

Eclipse下BPEL开发实例分享

大纲

- 环境准备
- 加减法 Web Service 开发部署
- BPEL实例开发、部署
- BPEL实例运行、测试
- 总结

一、环境准备

1.1 预置环境

JDK1.6

Tomcat 6.0

Eclipse 3.6 Helios

<http://www.eclipse.org/downloads/>



Eclipse IDE for Java EE Developers, 206 MB

Downloaded 468,290 Times

[Details](#)



Windows 32 Bit

Windows 64 Bit

环境准备

1.2 安装Apache ODE (BPEL 解析器)

1) 下载

- <http://ode.apache.org/gettingode.html>

选择 1.3.4 release版本

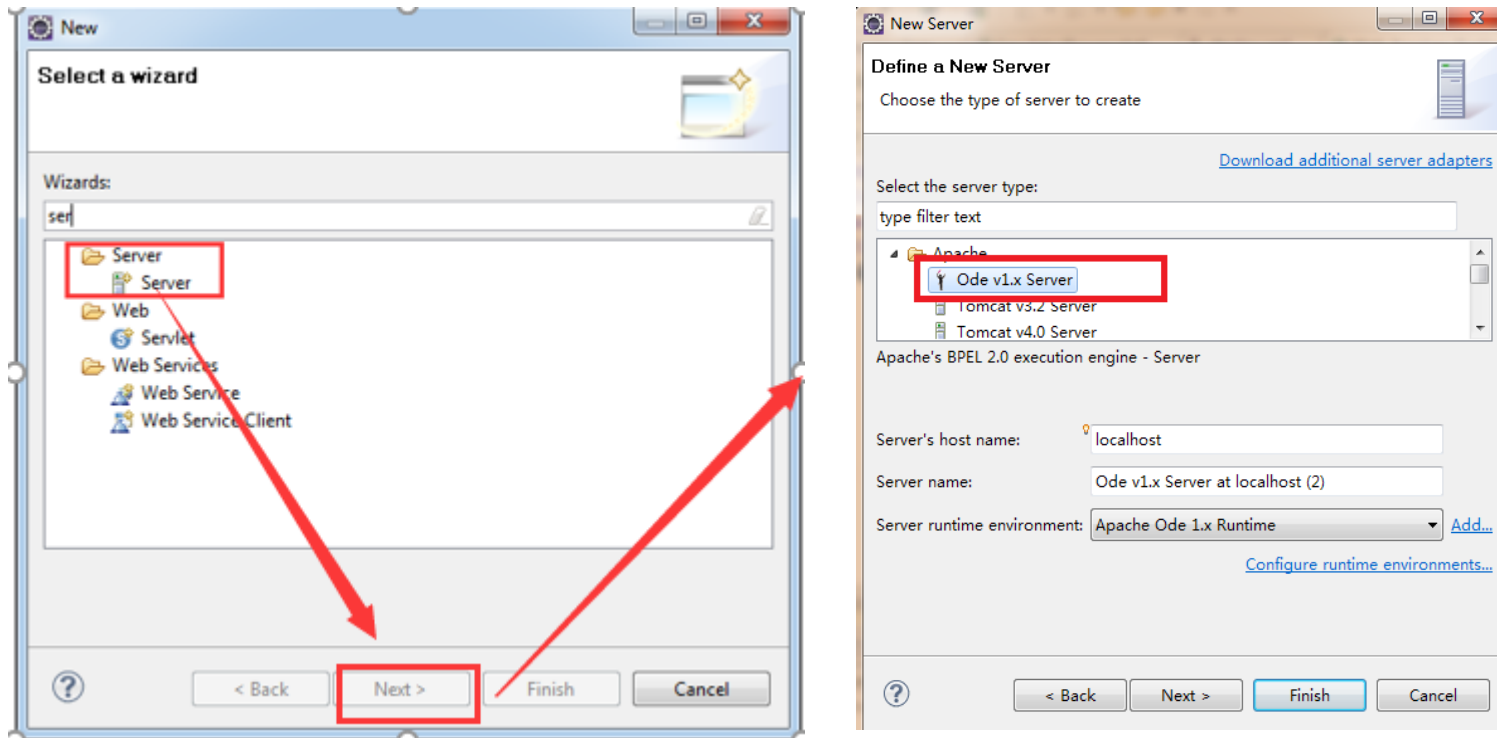
2) 解压zip文件, 找到启动的ode.war将其拷贝至路径 TOMCAT_INSTALL_DIR\webapps directory

环境准备

- 至Tomcat安装目录下bin目录启动Tomcat服务器，待启动完毕，会发现TOMCAT_INSTALL_DIR\webapps路径下会出现ODE文件夹。此时，启动浏览器输入地址<http://localhost:8080/ode>看到ODE的启动界面
- 至此ODE安装完毕！

