

Model-Driven Software Migration into SOA

SE 2010, Paderborn

Andreas Fuhr,

Tassilo Horn

University of Koblenz-Landau

{afuhr,horn}@uni-koblenz.de

Andreas Winter

Carl von Ossietzky University

Oldenburg

winter@se.uni-oldenburg.de

Motivation



- **SOAMIG** (SOA Migration) combines
 - Reverse-engineering and migration techniques
 - Model-driven software development techniques
 - SOA development methods
- Objective
 - Definition and application of model driven methodologies and techniques to software migration into SOA
 - Migration by transforming legacy functionality as opposed to wrapping

GEFÖRDERT VOM

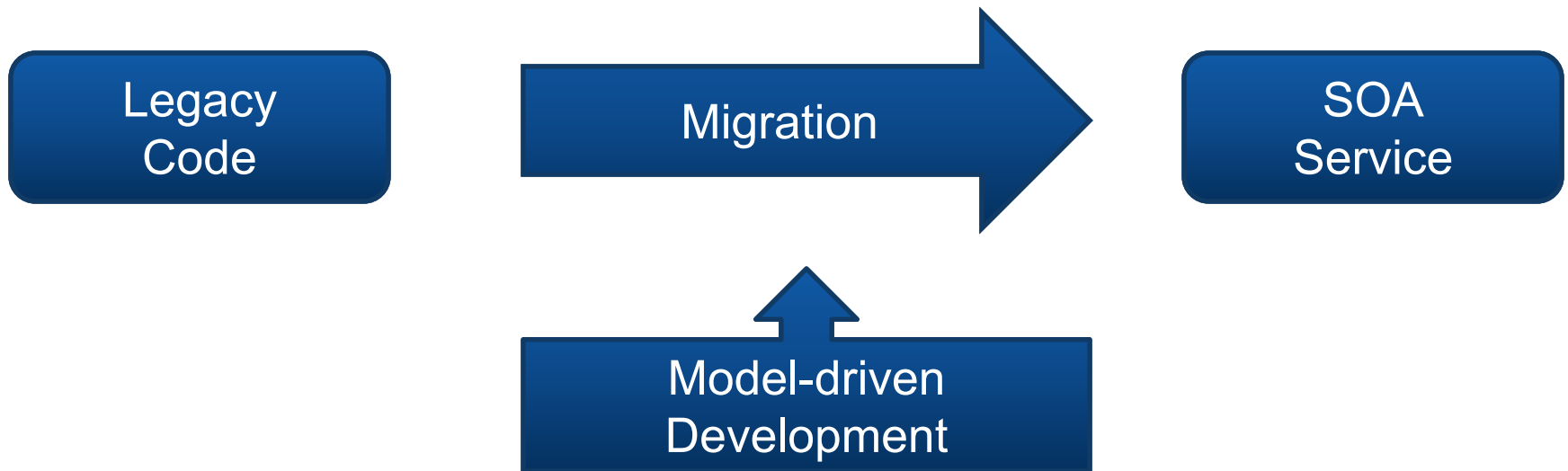


Bundesministerium
für Bildung
und Forschung

Project Partners



Goal of this presentation

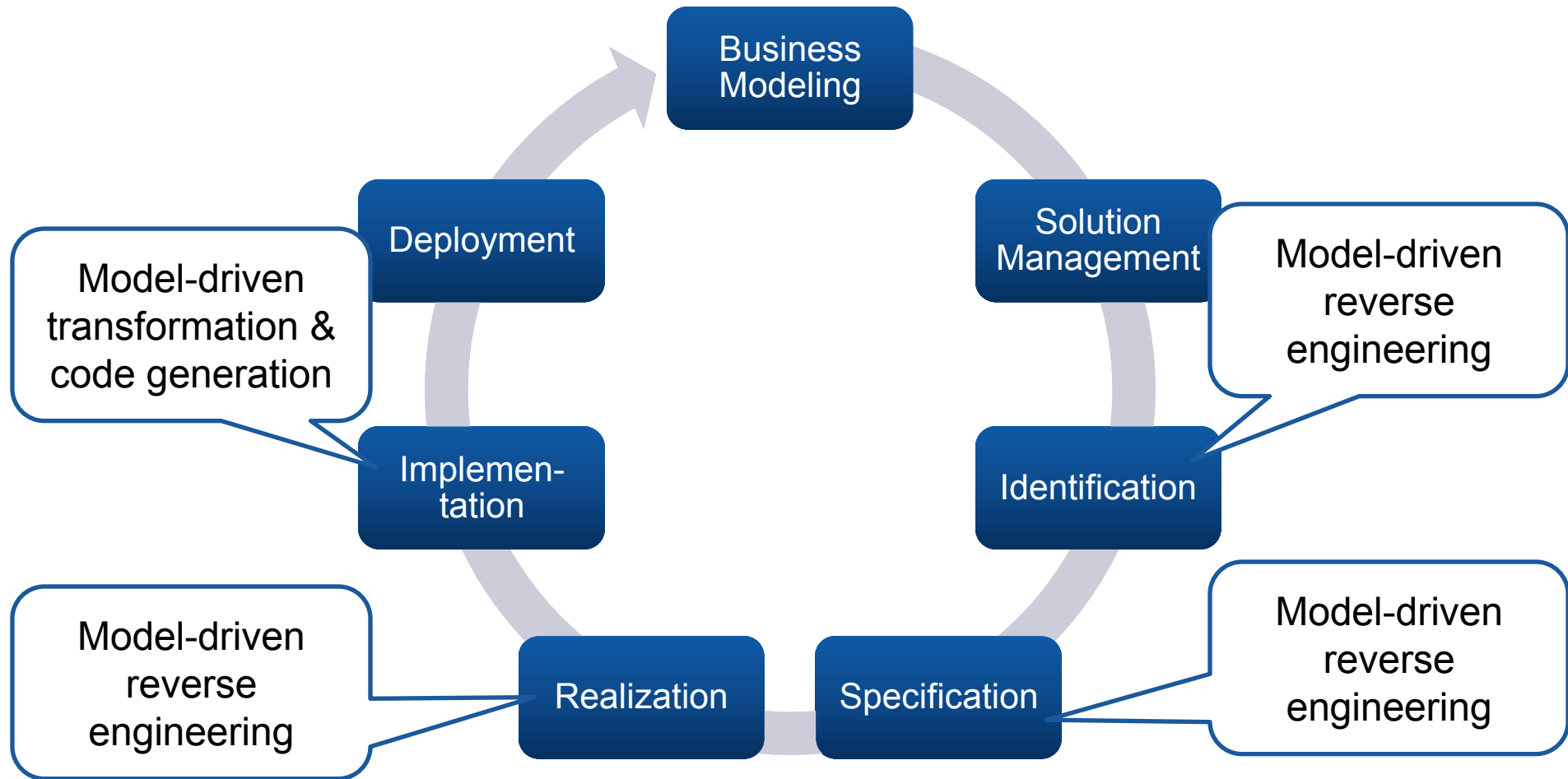


Service-oriented Modeling and Architecture (SOMA)

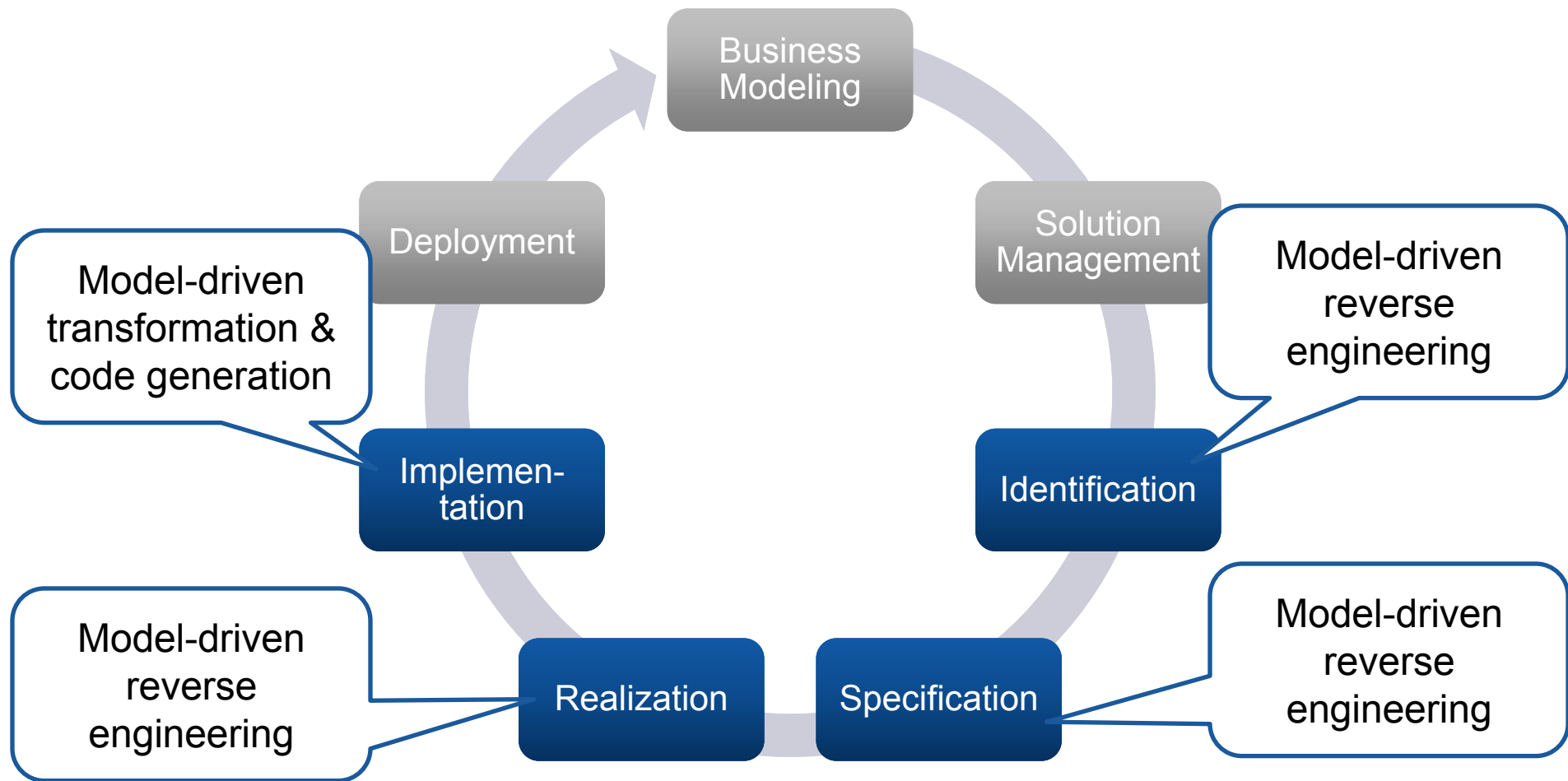
- SOA development method
- Developed by IBM (official version: 2.4)
- Covers **full SOA lifecycle** (7 phases)
- **Extensible** design
 - Include additional methods
 - Adapt to project needs



