



BPEL 2.0 服务契约

BPEL 2.0 服务契约之基础篇：什么是编制和编排？

本文选自《SOA and WS-BPEL》一书，旨在探讨如何为封装 WS-BPEL 流程逻辑所需的 Web 服务设计 WSDL 定义。因为 SOA 提倡用“契约优先”的方式来设计服务，所以理解由 WS-BPEL 引发的这种独特服务契约设计理念，是成功构建有效流程和服务的关键因素。

目前，SOA 作为一种架构平台已经被很多公司所采用。SOA 是一种有效的方法，能够把建立在 Web 服务的企业应用进行集成。Web 服务是把业务逻辑封装在清楚的上下文中，并且容易组合到合成解决方案之中的松耦合软件。虽然创建远程访问资源和功能的应用不是什么新鲜事，但是，能够根据面向服务原则（如松耦合）去做，却代表一种相对而言比较新的方法。

Web 服务是一种技术，它规定了一种标准机制，即如何通过互联网协议（如 HTTP）暴露并使用数据和应用逻辑。而 WS-BPEL 则是一种编制语言，用来规定业务流程，以描述 Web 服务间的交互，从而为建立基于 Web 服务的面向服务解决方案打下了基础。因此，为了利用 Web 服务和 WS-BPEL 来建立面向服务的解决方案，必须遵循以下两大步骤：

- 创建并发布 Web 服务，以运用到解决方案中
- 使用 WS-BPEL 把 Web 服务合成到业务流中

本文着重探讨 WS-BPEL 和 Web 服务之间的关系，同时重点介绍 WSDL 定义的设计。

服务合成的基本要素

实际上有很多方法可以把服务组织起来，形成一个合成方案。比如说，你可以创建一个合成服务，采用 PHP 脚本的形式，作为 Web 服务发布；如果有需要的话，通过编程的方式调用其他服务。然而，创建合成服务的最常见方式还是通过使用 WS-BPEL，尤其是在更加复杂的合成中。WS-BPEL 是一种编制语言，可以用来创建编制。编制是一种合成、控制器服务，规定使用的服务将如何通过互相协作来完成工作。

编制 (Orchestration)

编制是把各种服务组装到可执行的业务流程中，然后通过一个编制引擎来执行业务流程。用示意图来表示的话，一个编制可能看似图 1：

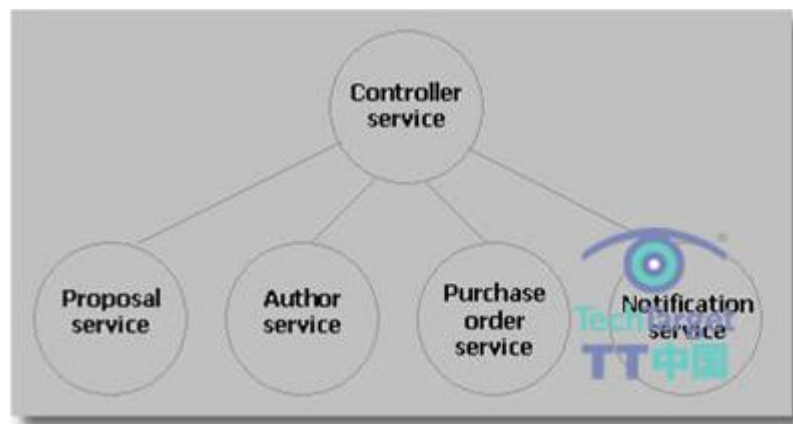


图 1：将其他服务组织到 SOA 合成中的控制服务器

从中可以看到，图 1 中的示意图说明了服务的组合，这些服务通过封装在控制器服务中的逻辑来进行协调。该控制器服务可能是一个 WS-BPEL 业务流程；当编制引擎运行时，它来完成某项业务任务。

可以把用 WS-BPEL 编制语言构建的控制器服务看成是编制任务服务 (Orchestrator Task Service)。作为 Web 服务，控制服务器应该还有一个相应的 WSDL 文档，用来向使用者描述这个服务。等下我们再讨论如何为由 WS-BPEL 构建的合成服务创建 WSDL 规范。

你可以创建一个编制，在另一个更大的编排中，把它作为服务来使用。例如，图 1 中描述的编制可能是另一个 WS-BPEL 编制的一部分。整个编辑过程，从接受方案到发布文章，是通过这个更大的编制来自动执行。

编排 (Choreography)