建立ODE Server

在Eclipse中选择File->New-> Other选项，Server选项。
如下图：



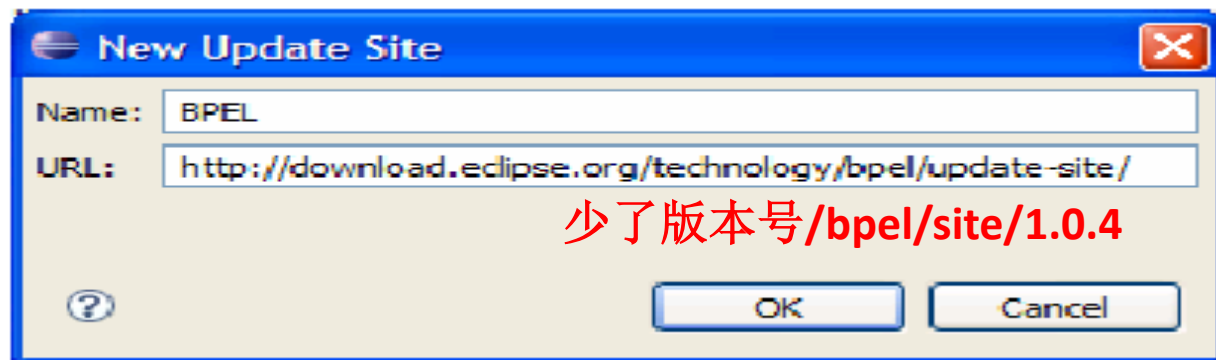
环境准备

1.3 在Eclipse中安装BPEL Desinger

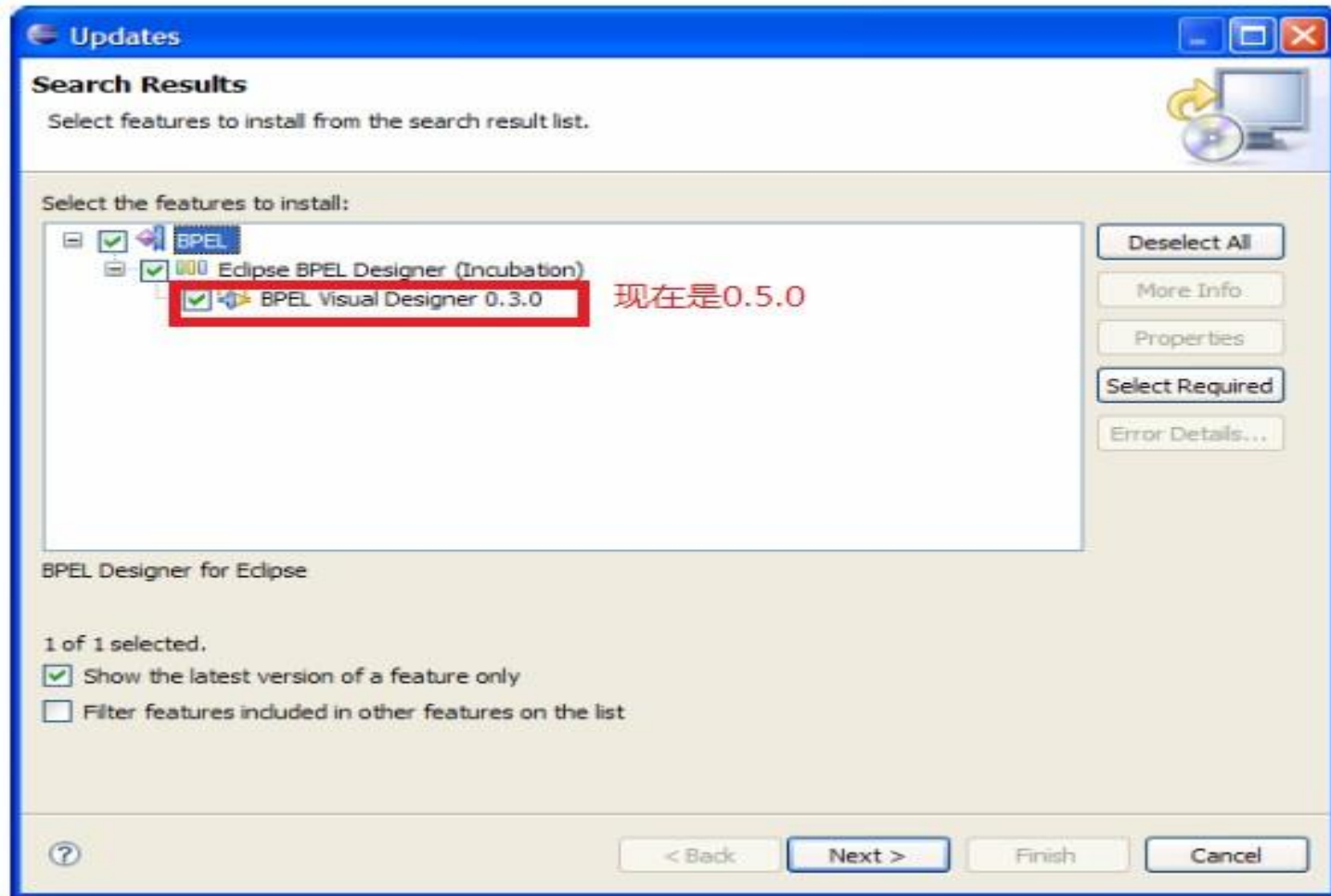
1) 启动Eclipse

Help -> Install New Software... Menu

2) 在弹出窗口中点击Add, 添加BPEL如下图

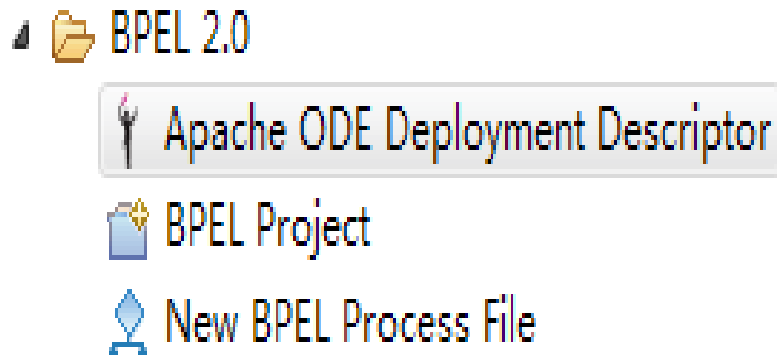


选择安装BPEL Designer



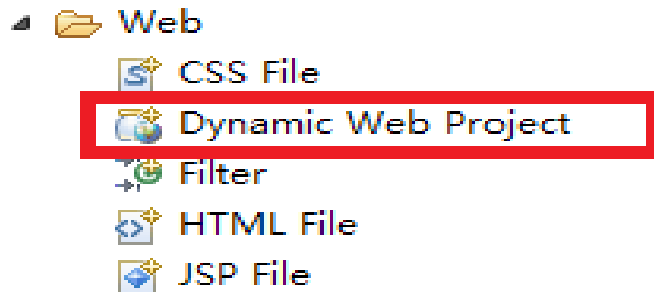
安装完成后查看

在Eclipse中选择File->New-> Other选项，查看是否有BPEL选项。如下图：



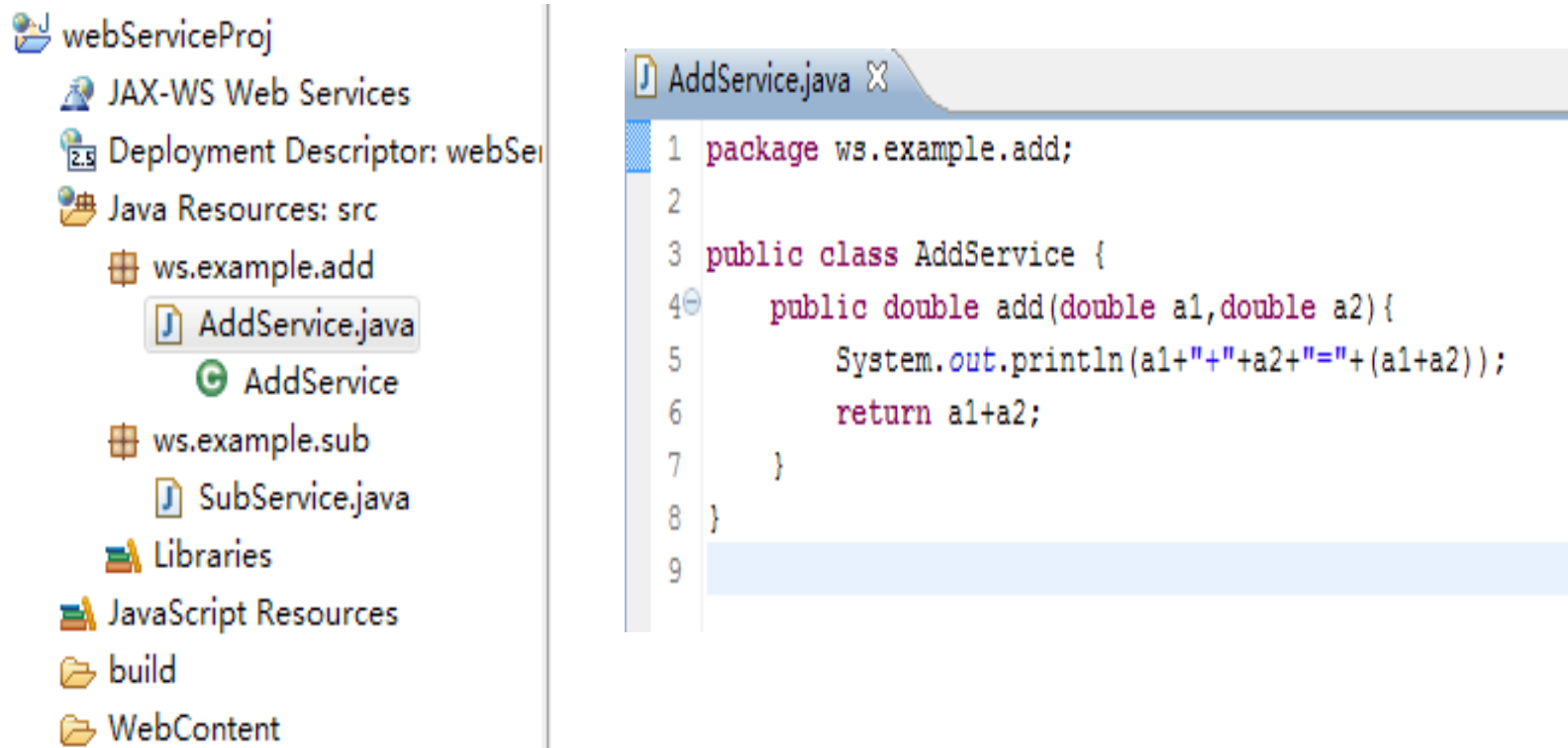
二、加减法WebService开发部署

1. 启动Eclipse， 建立一个动态web Project



工程名为webServiceProj. 工程下面建立两个包ws.example.add和ws.example.sub

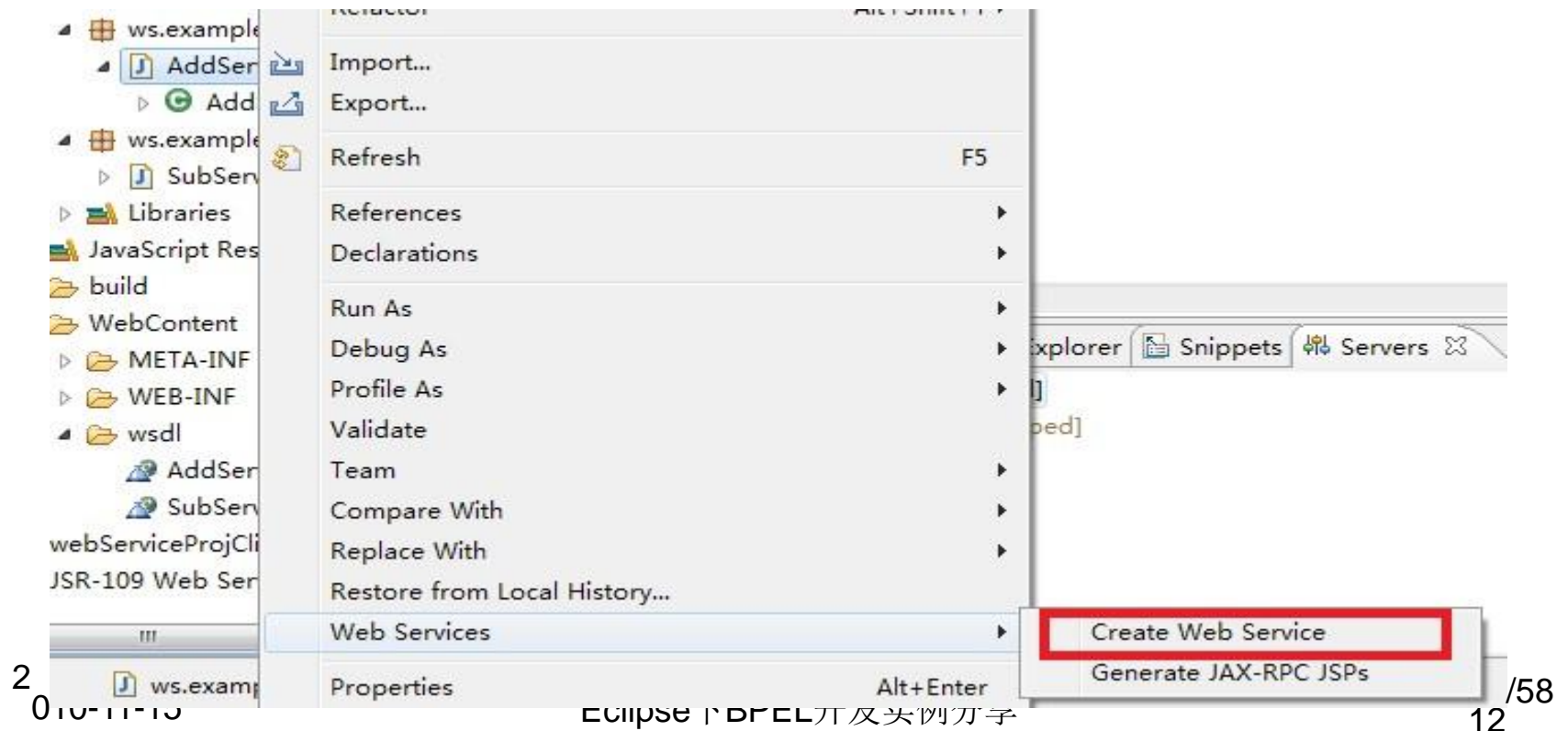
加減法WebService开发



在此以加法为例，SubService即是減法的实现流程。

加减法WebService部署

选择 AddService.java，右键WebService选项，选择 Create WebService。如图所示。



Web Services

Select a service implementation or definition and move the sliders to set the level of service and client generation.



Web service type: Bottom up Java bean Web Service

Service implementation: ws.example.add.AddService [Browse...](#)



Configuration:

[Server: Tomcat v6.0 Server](#)

[Web service runtime: Apache Axis](#)

[Service project: webServiceProj](#)

Client type: Java Proxy



Configuration:

[Server: Tomcat v6.0 Server](#)

[Web service runtime: Apache Axis](#)

[Client project: webServiceProjClient](#)

☐ Publish the Web service

☒ Monitor the Web service

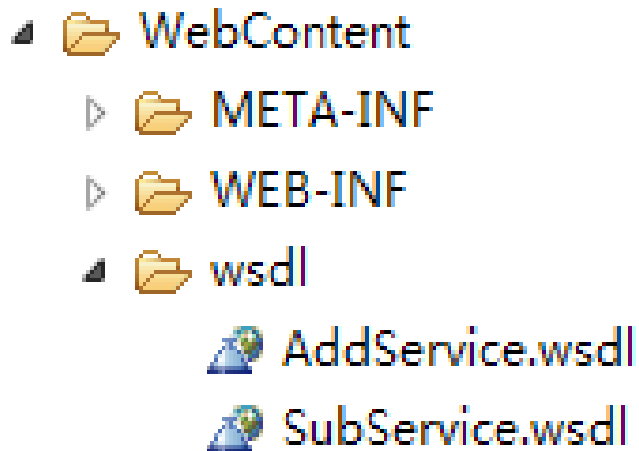
滑片选在 Start Service

滑片选在 Test client,
会生成一个客户端调用

复选上最后一个, 可以
看到交互过程中的
SOAP 消息

加减法WebService部署

点击finish 就会生成java类对应的wsdl和配置文件，并发布到tomcat中，同时会生成调用服务的客户端。



这里的WebService是为BPEL调用WebService实现流程做准备。

The screenshot displays the Eclipse IDE interface. On the left, the **Project Explorer** shows a project named **webServiceProj** with a **Deployment Descriptor: webServiceProc**. Under **Java Resources: src**, there are files **ws.example.add** (containing **AddService.java**) and **ws.example.sub**. A **WSDL** file, **AddService.wsdl**, is also visible. The main editor area is divided into two panes: **Methods** and **Inputs**. The **Methods** pane lists four methods: [getEndpoint\(\)](#), [setEndpoint\(java.lang.String\)](#), [getAddService\(\)](#), and [add\(double,double\)](#). The **Inputs** pane contains the text "Select a method to test." Below these panes is a **Result** section. At the bottom of the IDE, the **Servers** tab is active, showing a table of running servers:

Server	State	Status
Ode v1.x Server at localhost	Stopped	
Tomcat v6.0 Server at localhost	Started	Synchronize

减法的WebService发布流程同加法。

2010-11-15

Eclipse下BPEL开发实例分享

15/58

加减法WebService部署

要想把生成的服务放到Tomcat目录下，则很简单，把这个工程Export成一个war包即可。

File->Export->Web->WAR File,选择保存位置，则会生成**webServiceProj.war**，把生成的war包复制到Tomcat安装路径下的webapps目录下（我的路径为**D:\apache-tomcat-6.0.37_ODE\webapps**）下，重启Tomcat，则完成服务的发布。

测试服务是否发布成功

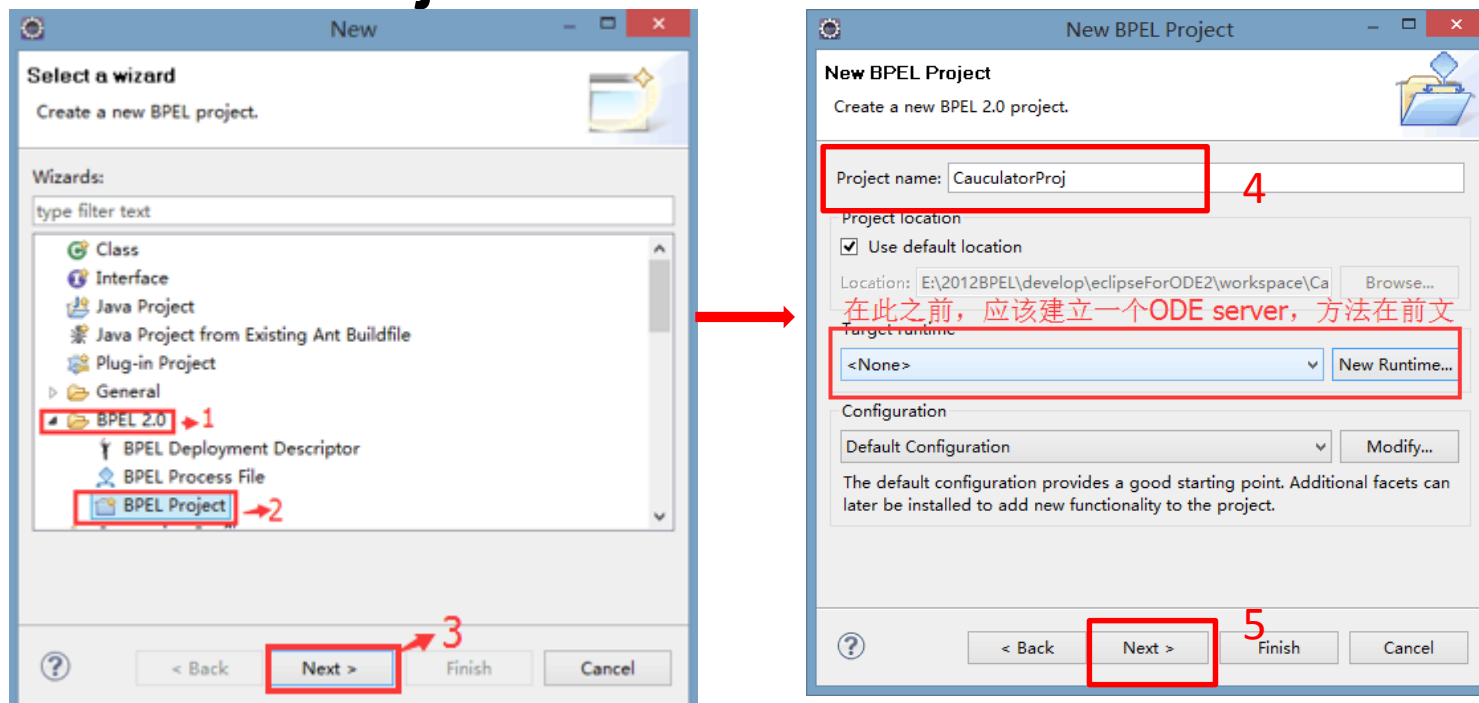
在浏览器里输入

<http://localhost:8080/webServiceProj/services/AddService?wsdl> 则能打开加法服务对应的wsdl.这也意味着服务发布成功了。

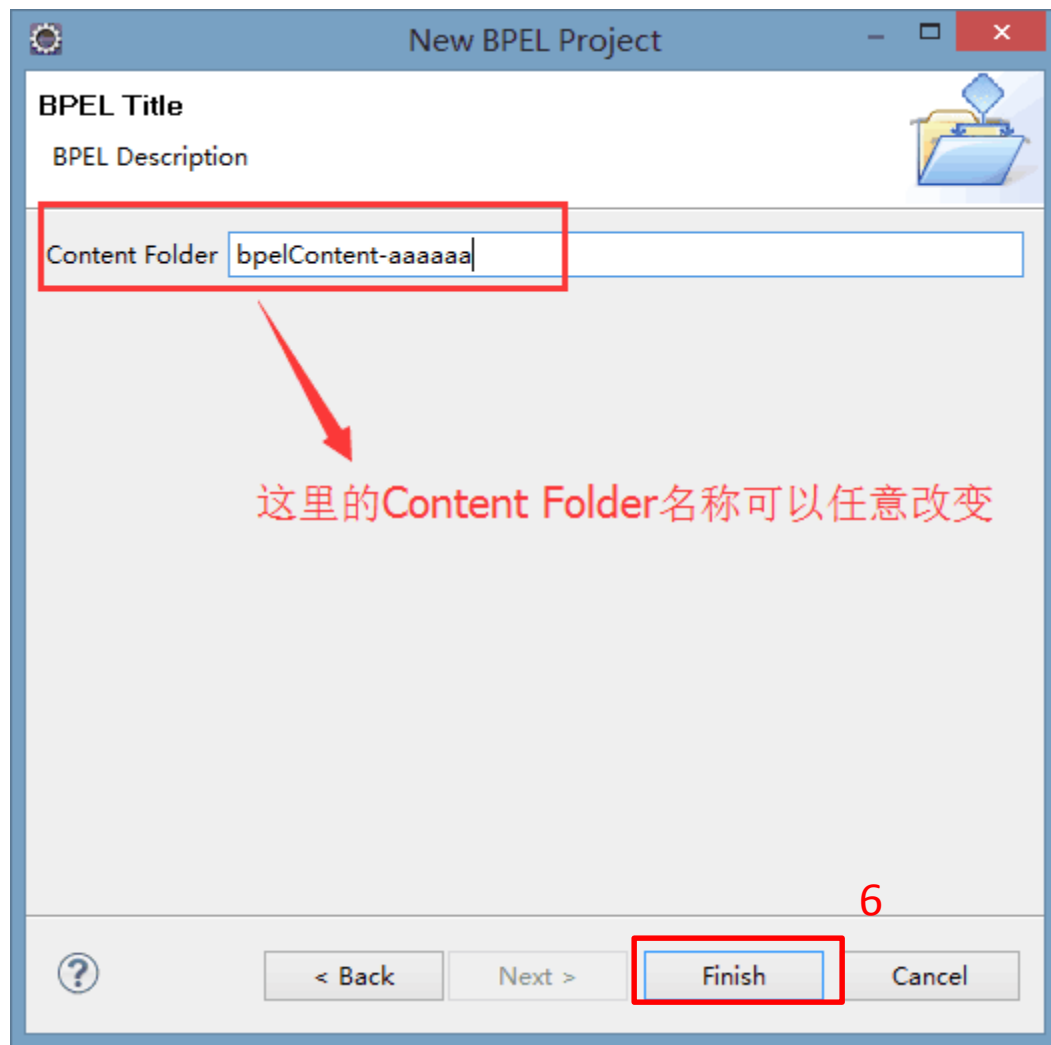
三、BPEL实例开发、部署

3.1 创建bpel工程

点File > New > Other，然后点BPEL 2.0 -> BPEL Project打开BPEL项目对话框，这里项目名Project name取为CaculatorProj

