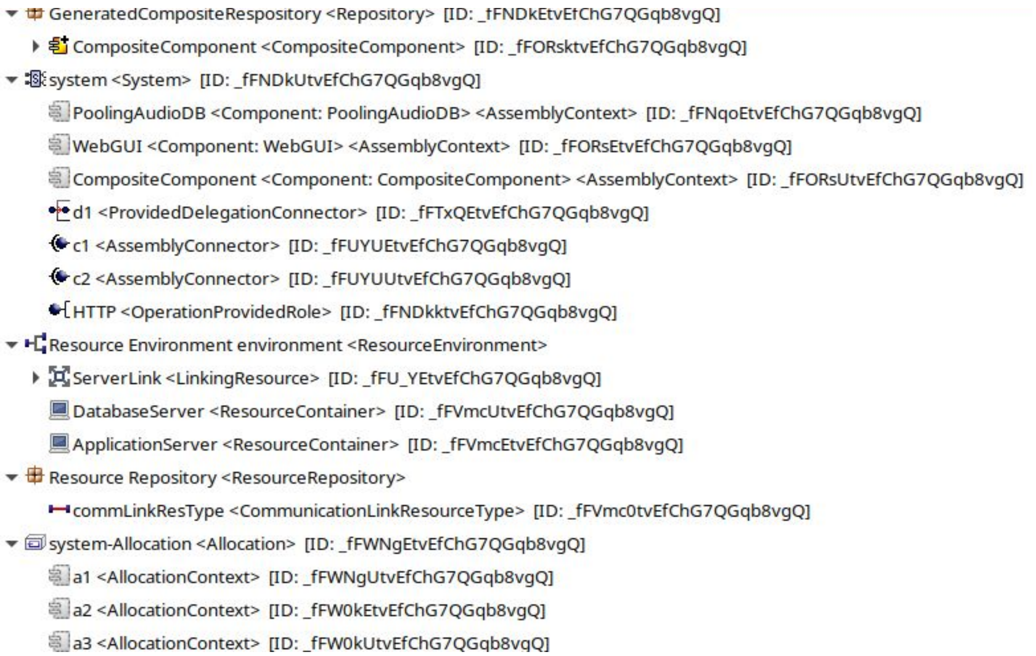
PCM



3 Unsere QVTo-Transformation: Instanz



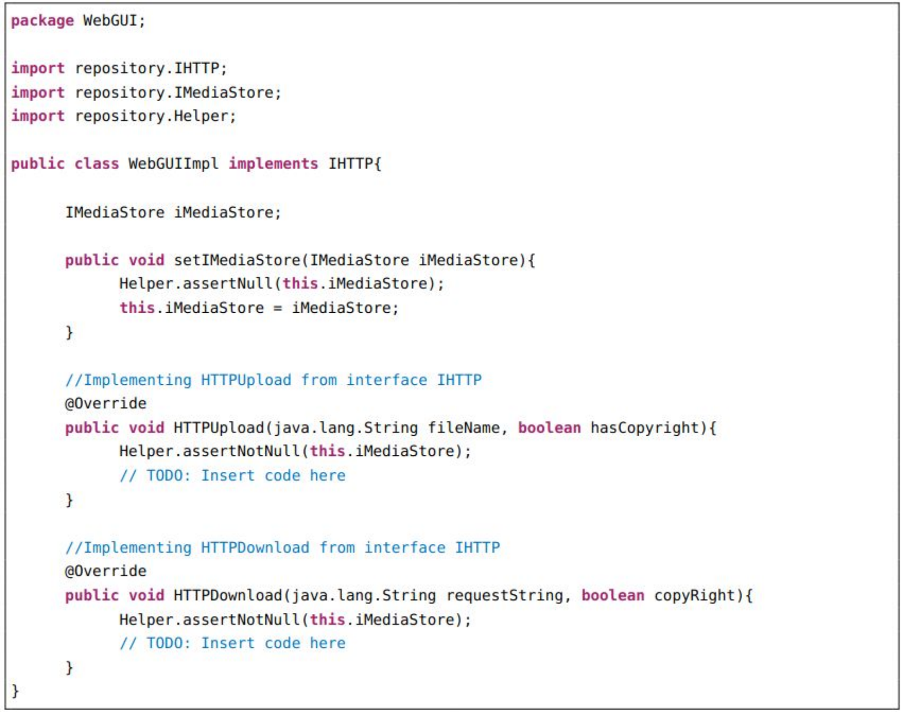
```
platform:/resource/MDSD2PCM/transforms/ComponentSystem2PCM.pcm
  repository <Repository>  [ID: _fFBdYEtvEfChG7QGqb8vgQ]
    WebGUI <BasicComponent>  [ID: _fFCEcEtvEfChG7QGqb8vgQ]
    MediaManager <BasicComponent>  [ID: _fFFHw0tvEfChG7QGqb8vgQ]
    DigitalWatermarking <BasicComponent>  [ID: _fFKnUktvEfChG7QGqb8vgQ]
    DBCache <BasicComponent>  [ID: _fFILEEtvEfChG7QGqb8vgQ]
    PoolingAudioDB <BasicComponent>  [ID: _fFL1cEtvEfChG7QGqb8vgQ]
    AudioDB <OperationInterface>  [ID: _fFFu0ktvEfChG7QGqb8vgQ]
    Sound <OperationInterface>  [ID: _fFG88EtvEfChG7QGqb8vgQ]
    HTTP <OperationInterface>  [ID: _fFDSkktvEfChG7QGqb8vgQ]
    MediaStore <OperationInterface>  [ID: _fFD5oUtvEfChG7QGqb8vgQ]
```



```
GeneratedCompositeRespository <Repository>  [ID: _fFNDkEtvEfChG7QGqb8vgQ]
  CompositeComponent <CompositeComponent>  [ID: _fFORsktvEfChG7QGqb8vgQ]
system <System>  [ID: _fFNDkUtvEfChG7QGqb8vgQ]
  PoolingAudioDB <Component: PoolingAudioDB> <AssemblyContext>  [ID: _fFNqoEtvEfChG7QGqb8vgQ]
  WebGUI <Component: WebGUI> <AssemblyContext>  [ID: _fFORsEtvEfChG7QGqb8vgQ]
  CompositeComponent <Component: CompositeComponent> <AssemblyContext>  [ID: _fFORsUtvEfChG7QGqb8vgQ]
  d1 <ProvidedDelegationConnector>  [ID: _fFTxQEtvEfChG7QGqb8vgQ]
  c1 <AssemblyConnector>  [ID: _fFUYUEtvEfChG7QGqb8vgQ]
  c2 <AssemblyConnector>  [ID: _fFUYUUtvEfChG7QGqb8vgQ]
  HTTP <OperationProvidedRole>  [ID: _fFNDkktvEfChG7QGqb8vgQ]
Resource Environment environment <ResourceEnvironment>
  ServerLink <LinkingResource>  [ID: _fFU_YEtvEfChG7QGqb8vgQ]
  DatabaseServer <ResourceContainer>  [ID: _fFVmcUtvEfChG7QGqb8vgQ]
  ApplicationServer <ResourceContainer>  [ID: _fFVmcEtvEfChG7QGqb8vgQ]
Resource Repository <ResourceRepository>
  commLinkResType <CommunicationLinkResourceType>  [ID: _fFVmc0tvEfChG7QGqb8vgQ]
system-Allocation <Allocation>  [ID: _fFWNgEtvEfChG7QGqb8vgQ]
  a1 <AllocationContext>  [ID: _fFWNgUtvEfChG7QGqb8vgQ]
  a2 <AllocationContext>  [ID: _fFW0kEtvEfChG7QGqb8vgQ]
  a3 <AllocationContext>  [ID: _fFW0kUtvEfChG7QGqb8vgQ]
```