Web 服务编排规范和 Web 服务编排描述语言 (WS-CDL) 提供了构建 SOA 合成应用的另一种方法。WS-BPEL 是用来把各种服务编制到合成解决方案中，通常表述了某公司的特定业务流程流。而 Web 服务编排描述语言 (WS-CDL) 描述了 Web 服务和参与者之间的点对点关系，这些参与者既可以在受信任范围内，也可以不在受信任范围内。

与编制不同的是，编排不包括一个中央控制机制，而是假定控制是在交互的参与者之间进行共享。这就是说，编制代表的是一个可执行流程，它在通过位于某个位置的一个编制引擎来执行。而编排实质上是代表一种描述，即如何在相互协同的参与者之间来分布控制，而不需要使用任何单个引擎来完成某项工作。

为了定义编排，就需要创建 WS-CDL 编排描述文档；它将是交互参与者之间的契约。具体来说，WS-CDL 文档描述的是相互协作的参与者之间的消息交换，规定这些参与者如何为完成一个共同的业务目标而必须一起工作。例如，可能会存在这样一种编排：在一个编制、一个代表 WS-BPEL 流程的控制器服务和一个与控制器服务交互的客户端服务之间进行协作。

用示意图来表示的话，可能看似图 2 所示：

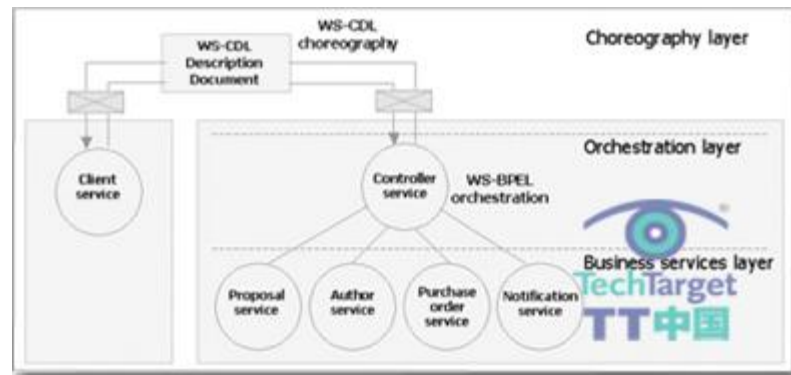


图 2：基于 WS-BPEL 业务流程的合成服务与该服务使用者之间的编排

在这种场景下，编排层的作用是具体规定两个服务之间的点对点协作。WS-CDL 编排文档特别描述了上文提到的合成服务和其中一个使用者之间的消息交换。

BPEL 2.0 服务契约之理解篇：如何使用 WS-BPEL？

之前提到过，WS-BPEL 是一种编制语言，通过定义控制流，来描述 Web 服务应用的执行逻辑，为伙伴服务提供了一种共享上下文的方法。伙伴服务指的是和 WS-BPEL 流程交互的服务。

WS-BPEL 是建立在几种规范之上，如 SOAP、WSDL 和 XML Schema。WSDL 可能是其中最重要的一个。在基于 WS-BPEL 的合成服务中，正是 WSDL 让服务起到作用。WS-BPEL 可以定义业务流程，通过 WSDL 描述，让流程与合作服务进行交互（稍后具体解释）。

有意思的是，虽然 WS-BPEL 是目前最受欢迎的可执行业务流程语言，但并不是对基于 Web 服务应用的执行逻辑进行定义的唯一方法。还有其他规范，如 XLANG、WSFL、XPDL 和 BPML。每一种都可以替代 WS-BPEL。

通过 WS-BPEL，你可以把一组 Web 服务集成到业务流程流中，从而构建业务流程。WS-BPEL 业务流程具体指定了如何在该流程的实例和伙伴服务之间协调交互。图 3 举的例子就是 WS-BPEL 业务流程的工作流示意图。