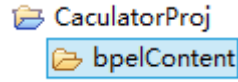


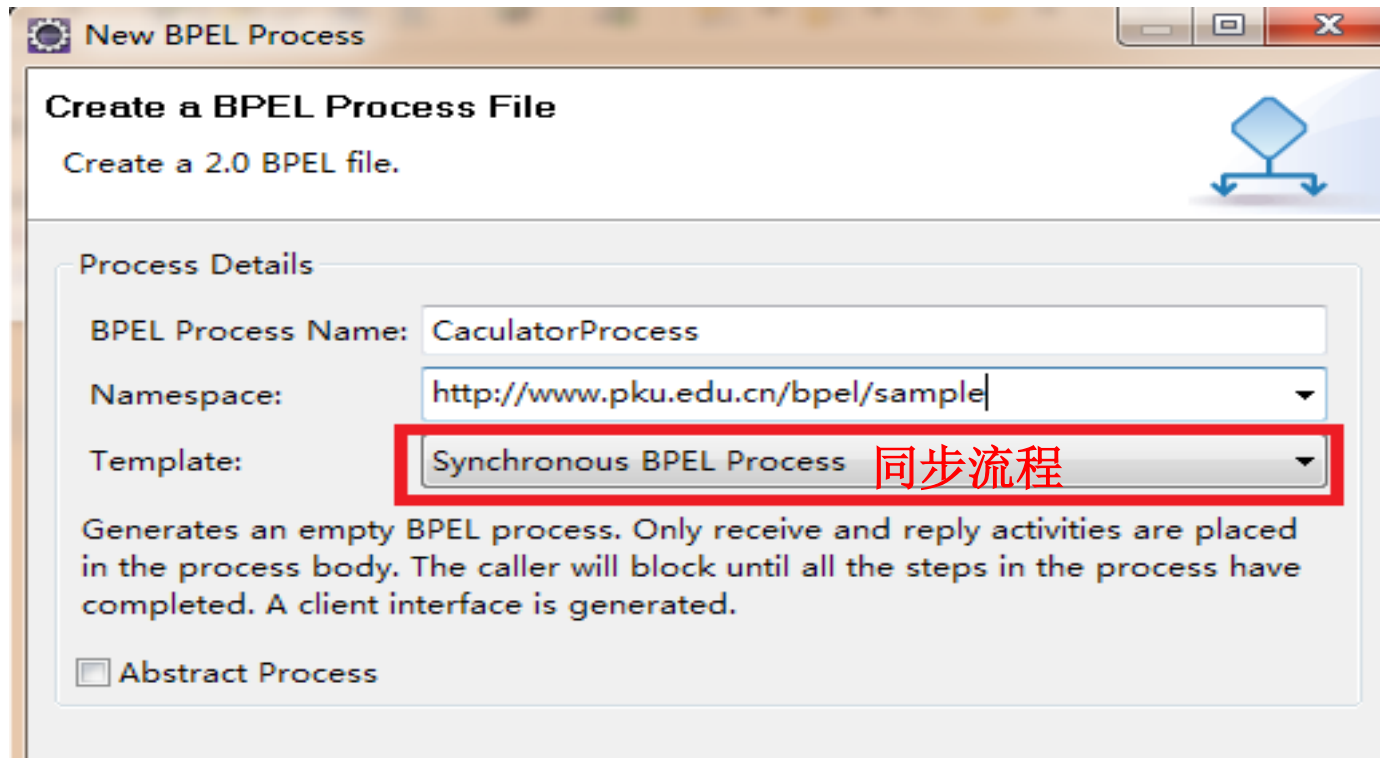
在第五步点击**Next**以后，弹出如下对话框。最后点击**Finish**完成创建BPEL工程



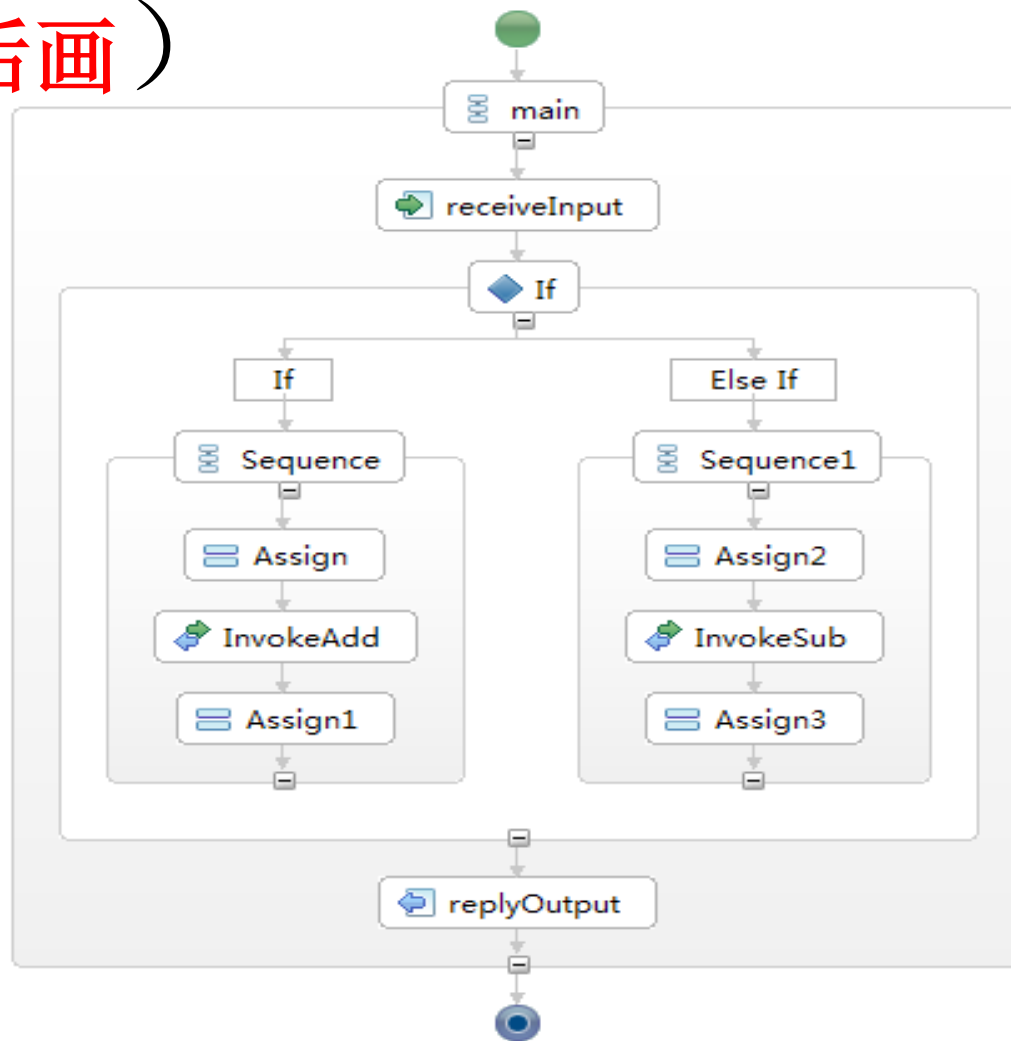
BPEL实例开发

3.2 创建流程

选上刚建立的BPEL项目（也就是 ），点File > New > Other，然后BPEL 2.0 > New BPEL Process File，打开流程创建对话框。按如下设置，点finish即可。流程名是 **CaculatorProcess**，名字空间设为：
<http://www.pku.edu.cn/bpel/sample>（名称空间可自己设置），选择同步流程（**如下图所示**）。

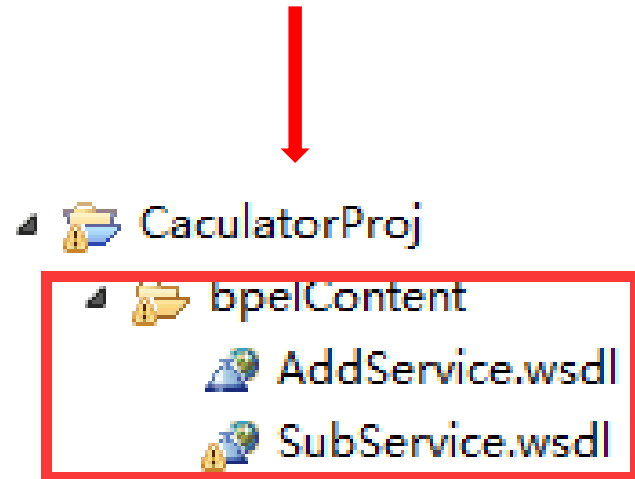


BPEL 开发后项目流程图（**流程项目图最后画**）



BPEL实例开发

3.3 导入加法服务和减法服务的**wsdl**文件。
很简单，就是把**AddService.wsdl**和
SubService.wsdl复制到工程目录里。



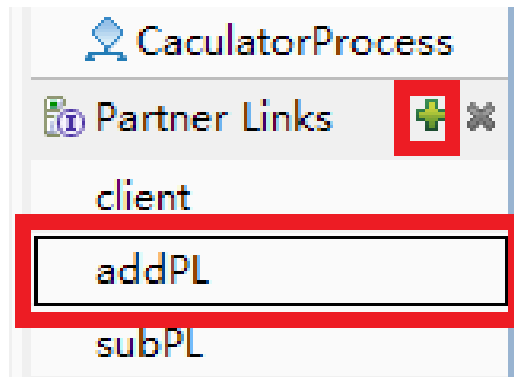
BPEL实例开发

- 3.4 设置伙伴链接

Bpel把涉及其中的所有服务都称之为伙伴链接。**Bpel**对应的**wsdl**本身也是一个伙伴链接。系统已经自动生成了。对于流程中要用到的加法服务，减法服务，应该生成对应的伙伴链接。

3.4 设置伙伴链接

在最右边的Partner Links中，增加两个链接：**addPL**，**subPL**，分别对应加法服务和减法服务。下面具体讲下**addPL**的配置。



3.4 设置伙伴链接（设置过程如下图所示）

1) 点选上addPL，在属性视图中Details页面中点Browser..按钮，打开一个对话框。

2) 点Add WSDL把用到的两个WSDL即AddService.wsdl和SubService.wsdl加进来

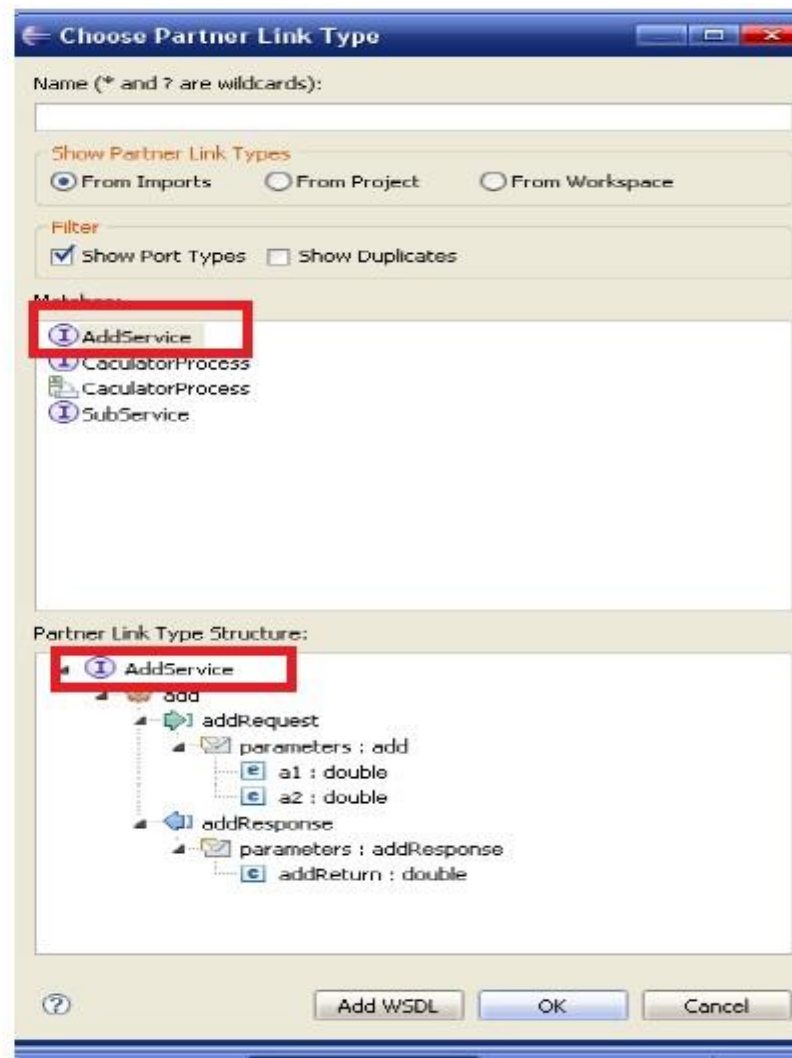
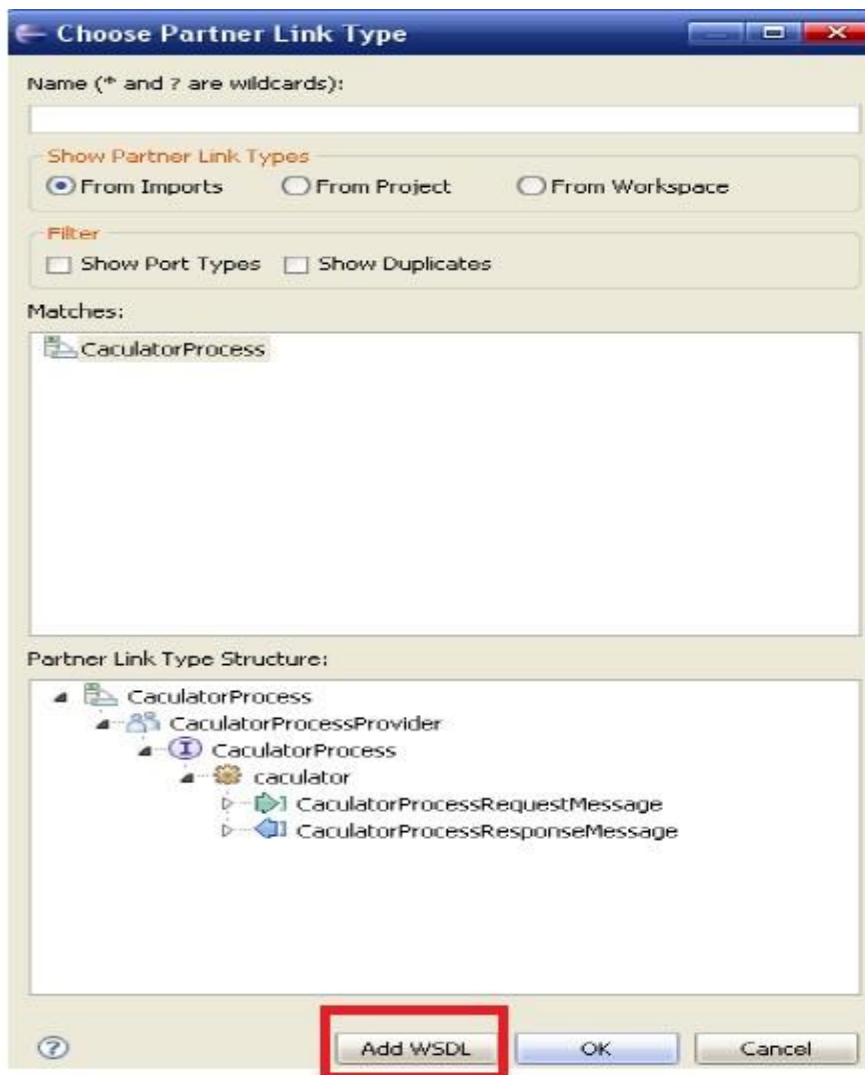
（这两个WSDL一定是BPEL工程里面的WSDL）

