

# 第6-2章作业

SOA - 22920212204392黄勛

## 1 什么是服务组合编程？

服务组合编程是用某种特定的服务组合编程语言把服务组织成能完成自动业务过程的程序。Web 服务的相关服务组合编程语言有：BPEL、BPML、WSFL、Xlang，其中 BPEL 是目前流行的 Web 服务组合语言，也是事实上的基于 web 服务开发可执行业务过程的业界标准。

## 2 简述一个BPEL程序的结构组成。

BPEL (Business Process Execution Language) 是一种用于描述业务流程的标准化语言。BPEL 程序的结构主要包括以下几个组成部分：（主要为三部分内容：合作伙伴链接声明、变量声明和流程定义，其中合作伙伴链接就是合作伙伴链接类型的实例声明）

### 1. 合作伙伴链接声明 (PartnerLink)：

- 在BPEL中，合作伙伴链接用于定义与外部实体（如其他Web服务或业务流程）的交互。合作伙伴链接声明包括合作伙伴链接类型的实例声明，指定了与外部实体之间的通信协议和规则。

```
<partnerLinks>
  <!--The 'client' role represents the requester of this service.-->
  <partnerLink name= "client"
    partnerLinkType= "tns:HelloWorld"
    myRole= "HelloWorldProvider" />
  <partnerLink name= "HelloWorldProviderLink"
    partnerLinkType= "nsxm10:HelloWorldLink"
    partnerRole= "HelloWorldProvider" />
</partnerLinks>
```

### 2. 变量声明 (Variable)：

- BPEL程序中使用的数据通常需要在流程执行过程中进行存储和处理。因此，变量声明用于定义BPEL流程中使用的变量，这些变量可以包含从外部实体接收的数据、在流程中使用的中间结果等。变量声明规定了变量的数据类型和作用域。

```
<variables>
  <!--BPEL程序接收消息和回复消息的变量-->
  <variable name= "input" messageType= "tns:HelloWorldRequestMessage" />
  <variable name= "output" messageType= "tns:HelloWorldResponseMessage" />
  <!--BPEL程序调用HelloWorld Web服务的输入和输出变量-->
  <variable name= "helloWorldResponse"
    messageType= "nsxm10:helloWorldResponse" />
  <variable name= "helloWorldRequest"
    messageType= "nsxm10:helloWorldRequest" />
</variables>
```

### 3. 流程定义 (Process)：

- 流程定义是BPEL程序的内核部分，描述了业务流程的执行逻辑。流程定义包括一系列活动（Activities）和结构化的控制流程。常见的活动包括串行（Sequence）、并行（Parallel）、选择（Switch）、循环（While）、以及调用外部服务的活动（Invoke）等。流程定义通过将这些活动组合起来，定义了业务流程中各个步骤的执行顺序和条件。

```
<sequence name= "main" >
  <!--接收消息语句-->
  <receive name= "receiveInput"
    partnerLink= "client" portType= "tns:HelloWorld"
    operation= "helloWorldProcess" variable= "input" createInstance= "yes" />

  <!--调用HelloWorld Web服务语句-->
  <invoke partnerLink= "HelloWorldPartnerLink"
    portType= "nsxm10:HelloWorld"
    operation= "helloWorld"
    outputVariable= "helloWorldResponse" inputVariable= "helloWorldRequest" />
```

```
<!--把HelloWorld 的返回结果复制到回复变量-->
<assign> <copy>
  <from variable= " helloWorldResponse" part= "helloWorldReturn" />
  <to variable= "output" part= "payload" query= "/tns: HelloWorldResponse
    /tns:result" />
</copy> </assign>
<!--回复语句-->
<reply name= "replyOutput" partnerLink= "client"
  portType= "tns:HelloWorld" operation= "process" variable= "output" />
</sequence>
```

#### 4. 消息（Message）：

- BPEL程序中的消息用于定义流程之间的通信。消息通常与合作伙伴链接相关联，描述了在流程执行过程中如何与合作伙伴进行消息交换。

#### 5. 事件处理器（Event Handlers）：

- BPEL程序可以定义事件处理器，用于处理异常或其他事件。事件处理器包括针对不同类型事件的处理逻辑，如故障处理、消息超时等。

## 第7章作业

### 1 比较服务组合与服务协作的异同。

定义：

#### 1.1 服务组合（Service Composition）：

- **定义：** 服务组合是将多个独立的服务组合在一起以实现更复杂的业务逻辑或功能的过程。这种组合可以是顺序的、并行的，或者二者结合。
- **目标：** 主要目标是将现有的服务组合成更高级别的服务，以满足特定的业务需求。这种组合可能涉及到串联和并联的服务调用，形成一个更大的服务。
- **实现：** 通常通过编排语言或工具来实现，例如BPEL（Business Process Execution Language）或使用工作流引擎。
- **通信方式：** 服务之间的通信通常是同步的，即一个服务的输出作为另一个服务的输入。