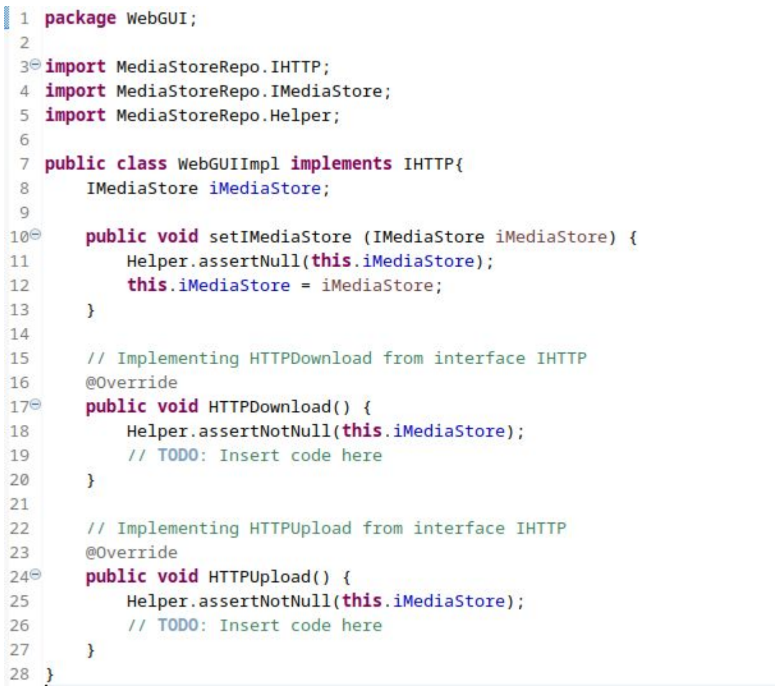
4 Unsere Codegenerierung: Ergebnis (1)

Die Anforderung



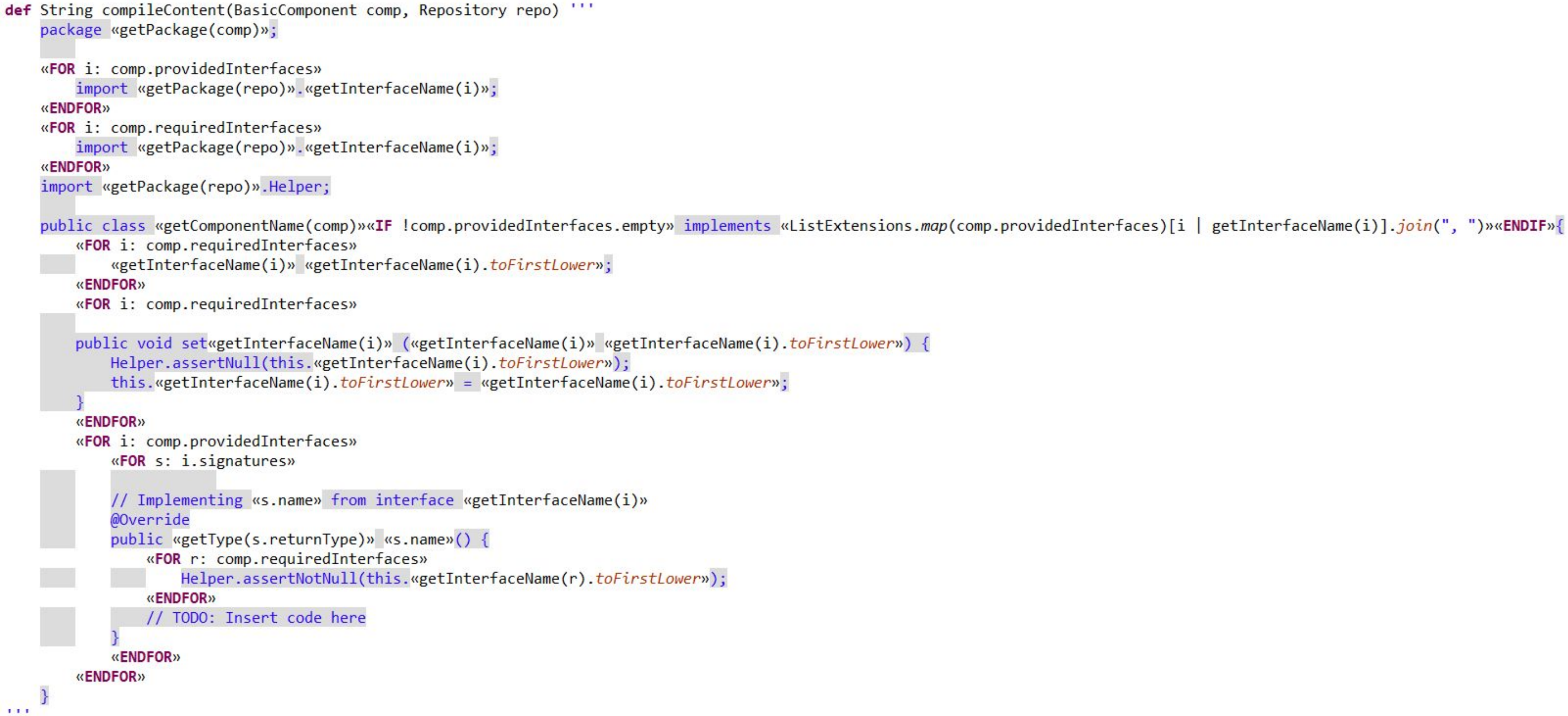
```java
package WebGUI;

import repository.IHTTP;
import repository.IMediaStore;
import repository.Helper;

public class WebGUIImpl implements IHTTP{

        IMediaStore iMediaStore;

        public void setIMediaStore(IMediaStore iMediaStore){
                Helper.assertNull(this.iMediaStore);
                this.iMediaStore = iMediaStore;
        }

        //Implementing HTTPUpload from interface IHTTP
        @Override
        public void HTTPUpload(java.lang.String fileName, boolean hasCopyright){
                Helper.assertNotNull(this.iMediaStore);
                // TODO: Insert code here
        }

        //Implementing HTTPDownload from interface IHTTP
        @Override
        public void HTTPDownload(java.lang.String requestString, boolean copyRight){
                Helper.assertNotNull(this.iMediaStore);
                // TODO: Insert code here
        }
}
```

