

# **Reusing Existing Object-oriented Code as Web Services in a SOA**

Harry M. Sneed &

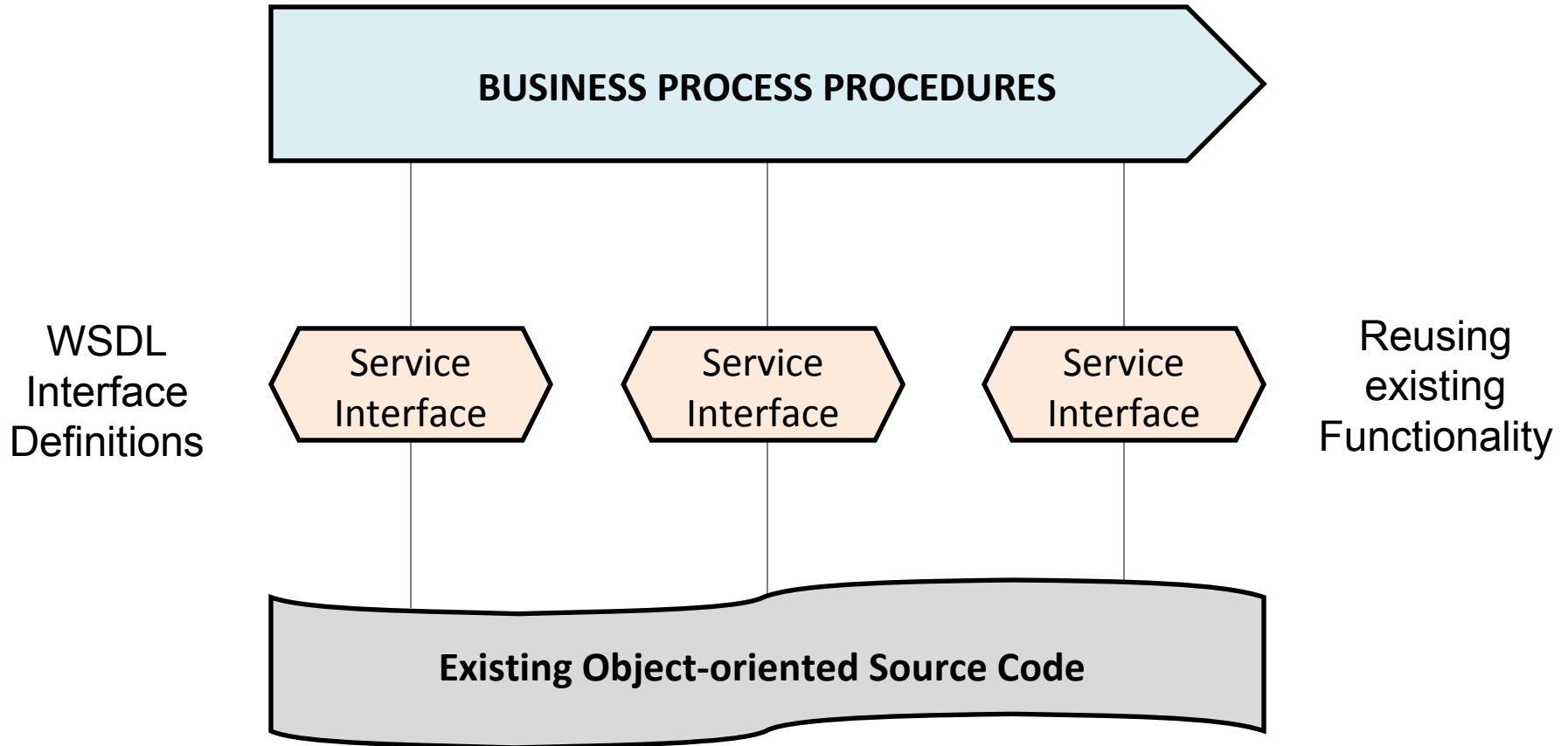
Stephan H. Sneed

for

MESOCA-2013

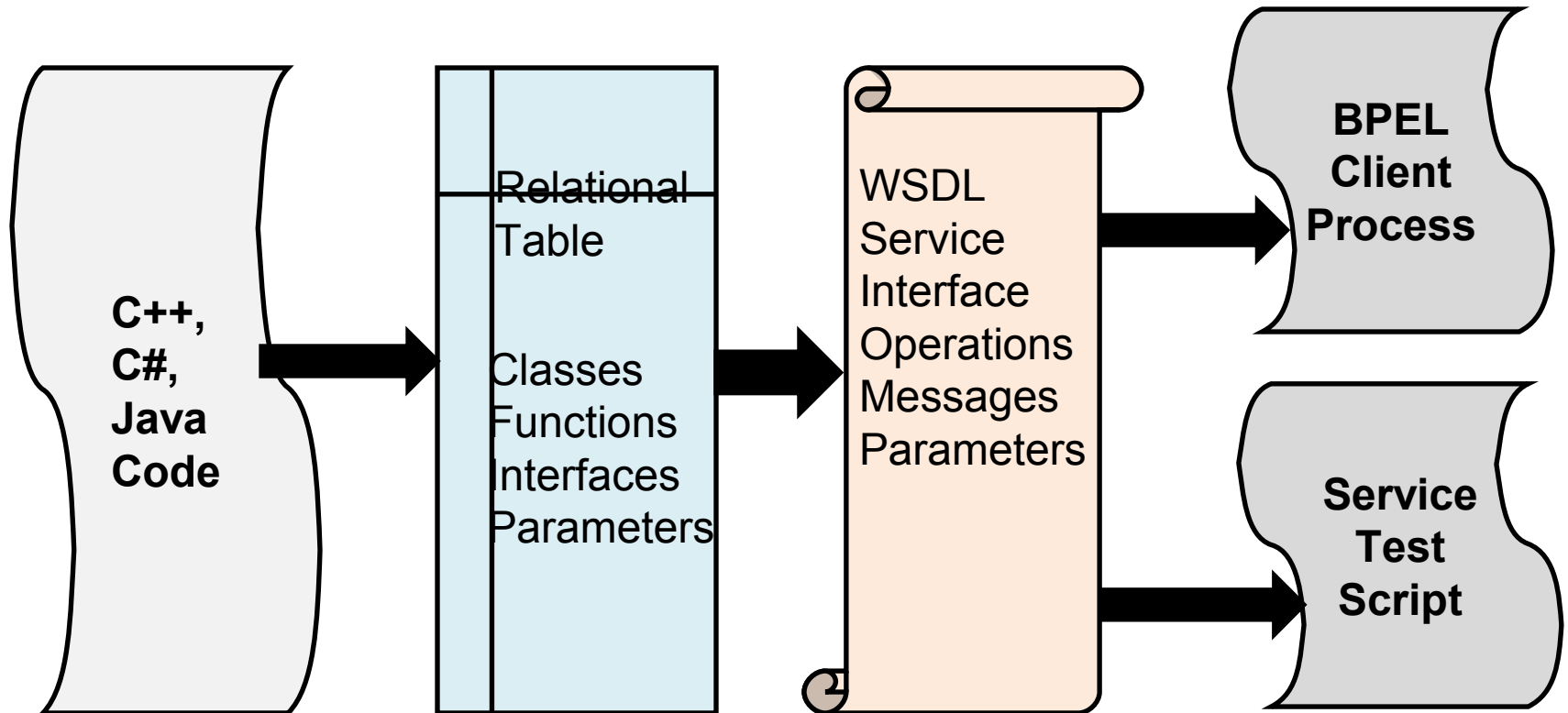
Eindhoven

# Bottom-Up Approach to collecting Web Services

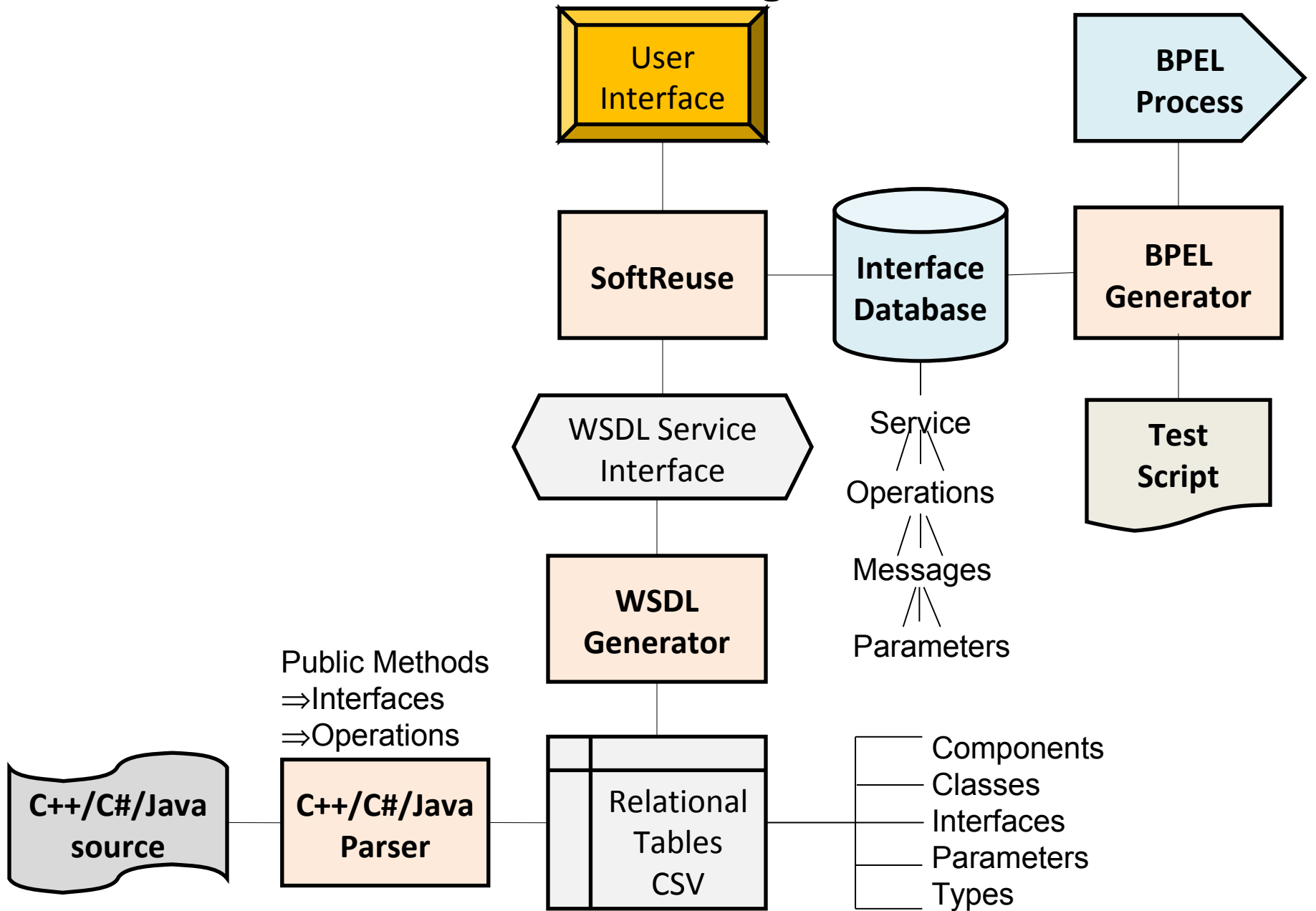


Reverse Engineering from Source Code to Business Process  
Using WSDL as a Link

# From C++, C#, Java to WSDL to BPEL & Test Script



# Web Service Mining Process



# Entity/Relationship Table

Type;Base Entity	;Rela;Type;Target Entity
CLAS ;Bausparer	;OWNS ;FUNC ;setNachname
FUNC ;setNachname	;OWNS ;INTR ;setNachname
INTR ;setNachname	;RECV ; PARM ;c
PARM ;c	;USES ;TYPE ;VarChar
CLAS ;Bausparer	;OWNS ;FUNC ;setVorname
FUNC ;setVorname	;OWNS ;INTR ;setVorname
INTR ;setVorname	;RECV ; PARM ;c
PARM ;c	;USES ;TYPE ;VarChar
CLAS ;Bausparer	;OWNS ;FUNC ;setKzAnrede