SOMA Phases



SOMA Phases: Focus





EXTENDING SOMA

Extending SOMA

- SOMA extended by model-driven
 - reverse engineering
 - transformation
- TGraph-Approach
 - TGraphs
 - Java 6 schema
 - JGraLab

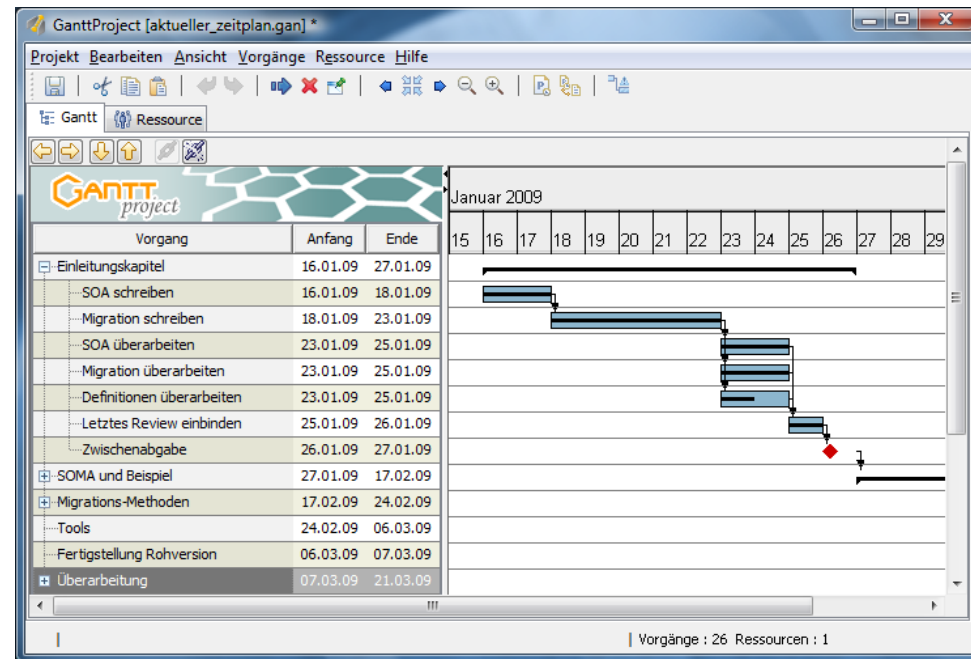


Requires MDA
framework

Example: GanttProject



- Open source project management tool
- Goal:
 - Identify and transform services to manage resources



Business Modeling and Solution Management

- Business Modeling
 - Capture **business processes**
 - (not covered in this presentation)
- Solution Management
 - **Adapt** method to **project needs**
 - Develop **metamodels**
 - Java 6 TGraph schema

Identification



JGraLab using Java 6 schema
 services from legal
 Generate visualization

Legacy Java code

TGraph Java 6

TGraph UML

Visualization

```
public void add(ProjectResource resource) {
    if (resource.getId() == -1) {
        resource.setId(nextFreeId);
    }
    if (resource.getId() >= nextFreeId) {
        nextFreeId = resource.getId() + 1;
    }
    resources.add(resource);
}
```