Die Umsetzung



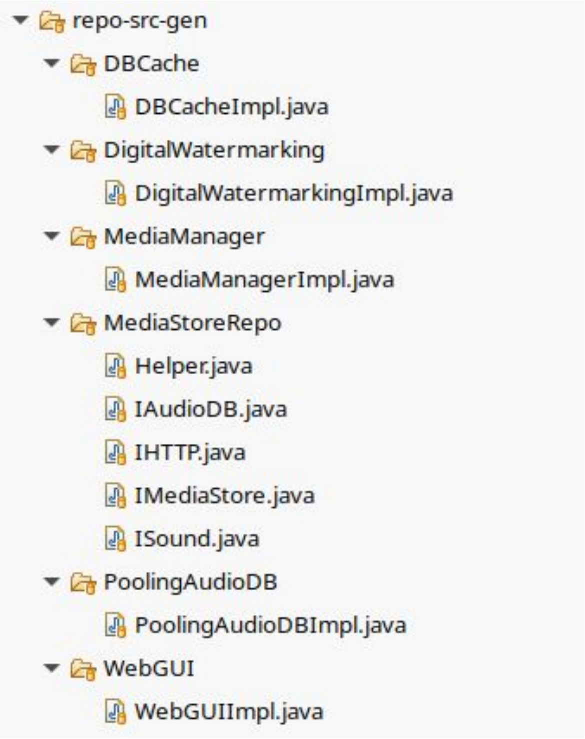
```java
package WebGUI;

import MediaStoreRepo.IHTTP;
import MediaStoreRepo.IMediaStore;
import MediaStoreRepo.Helper;

public class WebGUIImpl implements IHTTP{
    IMediaStore iMediaStore;

    public void setIMediaStore (IMediaStore iMediaStore) {
        Helper.assertNull(this.iMediaStore);
        this.iMediaStore = iMediaStore;
    }

    // Implementing HTTPDownload from interface IHTTP
    @Override
    public void HTTPDownload() {
        Helper.assertNotNull(this.iMediaStore);
        // TODO: Insert code here
    }

    // Implementing HTTPUpload from interface IHTTP
    @Override
    public void HTTPUpload() {
        Helper.assertNotNull(this.iMediaStore);
        // TODO: Insert code here
    }
}
```

4 Unsere Codegenerierung: Code

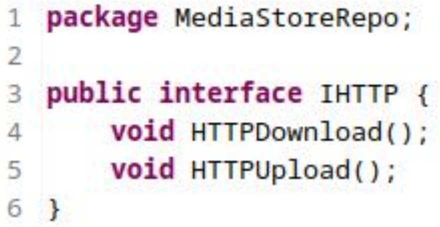


```
def String compileContent(BasicComponent comp, Repository repo) '''
    package «getPackage(comp)»;

    «FOR i: comp.providedInterfaces»
        import «getPackage(repo)».«getInterfaceName(i)»;
    «ENDFOR»
    «FOR i: comp.requiredInterfaces»
        import «getPackage(repo)».«getInterfaceName(i)»;
    «ENDFOR»
    import «getPackage(repo)».Helper;

    public class «getComponentName(comp)»«IF !comp.providedInterfaces.empty» implements «ListExtensions.map(comp.providedInterfaces)[i | getInterfaceName(i)].join(", ")»«ENDIF»{
        «FOR i: comp.requiredInterfaces»
            «getInterfaceName(i)» «getInterfaceName(i).toFirstLower»;
        «ENDFOR»
        «FOR i: comp.requiredInterfaces»

        public void set«getInterfaceName(i)» («getInterfaceName(i)» «getInterfaceName(i).toFirstLower») {
            Helper.assertNull(this.«getInterfaceName(i).toFirstLower»);
            this.«getInterfaceName(i).toFirstLower» = «getInterfaceName(i).toFirstLower»;
        }
        «ENDFOR»
        «FOR i: comp.providedInterfaces»
            «FOR s: i.signatures»

            // Implementing «s.name» from interface «getInterfaceName(i)»
            @Override
            public «getType(s.returnType)» «s.name»() {
                «FOR r: comp.requiredInterfaces»
                    Helper.assertNotNull(this.«getInterfaceName(r).toFirstLower»);
                «ENDFOR»
                // TODO: Insert code here
            }
            «ENDFOR»
        «ENDFOR»
    }
'''
```

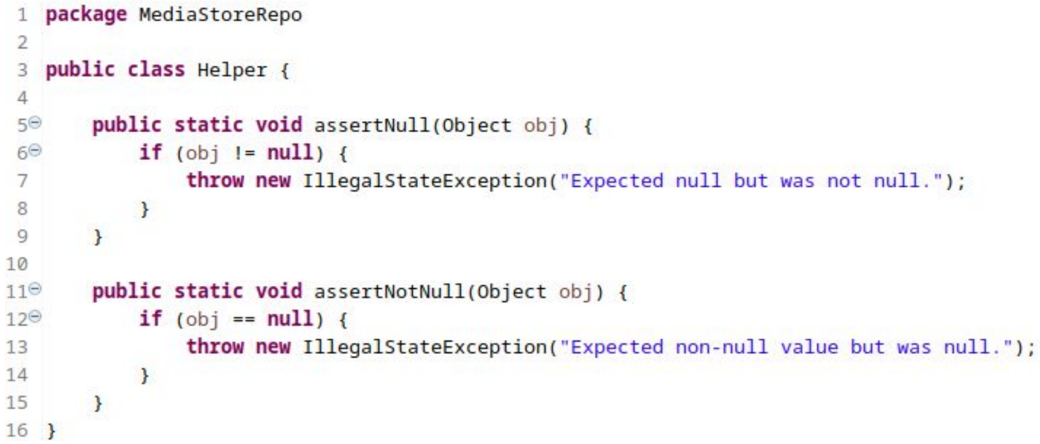
4 Unsere Codegenerierung: Ergebnis (2)



- repo-src-gen
  - DBCache
    - DBCacheImpl.java
  - DigitalWatermarking
    - DigitalWatermarkingImpl.java
  - MediaManager
    - MediaManagerImpl.java
  - MediaStoreRepo
    - Helper.java
    - IAudioDB.java
    - IHTTP.java
    - IMediaStore.java
    - ISound.java
  - PoolingAudioDB
    - PoolingAudioDBImpl.java
  - WebGUI
    - WebGUIImpl.java