FUNC ;setKzAnrede	;OWNS ;INTR ;setKzAnrede
INTR ;setKzAnrede	;RECV ; PARM ;c
PARM ;c	;USES ;TYPE ;VarChar
CLAS ;Bausparer	;OWNS ;FUNC ;setKzTitel
FUNC ;setKzTitel	;OWNS ;INTR ;setKzTitel
INTR ;setKzTitel	;RECV ; PARM ;c
PARM ;c	;USES ;TYPE ;VarChar
CLAS ;Bausparer	;OWNS ;FUNC ;setKzArchiv
FUNC ;setKzArchiv	;OWNS ;INTR ;setKzArchiv
INTR ;setKzArchiv	;RECV ; PARM ;b
PARM ;b	;USES ;TYPE ;boolean
CLAS ;Bausparer	;OWNS ;FUNC ;setGeburtsdatum
FUNC ;setGeburtsdatum	;OWNS ;INTR ;setGeburtsdatum
INTR ;setGeburtsdatum	;RECV ; PARM ;d
PARM ;d	;USES ;TYPE ;Datum
CLAS ;Bausparer	;OWNS ;FUNC ;setAdresse
FUNC ;setAdresse	;OWNS ;INTR ;setAdresse
INTR ;setAdresse	;RECV ; PARM ;o
PARM ;o	;USES ;TYPE ;Adresse

```

public void Bausparer ( VarChar  cRole,
                        Nummer  nVersion,
                        TimeStamp tsReadTime,
                        int      iSeqNumber,
                        Nummer  nGrdNr,
                        Nummer  nBBZS,
                        VarChar  cNachname,
                        VarChar  cVorname,
                        VarChar  cKzAnrede,
                        VarChar  cKzTitel,
                        Datum    dGeburtsdatum,
                        boolean  bKzArchiv,
                        Adresse  oAdresse
                        ){

```

```

public Person Bausparer () {
    return new Person( getBBZS(),
                      getGeburtsdatum(),
                      getGrdNr(),
                      getNachname(),
                      getVorname(),
                      null, // KzSparer - woher?
                      getKzArchiv(),
                      getAdresse().getKzAusland(),
                      getAdresse().getOrt(),
                      getAdresse().getPlz(),
                      getAdresse().getStrasse( )
                      );
}