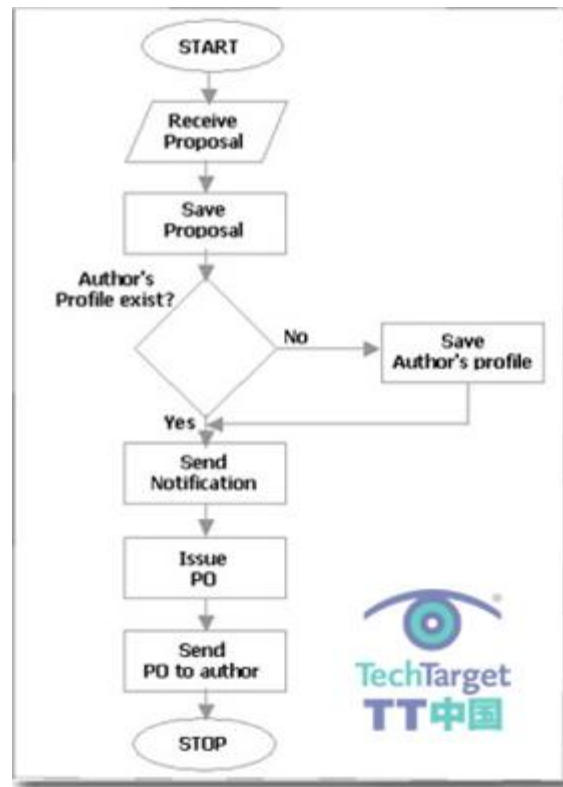


图 3: 把 WS-BPEL 流程封装到图 1 描述的合成服务之工作流示意图

图中所描述的 WS-BPEL 流程把所有的服务都集成到一个点对点流程中，这些服务都是从接受方案开始到最终完成所有步骤所需的服务。在这个具体的例子中，该流程集成了四个 Web 服务，如图 1 中所示。在下文我们将介绍 WS-BPEL 流程如何通过 WSDL 文档定义的一个伙伴链接来连接 Web 服务。

WS-BPEL 流程除了具有调用多个服务的能力外，还可能对 XML 数据进行操作，运行并行处理、条件分支和循环来控制流程流。例如，在上面的流程中，你可以使用一个开关动作，建立两个分支。如果提交方案的是一名新作者，那么这个流程就会调用“Author”服务，把作者的信息保存到数据库。否则，就跳过这一步。

BPEL 2.0 服务契约之实战篇： 如何为封装 WS-BPEL 的服务设计 WSDL 定义？

鉴于 WS-BPEL 编制与 WSDL 定义之间的联系，这样就可以把编制本身当成一个服务，这个服务可以被另一个服务调用，或者作为另一个编制或编排的一部分。

如果 WS-BPEL 流程作为一个服务的形式存在的话，那么应该有相应的 WSDL 文档，客户服务才能调用这个流程。例如，图 3 中描述的 WS-BPEL 流程可能与下面的 WSDL 定义相关联。

查看附录中的例 1。

你会看到，该 WSDL 文档没有包含 binding 或 service 元素。事实上，WS-BPEL 流程服务的 WSDL 文档只包含了服务的抽象定义和 partnerLinkType 部分，该 partnerLinkType 部分代表流程服务与其客户端服务之间的交互。在这个具体的例子中，WSDL 定义只包含了一个 partnerLinkType 部分，支持客户端使用的一个操作来初始化流程。

partnerLinkType 部分最多定义两种角色，每一种角色结果都和之前提及的 WSDL 文档中定义的 portType 有关。WS-BPEL 使用伙伴链接机制对 WS-BPEL 流程和参与者之间的关系进行定义。在本章的稍后部分你会看到，WS-BPEL 流程的定义包含了 partnerLink 元素，用来指定 WS-BPEL 流程和客户端，及合作伙伴者之间的交互。WS-BPEL 流程定义文档中的每个 partnerLink 都和对应的 WSDL 文档中的 partnerLinkType 相关联。用示意图表示的话，如图 4 所示。

一旦你为流程服务创建了 WSDL 定义，那么对于流程执行时将被调用的伙伴服务，就要确保修改这些服务的 WSDL 文档。为了能够让一个服务成为 WS-BPEL 编制的一部分，你可能想在对应的 WSDL 文档中增加一个 `partnerLinkType` 元素。例如，为了能够让图 3 中的 Notification 服务参与到编制中，你需要创建一个如下的 WSDL 文档。

查看附录中的例 2。

注意高亮加粗的 `partnerLinkType` 部分。在描述服务的 WSDL 文档末加入这一部分，可以让这个服务成为一个伙伴链接，让这个服务可能成为编制的一部分。

之前提到过，WS-BPEL 使用伙伴链接机制为业务流程中交互的服务进行建模。这有一段图 3 所示的 WS-BPEL 业务流程定义，描述了使用伙伴链接的 `proposalProcessingService` 流程服务：

查看附录中的例 3。

上面的例子只显示了流程定义中的两个 `partnerLink`。第一个 `partnerLink` 指定了 WS-BPEL 流程和它的客户端之间的交互。另一个 `partnerLink` 指定了 WS-BPEL 流程与在这里作为伙伴服务的 Notification 服务之间的交互。