，这样会在CaculatorProcessArtifacts.wsdl文件中生成两条对应的<import>语句。

```
<bpel:import namespace="http://sub.example.ws" location="SubService.wsdl" importType="http://schemas.xmlsoap.org/wsdl/" />
<bpel:import namespace="http://add.example.ws" location="AddService.wsdl" importType="http://schemas.xmlsoap.org/wsdl/" />
```

加完后这个对话框也会把这两个wsdl文件中的接口都显示出来。

3.4 设置伙伴链接

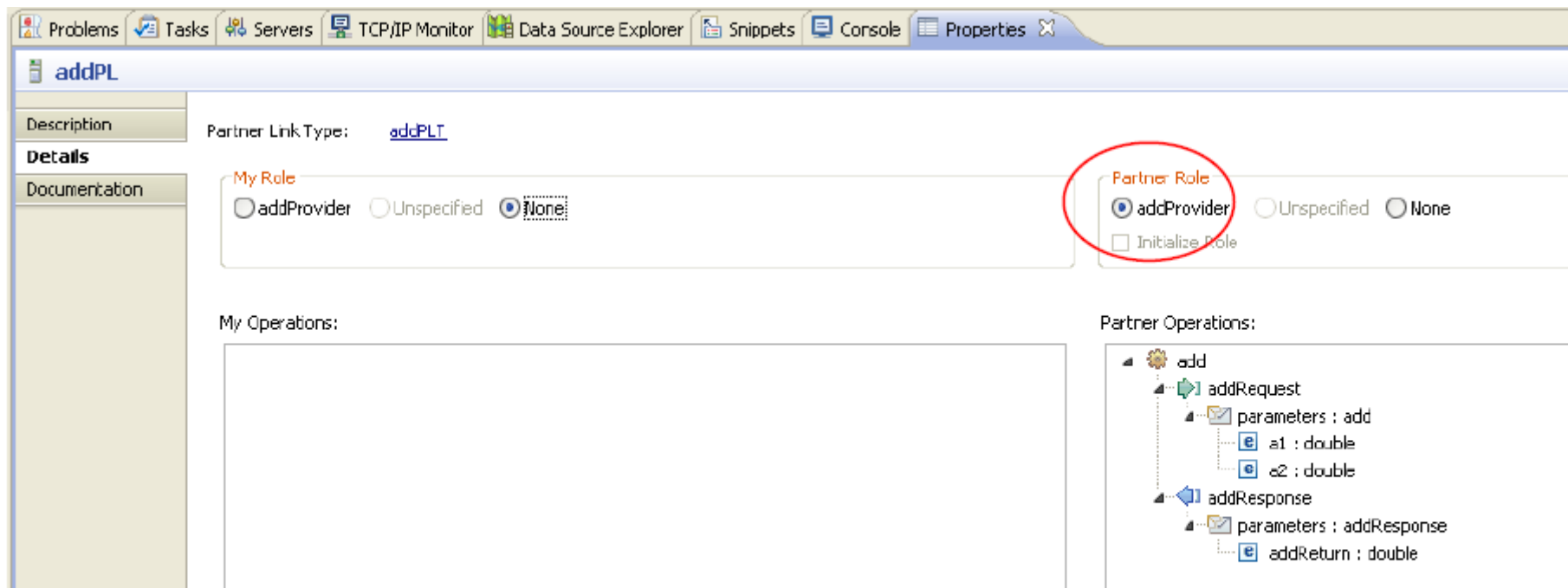


3.4 设置伙伴链接

一定要记住，所谓的伙伴链接一定是针对某个**wSDL**文件的某个接口而言的。 选上**AddService**接口，点**Ok**,弹出伙伴链接类型定义，把名字取为**addPLT**，点**Next**。要求输入**Role Name**，这里输入“**addProvider**”，点**Finish**完成。同时在属性视图中把**PartnerRole**选为**addProvider**

3.4 设置伙伴链接

ParterRole选为addProvider—这一步很重要



同样的方式配置 subPL，只不过把名字分别换成 subPLT，subProvider.，同时别忘记

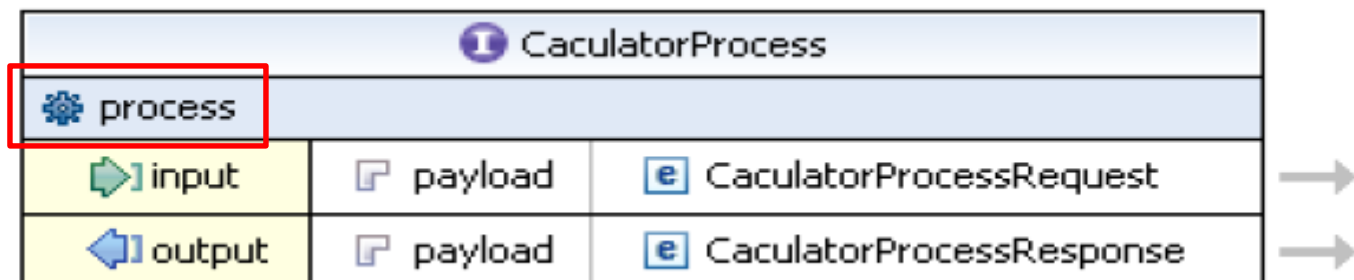
Partner Role 要选上 subProvider.

3.5修改**CaculatorProcessArtifacts.wsdl**文件

主要是改输入输出，使之符合我们的要求。

3.5.1

用open with WSDL Editor打开该文件，转到设计视图里. 可以看到，默认生成的WSDL很简单，只有一个操作**process**, 该操作有输入**input**，输出**output**. 如下图所示。



3.5.2 (初学者我的建议是不改这里的操作名) 为了让操作名更有意义，首先把操作名process改为caculator,可以在图上改也 可以在属性视图里改。(一定要注意: xxx.bpel文件默认的receive operation是process，这里如果改成了caculator，xxx.bpel的代码也要改，xxx.bpel文件中operation一共有两处，都要改成和你所改的一致)

这里是改的第一处

```
<bpel:receive name="receiveInput" partnerLink="client"
  portType="tns:CaculatorProcess"
  operation="process" variable="input"
  createInstance="yes"/>
```

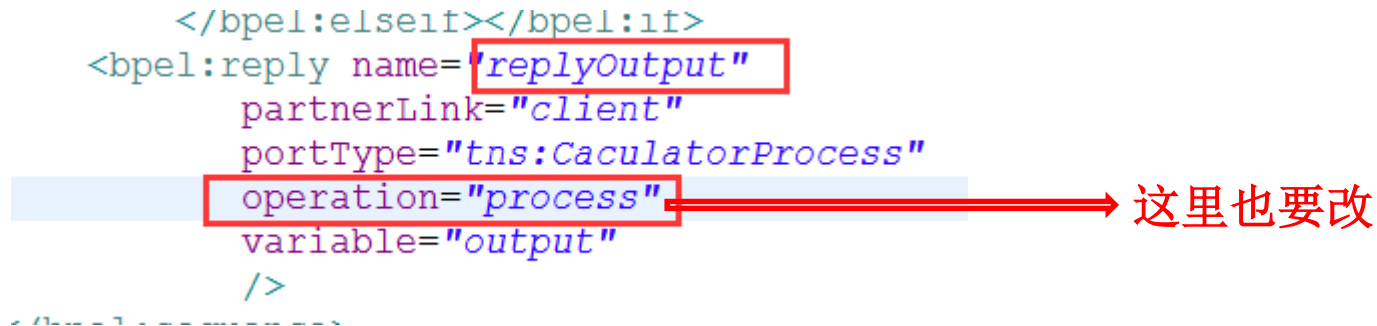
xxx.bpel文件默认的operation是process

改成

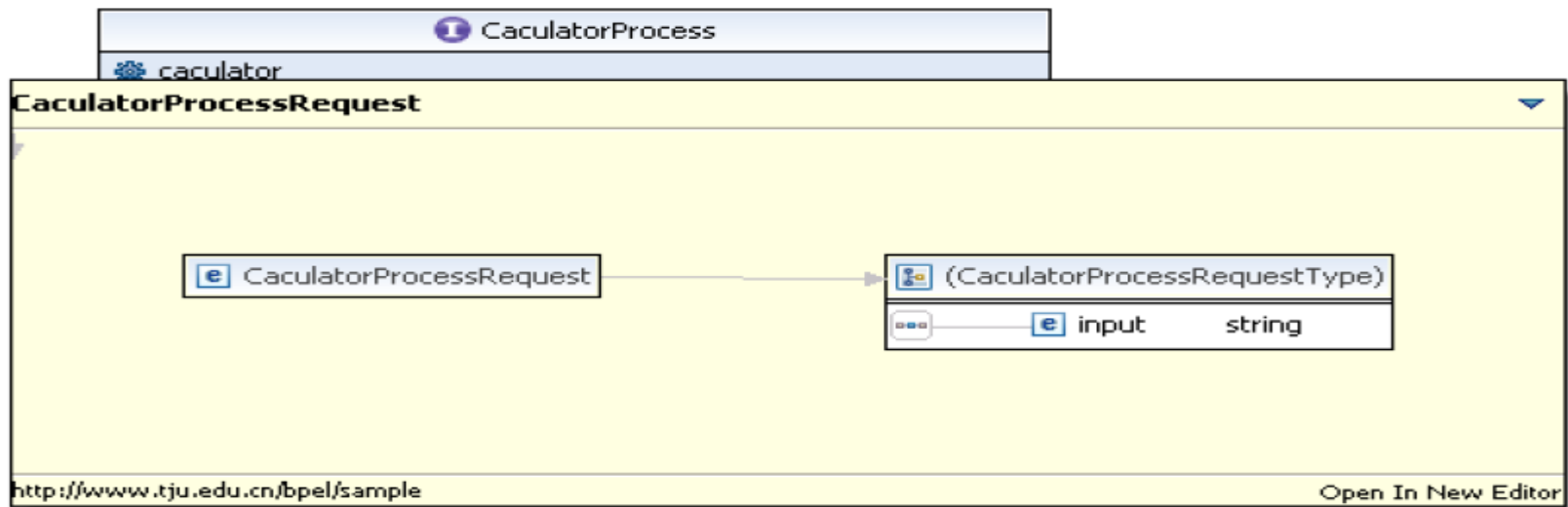
```
<bpel:receive name="receiveInput" partnerLink="client"
  portType="tns:CaculatorProcess"
  operation="caculator" variable="input"
  createInstance="yes"/>
```

这里是改的第二处

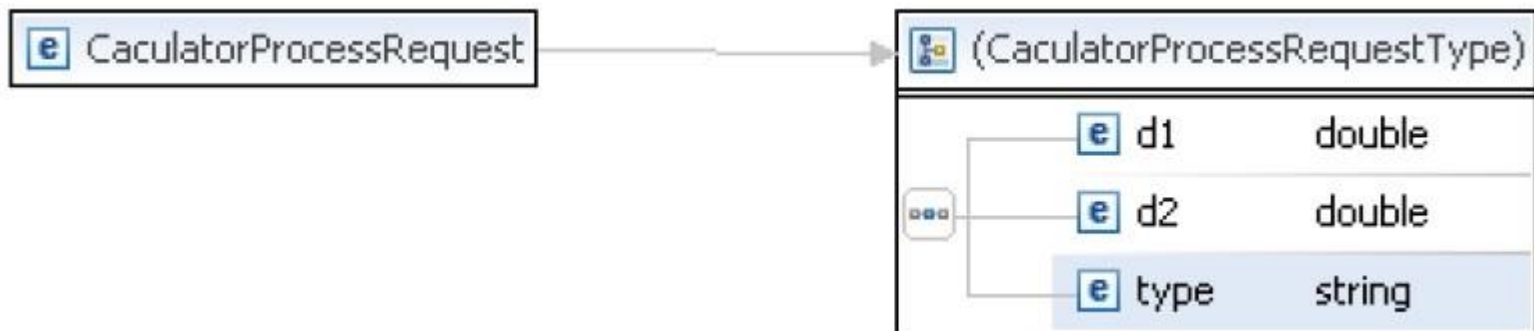
```
        </bpel:elseif></bpel:if>
        <bpel:reply name="replyOutput"
            partnerLink="client"
            portType="tns:CalculatorProcess"
            operation="process"
            variable="output"
        />
    </bpel:reply>
</bpel:process>
```



3.5.3 当把鼠标移到输入输出所在行右边的 箭头时，可以查看输入输出消息的详细定义。下图是输入消息：



可以看到该输入消息只有一个元素input，和我们的要求是不符合的，点击右下角的Open In New Editor, 打开一个新的Editor对输入消息进行配置。进去后选中类型，右键，就可以增加元素，并设置元素的类型，结果如下。



咱们要做的流程是接受三个参数，前两个是**double**型数据，第三个表示是调用加法服务还是减法服务，当**type='add'**，时，调用加法服务，当**type='sub'**时调用减法服务。同样的方法，改一下输出，把类型改为**double**。