```

```

CLAS;Bausparer      ;OWNS;FUNC;Bausparer
FUNC;Bausparer      ;OWNS;INTR;Bausparer
INTR;Bausparer      ;RECV;PARAM;cRole
PARAM;cRole         ;USES;TYPE;VarChar
INTR;Bausparer      ;RECV;PARAM;nVersion
PARAM;nVersion      ;USES;TYPE;Nummer
INTR;Bausparer      ;RECV;PARAM;tsReadTime
PARAM;tsReadTime    ;USES;TYPE;TimeStamp
INTR;Bausparer      ;RECV;PARAM;iSeqNumber
PARAM;iSeqNumber    ;USES;TYPE;int
INTR;Bausparer      ;RECV;PARAM;nGrdNr
PARAM;iGrdNr        ;USES;TYPE;Nummer
INTR;Bausparer      ;RECV;PARAM;nBBZS
PARAM;iBBZS         ;USES;TYPE;Nummer

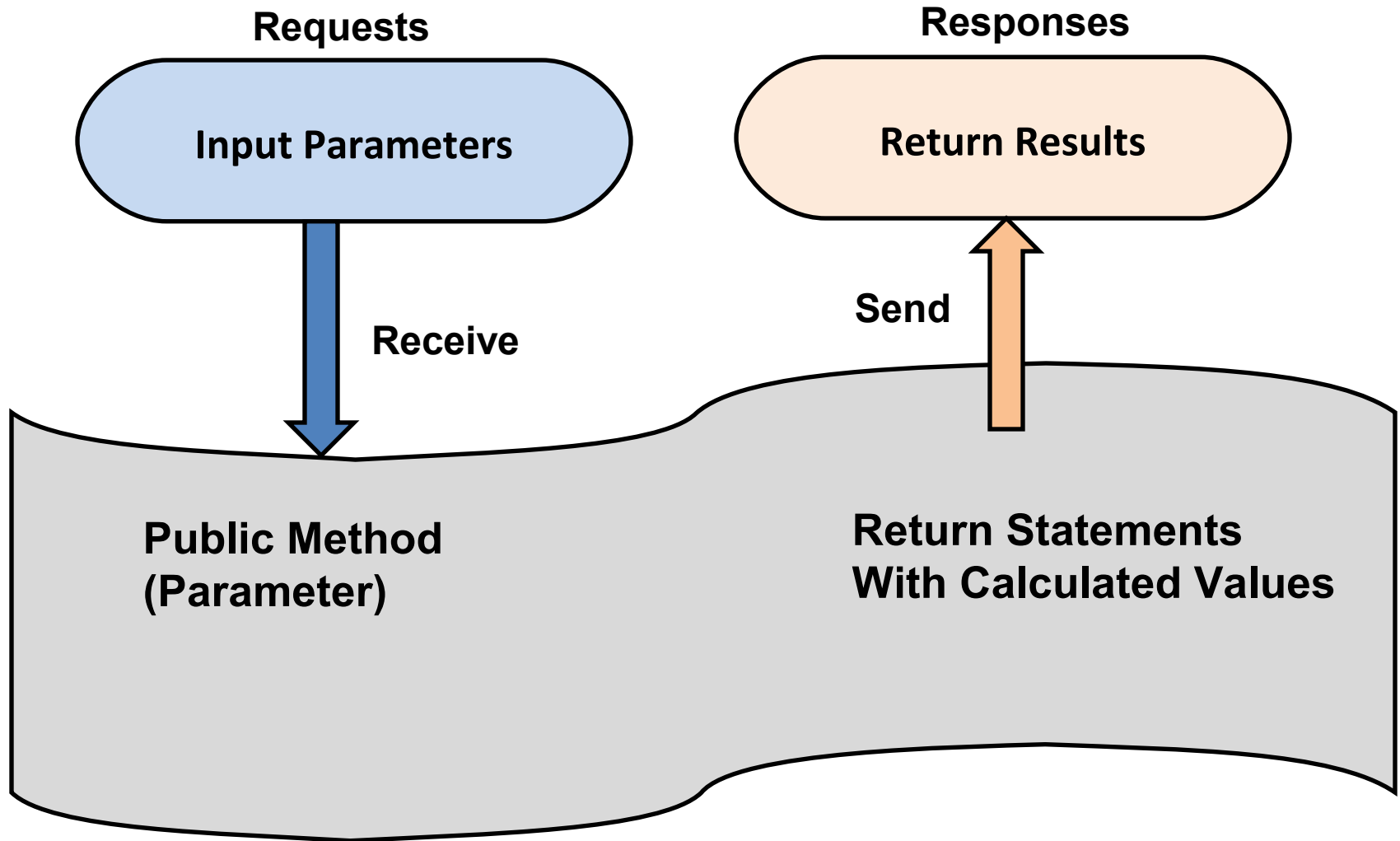
```

```

CLAS;Bausparer      ;OWNS;FUNC;toPerson
FUNC;toPerson       ;OWNS;INTR;Person
INTR;Person         ;OWNS;FUNC;getBBZS
FUNC;Bausparer      ;OWNS;INTR;getBBZS
INTR;getBBZS       ;SEND;PARAM;BBZS
PARAM;BBZS         ;USES;TYPE;Nummer
INTR;Person         ;OWNS;FUNC;getGrdNr
FUNC;getGrdNr      ;OWNS;INTR;getGrdNr
INTR;getGrdNr      ;SEND;PARAM;m_nGrdNr
PARAM;GrdNr        ;USES;TYPE;Nummer
INTR;Person
;OWNS;FUNC;getNachname
FUNC;getNachname   ;OWNS;INTR;getNachname
INTR;getNachname   ;SEND;PARAM;Nachname
PARAM;Nachname     ;USES;TYPE;varChar