图 4 也许能帮你更好地理解在 WS-BPEL 中如何使用伙伴链接机制。

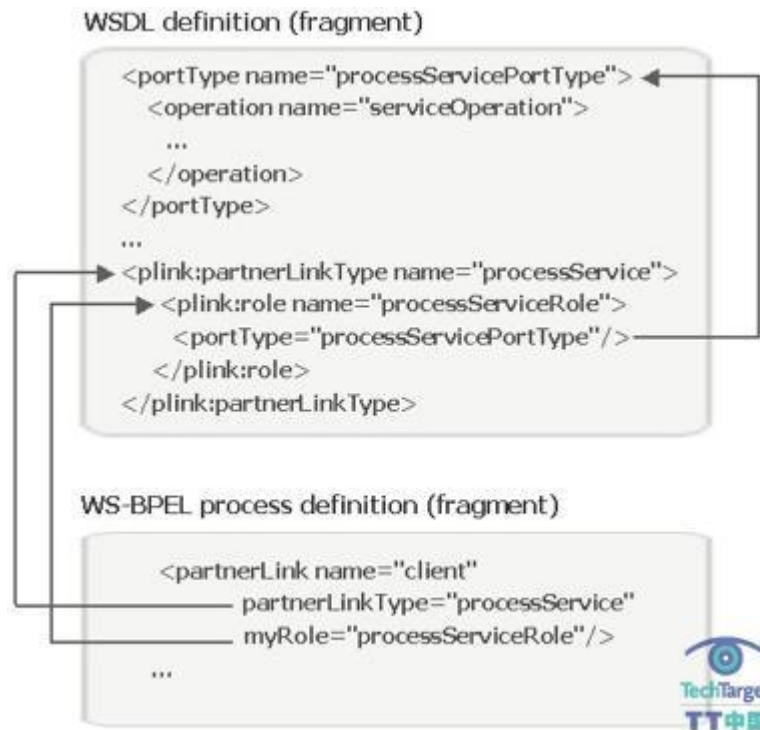


图 4：WSDL 文档中指定的 partnerLinkType 如何和 WS-BPEL 流程定义中的 partnerLink 之间相关联

图中的这个例子表示了描述 WS-BPEL 流程服务的 WSDL 文档中的 partnerLinkType 和流程定义中的 partnerLink 之间的关系。当定义 partnerLinkType 时，使用 myRole 属性来指定 WS-BPEL 流程的角色。为了指定伙伴的角色，则需要使用 partnerRole 属性，参见之前讨论过的流程定义文档。

看一下这里讨论的流程定义文档中的 partnerLink 部分，你会发现它们其实没有提供任何关于 WSDL 文档位置的信息。这些 WSDL 文档包含相应的伙伴链接类型。这些信息其实储存在流程部署描述符（descriptor）文件中。这个文件的格式会因为你所用的 WS-BPEL 工具有所不同。

下面是两个不同描述符的例子，它们是由不同的工具创建的：

查看附录中的例 4。

查看附录中的例 5。

BPEL 2.0 服务契约之实战篇：如何创建 WS-BPEL 流程服务？

通常的面向服务设计都建议流程服务的设计应该从服务契约开始。所以为了提供更多的 WSDL 和 WS-BPEL 定义的例子，我们就来遵循这个过程。

Hello 服务的 WSDL 定义（抽象描述）

我们从一个简单的 Hello 服务的新 WSDL 文档开始：

查看附录中的例 6。

例 6 中的高亮显示部分表示的是 `partnerLinkType` 结构。在该结构中，对 WS-BPEL 流程和它的使用者（客户端服务）的关系进行了定义。

注意，必须为所涉及的每一个伙伴服务指定 `partnerLinkType` 的结构。不过在这个例子中，WS-BPEL 流程只有一个伙伴，即流程的使用者。因此，只需在描述这个 WS-BPEL 流程服务的 WSDL 文档中指定一个 `partnerLinkType` 结构。

Hello 服务 WSDL 定义（抽象和具体描述）

也许你已经注意到，例 6 中的 WSDL 定义没有包含部署信息（绑定和地址信息）。事实上客户端服务使用的是这个文档的修改版。基于上面的 WSDL 代码，WS-BPEL 引擎可以动态地生成包含绑定和地址信息的文档。因此，当我们用浏览器请求这个 WSDL 定义时，我们应该收到下面的输出信息：