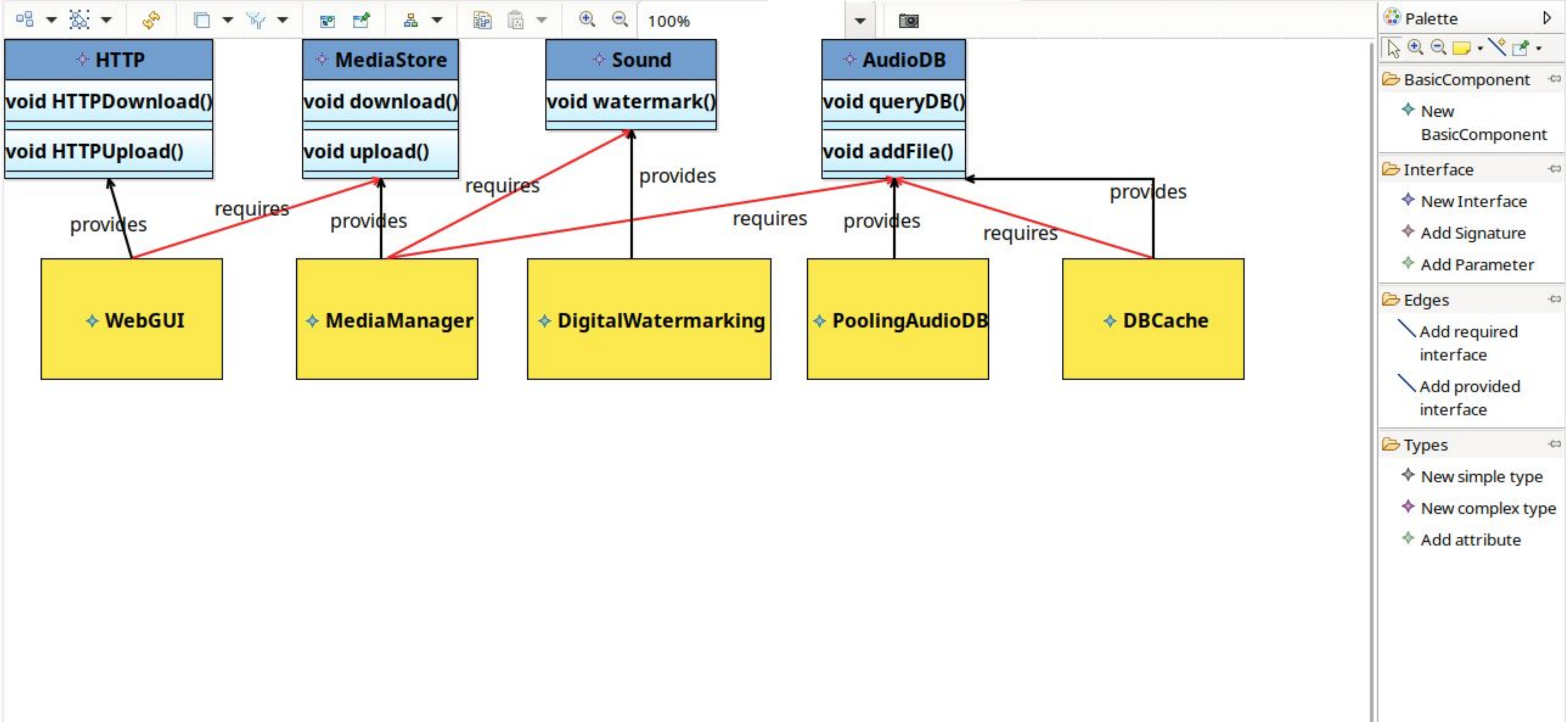


```java
package MediaStoreRepo;

public interface IHTTP {
    void HTTPDownload();
    void HTTPUpload();
}
```

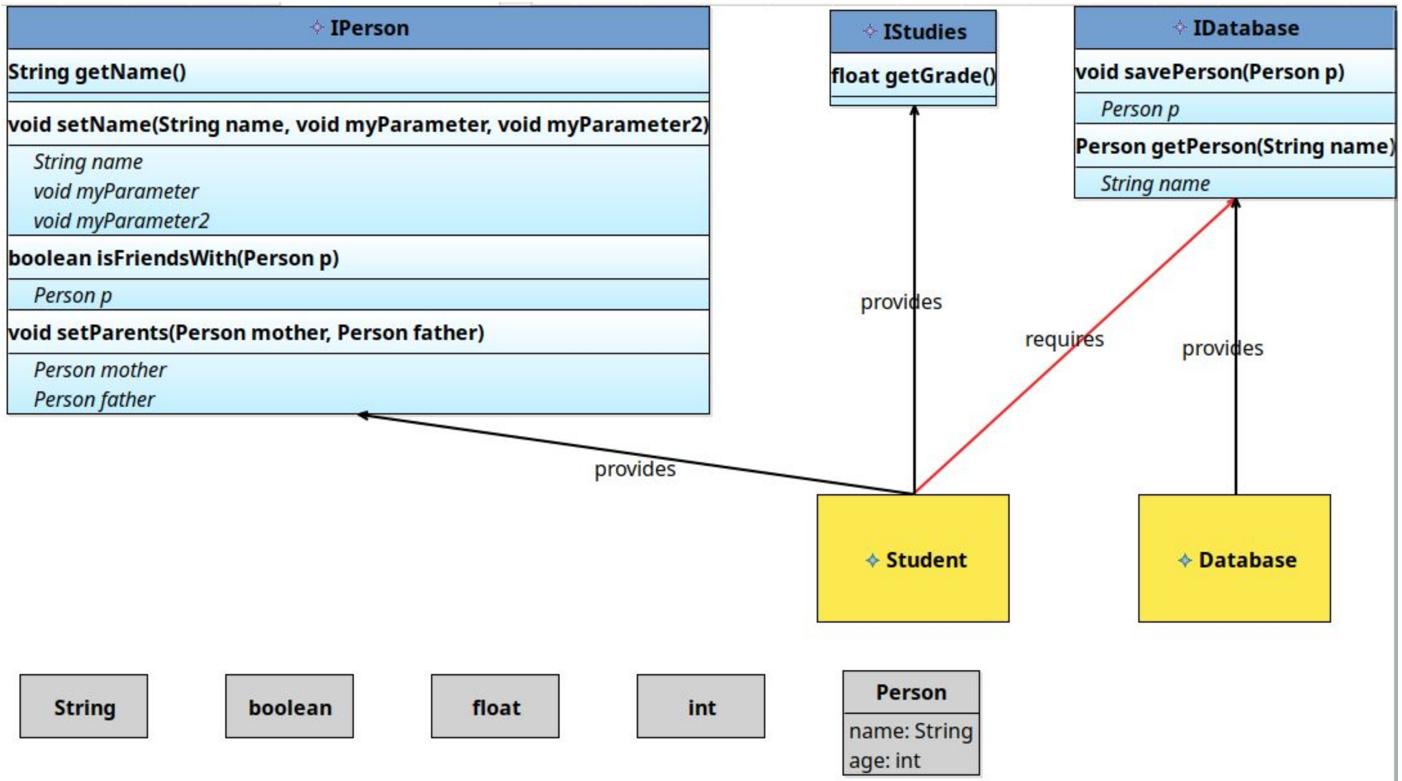


```java
package MediaStoreRepo

public class Helper {

    public static void assertNull(Object obj) {
        if (obj != null) {
            throw new IllegalStateException("Expected null but was not null.");
        }
    }

    public static void assertNotNull(Object obj) {
        if (obj == null) {
            throw new IllegalStateException("Expected non-null value but was null.");
        }
    }
}
```