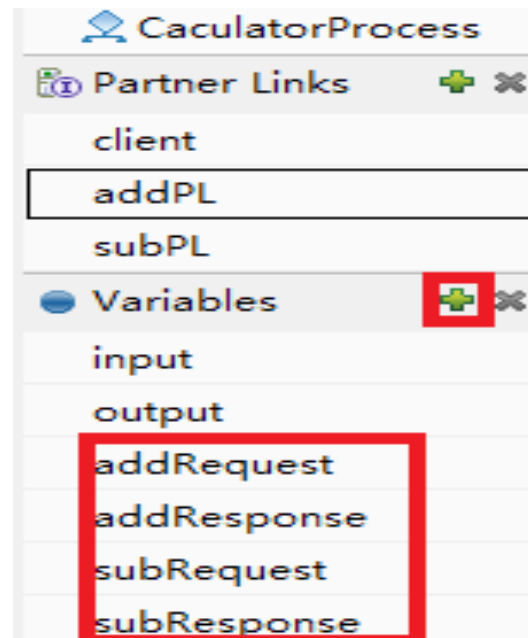


3.6创建调用服务时需要的变量（两种方法，推荐第二种）

第一种：

每次invoke一个服务时都要有对应的变量，同时要进行赋值。我们这里要调用加法服务和减法服务。因此需要额外加入addRequest,addResponse,subRequest,subResponse。

加完后的效果如下：



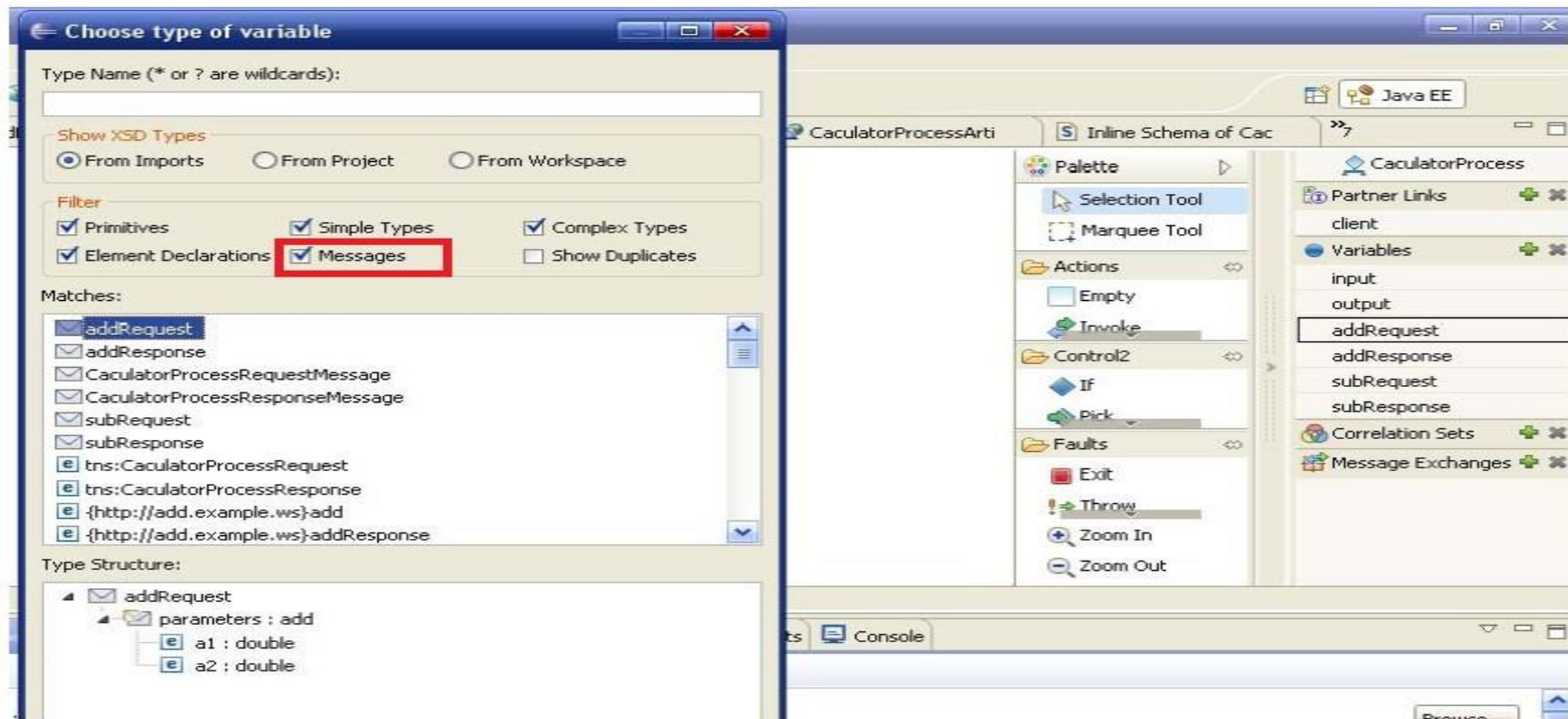
3.7配置变量

- Input,output变量都自动配置好了，对应CaculatorProcessArtifacts.wsdl定义的消息。下面以addRequest为例讲解配置过程。
- 点上addRequest变量，属性视图中切换到Details页，点右上角的Browse..按钮，打开类型选择对话框。addRequest是加法服务的输入，因此把它的类型设为加法服务里的输入定义就可以了。

3.7配置变量

- 一个变量的类型可以是简单类型，也可以是复杂类型，还可以是wsdl中的Message定义。简单起见，我们这里就把addRequest的类型设为AddService服务中的addRequest Message

3.7配置变量

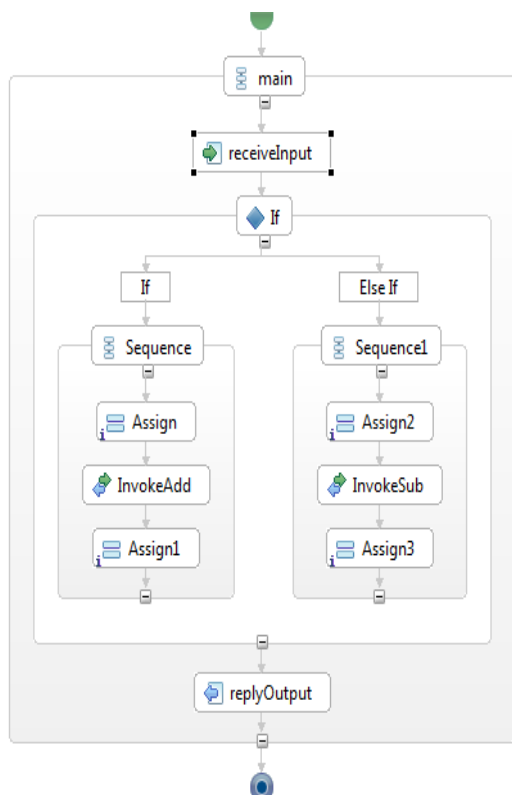


点ok即完成变量的定义，同样的方式定义其它三个变量。至此，准备工作就算是做好了。下一步开始真正的bpel流程设计。

第二种： 注意：对于变量的配置，可以在BPEL的项目流程图的invoke自动生成，不需要上述这么麻烦，具体过程如下：

第一步：项目流程图画完以后，选择invokeadd

第二步：在右侧QuickPick中，单击addRequest，自动生成变量的配置，同时addResponse变量也会自动生成



InvokeAdd
Description
Details
Join Behavior
Correlation
Namespaces
Documentation

Partner Link: addPL
Operation: add

Quick Pick:
addPL
1 AddService
addRequest
parameters: add
addResponse
parameters: addResponse
subPL
SubService
sub
subRequest
parameters: sub
subResponse
parameters: subResponse

Invoke设置完成后，bpel文件会加入红框所示两条语句，可以理解为invoke的活动ns标识

```
<bpel:process name="CaculatorProcess"
  targetNamespace="http://www.pku.edu.cn/bpel/sample"
  suppressJoinFailure="yes"
  xmlns:tns="http://www.pku.edu.cn/bpel/sample"
  xmlns:bpel="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
  xmlns:ns="http://add.example.ws"
  xmlns:ns0="http://sub.example.ws">
```

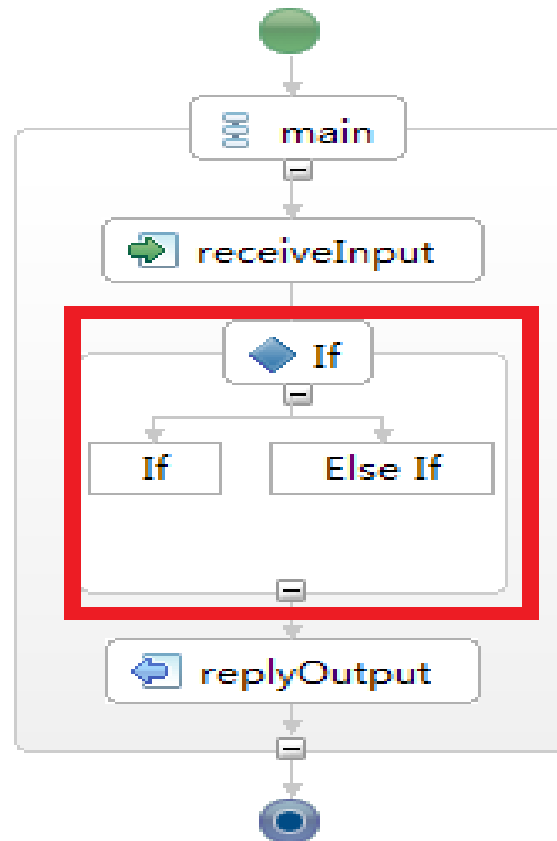
注意：如果此处是VxBPEL，还应该加入
xmlns:vxbpel=<http://vxbpel.rug.org>
例如：

```
<bpel:process name="CarAssembleProcess"
  targetNamespace="cn.edu.ustb.saas"
  suppressJoinFailure="yes"
  xmlns:tns="cn.edu.ustb.saas"
  xmlns:vxbpel="http://vxbpel.rug.org"
  xmlns:bpel="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
  xmlns:ns="http://C.wiring.saas.ustb.edu.cn"
  xmlns:ns0="http://C.tank.saas.ustb.edu.cn"
  xmlns:ns1="http://C.interior.saas.ustb.edu.cn"
  xmlns:ns2="http://A.console.saas.ustb.edu.cn"
  xmlns:ns3="http://A.seat.saas.ustb.edu.cn"
  xmlns:ns4="http://A.steeringwheel.saas.ustb.edu.cn"
```


3.8 BPEL流程设计

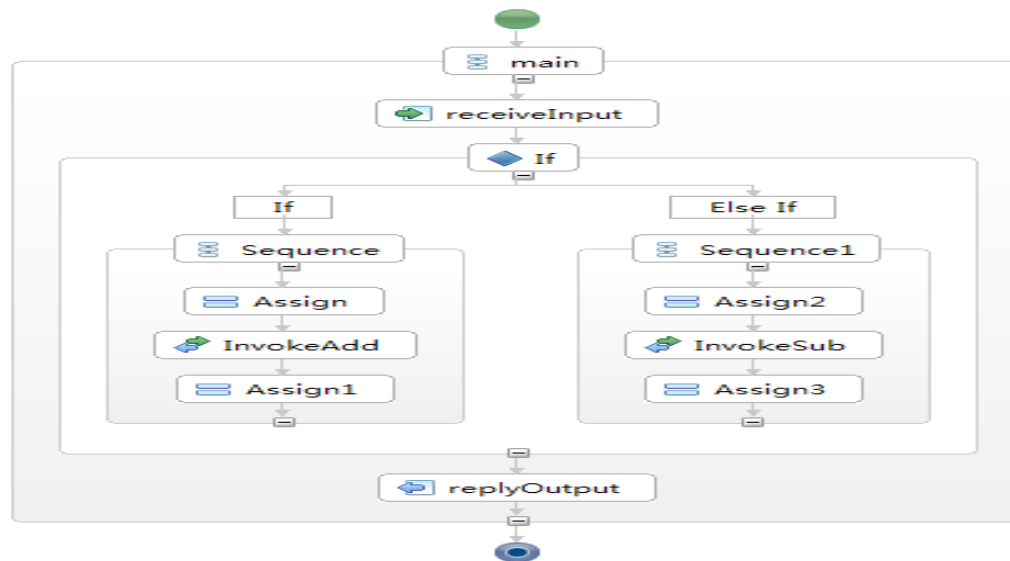
- 根据需要，我们首先是根据输入的第三个变量的值来决定调用加法服务还是减法服务，所以，显然需要一个条件**IF**语句，拖到**receiveInput**后面
- 选中刚建好的**if**模块，点右键，依次点**Add Elself**。基本结构如下

3.8 BPEL流程设计



3.8 BPEL流程设计

- 思路应该比较清晰，如果type='add'，则调用加法服务，如果type='sub'则调用减法服务。进一步完善后，如下。



3.8 BPEL流程设计

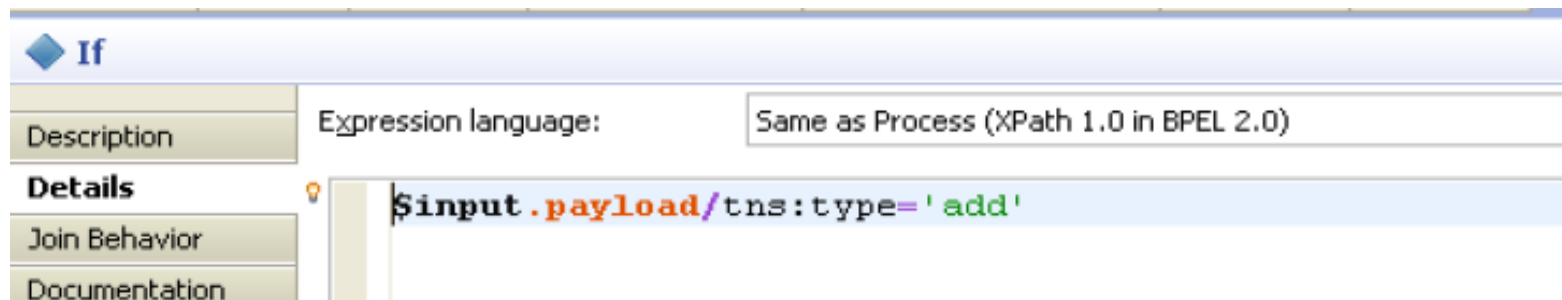
接下来就是对各个结点进行具体的配置了。

3.8.1 If语句的配置选择If,在属性视图中点开Details页,做如下配置: 输入条件语句:

`$input.payload/tns:type='add'`

(这里的**type**是前面在修CalculatorProcessArtifacts.wsdl文件的输入输出  中的**type**一致, 如果这里**type**改成了**typee**, 那么**if**的输入条件语句也要改成**`$input.payload/tns:typee='add'`**)

