



UNIA EUROPEJSKA  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY



Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego

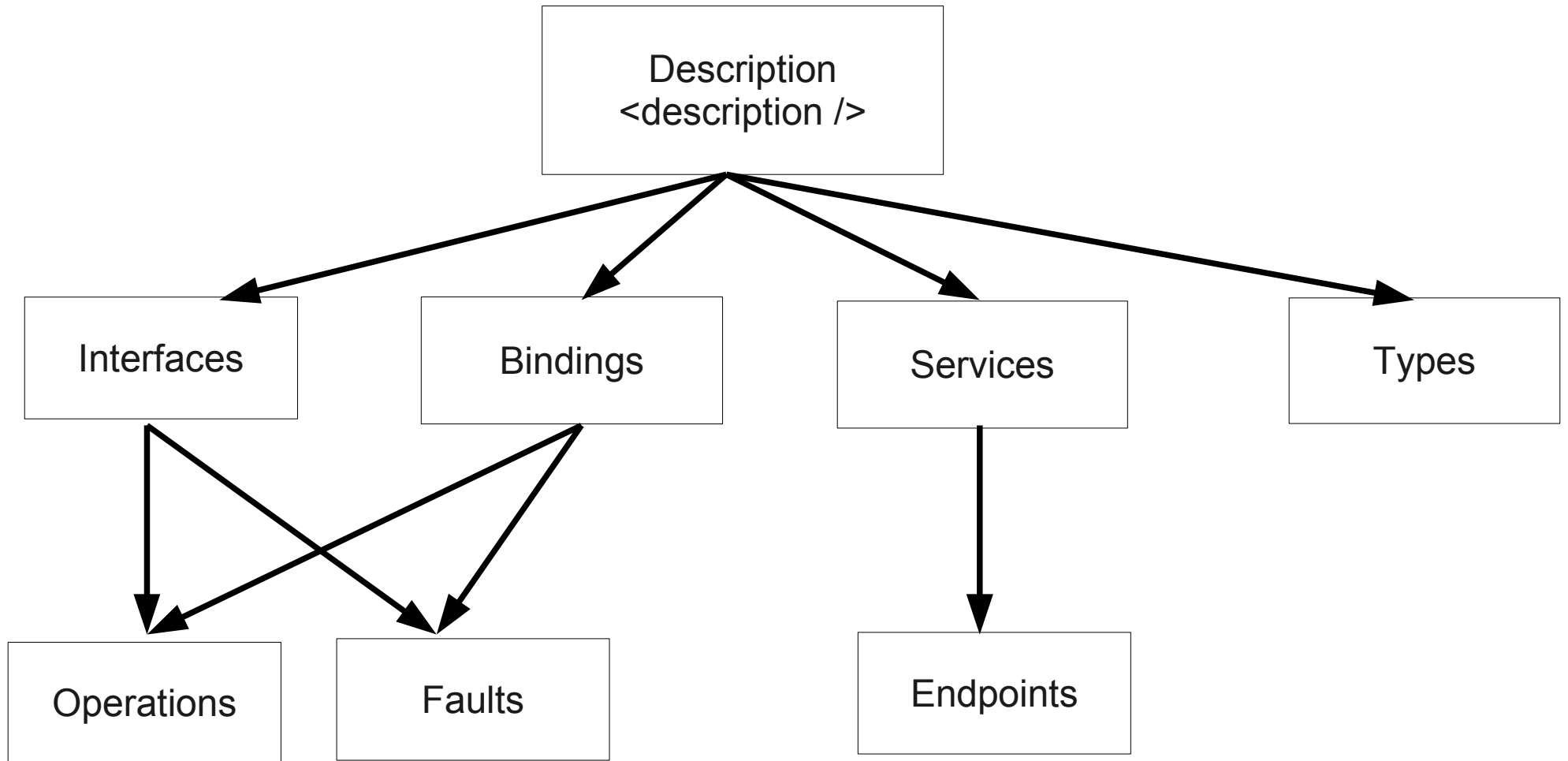
# Rozproszone systemy Internetowe

## Web Service Description Language

„Podniesienie potencjału uczelni wyższych jako czynnik rozwoju gospodarki opartej na wiedzy”