5 Unser Sirius Editor: Instanz vom Übungsblatt



5 Unser Sirius Editor: Eigene Instanz



IPerson

String getName()

void setName(String name, void myParameter, void myParameter2)

String name

void myParameter

void myParameter2

boolean isFriendsWith(Person p)

Person p

void setParents(Person mother, Person father)

Person mother

Person father

IStudies

float getGrade()

IDatabase

void savePerson(Person p)

Person p

Person getPerson(String name)

String name

provides

requires

provides

provides

Student

Database

String

boolean

float

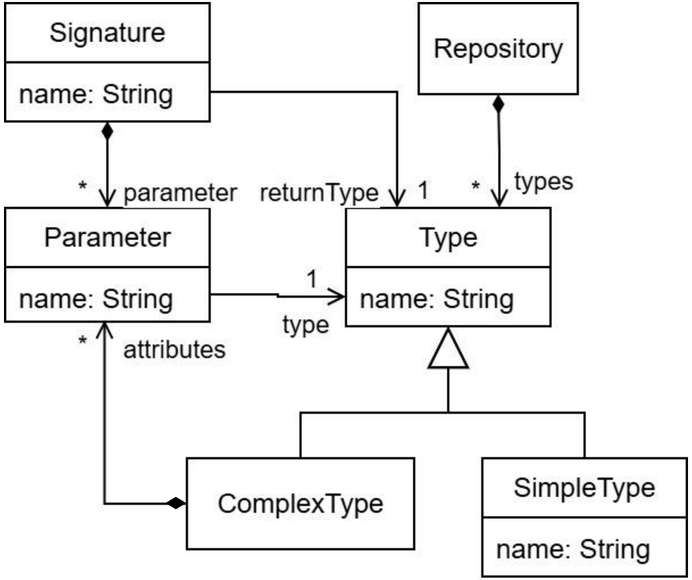
int

Person

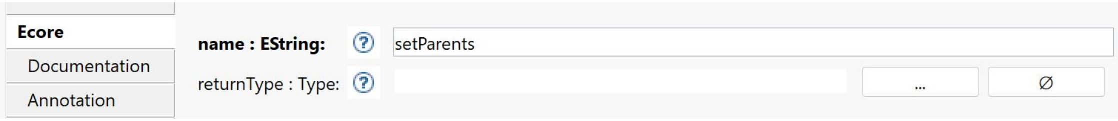
name: String

age: int

5 Unser Sirius Editor: Types bearbeiten



1. Versuch: Typen in Properties setzen ✗



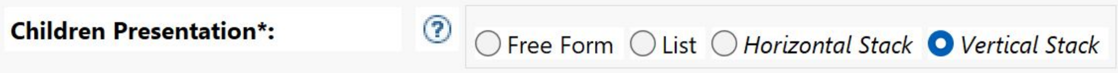
- Ecore
- Documentation
- Annotation

name : EString: setParents

returnType : Type: [...] [Ø]

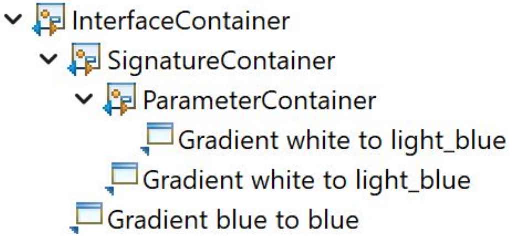
2. Versuch: Typen als Kanten der Parameter ✗

Problem: List-Elemente können nicht über Kanten verbunden werden, Vertical Stack wird nicht angezeigt

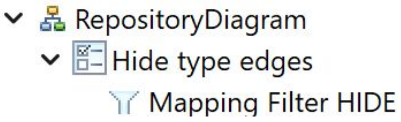


Children Presentation*: ○ Free Form ○ List ○ *Horizontal Stack* ● *Vertical Stack*

3. Versuch: Parameter als Container + Kanten + Filter (✓)



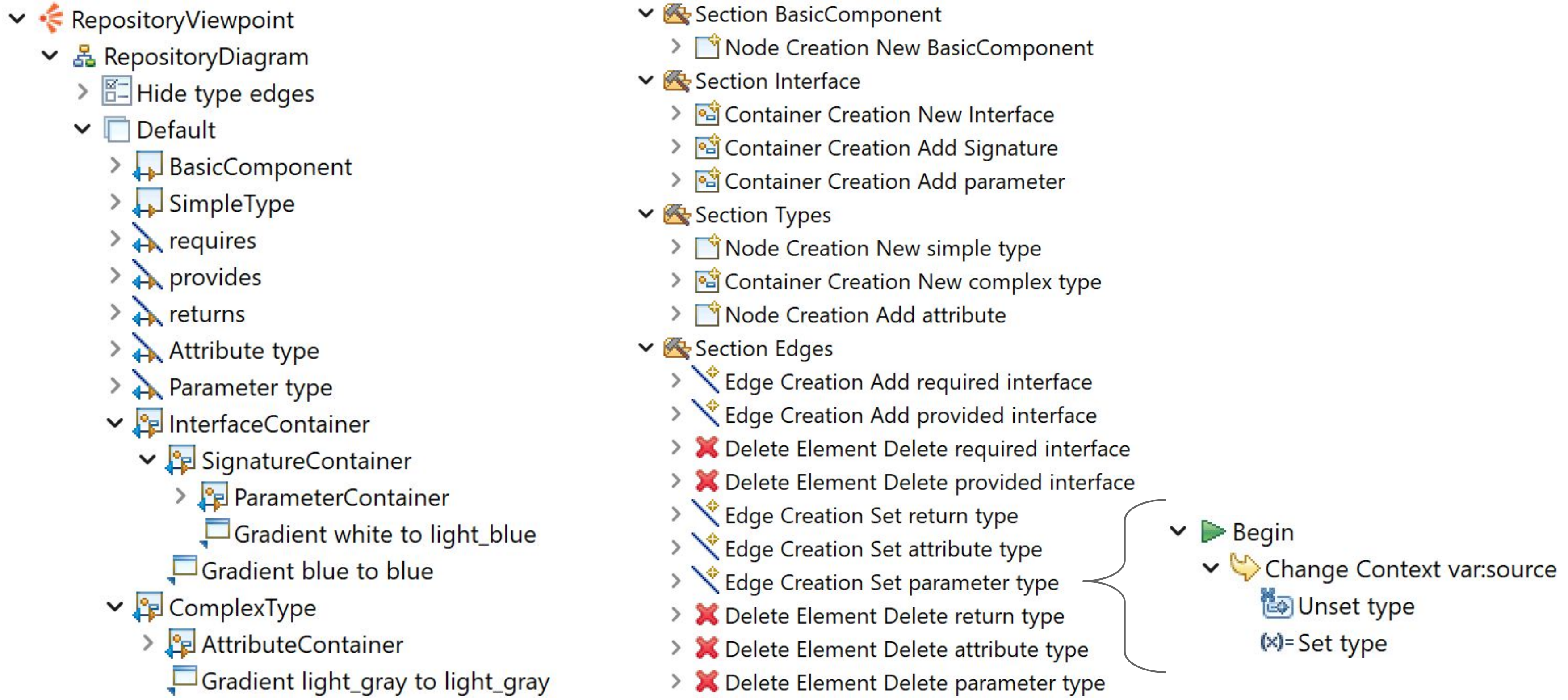
- InterfaceContainer
  - SignatureContainer
    - ParameterContainer
      - Gradient white to light_blue
    - Gradient white to light_blue
  - Gradient blue to blue