GReTL transformation:
 All classes/interfaces
 named *resource*

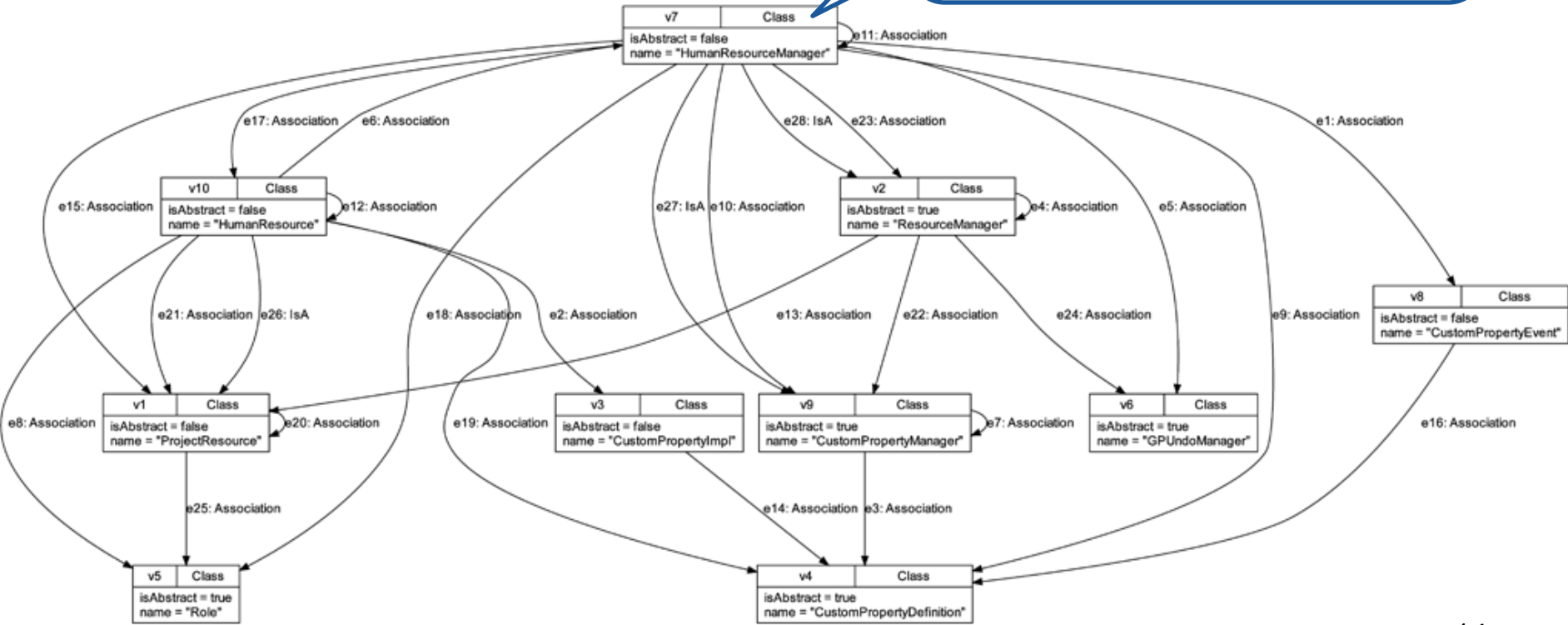


GReTL: Transformation (partly)

```
VertexClass umlClass = createVertexClass("uml.Class",
    "  from t : V{Type}"
+ "  with t.name = \".*resource.*\" or "
+ "    count(t -->{IsTypeDefinitionOf} "
+ "      -->
        {AttributedEdge, ^IsBreakTargetOf,
        ^IsContinueTargetOf, ^IsTypeDefinitionOf}* "
+ "      -->{IsMemberOf} -->{IsBlockOf} "
+ "      & {hasType(thisVertex, \"Type\") ? "
+ "        thisVertex.name =~ \".*resource.*\" : false :
false }) > 0 "
+ "reportSet t end ");
```

GReTL Result: Visu

Class
HumanResourceManager
(HRM) may be able to
implement service



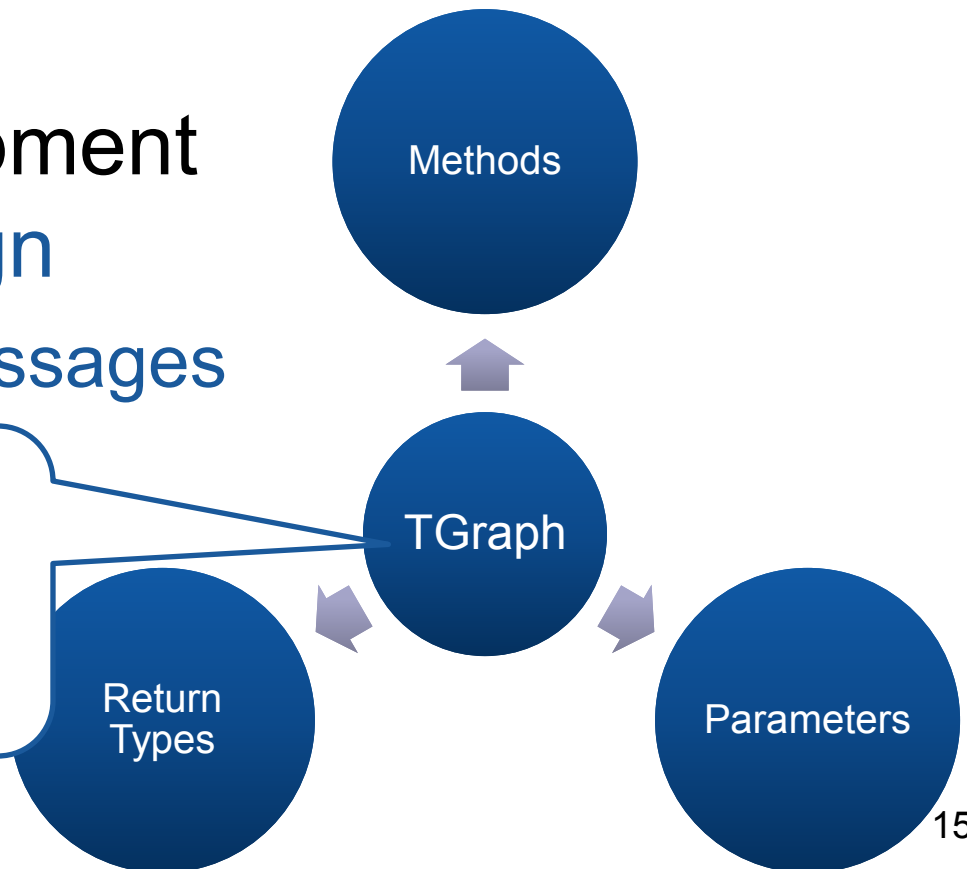
Specification



- Support development of **Service Design**
 - e.g. **Service Messages**

GReQL query:

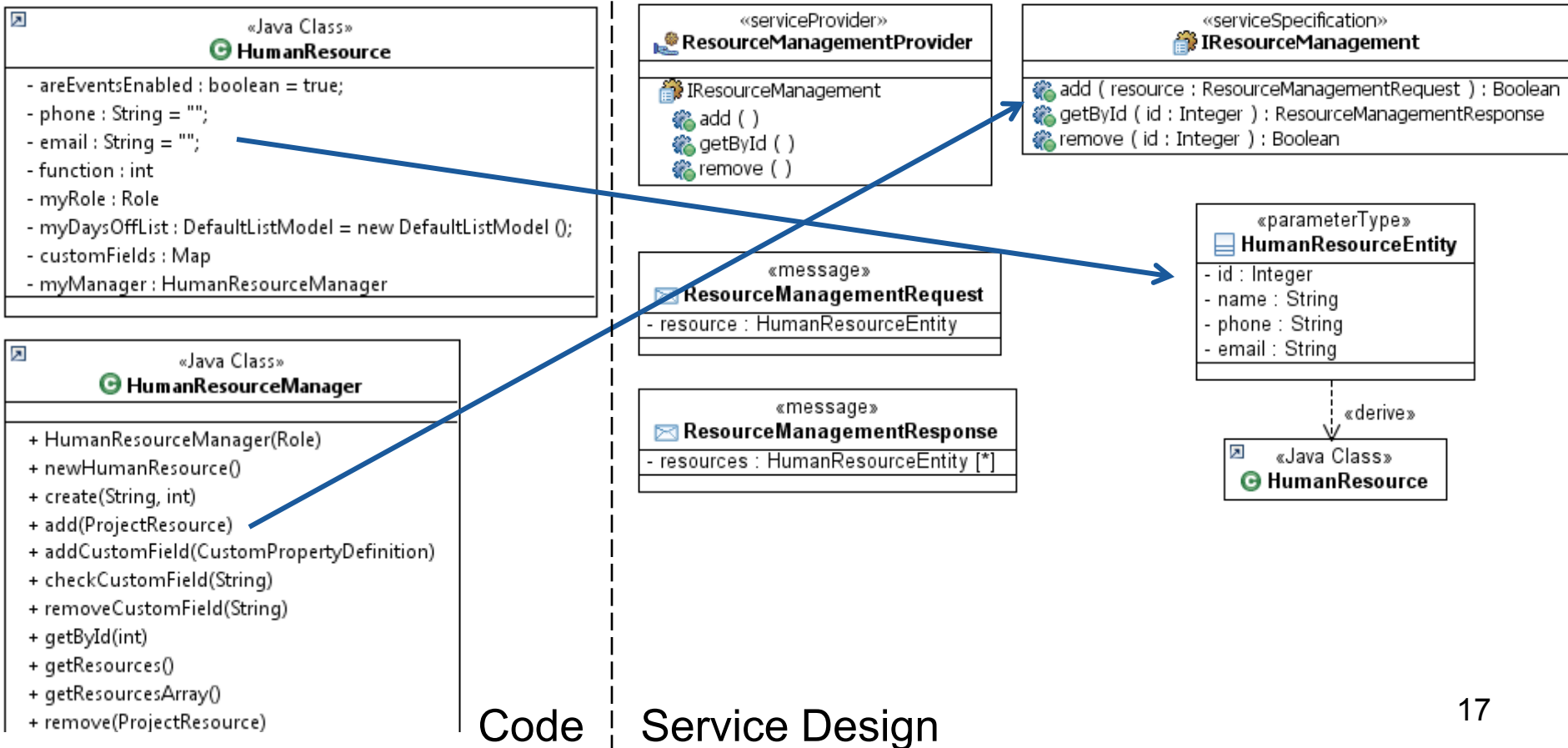
```
HRM -->{hasMethod} method  
-->{hasParameterType} |  
-->{hasReturnType} type
```



GReQL Query: Retrieve Method Parameters and Returntypes

```
let classname := "HumanResourceManager" in tup(  
  from hrmClass : V{ClassDefinition}, usedType : V{Type, BuiltInType}  
  with hrmClass.name = classname and hrmClass  
    <-- {IsClassBlockOf} <-- {IsMemberOf} <-- {IsParameterOfMethod} <--  
      {IsTypeOfParameter} [<-- {IsTypeDefOf}] usedType  
  reportSet (hasType(usedType, "BuiltInType")) ?  
    usedType.type : theElement(usedType<--&{Identifier}).name end,  
  
  from hrmClass : V{ClassDefinition}, usedType : V{Type, BuiltInType}  
  with hrmClass.name = classname and hrmClass  
    <-- {IsClassBlockOf} <-- {IsMemberOf} <-- {IsReturnTypeOf} [<--  
      {IsTypeDefOf}] usedType  
  reportSet (hasType(usedType, "BuiltInType")) ?  
    usedType.type : theElement(usedType<--&{Identifier}).name end)
```