```

# Public Method Parameters & Return Results



# Generated WSDL Data Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="DBBSAG-BAUSPAR" xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://corba"
  xmlns:DBBSAG-BAUSPAR= http://wbmservicetypes10.generated.wbm.websf4j.at/>
<types>
  <complexType name = „Person">
    <element name = "Role" type = "xs:cRole"/>
    <element name = „Version" type = "xs:nVersion"/>
    <element name = „ReadTime" type = "xs:tsReadTimer"/>
    <element name = „SeqNumber" type = "xs:iSeqNumber"/>
    <element name = "GrdNr_ELEM" type = "xs:Nummer"/>
    .....
  </complexType>
  <complexType name = "cRole">
    <sequence>
      <element name = "cRole_ELEM" type = "xs:string"/>
    </sequence>
  </complexType>
  <complexType name = "nVersion">
    <sequence>
      <element name = "nVersion_ELEM" type = "xs:Nummer"/>
    </sequence>
  </complexType>
  <complexType name = "tsReadTime">
    <sequence>
      <element name = "tsReadTime_ELEM" type = "xs:TimeStamp"/>
    </sequence>
  </complexType>
  <complexType name = "iSeqNumber">
    <sequence>
      <element name = "iSeqNumber_ELEM" type = "xs:int"/>
    </sequence>
  </complexType>
```



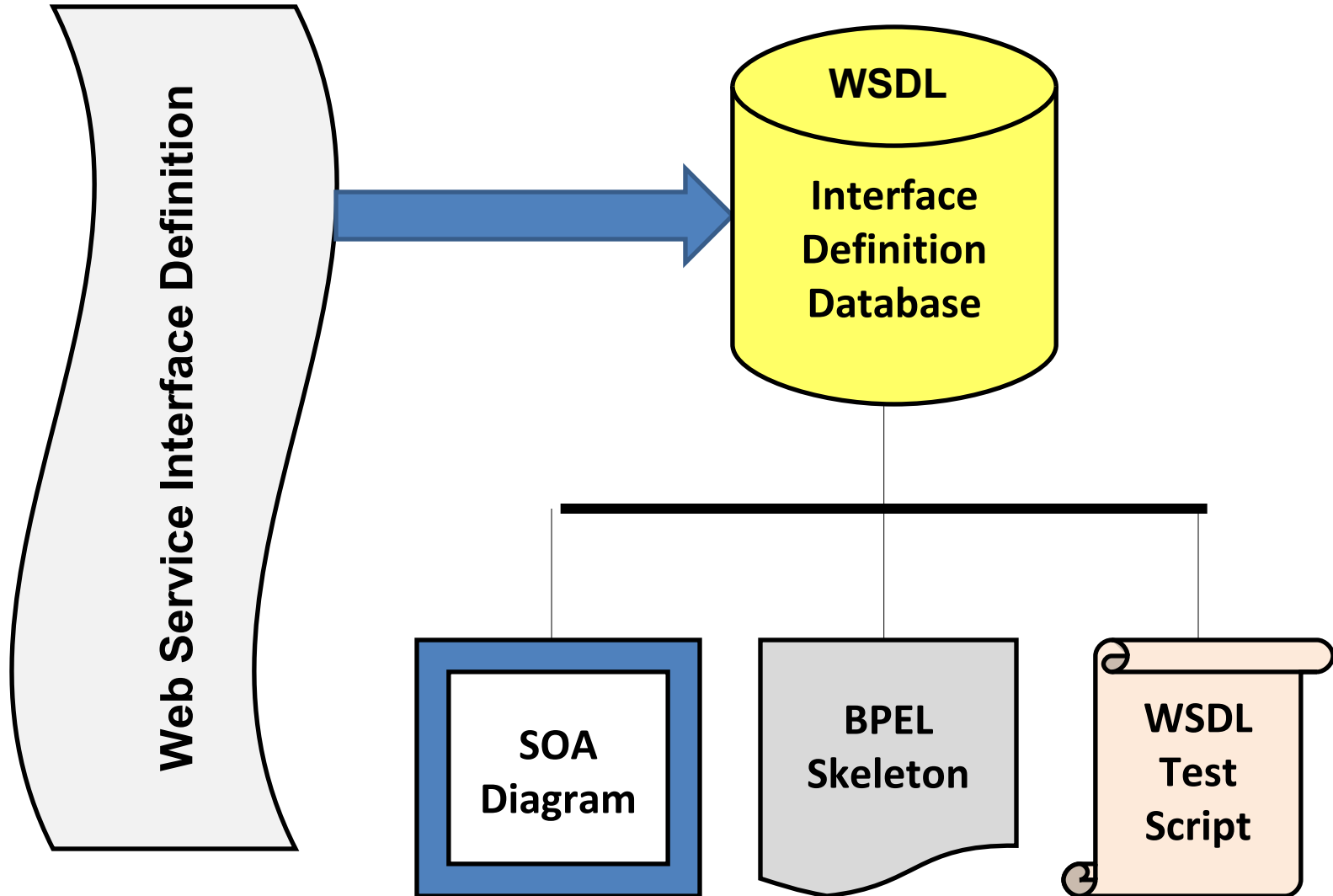
# Generated XML Data Type Definitions