Nr projektu: UDA-POKL.04.01.01-00-143/09-00

# Web Services Description Language



# Elementy WSDL

- Elementy składowe
  - Opis (description) - pojęcie "targetNamespace"
  - Interfejs:

```
<interface name="" extends="" styleDefault="" >
  <operation name="" pattern="" style="" >
    <input messageLabel="" element="" />
    <output messageLabel="" element="" />
    <infault messageLabel="" ref="" />
    <outfault messageLabel="" ref="" />
  </operation>
  <fault name="" element="" />
</interface>
```

- Predefiniowane wzorce wymiany komunikatów:
  - *in-only, robust-in-only, in-out, in-optional-out, out-only, robust-out-only, out-in, out-optional-in*

# Elementy WSDL

```
<output messageLabel="" element="" />
```

```
<outfault messageLabel="" ref="" />
```

*definiowane przez MEP*

**No Fault**

**#none  
#any  
#other**

## Message Triggers Fault

Dowolny komunikat może spowodować wysłanie błędu (w przeciwnym kierunku)

## Fault Replaces Message

Dowolny komunikat (poza pierwszym) może zostać zastąpiony błędem (w przeciwnym kierunku)

# Wzorce wymiany komunikatów

MEP	Nazwy komunikatów	Model propagacji błędu
In-Only	In	No Fault
Robust-In-Only	In	Message Triggers Fault
In-Out	In, Out	Fault Replaces Message
In-Optional-Out	In, Out	Fault Replaces Message
Out-Only	Out	No Fault
Robust-Out-Only	Out	Message Triggers Fault
Out-In	Out, In	Fault Replaces Message
Out-Optional-In	Out, In	Message Triggers Fault

# Style operacji

- **RPC** (*<http://www.w3.org/ns/wsd1/style/rpc>*)
  - MEP: in-only lub in-out
  - Komunikaty zawierają konstrukcje XML (postawione są pewne warunki)
  - możliwość użycia rozszerzenia wrpc:signature
- **IRI** (*<http://www.w3.org/ns/wsd1/style/iri>*)
  - konstrukcja, która ułatwia przekształcenie komunikatu do notacji IRI
- **Multipart** (*<http://www.w3.org/ns/wsd1/style/multipart>*)
  - komunikaty serializowane jako "multipart/form-data"
  - przeznaczony dla klientów używających XForms z dowiązaniem HTTP

# Elementy WSDL

- Dowiązanie protokołów (binding)

```
<binding name="" interface="" type="" >  
  <operation ref="" >  
    <input />  
    <output />  
    <infault />  
    <outfault />  
  </operation>  
  <fault ref="" />  
</binding>
```

- Usługa

```
<service name="" interface="" >  
  <endpoint name="" binding="" address="" />  
</service>
```

# Przykład

```
<?xml version="1.0" encoding="utf-8" ?>
<description
  xmlns="http://www.w3.org/ns/wsd1"
  targetNamespace= "http://wyklady.com/wsd1/ktel"
  xmlns:kns= "http://wyklady.com/wsd1/ktel"
  xmlns:wykns = "http://wyklady.com/schemas/ktel"
  xmlns:wsoap= "http://www.w3.org/ns/wsd1/soap"
  xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsd1x= "http://www.w3.org/2006/01/wsd1-extensions">
```

## <documentation>

Tutaj może być zapisana dodatkowa informacja tekstowa.

Przestrzenie nazewnicze wykorzystane w przykładzie:

kns - przestrzeń WSDL dla książki telefonicznej

wykns - przestrzeń dla XMLSchema opisującego elementy

wiadomości

w książce telefonicznej

wsoap - powiązanie pomiędzy WSDL a SOAP

wsd1x - rozszerzenia WSDL

soap - SOAP

## </documentation>



# Przykład

```
<types>
  <xs:schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://wyklady.com/schemas/kte1"
    xmlns="http://wyklady.com/schemas/kte1">

    <xs:element name="Szukaj" type="tSzukaj"/>
    <xs:complexType name="tSzukaj">
      <xs:sequence>
        <xs:element name="imie" type="xs:string"/>
        <xs:element name="nazwisko" type="xs:string"/>
        <xs:element name="ulica" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>

    <xs:element name="SzukajWynik" type="xs:string"/>

    <xs:element name="Blad" type="xs:string"/>

  </xs:schema>
</types>
```

# Przykład

```
<interface name = "SzukajAbonenta" >

  <fault name = "BladDanych"
    element = "wykns:Blad"/>

  <operation name="opSzukaj"
    pattern="http://www.w3.org/2006/01/wsdl/in-out"
    style="http://www.w3.org/2006/01/wsdl/style/iri"
    wsdlx:safe = "true">

    <input messageLabel="In"
      element="wykns:Szukaj" />

    <output messageLabel="Out"
      element="wykns:SzukajWynik" />

    <outfault ref="kns:BladDanych" messageLabel="Out"/>

  </operation>

</interface>
```