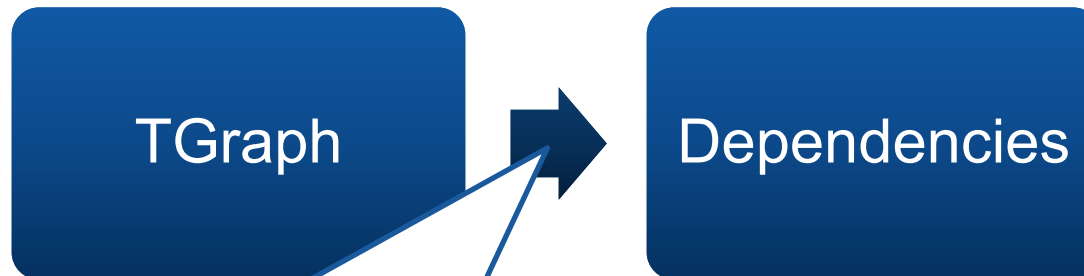
Specification: Results



Realization



- Explore **how to implement** the service



GReQL query:

```
HRM -->{Uses} class
```

GReQL Query: Dependencies

```

from hrmClass : V{ClassDefinition}, hrmMethod : V{MethodDefinition}, usedType : V{Type}
with hrmClass.name = "HumanResourceManager"
  and hrmClass <-- {IsClassBlockOf} <-- {IsMemberOf} hrmMethod
  and (hrmMethod (

```

```

    (<-- {IsBodyOfMethod} <-- {IsStatementOfBody}
     (<-- {AttributedEdge, ^IsBreakTargetOf, ^IsContinueTargetOf, ^IsTypeDefinitionOf}) *
     <-- {IsDeclarationOfInvokedMethod}
     --> {IsMemberOf} --> {IsClassBlockOf}

```

method invocations

```

    ) | (
     <-- {IsParameterOfMethod} <-- {IsTypeOf} +
     <-- {IsTypeDefinitionOf}

```

method parameters

```

    ) | (
     <-- {IsBodyOfMethod} <-- {IsStatementOfBody}
     (<-- {AttributedEdge, ^IsBreakTargetOf, ^IsContinueTargetOf, ^IsTypeDefinitionOf}) *
     <-- {IsTypeOfVariable} <-- {IsTypeDefinitionOf}

```

local variables

```

    ) | (
     <-- {IsReturnTypeOf} <-- {IsTypeDefinitionOf}

```

return types

```

) usedType or hrmClass ( (
  <-- {IsClassBlockOf} <-- {IsMemberOf}
  <-- {IsFieldCreationOf} <-- {IsTypeOfVariable}
  <-- {IsTypeDefinitionOf}

```

fields

```

) | (
  (<-- {IsSuperClassOfClass} | <-- {IsInterfaceOfClass})
  <-- {IsTypeDefinitionOf}

```

superclasses

```

) | (
  (<-- {IsClassBlockOf} <-- {IsMemberOf}) +

```

local classes

```

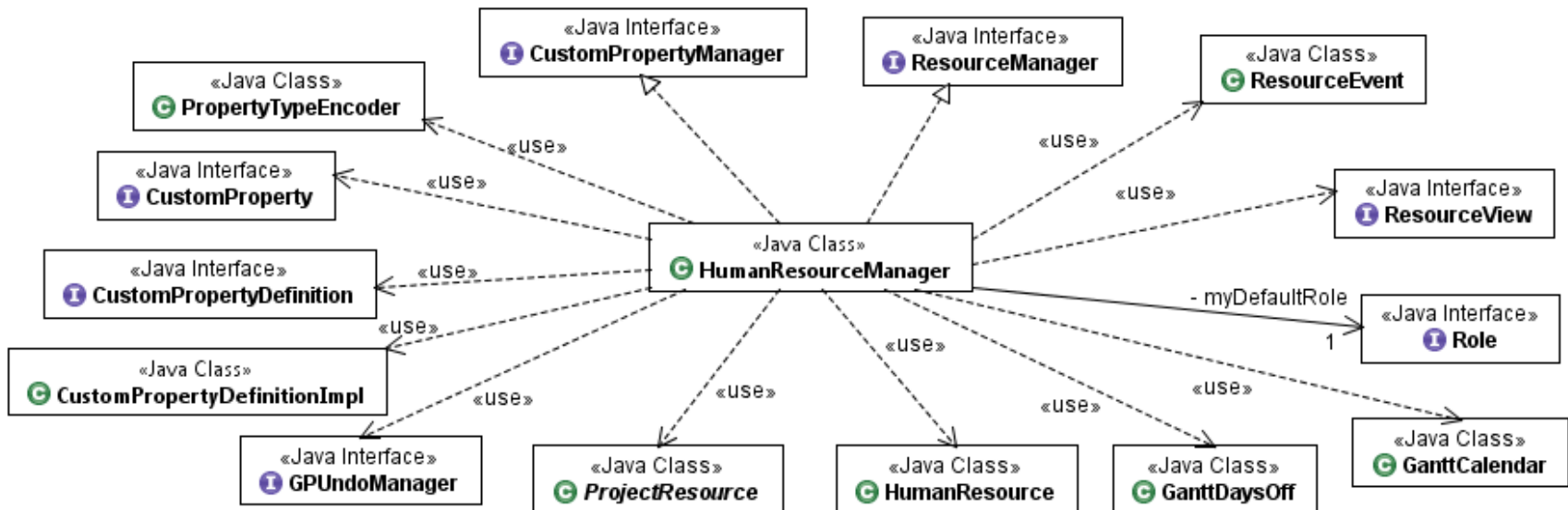
) usedType
)
reportSet theElement(usedType <-- & {Identifier}).name, typeName(usedType)
end