```
<complexType name = "Person_Input_Params">
  <sequence>
    <element name = "cRole " type = "ns:string"/>
    <element name = "nVersion " type = "ns:iNummer"/>
    <element name = "tsReadTime " type = "ns:TimeStamp"/>
    <element name = "iSeqNumber " type = "ns:int"/>
    <element name = "nBBZS " type = "ns:string"/>
    <element name = "dGeburtsdatum " type = "ns:Datum"/>
    <element name = "nGrdNr " type = "ns:Number"/>
    <element name = "cNachname " type = "ns:string"/>
    <element name = "cVorname " type = "ns:string"/>
    <element name = "cKzSparer" type = "ns:string"/>
    <element name = "bKzArchiv " type = "ns:boolean"/>
    <element name = "cOrt " type = "ns:string"/>
    <element name = "cPlz " type = "ns:Nummer"/>
    <element name = "cStrasse_Param " type = "ns:string"/>
  </sequence>
</complexType>
<complexType name = "Person_Output_Params">
  <sequence>
    <element name = "cRole " type = "ns:string"/>
    <element name = "nVersion " type = "ns:Nummer"/>
    <element name = "tsReadTime " type = "ns:TimeStamp"/>
    <element name = "iSeqNumber " type = "ns:int"/>
    <element name = "nBBZS " type = "ns:Nummer"/>
    <element name = "dGeburtsdatum " type = "ns:Datum"/>
    <element name = "nGrdNr " type = "ns:Nummer"/>
    <element name = "cNachname " type = "ns:string"/>
    <element name = "cVorname " type = "ns:string"/>
    <element name = "cKzSparer " type = "ns:string"/>
    <element name = "bKzArchiv " type = "ns:boolean"/>
    <element name = "cOrt " type = "ns:string"/>
    <element name = "cPlz " type = "ns:Nummer"/>
    <element name = "cStrasse " type = "ns:string"/>
  </sequence>
</complexType>
```

# Generated WSDL Operations