查看附录中的例 7。

WS-BPEL 引擎隐式地生成了绑定和服务的定义，因此客户端伙伴服务能够使用这里讨论的流程服务。

Hello WS-BPEL 流程定义

下一步要实施 WS-BPEL 流程定义。这些是相应 hello.bpel 文件的内容：

查看附录中的例 8。

partnerLinks 部分包含与伙伴服务建立关系的 partnerLink 部分；在执行过程中，这些伙伴服务与业务流程进行交互。在例 8 中，partnerLinks 部分只包含了一个入口，即表示流程服务和客户端服务之间关系的入口。这种情况下，需要使用 partnerLink 元素中 myRole 属性，因为 WS-BPEL 流程服务是客户端服务的提供者。

注意 partnerLink 元素中 partnerLinkType 属性指定的名字。这个很重要，它就是之前提到 hello.wsdl 文档中对应的 partnerLinkType 部分指定的名字。

序列结构中定义的活动顺序非常直接明了。第一个是 receive（接收）活动，用来处理客户端服务发送过来的请求消息。把 receive 的 createInstance 属性设置为“yes”，就是告诉 WS-BPEL 引擎：当活动接收到请求消息时，创建一个流程实例。

序列活动中的下一个活动是 assign（赋值）。这里实际上是基于输入消息中的信息，生成了输出消息。

最后，定义了一个 reply（回答）活动，它负责向客户端服务发送输出消息。部署完 hello 流程服务，知道了如何去获得向客户端描述这个服务的 WSDL 文档，接下来就是测试。为此，你可能需要创建并运行下面这段 PHP 脚本：

查看附录中的例 9。

当执行时，上面的脚本应该会调用之前讨论的 hello 流程服务，并输出下面的 hello 消息：

Hello, Larry!

BPEL 2.0 服务契约之实战篇：

WS-BPEL 流程服务如何与伙伴服务产生交互？

poInfo WSDL 定义

我们介绍的这个 Hello 流程服务只是 WS-BPEL 流程服务的一个简化例子。它没有使用伙伴服务来完成这个工作，仅仅是根据客户端发来的数据基础上合成了一个 hello 消息。相比之下，在现实中的 WS-BPEL 流程在执行期间可能会调用很多伙伴服务。

接下来的例子描述的是 poInfo WS-BPEL 流程服务，它和另外两个伙伴服务产生交互：

poOrderDocService 和 poOrderStatusService。

我们从 poInfo 的 WSDL 定义开始：

查看附录中的例 10。

当客户端发送包含两个参数（pono 和 par）的请求消息时，poInfo 流程服务即被调用。第一个参数指定了你要从订单文档中获取的 pono 信息，第二个指定了信息需要返回什么样的信息，意味着两种可能的选择：整个文档或文档的状态。

例 10 还说明了 WSDL 定义如何导入两个描述 poOrderDocService 和 poOrderStatusService 伙伴服务的定义。这个例子假设已经创建好了 po_orderdoc.wsdl 和 po_orderstatus.wsdl。

当对输入消息进行定义时，注意 `poInfoRequest` 复杂类型的用法。这个结构允许让客户端发送两个参数成为可能，即在一个请求消息中的 `pono` 和 `par`。

这里还需要注意的一点是三个伙伴链接的用法。第一个定义了 WS-BPEL 流程服务和它的客户端之间的交互，而另外两个定义了 WS-BPEL 服务分别与 `poOrderDocService` 和 `poOrderStatusService` 伙伴服务的关系。

poInfo WS-BPEL 的定义

下面的 WS-BPEL 定义比我们之前讨论的稍稍复杂些。这是因为下面展示的 `poInfo.bpel` 定义包含了条件逻辑：

查看附录中的例 11。

例 11 中 WS-BPEL 流程定义最有意思的部分是 `if/elseif/else` 结构。

例 12 用示意图来诠释：

查看附录中的例 12。

注意例 12 内部序列结构的用法。WS-BPEL 其实不允许在 `if` 或 `elseif` 或 `else` 块中使用一个以上的活动。这就是为什么不得不按顺序在这些块中包装一组活动。

假设你把剩下的部分也放在合适的位置，流程也能够成功执行，接下来可以用下面的 PHP 脚本来测试。

查看附录中的例 13。

这应该输出整个 po 文档，pono 为 108128476。如果把上面脚本中的\$xmldoc 变量设置如下：`$xmldoc = '108128476status'`；

那么你应该会收到一条短消息：shipped，这表示文档目前的状态。

总结

流程服务可能看似建立了一种 Web 服务的终端，正如 SOA 实施中的其他服务一样。然而，由于 WS-BPEL 语言对 WSDL 定义有着很具体的要求，所以需要增加额外的结构，并进行优化。理解这些新元素，理解它们如何把一个服务作为更大的服务合成中一个潜在的众多伙伴服务之一，是非常重要的。因为尽管任何一个 WS-BPEL 流程将合成其他服务，但是通过创建自己的 WSDL 服务契约，WS-BPEL 流程就可以像任何其他服务一样具有可合成性。即作为一个流程，同时也作为一个服务存在。