以**`$input.payload/tns:type='add'`**为例解析这句代码: 首先**\$**是固定字符串, **Input**是



以\$input.payload/tns:type='add'为例解析这句代码:

1.首先\$是固定字符串, 接下来访问 input

2.input是<bpel:variables>标签里面的变量 (variables存放了输入input、输出output、其他partnerlinkwsdl里面的变量)

3.在解析到变量input时, messageType="tns:CaculatorProcessRequestMessage"

```
<bpel:variables>
  <!-- Reference to the message passed as input during initiation -->
  <bpel:variable name="input"
    messageType="tns:CaculatorProcessRequestMessage"/>
```

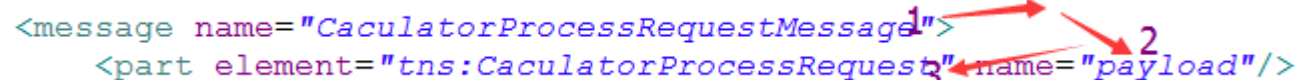
4.接下来根据tns访问到xmlns:tns=<http://eclipse.org/bpel/sample>

5.根据tns的命名空间访问到import的import标签, 其location="CaculatorProcessArtifacts.wsdl"

6.根据location (代表位置), 访问CaculatorProcessArtifacts.wsdl文件, 访问文件中的CaculatorProcessRequestMessage (上图)

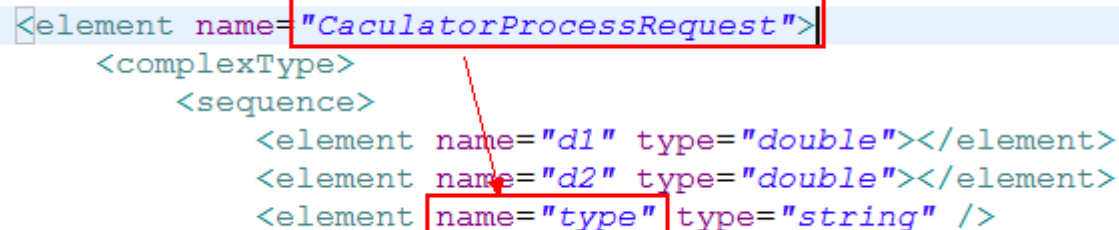
7.根据payload访问到其元素element="tns:CaculatorProcessRequest "

```
~~~~~  
<message name="CaculatorProcessRequestMessage"1>  
  <part element="tns:CaculatorProcessRequest"3 name="payload"/>  
~~~~~
```



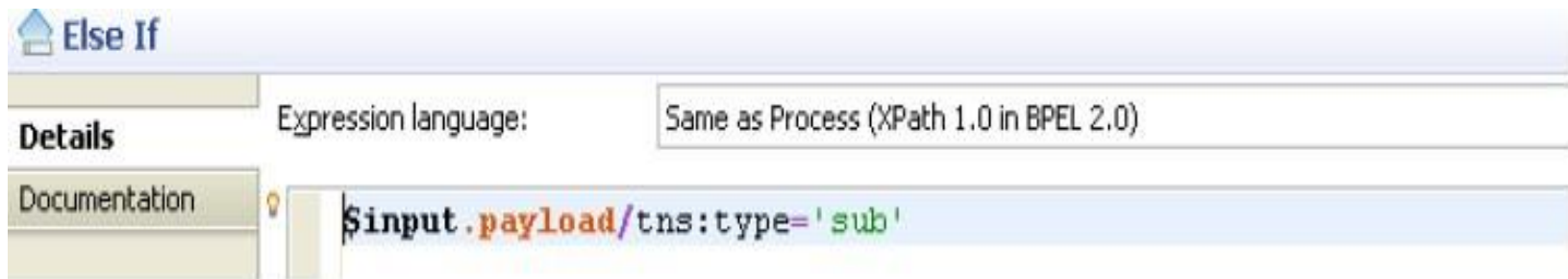
8.最后根据CaculatorProcessRequest访问到name="type"的元素

```
<element name="CaculatorProcessRequest">  
  <complexType>  
    <sequence>  
      <element name="d1" type="double"/>  
      <element name="d2" type="double"/>  
      <element name="type" type="string" />  
    </sequence>  
  </complexType>  
</element>
```



3.8 BPEL流程设计

- 同样，选择Else If

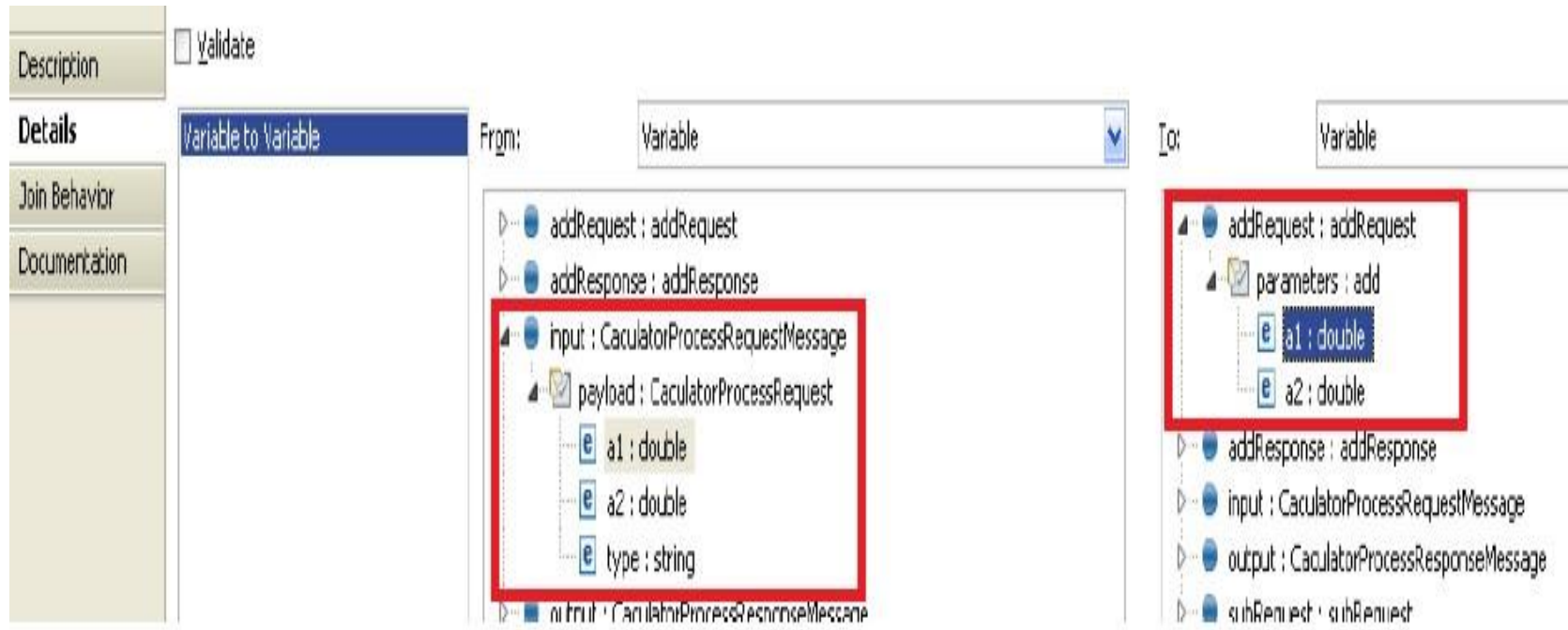


3.8 BPEL流程设计

3.8.2 Assign语句的设置

- 四个Assign语句的配置过程都一样，这里讲解下第一个Assign的配置。
- 第一个Assign的作用是把输入变量input中的前两个参数传给addRequest变量。需要注意的是，**BEPL**中的变量赋值只能是一个一个的赋，不能直接把**input**赋给**addRequest**（当然这里也不匹配，即使匹配也不行），而只能把**input**下的参数挨个赋给**addRequest**..

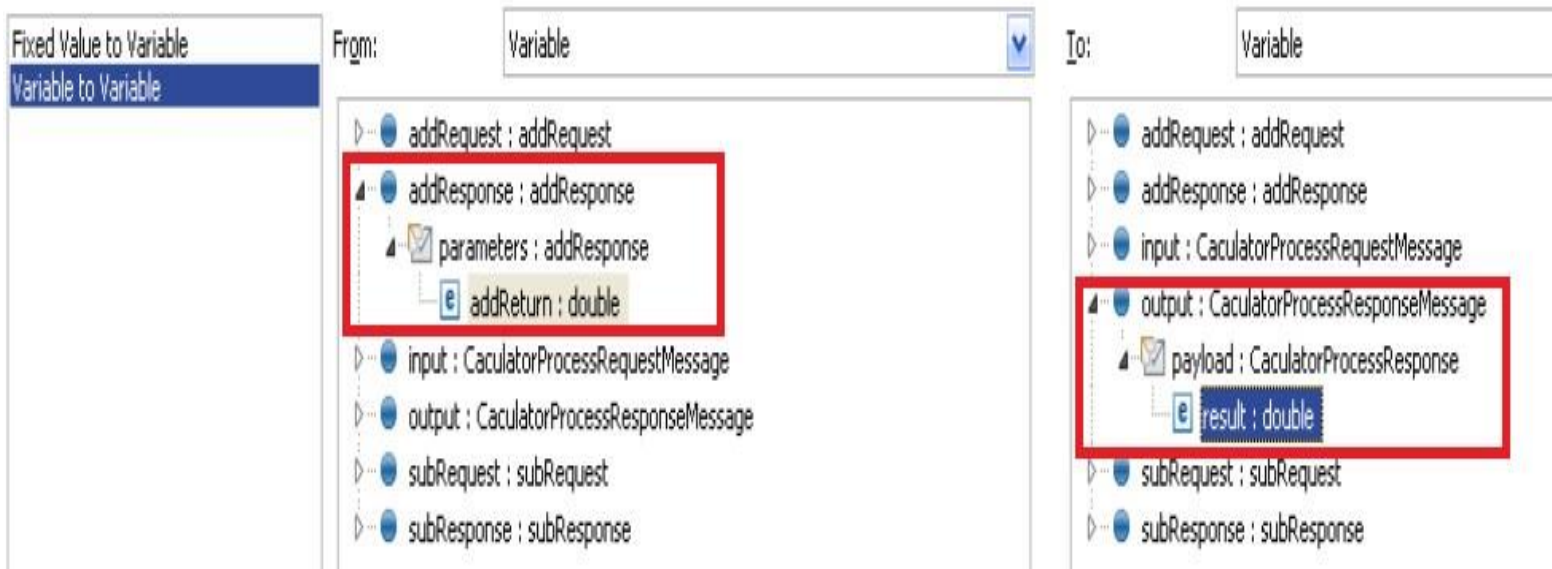
3.8 BPEL流程设计



意思应该很明了，把input下的a1赋给addRequest下的a1. 然后会弹出一个对话框，问是否需要初始化addRequest,你点Yes就是了。 再次点New，把input下的a2赋给addRequest下的a2.