# Przykład

```
<binding name="KTELSOAPBinding"
  interface="kns:SzukajAbonenta"
  type="http://www.w3.org/2006/01/wsdl/soap"
  wsoap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP" >

  <fault ref="kns:BladDanych" wsoap:code="soap:Sender"/>

  <operation ref="kns:opSzukaj"
    wsoap:mep="http://www.w3.org/2003/05/soap/mep/soap-response"/>

</binding>

<service name="KTEL" interface="kns:SzukajAbonenta">

  <endpoint name="ktelEndpoint"
    binding="kns:KTELSOAPBinding"
    address="http://wyklady.com/ktel"/>

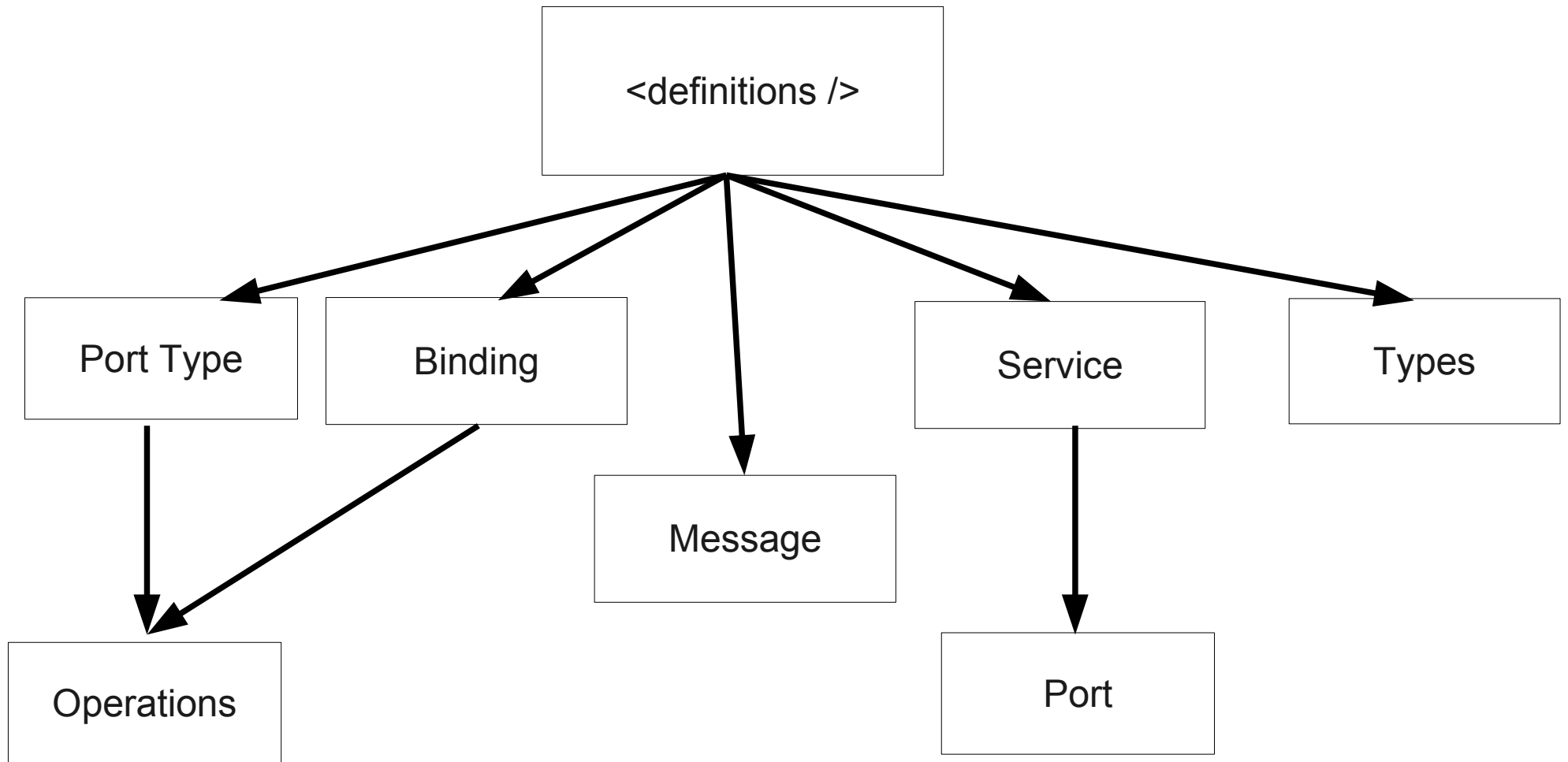
</service>
</decription>
```