关于作者

Yuli Vasiliev 是软件开发者、自由撰稿人和咨询师。目前专注于开源开发、Oracle 技术和 SOA。他具有十多年的软件开发经验和七年多的技术写作经验。他出版的新书《SOA and WS-BPEL》具体详述了如何使用开源产品 PHP 和 ActiveBPEL 引擎创建和部署你自己的面向服务解决方案。

(作者: Yuli Vasilev 译者: 杨晓明, Shirley 来源: TechTarget 中国)

例 1

```
<?xml version="1.0" encoding="utf-8"?>
<definitions name="proposalProcessingService"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsd1="http://localhost/Webservices/schema/"
  xmlns:plink="http://schemas.xmlsoap.org/ws/2004/03/partner-link/"
  targetNamespace="http://localhost/Webservices/ch1/proposalProcessing/">
  <types>
    <xsd:schema
targetNamespace="http://localhost/Webservices/ch1/proposalProcessing.wsdl">
      <xsd:import namespace="http://localhost/Webservices/schema/"
        schemaLocation="http://localhost/Webservices/schema/propProc.xsd"/>
    </xsd:schema>
  </types>
  <message name="receiveProposalInput">
    <part name="body" element="xsd1:proposalDocType"/>
  </message>
  <message name="receiveProposalOutput">
    <part name="body" element="xsd:string"/>
  </message>
  <portType name="proposalProcessingServicePortType">
```

```
<operation name="receiveProposal">
  <input message="tns:receiveProposalInput"/>
  <output message="tns:receiveProposalOutput"/>
</operation>
</portType>
<plink:partnerLinkType name="proposalProcessingService">
  <plink:role name="proposalProcessingServiceRole">
    <portType="tns:proposalProcessingServicePortType"/>
  </plink:role>
</plink:partnerLinkType>
```

例 2

```
<?xml version="1.0" encoding="utf-8"?>
<definitions name="notificationService"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:plink="http://schemas.xmlsoap.org/ws/2004/03/partner-link/"
  targetNamespace="http://localhost/Webservices/ch1/notification/">
  <message name="sendMessageInput">
    <part name="body" element="xsd:string"/>
  </message>
  <message name="sendMessageOutput">
    <part name="body" element="xsd:string"/>
  </message>
  <portType name="notificationServicePortType">
    <operation name="sendMessage">
      <input message="tns:sendMessageInput"/>
      <output message="tns:sendMessageOutput"/>
    </operation>
  </portType>
  <binding name="notificationServiceBinding"
type="tns:notificationServicePortType">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
```

```
<operation name="sendMessage">
  <soap:operation
soapAction="http://localhost/WebServices/ch1/sendMessage"/>
  <input>
    <soap:body use="literal"/>
  </input>
  <output>
    <soap:body use="literal"/>
  </output>
</operation>
</binding>
<service name="notificationService">
  <port name="notificationServicePort"
binding="tns:notificationServiceBinding">
    <soap:address
location="http://localhost/WebServices/ch1/SOAPserver.php"/>
  </port>
</service>
<plink:partnerLinkType name="notificationService">
  <plink:role name="notificationServiceRole">
    <portType="tns:notificationServicePortType"/>
  </plink:role>
</plink:partnerLinkType>
</definitions>
```

例 3

```
<process name="BusinessTravelProcess"
  targetNamespace="http://localhost/WebServices/ch1/proposalProcessing/"
  xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
  xmlns:prc="http://localhost/WebServices/ch1/proposalProcessing/"
  xmlns:ntf="http://localhost/WebServices/ch1/notification/"

  <partnerLinks>
    <partnerLink name="client"
      partnerLinkType="prc:proposalProcessingService"
      myRole="proposalProcessingServiceRole"/>
    <partnerLink name="sendingNotification"
      partnerLinkType="ntf:notificationService"
      partnerRole="notificationServiceRole"/>
    ...
  </partnerLinks>
  ...
</process>
```

例 4

```
<?xml version="1.0" encoding="UTF-8"?>
<BPELSuitcase>
  <BPELProcess src="proposalProcess.bpel" id="proposalProcess">
    <partnerLinkBindings>
      <partnerLinkBinding name="client">
        <property name="wsdlLocation">proposalProcess.wsdl</property>
      </partnerLinkBinding>
      <partnerLinkBinding name="sendingNotification">
        <property
name="wsdlLocation">http://localhost/Webservices/ch1/notification?wsdl</proper
ty>
      </partnerLinkBinding>
      ...
    </partnerLinkBindings>
  </BPELProcess>
</BPELSuitcase
```