```

Realization: Result of Dependencies Query

```
{CustomProperty, OutputStream,  
CustomPropertyListener,  
CustomPropertyDefinition,  
HumanResourceManager, CustomPropertyManager,  
ArrayList, EventObject, ResourceView,  
Iterator, ResourceEvent, ResourceManager,  
HumanResource, GPUndoManager, String,  
ProjectResource, GanttDaysOff, GanttCalendar,  
Role, CustomPropertyEvent, HashMap}
```

Realization: Visualization of Dependencies



Implementation

Identifi-
cation

Specifi-
cation

Realiz-
ation

Implem-
entation

- Transform legacy code into service implementation

TGraph
Java 6

Service
implementation

Generate Java code
using **GReQL** query
result

```
public void add(ProjectResource resource) {  
    if (resource.getId() == -1) {  
        resource.setId(nextFreeId);  
    }  
    if (resource.getId() >= nextFreeId) {  
        nextFreeId = resource.getId() + 1;  
    }  
    resources.add(resource);  
}
```

Service Implementation

```

@WebService(name = "IResource", targetNamespace =
    "http://ProjectManagement/ResourceProvider/")
@SOAPBinding(parameterStyle = SOAPBinding.ParameterStyle.BARE)
@XmlSeeAlso({
    messages.ObjectFactory.class,
    projectmanagement.resourceprovider.ObjectFactory.class,
    parametertypes.ObjectFactory.class
})
public interface IResource {

    @WebMethod(action =
        "http://ProjectManagement/ResourceProvider/addResource/")
    @WebResult(name = "IResourceaddResourceResponse1",
        targetNamespace =
            "http://ProjectManagement/ResourceProvider/", partName =
                "Parameter1")
    public boolean addResource(
        @WebParam(name = "IResourceaddResourceRequest1",
            targetNamespace =
                "http://ProjectManagement/ResourceProvider/", partName =
                    "rsc")
            ResourceMsg rsc);

    @WebMethod(action =
        "http://ProjectManagement/ResourceProvider/getResource/")
    @WebResult(name = "IResourcegetResourceResponse1",
        targetNamespace =
            "http://ProjectManagement/ResourceProvider/", partName =
                "Parameter1")
    public ResourceMsg getResource(