# Dodatkowe zadania do przebadania/przemyślenia

- Porównanie specyfikacji WSDL 1.1 z prezentowaną na wykładzie
- Mechanizm dziedziczenia interfejsów
- Definiowanie własnych wzorców wymiany komunikatów (MEP)
- Opis semantyczny dla WSDL - SAWSDL

# WSDL 1.1

*<http://www.w3.org/TR/wsd1>*



# WSDL 1.1

- **WSDL:** *http://schemas.xmlsoap.org/wsdl/*

- **SOAP:**

*http://schemas.xmlsoap.org/wsdl/soap/*

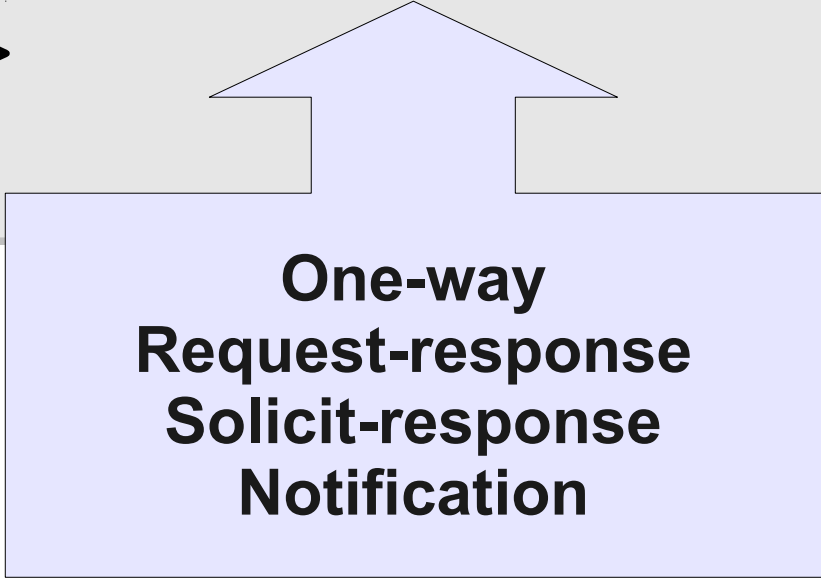
- **HTTP:**

*http://schemas.xmlsoap.org/wsdl/http/*

```
<message name="">  
  <part name="" element="" />  
  <part name="" element="" />  
</message>
```

# WSDL 1.1

```
<portType name="">  
  <operation name="">  
    <input name="" message="" />  
    <output name="" message="" />  
    <fault name="" message="" />  
  </operation>  
</portType>
```



**One-way  
Request-response  
Solicit-response  
Notification**

# WSDL 1.1

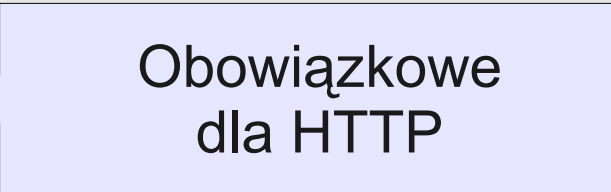
```
<binding name="" type="portType">  
  <operation name="">  
    [...]   
    <input name=""> [...] </input>  
    <output name=""> [...] </output>  
    <fault name=""> [...] </fault>  
  </operation>  
</binding>
```

[...] - *parametry związane  
z konkretnym dowiązaniem*



# WSDL 1.1

```
<binding name="" type="portType">
  <soap:binding
    style="document/rpc"
    transport="http://schemas.../soap/http" />
  <http:binding verb="GET/POST" />
  <operation name="">
    <soap:operation
      style="..."
      soapAction="...">
    <input name="">
      <soap:body use="literal/encoded" />
      <soap:header ... />
    </input>
```



Obowiązkowe dla HTTP

# WSDL 1.1

```
<service name="" >  
  <port name="" binding="...">  
    [...]  
  </port>  
</service>
```

[...] – *parametry związane z konkretnym dowiązaniem*  
*np: <soap:address location="uri"/>*