- RepositoryDiagram
  - Hide type edges
    - Mapping Filter HIDE

5 Unser Sirius Editor: Types bearbeiten

5 Unser Sirius Editor: Odesign Datei



- RepositoryViewpoint
  - RepositoryDiagram
    - Hide type edges
    - Default
      - BasicComponent
      - SimpleType
      - requires
      - provides
      - returns
      - Attribute type
      - Parameter type
      - InterfaceContainer
        - SignatureContainer
          - ParameterContainer
          - Gradient white to light_blue
        - Gradient blue to blue
      - ComplexType
        - AttributeContainer
        - Gradient light_gray to light_gray

- Section BasicComponent
  - Node Creation New BasicComponent
- Section Interface
  - Container Creation New Interface
  - Container Creation Add Signature
  - Container Creation Add parameter
- Section Types
  - Node Creation New simple type
  - Container Creation New complex type
  - Node Creation Add attribute
- Section Edges
  - Edge Creation Add required interface
  - Edge Creation Add provided interface
  - Delete Element Delete required interface
  - Delete Element Delete provided interface
  - Edge Creation Set return type
  - Edge Creation Set attribute type
  - Edge Creation Set parameter type
  - Delete Element Delete return type
  - Delete Element Delete attribute type
  - Delete Element Delete parameter type

- Begin
  - Change Context var:source
    - Unset type
    - Set type

Herausforderungen

1. **Metamodell**
   - Mehrdeutige Anforderungen
2. **Xtext-Grammatik, DSL**
   - Editor generieren zum Type-Highlighting
3. **QVTo-Transformation**
   - Verstehen des PCM-Metamodells (Viele Vererbungen, zu groß für Visualisierung, keine automatischen Vorschläge in QVTo)
   - Nutzlose SEFFs
   - PCM Random Variables
4. **Xtend Code-Generierung**
   - Dispatch
5. **Sirius-Editor**
   - Diagramm erstellen
   - Types
   - Composite Components sind nicht Teil des Repositories, Interfaces können nicht voneinander erben

Insgesamt: **Eclipse**

- Unhilfreiche Fehlermeldungen
- Funktioniert durchschnittlich auf 40% der Geräte
- Wenig Dokumentation

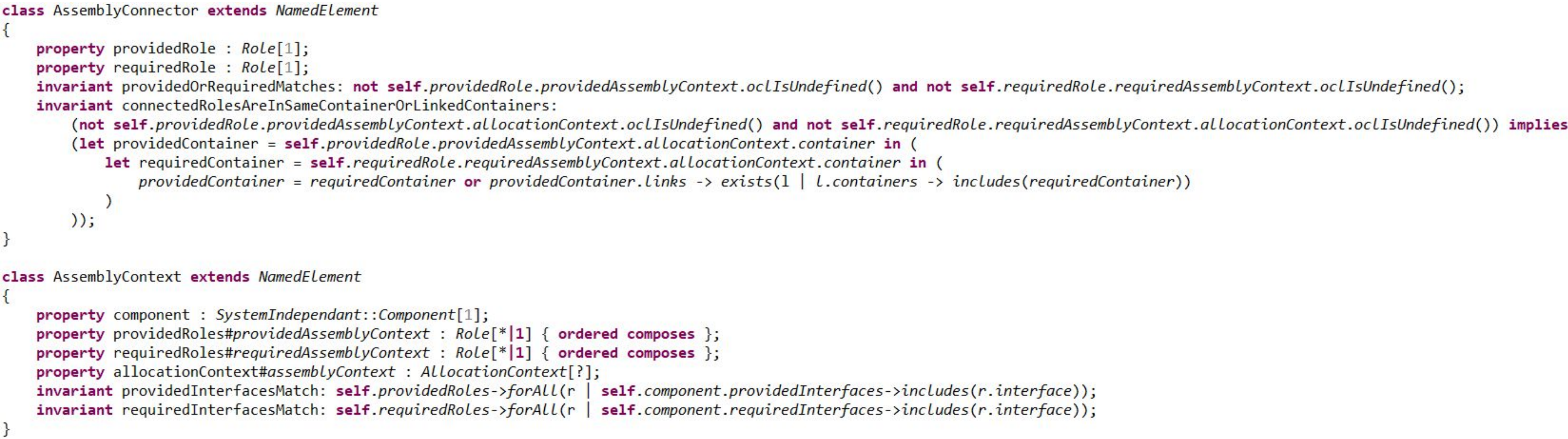
Warum hat das Ecore-Modell keinen Humor?
Weil die Constraints zu streng sind.

(AI generated MDSD humor)

Vielen Dank für eure Aufmerksamkeit!