3.8 BPEL流程设计

- 对于Assign1，它的作用是把addResponse的值传给流程的输出变量output.如图：



3.8 BPEL流程设计

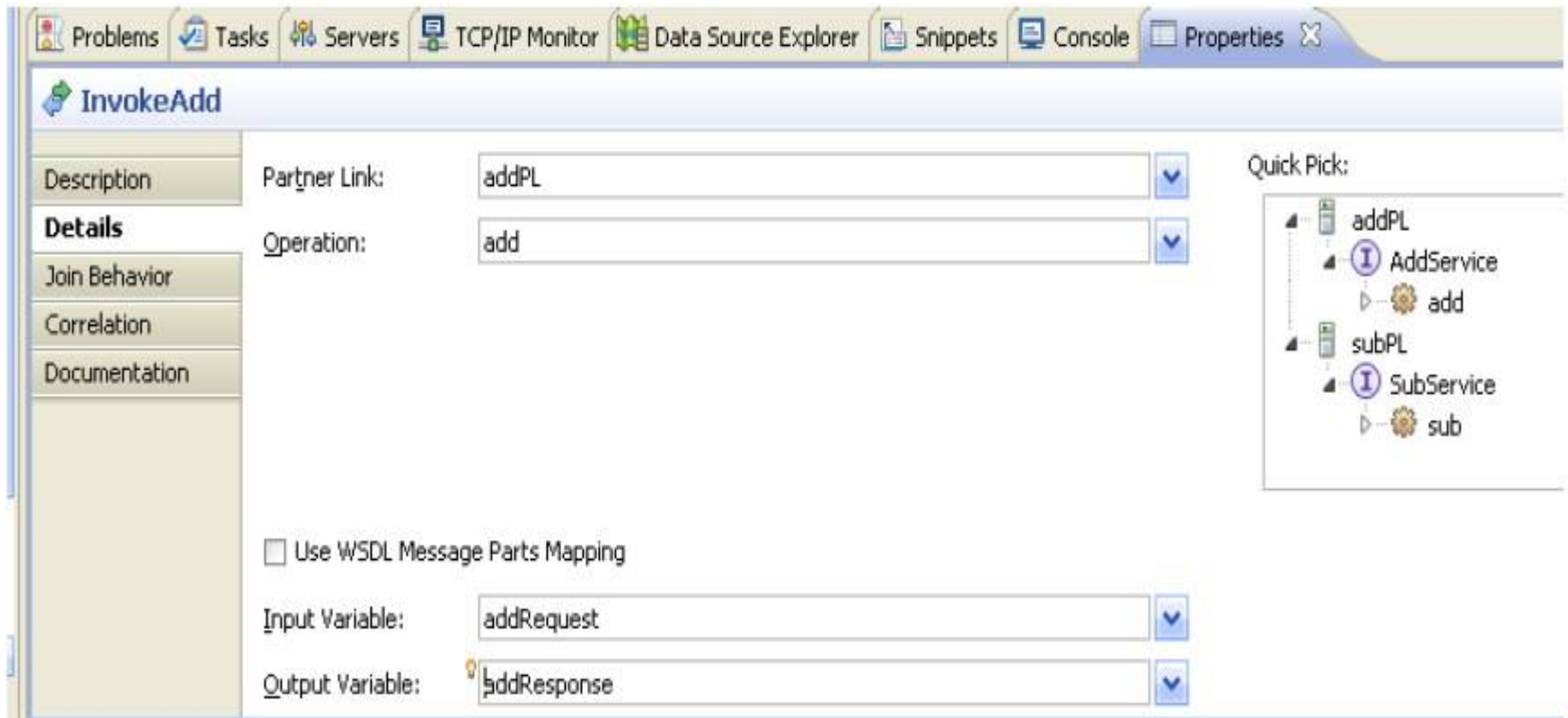
3.8.3 Invoke语句的设置

Invoke的作用是调用伙伴链接对应接口下的对应操作，还需要指定输入输出变量。

选上**InvokeAdd**,同样在属性视图的**Details**页面中进行如下配置：

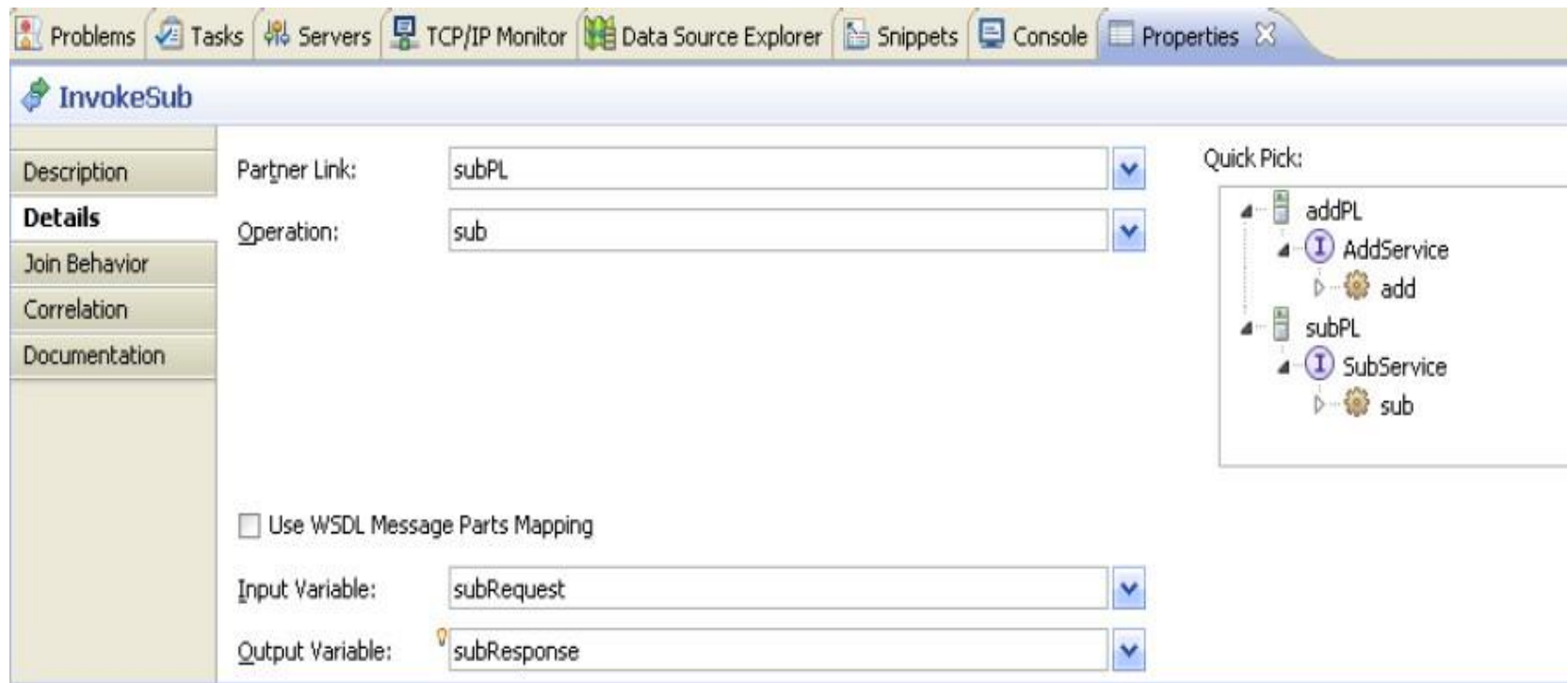
3.8 BPEL流程设计

InvokeAdd的配置，右侧QuickPick单击add下面的addRequest



3.8 BPEL流程设计

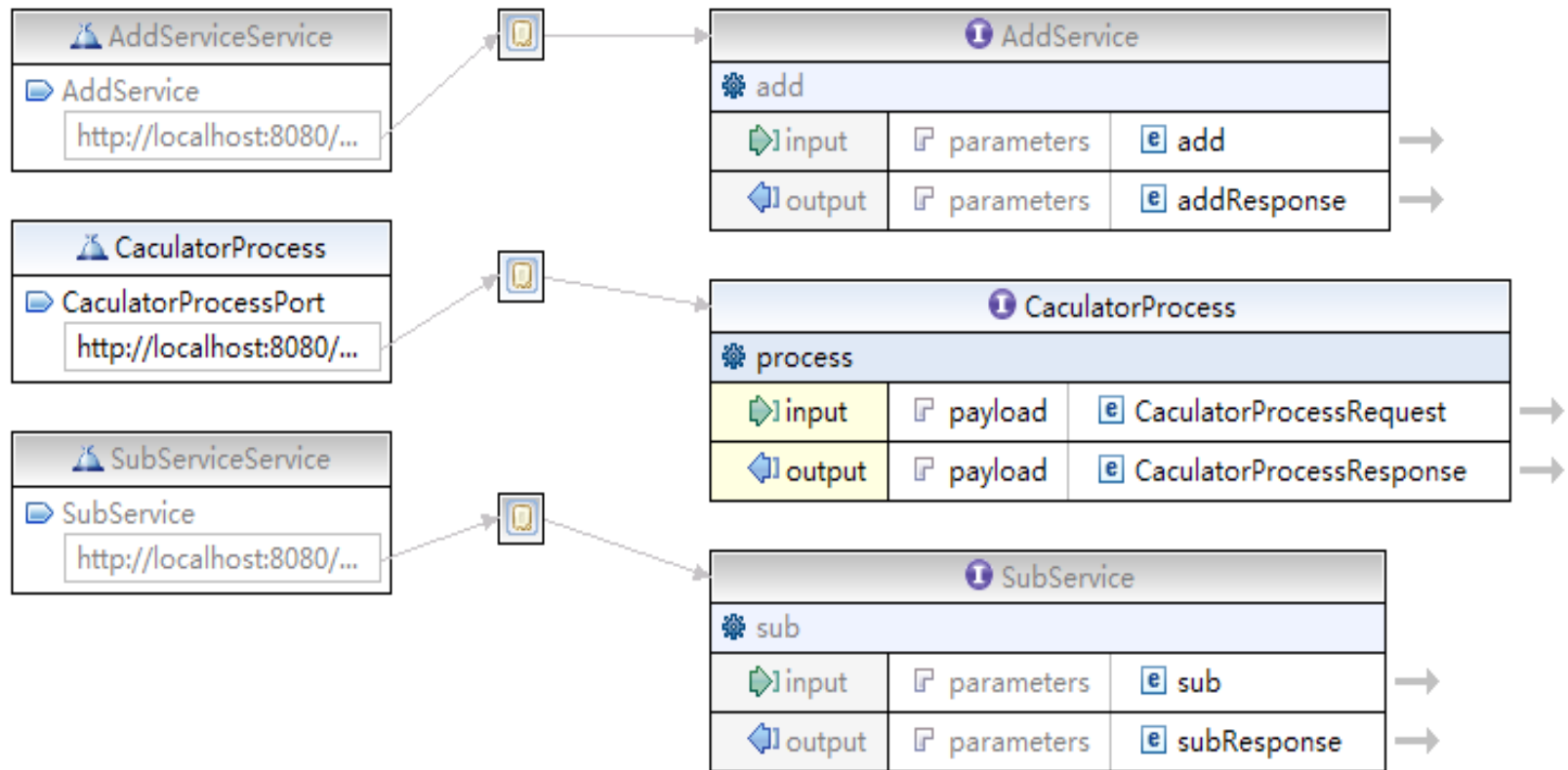
InvokeSub的配置



3.9 流程发布

BPEL流程最终的目的也是发布成一个**web-service**，只不过这个**web**服务自己基本上不实现功能，主要是通过调用别人的功能来完成任务。因此，最后也要把流程对应的**WSDL**发布出去。默认生成的**WSDL**只有类型定义，消息定义，和操作定义，缺少绑定信息和服务定义。

3.9.1 CaculatorProcessArtifacts.wSDL文件



从图中可以看到，`import`进来的wSDL也会显示在图形视图中。

3.9.2 创建发布ODE发布文件

- 在项目名上选择File->New Other BPEL 2.0 Apache ODE Deployment Descriptor, 直接点Finish即会生成一个deploy.xml文件
- 打开并配置伙伴链接的端口

Process CaculatorProcess - http://www.pku.edu.cn/bpel/sample

▼ General

This process is activated

☐ Run this process in memory

▼ Inbound Interfaces (Services)

The table contains interfaces the process provides. Specify the service, port and binding you want to use for each PartnerLink listed

Partner Link	Associated Port	Related Service	Binding Used
client	CaculatorProcessPort	{http://www.pku.edu.cn/bpel/sample}CaculatorProcess	CaculatorProcessBinding

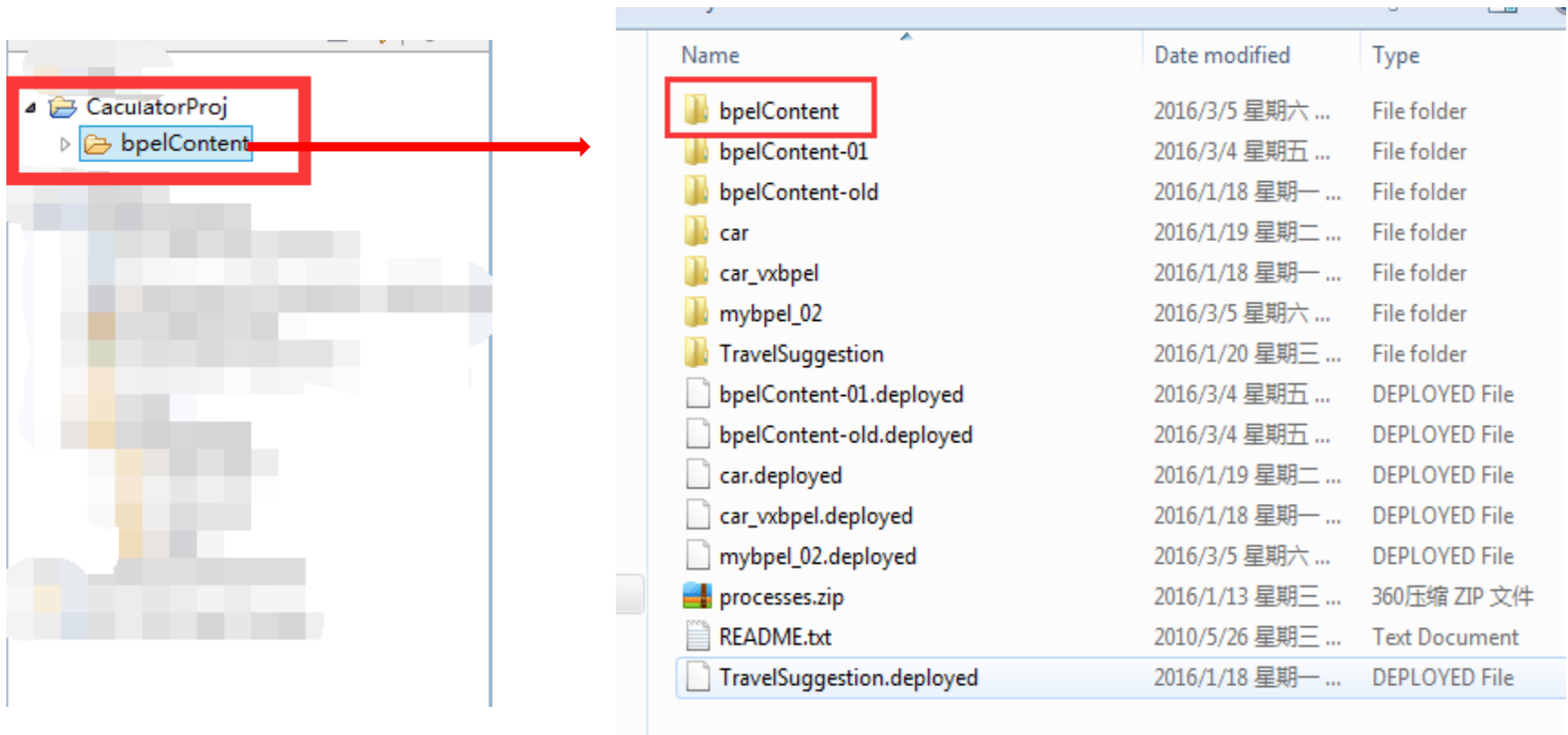
▼ Outbound Interfaces (Invokes)

The table contains interfaces the process invokes. Specify the service, port and binding you want to use for each PartnerLink listed

Partner Link	Associated Port	Related Service	Binding Used
addPL	AddService	{http://add.example.ws}AddServiceService	AddServiceSoapBinding
subPL	SubService	{http://sub.example.ws}SubServiceService	SubServiceSoapBinding

CaculatorProcess

部署到ODE当中，也就是将工程下的项目
（我这里是bpelContent）全部复制到
`ode\WEB-INF\processes`路径下



因为你是发布到ODE引擎下面的xxxxxxxArtifacts.wsdl中的address地址默认的是<http://localhost:8080/CaculatorProcess>，这里要改成

<http://localhost:8080/ode/processes/CaculatorProcess>

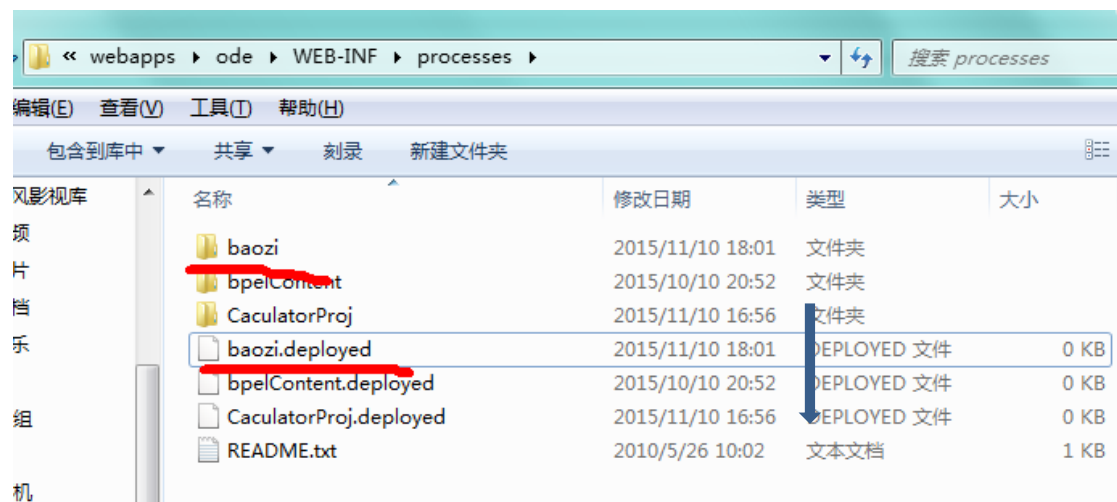
```
-----  
<service name="CaculatorProcessService">  
  <port binding="tns:CaculatorProcessBinding" name="CaculatorProcessPort">  
    <soap:address location="http://localhost:8080/ode/processes/CaculatorProcess" />  
  </port>  
</service>  
</definitions>
```