```
<portType name="BAUSPAR_DBBSAG">
  <operation name="Bausparer">
    <input message="tns:Bausparer_Input"/>
    <output message="tns:Bausparer_Output"/>
  </operation>
  <operation name="BausparerMarshalling">
    <input message="tns:BausparerMarshalling_Input"/>
    <output message="tns:BausparerMarshalling_Output"/>
  </operation>
  <operation name="BausparerMarshalling">
    <input message="tns:BausparerMarshalling_Input"/>
    <output message="tns:BausparerMarshalling_Output"/>
  </operation>
  <operation name="selectMatchcodeString">
    <input message="tns:selectMatchcodeString_Input"/>
    <output message="tns:selectMatchcodeString_Output"/>
  </operation>
  <operation name="unmarshalSelectGrundnummer">
    <input message="tns:unmarshalSelectGrundnummer_Input"/>
    <output message="tns:unmarshalSelectGrundnummer_Output"/>
  </operation>
  <operation name="unmarshalSelectMatchcode">
    <input message="tns:unmarshalSelectMatchcode_Input"/>
    <output message="tns:unmarshalSelectMatchcode_Output"/>
  </operation>
  <operation name="Person">
    <input message="tns:Person_Input"/>
    <output message="tns:Person_Output"/>
  </operation>
</portType>
```

# Outputs of the WSDL Database



# WSDL Database Viewer (Input)

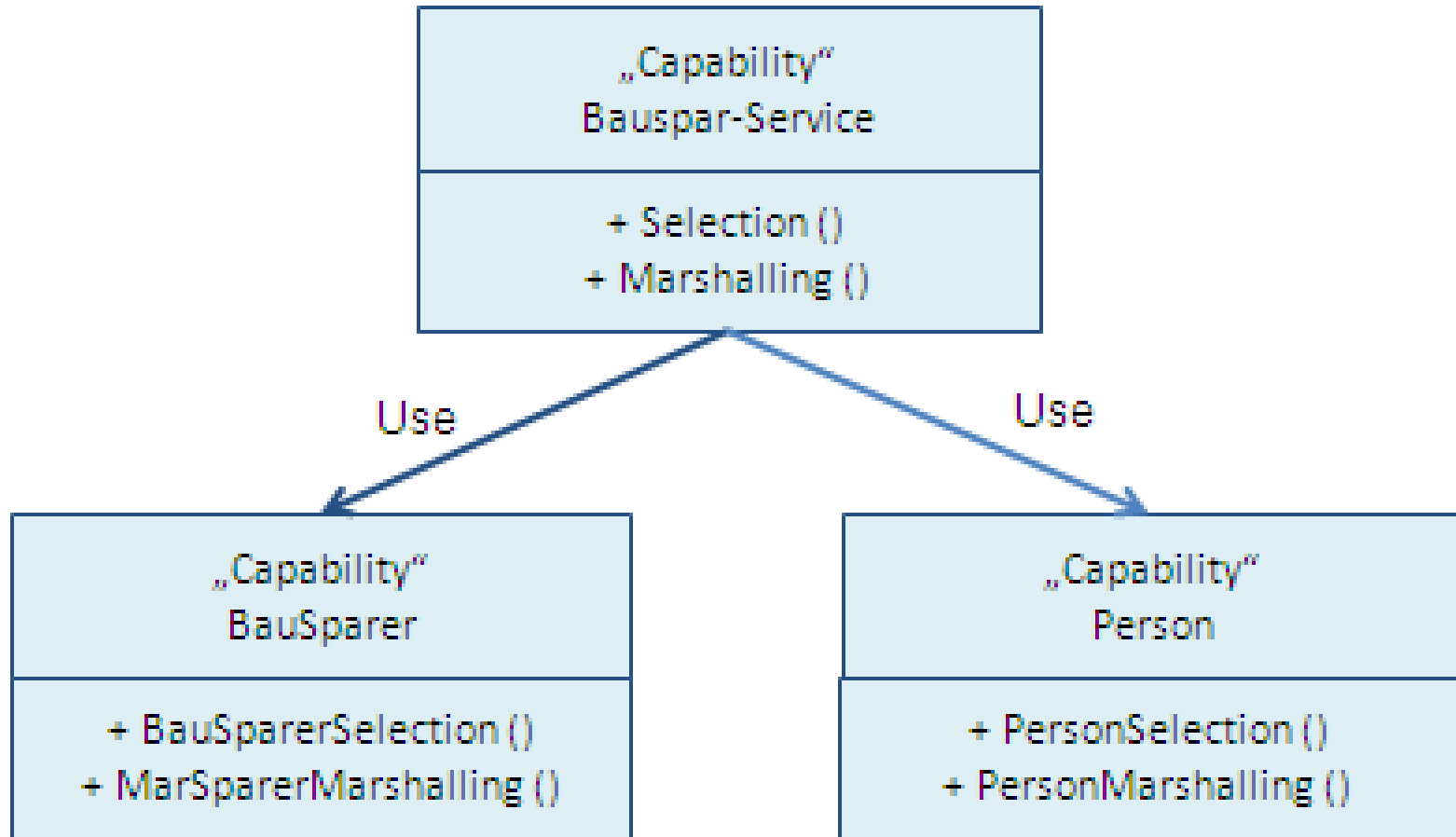
Interfaces		Operations		Inputs/Outputs		Parameters		
Name	Name	Type	Name	Level	LevelGraf	Type		
ATM_DEMO	Bausparer	input	Bausparer_Input	1	01 Bausparer_Input_Type	Bausparer_Input_Params		
BACKEND_ORDERING	BausparerMarshalling	output	Bausparer_Output	2	02 cRole_Param	cRole		
BAUSPAR_DBBSAG	BausparerMarshalling			3	03 cRole_ELEM	string		
FRONTEND_ORDERING	selectMatchcodeString			2	02 nVersion_Param	nVersion		
IWarehouseWebService	unmarshalSelectGrundnummer			3	03 nVersion_ELEM	string		
	unmarshalSelectMatchcode			2	02 tsReadTime_Param	tsReadTime		
	Person			3	03 tsReadTime_ELEM	string		
				2	02 iSeqNumber_Param	iSeqNumber		
				3	03 iSeqNumber_ELEM	string		
				2	02 nGrdNr_Param	nGrdNr		
				3	03 nGrdNr_ELEM	string		
				2	02 nBBZS_Param	nBBZS		
				3	03 nBBZS_ELEM	string		
				2	02 cNachname_Param	cNachname		
				3	03 cNachname_ELEM	string		
				2	02 cVorname_Param	cVorname		
				3	03 cVorname_ELEM	string		
				2	02 cKzAnrede_Param	cKzAnrede		
				3	03 cKzAnrede_ELEM	string		
				2	02 cKzTitel_Param	cKzTitel		
				3	03 cKzTitel_ELEM	string		
				2	02 dGeburtsdatum_Param	dGeburtsdatum		
				3	03 dGeburtsdatum_ELEM	string		
				2	02 bKzArchiv_Param	bKzArchiv		
				3	03 bKzArchiv_ELEM	string		
				2	02 nGrdNr_Param	nGrdNr		
BAUSPAR_DBBSAG	Bausparer	Bausparer_Input	Bausparer_Input_Type					

Navigation controls: Back, Forward, Home, End, Search (Click to search), Refresh, Close

# WSDL Database Viewer (Output)

Interfaces	Operations	Inputs/Outputs		Parameters	
Name	Name	Type	Name	Level	LevelGraf
BAUSPAR_ProdNm	setBBZS.setBBZS	input	selectMatchcode.selectMatchcode_Input	1	01 selectMatchcode.selectMatchcode_Output_Type
FLBAUS_ProdNm	setNachname.setNachname	output	selectMatchcode.selectMatchcode_Output	2	02 unmarshalSelectMatchcode_Param
PERSON_ProdNm	setVorname.setVorname			3	03 unmarshalSelectMatchcode_ELEM
	setKzAnrede.setKzAnrede				
	setKzTitel.setKzTitel				
	setKzArchiv.setKzArchiv				
	setGeburtsdatum.setGeburtsdatum				
	setAdresse.setAdresse				
	o2x.o2x				
	BausparerMarshalling.BausparerMarshalling				
	selectGrundnummer.selectGrundnummer				
	selectMatchcode.selectMatchcode				
	unmarshalSelectGrundnummer.unmarshalSelectGrundnummer				
	unmarshalSelectMatchcode.unmarshalSelectMatchcode				
	BausparerStub.BausparerStub				
	selectMatchcode.selectMatchcode				
	BausparerListe.BausparerListe				
	getListe.getListe				
	setListe.setListe				
	add.add				
	remove.remove				
	o2x.o2x				
	FLBausparer.FLBausparer				
	getStammdaten.getStammdaten				
	deleteChangeAdress.deleteChangeAdress				
	startChangeAdress.startChangeAdress				
BAUSPAR_ProdNm	selectMatchcode.selectMatchcode		selectMatchcode.selectMatchcode_Output		selectMatchcode.selectMatchcode_Output_Type

## Service Hierarchy



Main Services invoke subordinate Services