Anhang

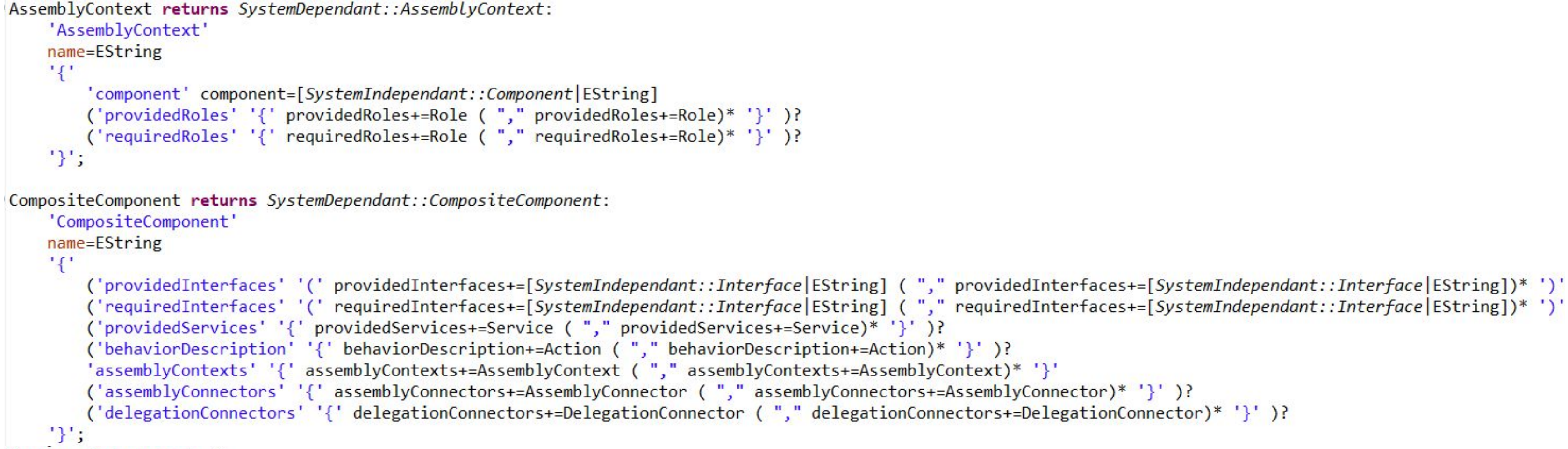
1 Unser Metamodell: OCL Constraints (2)



```
class AssemblyConnector extends NamedElement
{
    property providedRole : Role[1];
    property requiredRole : Role[1];
    invariant providedOrRequiredMatches: not self.providedRole.providedAssemblyContext.oclIsUndefined() and not self.requiredRole.requiredAssemblyContext.oclIsUndefined();
    invariant connectedRolesAreInSameContainerOrLinkedContainers:
        (not self.providedRole.providedAssemblyContext.allocationContext.oclIsUndefined() and not self.requiredRole.requiredAssemblyContext.allocationContext.oclIsUndefined()) implies
        (let providedContainer = self.providedRole.providedAssemblyContext.allocationContext.container in (
            let requiredContainer = self.requiredRole.requiredAssemblyContext.allocationContext.container in (
                providedContainer = requiredContainer or providedContainer.links -> exists(l | l.containers -> includes(requiredContainer))
            )
        ));
}

class AssemblyContext extends NamedElement
{
    property component : SystemIndependant::Component[1];
    property providedRoles#providedAssemblyContext : Role[*|1] { ordered composes };
    property requiredRoles#requiredAssemblyContext : Role[*|1] { ordered composes };
    property allocationContext#assemblyContext : AllocationContext[?];
    invariant providedInterfacesMatch: self.providedRoles->forAll(r | self.component.providedInterfaces->includes(r.interface));
    invariant requiredInterfacesMatch: self.requiredRoles->forAll(r | self.component.requiredInterfaces->includes(r.interface));
}
```

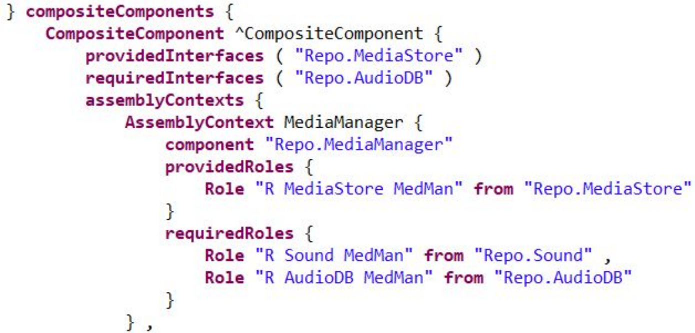
2 Unsere Xtext Grammatik: Assembly Context & Composite Component

Code



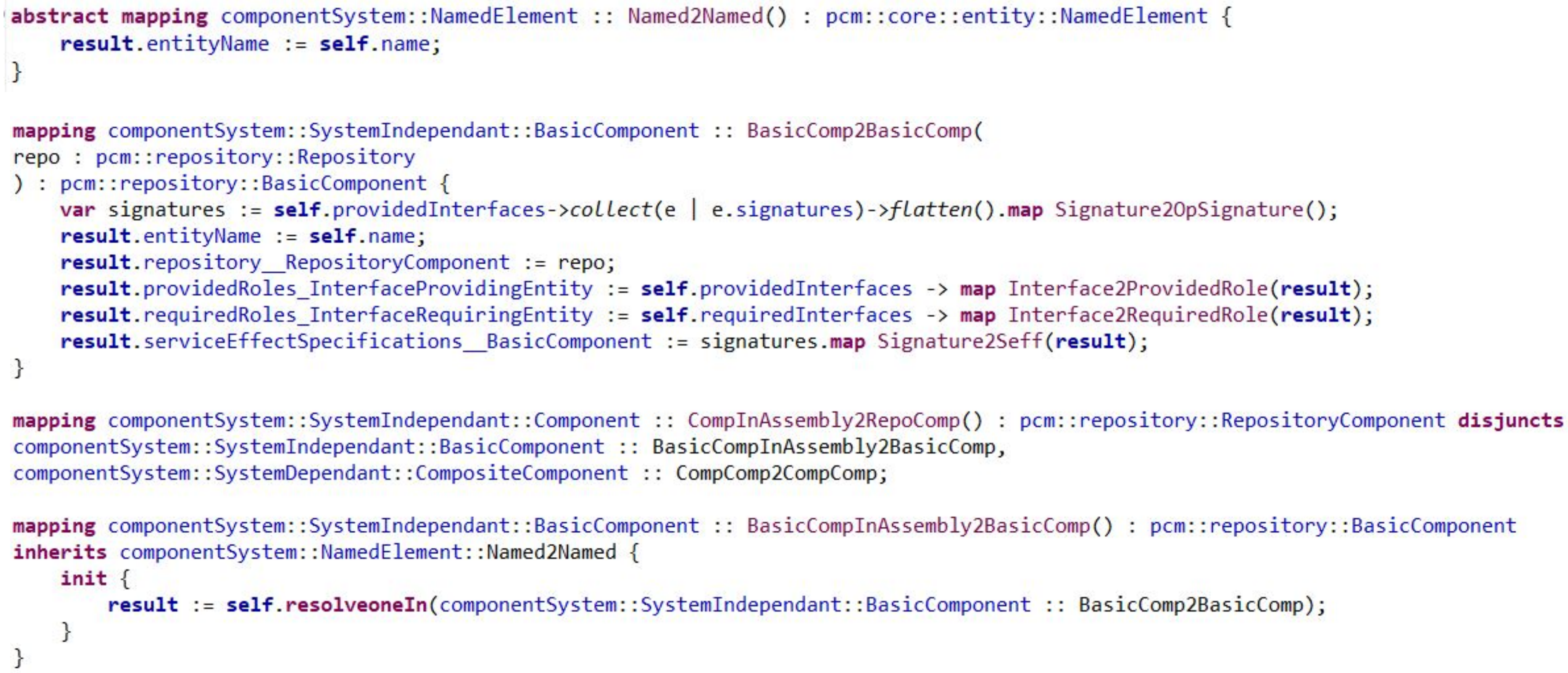
```
AssemblyContext returns SystemDependant::AssemblyContext:
    'AssemblyContext'
    name=EString
    '{'
        'component' component=[SystemIndependant::Component|EString]
        ('providedRoles' '{' providedRoles+=Role ( "," providedRoles+=Role)* '}' )?
        ('requiredRoles' '{' requiredRoles+=Role ( "," requiredRoles+=Role)* '}' )?
    '}';

CompositeComponent returns SystemDependant::CompositeComponent:
    'CompositeComponent'
    name=EString
    '{'
        ('providedInterfaces' '(' providedInterfaces+=[SystemIndependant::Interface|EString] ( "," providedInterfaces+=[SystemIndependant::Interface|EString])* ')'
        ('requiredInterfaces' '(' requiredInterfaces+=[SystemIndependant::Interface|EString] ( "," requiredInterfaces+=[SystemIndependant::Interface|EString])* ')'
        ('providedServices' '{' providedServices+=Service ( "," providedServices+=Service)* '}' )?
        ('behaviorDescription' '{' behaviorDescription+=Action ( "," behaviorDescription+=Action)* '}' )?
        'assemblyContexts' '{' assemblyContexts+=AssemblyContext ( "," assemblyContexts+=AssemblyContext)* '}'
        ('assemblyConnectors' '{' assemblyConnectors+=AssemblyConnector ( "," assemblyConnectors+=AssemblyConnector)* '}' )?
        ('delegationConnectors' '{' delegationConnectors+=DelegationConnector ( "," delegationConnectors+=DelegationConnector)* '}' )?
    '}';
```