```

```

    @WebParam(name = "IResourcegetResourceRequest1", targetNamespace =
        "http://ProjectManagement/ResourceProvider/", partName =
            "rsc")
            ResourceId rsc);

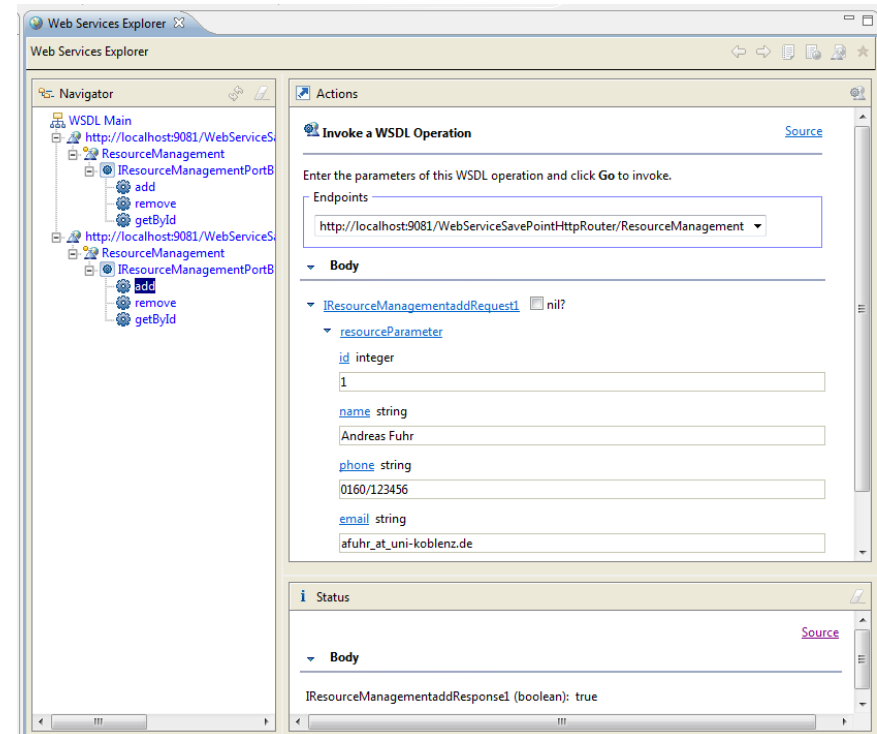
    @WebMethod(action =
        "http://ProjectManagement/ResourceProvider/deleteResource/")
    @WebResult(name = "IResourcedeleteResourceResponse1",
        targetNamespace =
            "http://ProjectManagement/ResourceProvider/", partName =
                "Parameter1")
    public boolean deleteResource(
        @WebParam(name = "IResourcedeleteResourceRequest1",
            targetNamespace =
                "http://ProjectManagement/ResourceProvider/", partName =
                    "rsc")
            ResourceId rsc);

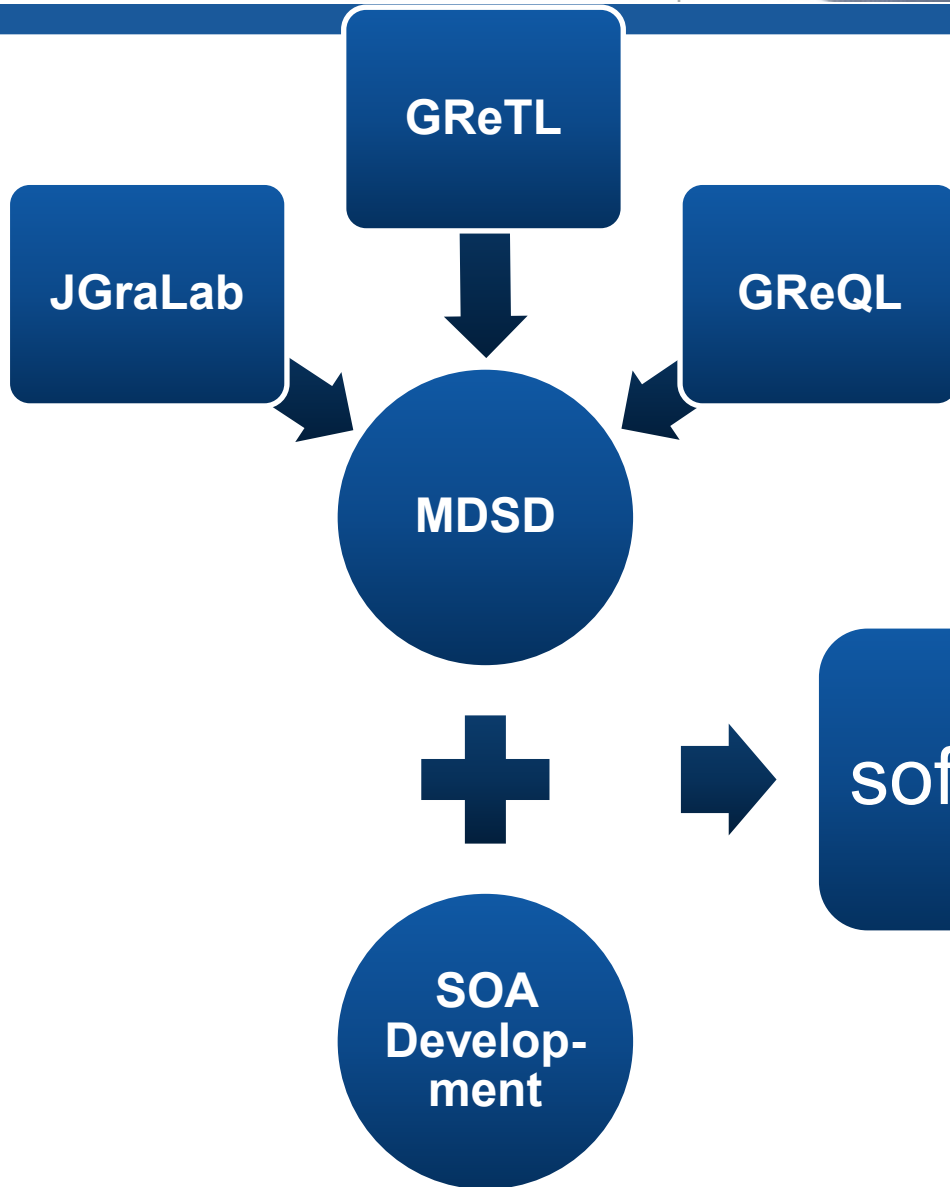
    @WebMethod(action =
        "http://ProjectManagement/ResourceProvider/importResource/")
    @WebResult(name = "IResourceimportResourceResponse1",
        targetNamespace =
            "http://ProjectManagement/ResourceProvider/", partName =
                "Parameter1")
    public boolean importResource(
        @WebParam(name = "IResourceimportResourceRequest1",
            targetNamespace =
                "http://ProjectManagement/ResourceProvider/", partName =
                    "file")
            ResourceFile file);
}

```

Service Deployment

- Deploy Services to customer environment
- Regression Testing





**Thank you for
your attention!**

Questions?