例 5

```
<?xml version="1.0" encoding="UTF-8"?>
<process ...>
  ...
  <references>
    <wsdl location="project:/proposalProcess/WSDL/proposalProcess.wsdl"
namespace="http://localhost/Webservices/ch1/proposalProcessing/" />
    <wsdl location="project:/proposalProcess/WSDL/notification.wsdl"
namespace="http://localhost/Webservices/ch1/notification/" />
  </references>
</process>
```


例 6

```
<?xml version="1.0" encoding="UTF-8" ?>
<definitions targetNamespace="http://localhost:8081/active-
bpel/services/hello"
    xmlns:tns="http://localhost:8081/active-bpel/services/hello"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:plnk="http://schemas.xmlsoap.org/ws/2003/05/partner-link/">
<!-- abstract characteristics of the WS-BPEL process service -->
<message name="inputMessage">
    <part name="firstName" type="xsd:string"/>
</message>
<message name="outputMessage">
    <part name="hello" type="xsd:string"/>
</message>
<portType name="helloServicePT">
    <operation name="hello">
        <input name="inputMessage" message="tns:inputMessage"/>
        <output name="outputMessage" message="tns:outputMessage"/>
    </operation>
</portType>
<!-- partnerLinkType section representing interaction between the WS-BPEL
service and its client -->
<plnk:partnerLinkType name="helloPartnerLinkType">
```

```
<plnk:role name="helloServiceRole">  
  <plnk:portType name="tns:helloServicePT"/>  
</plnk:role>  
</plnk:partnerLinkType>  
</definitions>
```

例 7

```
<?xml version="1.0" encoding="UTF-8" ?>
<definitions targetNamespace="http://localhost:8081/active-
bpel/services/hello"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:plnk="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:tns="http://localhost:8081/active-bpel/services/hello"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <message name="outputMessage">
    <part name="hello" type="xsd:string" />
  </message>
  <message name="inputMessage">
    <part name="firstName" type="xsd:string" />
  </message>
  <portType name="helloServicePT">
    <operation name="hello">
      <input message="tns:inputMessage" name="inputMessage" />
      <output message="tns:outputMessage" name="outputMessage" />
    </operation>
  </portType>
  <binding name="helloServicePTBinding" type="tns:helloServicePT">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
    <operation name="hello">
```

```
<soap:operation soapAction="" style="rpc"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
  <input>
    <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
use="encoded" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
  </input>
  <output>
    <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
use="encoded" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
  </output>
</operation>
</binding>
<service name="helloServicePT">
  <port binding="tns:helloServicePTBinding" name="helloServicePTPort">
    <soap:address location="http://localhost:8081/active-
bpel/services/helloServicePT"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" />
  </port>
</service>
</definitions>
```

例 8

```
<?xml version="1.0" encoding="UTF-8"?>
<process name="hello"
  xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
  xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
  xmlns:lns="http://localhost:8081/active-bpel/services/hello"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  suppressJoinFailure="yes"
  targetNamespace="http://hello">
  <partnerLinks>
    <partnerLink myRole="helloServiceRole" name="customer"
partnerLinkType="lns:helloPartnerLinkType"/>
  </partnerLinks>
  <variables>
    <variable messageType="lns:inputMessage" name="inputMessage"/>
    <variable messageType="lns:outputMessage" name="outputMessage"/>
  </variables>
  <sequence>
    <receive createInstance="yes"
      operation="hello"
      partnerLink="customer"
      portType="lns:helloServicePT"
      variable="inputMessage"/>
  <assign>
    <copy>
```

```
        <from expression="concat( 'Hello, ',
bpws:getVariableData(' inputMessage', ' firstName'), '!' )"/>
        <to part="hello" variable="outputMessage"/>
    </copy>
</assign>
<reply operation="hello"
    partnerLink="customer"
    portType="lns:helloServicePT"
    variable="outputMessage"/>
</sequence>
</process>
```

例 9

```
<?php
//File: SoapClient_hello.php
$client = new SoapClient("http://localhost:8081/active-
bpel/services/helloServicePT?wsdl");
try {
    print($client->hello('Larry'));
}
catch (SoapFault $e) {
    print $e->getMessage();
}
?>
```