Instance



```
} compositeComponents {
    CompositeComponent ^CompositeComponent {
        providedInterfaces ( "Repo.MediaStore" )
        requiredInterfaces ( "Repo.AudioDB" )
        assemblyContexts {
            AssemblyContext MediaManager {
                component "Repo.MediaManager"
                providedRoles {
                    Role "R MediaStore MedMan" from "Repo.MediaStore"
                }
                requiredRoles {
                    Role "R Sound MedMan" from "Repo.Sound" ,
                    Role "R AudioDB MedMan" from "Repo.AudioDB"
                }
            } ,
```

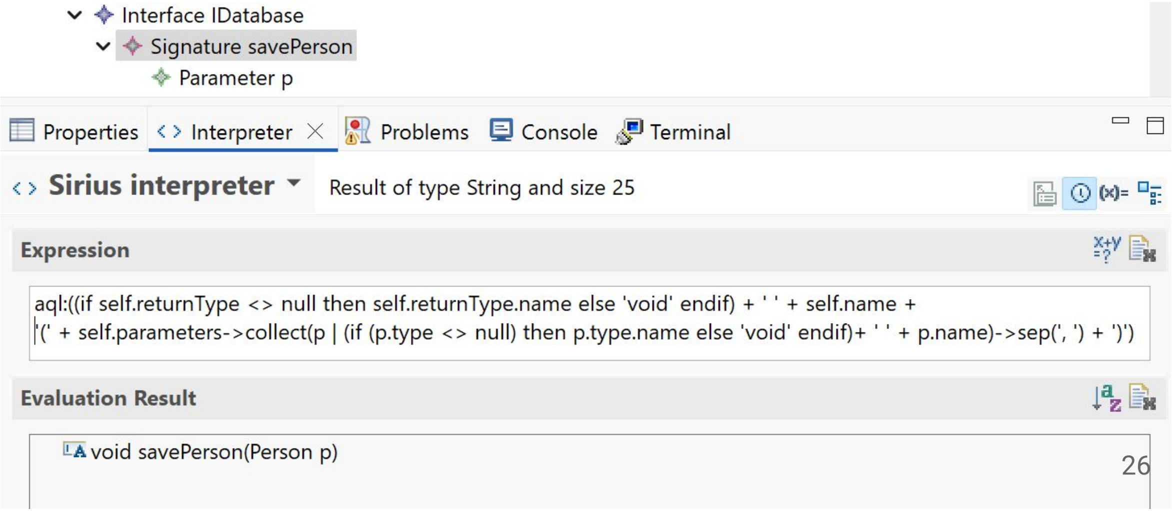
3 Unsere QVTo-Transformation: Verschiedene Arten von Mappings



```
abstract mapping componentSystem::NamedElement :: Named2Named() : pcm::core::entity::NamedElement {
    result.entityName := self.name;
}

mapping componentSystem::SystemIndependant::BasicComponent :: BasicComp2BasicComp(
repo : pcm::repository::Repository
) : pcm::repository::BasicComponent {
    var signatures := self.providedInterfaces->collect(e | e.signatures)->flatten().map Signature2OpSignature();
    result.entityName := self.name;
    result.repository__RepositoryComponent := repo;
    result.providedRoles_InterfaceProvidingEntity := self.providedInterfaces -> map Interface2ProvidedRole(result);
    result.requiredRoles_InterfaceRequiringEntity := self.requiredInterfaces -> map Interface2RequiredRole(result);
    result.serviceEffectSpecifications__BasicComponent := signatures.map Signature2Seff(result);
}

mapping componentSystem::SystemIndependant::Component :: CompInAssembly2RepoComp() : pcm::repository::RepositoryComponent disjuncts
componentSystem::SystemIndependant::BasicComponent :: BasicCompInAssembly2BasicComp,
componentSystem::SystemDependant::CompositeComponent :: CompComp2CompComp;

mapping componentSystem::SystemIndependant::BasicComponent :: BasicCompInAssembly2BasicComp() : pcm::repository::BasicComponent
inherits componentSystem::NamedElement::Named2Named {
    init {
        result := self.resolveoneIn(componentSystem::SystemIndependant::BasicComponent :: BasicComp2BasicComp);
    }
}
```

5 Unser Sirius Editor: Types anzeigen

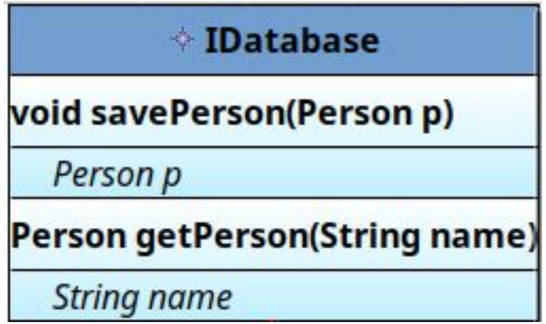
```
aql:((if self.returnType <> null then self.returnType.name else 'void' endif)
+ ' ' + self.name + '(' + self.parameters -> collect(p |
     (if (p.type <> null) then p.type.name else 'void' endif)
     + ' ' + p.name)
->sep(', ') + ')')
```



Result of type String and size 25

```
aql:((if self.returnType <> null then self.returnType.name else 'void' endif) + ' ' + self.name +
'(' + self.parameters->collect(p | (if (p.type <> null) then p.type.name else 'void' endif)+ ' ' + p.name)->sep(', ') + ')')
```

void savePerson(Person p)


IDatabase
void savePerson(Person p)
Person p
Person getPerson(String name)
String name