## 1.2 服务协作 (Service Collaboration) :

- **定义:** 服务协作是指不同的服务共同协作以完成一个更大的任务或业务流程。每个服务都保持独立性, 但通过协作实现更大的业务目标。
- **目标:** 主要目标是促使各个服务相互协作, 以共同实现一个复杂的业务目标。每个服务可能负责特定的子任务。
- **实现:** 可以通过消息传递、事件驱动等机制实现服务之间的协作。
- **通信方式:** 服务之间的通信通常是异步的, 一个服务通过消息传递触发另一个服务执行相应的操作。

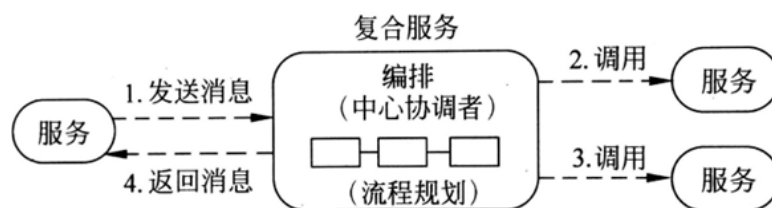
## 1.3 共性

两者都涉及一组服务以及这些服务之间的交互和协作。

## 1.4 差异

### 1. 独立性差异:

- **服务组合:** 在SOA中的服务组合是指以流程的方式完成服务的编排。服务的编排由一个中心协调者(指挥)完成。中心协调者依据流程规划(schema)协调对不同服务的调用, 完成既定业务目标, 参与流程的服务并不知道流程的全貌。



- **服务协作:** 服务协作中并不存在一个中心协调者, 所有服务以对等方式互相协作, 因此每个服务都需要知道和说明自己接收和发送消息的约定。

### 2. 目标差异:

- **服务组合:** 目标是将现有服务组合成一个更大的服务, 以实现特定的业务逻辑。
- **服务协作:** 目标是多个独立服务共同协作, 以实现一个更大的业务目标。

### 3. 实现方式差异:

- **服务组合:** 通过编排语言或工具, 以串联或并联的方式组合服务。
- **服务协作:** 通过消息传递、事件驱动等机制实现服务之间的协作。

### 4. 通信方式差异:

- **服务组合:** 通常涉及同步通信, 一个服务的输出作为另一个服务的输入。
- **服务协作:** 通常涉及异步通信, 通过消息传递等机制实现服务之间的协作。

## 2 比较服务编舞与服务编排的异同。

### 2.1 服务编舞

描述参与某次协作多个服务之间的对等消息交互协议。该协议一般是公开发布的，由协作的所有参与者共享。//peer-to-peer,autonomous

- **定义：** 服务编舞强调的是参与者之间的协作和通信，而不是集中式的控制。在服务编舞中，各个服务之间通过定义协议和消息传递的方式相互协作，形成一种分散的、去中心化的协作结构。
- **目标：** 目标是描述服务之间的交互模式，强调各个服务之间的对等关系，让参与者之间协作来达到共同的业务目标。
- **实现：** 实现服务编舞的方式通常包括使用标准的协议（如Web服务协议）以及定义消息的格式和传递规则。

### 2.2 服务编排

描述服务组合的一个过程模型，该过程模型描述如何按一定的次序和参与过程的服务进行消息交互以完成组合服务的应用逻辑。服务编排逻辑受全局服务编舞逻辑的约束。

- **定义：** 服务编排强调的是中心化的控制，一个中心实体（通常是编排引擎或者流程引擎）负责协调和管理各个服务的执行顺序，形成一种集中式的控制结构。
- **目标：** 目标是通过一个中心化的控制点来定义和管理服务之间的执行顺序，以实现复杂的业务流程。
- **实现：** 实现服务编排通常包括使用编排语言（如BPEL）来定义流程、规定服务的执行顺序、以及处理异常情况。

### 2.3 共性

两者都旨在实现服务协作和协调，都关注定义和管理业务流程，都需要服务之间的交互和通信，都适用于松散耦合的服务架构。

### 2.4 差异

服务编舞和服务编排的主要区别在于应用层面不同。

服务编舞体现一个全局的、服务之间要开展协作必须遵守的公共协议，该协议凌驾于参与服务编舞的服务编排逻辑之上；服务编排是一种服务组合的编程手段，是组织实现业务逻辑的方法和途径，一般由组织私有，且其行为受到全局服务编舞协议的约束。