四、BPEL运行与测试

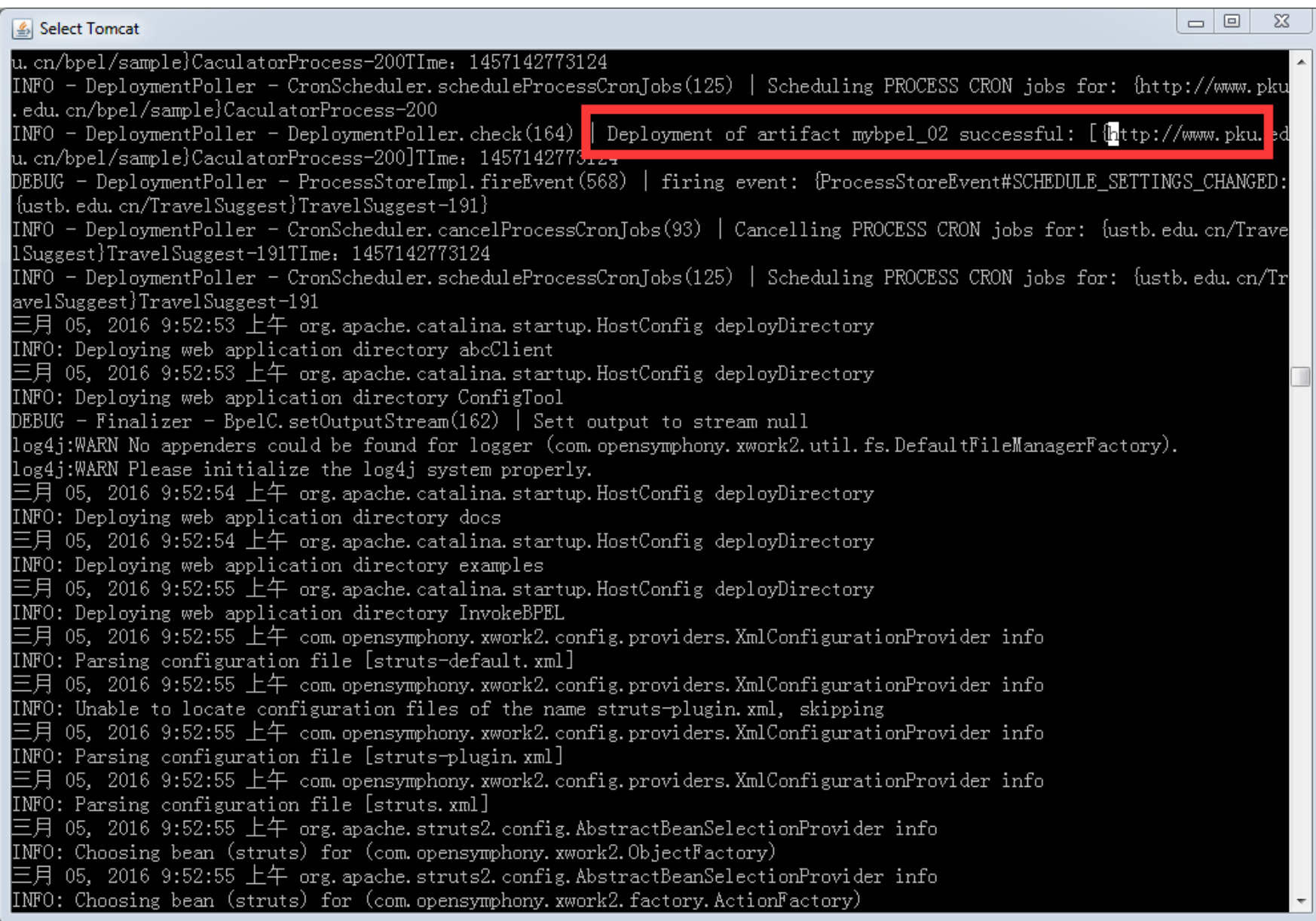
4.1 启动ODE服务器 讲BPEL工程部署到ODE服务器上， Console 信息如下图

```
14:15:10,847 INFO [BpelServerImpl] BPEL Server Started.  
14:15:10,862 INFO [SystemSchedulesConfig] SYSTEM CRON configuration: C:\Tomcat 6.0\webapps\ode\WEB-INF\conf\schedules.xml  
14:15:11,572 INFO [DeploymentPoller] Poller started.  
14:15:11,572 INFO [ODEServer] Process deployment polling started on path C:\Tomcat 6.0\webapps\ode\WEB-INF\processes.  
14:15:11,572 INFO [ODEServer] ODE Service Engine has been started.  
14:15:11,577 INFO [CronScheduler] Cancelling PROCESS CRON jobs for: {http://www.pku.edu.cn/bpel/sample}CaculatorProcess-8  
14:15:11,582 INFO [CronScheduler] Scheduling PROCESS CRON jobs for: {http://www.pku.edu.cn/bpel/sample}CaculatorProcess-8  
14:15:11,582 INFO [CronScheduler] Cancelling PROCESS CRON jobs for: {http://ode/bpel/unit-test>HelloWorld2-1  
14:15:11,582 INFO [CronScheduler] Scheduling PROCESS CRON jobs for: {http://ode/bpel/unit-test>HelloWorld2-1
```

文件复制到tomcat的路径下以后会自动生成deployed文件，标识部署成功



ODE引擎部署成功，console信息如下图，红色框里显示就是部署成功的标致



```
Select Tomcat
u.cn/bpel/sample}CaculatorProcess-200Time: 1457142773124
INFO - DeploymentPoller - CronScheduler.scheduleProcessCronJobs(125) | Scheduling PROCESS CRON jobs for: {http://www.pku.edu.cn/bpel/sample}CaculatorProcess-200
INFO - DeploymentPoller - DeploymentPoller.check(164) | Deployment of artifact mybpel_02 successful: [http://www.pku.edu.cn/bpel/sample]CaculatorProcess-200Time: 1457142773124
DEBUG - DeploymentPoller - ProcessStoreImpl.fireEvent(568) | firing event: {ProcessStoreEvent#SCHEDULE_SETTINGS_CHANGED: {ustb.edu.cn/TravelSuggest}TravelSuggest-191}
INFO - DeploymentPoller - CronScheduler.cancelProcessCronJobs(93) | Cancelling PROCESS CRON jobs for: {ustb.edu.cn/TravelSuggest}TravelSuggest-191Time: 1457142773124
INFO - DeploymentPoller - CronScheduler.scheduleProcessCronJobs(125) | Scheduling PROCESS CRON jobs for: {ustb.edu.cn/TravelSuggest}TravelSuggest-191
三月 05, 2016 9:52:53 上午 org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory abcClient
三月 05, 2016 9:52:53 上午 org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory ConfigTool
DEBUG - Finalizer - BpelC.setOutputStream(162) | Set output to stream null
log4j:WARN No appenders could be found for logger (com.opensymphony.xwork2.util.fs.DefaultFileManagerFactory).
log4j:WARN Please initialize the log4j system properly.
三月 05, 2016 9:52:54 上午 org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory docs
三月 05, 2016 9:52:54 上午 org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory examples
三月 05, 2016 9:52:55 上午 org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory InvokeBPEL
三月 05, 2016 9:52:55 上午 com.opensymphony.xwork2.config.providers.XmlConfigurationProvider info
INFO: Parsing configuration file [struts-default.xml]
三月 05, 2016 9:52:55 上午 com.opensymphony.xwork2.config.providers.XmlConfigurationProvider info
INFO: Unable to locate configuration files of the name struts-plugin.xml, skipping
三月 05, 2016 9:52:55 上午 com.opensymphony.xwork2.config.providers.XmlConfigurationProvider info
INFO: Parsing configuration file [struts-plugin.xml]
三月 05, 2016 9:52:55 上午 com.opensymphony.xwork2.config.providers.XmlConfigurationProvider info
INFO: Parsing configuration file [struts.xml]
三月 05, 2016 9:52:55 上午 org.apache.struts2.config.AbstractBeanSelectionProvider info
INFO: Choosing bean (struts) for (com.opensymphony.xwork2.ObjectFactory)
三月 05, 2016 9:52:55 上午 org.apache.struts2.config.AbstractBeanSelectionProvider info
INFO: Choosing bean (struts) for (com.opensymphony.xwork2.factory.ActionFactory)
```

4.2测试BPEL

The screenshot displays the Eclipse IDE interface. On the left, the Project Explorer shows a project named 'CaculatorProj' with a sub-project 'bpelContent'. Inside 'bpelContent', there are files: 'AddService.wsdl', 'CaculatorProcess', 'CaculatorProcess', 'CaculatorProcess', 'deploy.xml', and 'SubService.wsdl'. A context menu is open over the 'CaculatorProcess' file, listing various actions like 'New', 'Open', 'Copy', 'Paste', 'Delete', 'Import...', 'Export...', 'Refresh', 'Validate', 'Run As', 'Debug As', 'Profile As', 'Team', 'Compare With', 'Replace With', 'Web Services', 'Source', 'WikiText', and 'Properties'. The 'Web Services' option is selected, and a sub-menu is visible with options: 'Test with Web Services Explorer' (highlighted with a red box), 'Publish WSDL File', 'Generate Java Bean Skeleton', 'Generate Client', and 'Generate WSIL'.

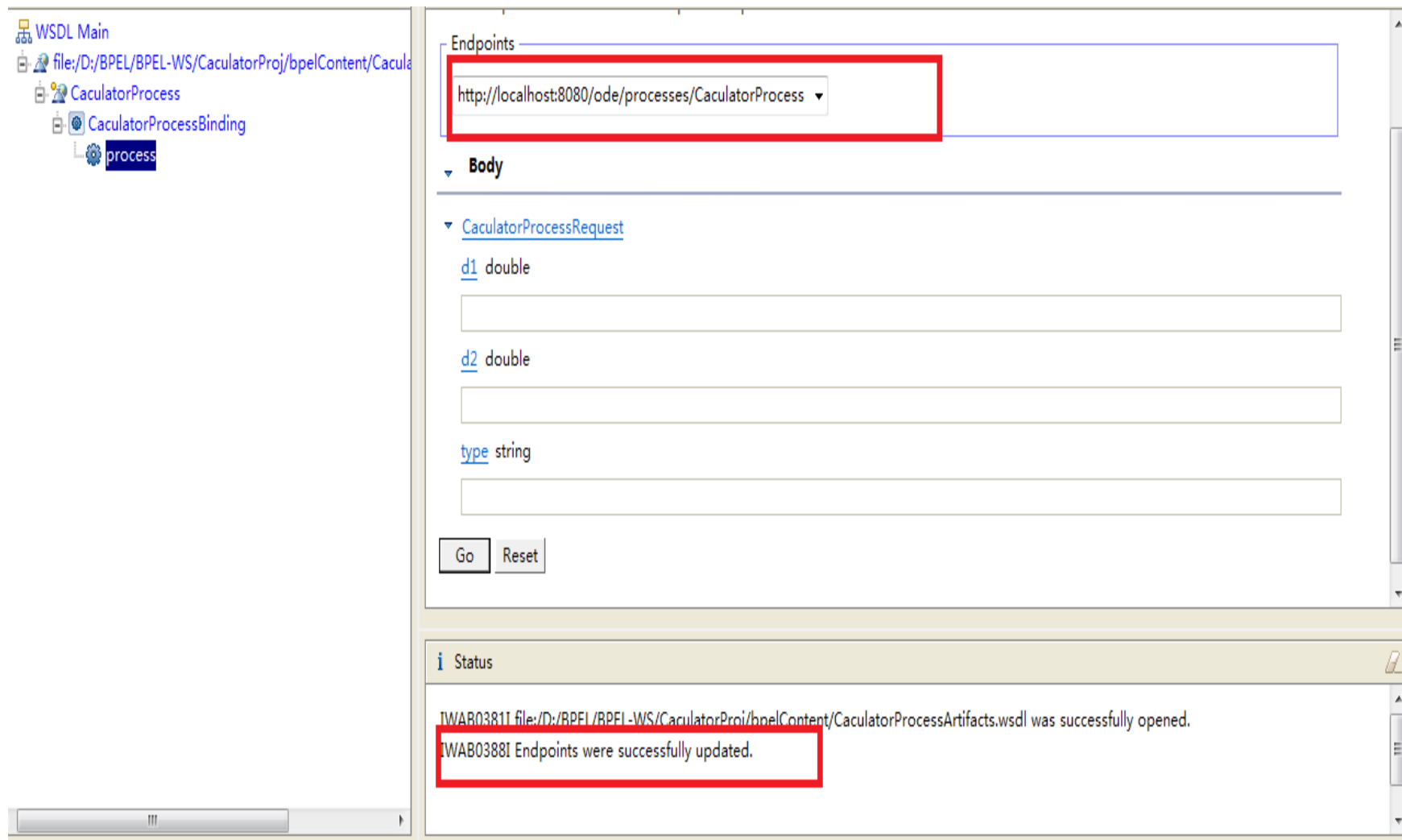
The main editor area shows the 'WSDL Binding Details' for a SOAP binding. It includes a table for 'Operations' and a section for 'Endpoints'.

Name	Documentation
process	--

Below the table, there is a section for 'Endpoints' with 'Add' and 'Remove' buttons.

The 'Status' bar at the bottom shows the file path: 'IWAB0381I file:/D:/BPEL/BPEL-WS/CaculatorProj/bpelContent/CaculatorProcessArtifact'.

The bottom status bar shows the date and time: '2010-11-12 下午02:14:40'.



4.2.1 测试加法

Actions

Endpoints

http://localhost:8080/ode/processes/CaculatorProcess

Body

CaculatorProcessRequest

d1 double

3

d2 double

5

type string

add

Go

Reset

Status

Body

CaculatorProcessResponse

result (double): 8.0

参考文献

- 1) HelloWorld-BPELDesignerAndODE.pdf
- 2) 一起学BPEL实例教程一V1.0.pdf
- 3) ODE 官方网站 <http://ode.apache.org/>
- 4) Eclipse BPEL Designer
<http://www.eclipse.org/bpel/install.php>