```
process name = "Bausparen"  
<PartnerLink name = "BausparerUser"  
  partnerLinkType = "calender:User"  
  myRole = "Provider"  
    partnerRole = "User" />  
</partnerLinks>  
<variables>  
  <!-- inputs to Bausparer Functions -->  
  <variable name = " cRole_ELEM " messageType = "Bausparer:string"/>  
  <variable name = "nGrdNr" messageType = "Bausparer:Nummer"/>  
  <variable name = "nBBZS" messageType = "Bausparer:Nummer"/>  
  <!-- outputs from Bausparer Functions -->  
  <variable name = "ResponseCode" messageType = "int"/>  
  <variable name = "Person" messageType = "Bausparer:Person"/>  
</variables>  
<assign>  
  <copy>  
    <from variable = "Role" part = "Client" />  
    <to variable = "cRole" part = "Bausparer" />  
  </copy>  
  <copy>  
    <from variable = "GrundId" part = "Client" />  
    <to variable = "nGrdNr" part = "Bausparer" />  
  </copy>  
</assign>  
<!-- call Bauparer Service to provide Person Data -->  
<invoke partnerLink = "BausparerUser"  
  portType = "BausparerStatusPT"  
  operation = "Bausparer"  
  inputVariable = "BausparerRequest"  
  output Variable = "BausparerResponse" />  
<assign>  
  <copy>  
    <from variable = "Person" part = "BausparerResponse" />  
    <to variable = "ThisClient" part = "Clients" />  
  </copy>  
</assign>  
</process>
```

# Generated Test Script for Web Service Test

```
service: BausparerSOAPServerservice;
  if (testcase = „Bausparer001“);
  // It should be possible to select a Bausparer by means of the ground number
  // and the BBZS
    if ( operation = „Bausparer“);
      if ( request = „Bausparer1Request“);
        if ( object = "Person" );
          assert inp.CRole = „Besitzer“;
          assert inp.nGrdNr = „4711“;
          assert inp.bBBZS = „120036“;
          assert inp.tsReadTime = „201303130700“;
          assert inp.iSeqNumber = "100";
        endObject;
      endRequest ;
      if ( response = „Bausparer1Response“);
        assert out.$ResponseTime < „1200“;
        if ( object = "return" );
          assert out.ResponseCode = „00“;
          assert out.cNachname = „Schmidt“;
          assert out.cVorname = „Karl“;
          assert out.dGeburtsdatum = „19220420“;
        endObject;
      endResponse ;
    endOperation;
  endCase;
```



# Summary

- Existing Code is an excellent source for obtaining web services quickly and cheaply
- Legacy classes are potential services and public methods are potential operations
- The code has only to be mined to extract the interface definitions for wrapping the services.
- From the extracted interface definitions both client frameworks and test scripts can be generated to test the wrapped services.
- The migration to Service Orientation can be quickened by reusing existing software!