例 10

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="poInfo"
  targetNamespace="http://localhost:8081/active-
bpel/services/poInfoService.wsdl"
  xmlns:tns="http://localhost:8081/active-bpel/services/poInfoService.wsdl"
  xmlns:plnk="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ns2="http://localhost/WebServices/wsdl/poOrderDoc"
  xmlns:ns3="http://localhost/WebServices/wsdl/poOrderStatus"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
  <import namespace="http://localhost/WebServices/wsdl/poOrderDoc"
    location="http://localhost/WebServices/wsdl/po_orderdoc.wsdl"/>
  <import namespace="http://localhost/WebServices/wsdl/poOrderStatus"
    location="http://localhost/WebServices/wsdl/po_orderstatus.wsdl"/>
  <types>
    <schema attributeFormDefault="qualified"
      elementFormDefault="qualified"
      targetNamespace="http://localhost:8081/active-
bpel/services/poInfoService.wsdl"
      xmlns="http://www.w3.org/2001/XMLSchema">
      <element name="poInfoRequest">
        <complexType>
          <sequence>
            <element name="pono" type="xsd:string"/>

```



```
        <element name="par" type="xsd:string"/>
    </sequence>
</complexType>
</element>
</schema>
</types>
<message name="poInfoResponseMessage">
    <part name="payload" type="xsd:string"/>
</message>
<message name="poInfoRequestMessage">
    <part name="payload" element="tns:poInfoRequest"/>
</message>
<portType name="poInfoPT">
    <operation name="getInfo">
        <input message="tns:poInfoRequestMessage"/>
        <output message="tns:poInfoResponseMessage"/>
    </operation>
</portType>
<plnk:partnerLinkType name="poInfoLT">
    <plnk:role name="poInfoProviderRole">
        <plnk:portType name="tns:poInfoPT"/>
    </plnk:role>
</plnk:partnerLinkType>
<plnk:partnerLinkType name="poDocLT">
    <plnk:role name="poDocProviderRole">
        <plnk:portType name="ns2:poOrderDocServicePortType"/>
    </plnk:role>
</plnk:partnerLinkType>
```

```
<plnk:partnerLinkType name="poStatusLT">
  <plnk:role name="poStatusProviderRole">
    <plnk:portType name="ns3:poOrderStatusServicePortType"/>
  </plnk:role>
</plnk:partnerLinkType>
</definitions>
```

例 11

```
<?xml version="1.0" encoding="UTF-8"?>
<process xmlns="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
  xmlns:ns1="http://localhost:8081/active-bpel/services/poInfoService.wsdl"
  xmlns:ns2="http://localhost/WebServices/wsdl/poOrderDoc"
  xmlns:ns3="http://localhost/WebServices/wsdl/poOrderStatus"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  name="poInfo.bpel"
  suppressJoinFailure="yes"
  targetNamespace="http://poInfo.bpel">
  <import importType="http://schemas.xmlsoap.org/wsdl/"
    location="wsdl/poInfo.wsdl"
    namespace="http://localhost:8081/active-
bpel/services/poInfoService.wsdl"/>
  <import importType="http://schemas.xmlsoap.org/wsdl/"
    location="http://localhost/WebServices/wsdl/po_orderdoc.wsdl"
    namespace="http://localhost/WebServices/wsdl/poOrderDoc"/>
  <import importType="http://schemas.xmlsoap.org/wsdl/"
    location="http://localhost/WebServices/wsdl/po_orderstatus.wsdl"
    namespace="http://localhost/WebServices/wsdl/poOrderStatus"/>

  <partnerLinks>
    <partnerLink myRole="poInfoProviderRole" name="poInfoProvider"
partnerLinkType="ns1:poInfoLT"/>
```

```
<partnerLink name="poDocRequester" partnerLinkType="ns1:poDocLT"
partnerRole="poDocProviderRole"/>
<partnerLink name="poStatusRequester" partnerLinkType="ns1:poStatusLT"
partnerRole="poStatusProviderRole"/>
</partnerLinks>
<variables>
  <variable messageType="ns1:poInfoRequestMessage"
name="poInfoRequestMessage"/>
  <variable messageType="ns1:poInfoResponseMessage"
name="poInfoResponseMessage"/>
  <variable messageType="ns2:getOrderDocInput"
name="poDocRequestMessage"/>
  <variable messageType="ns2:getOrderDocOutput"
name="poDocResponseMessage"/>
  <variable messageType="ns3:getOrderStatusInput"
name="poStatusRequestMessage"/>
  <variable messageType="ns3:getOrderStatusOutput"
name="poStatusResponseMessage"/>
</variables>
<sequence>
  <receive createInstance="yes"
    operation="getInfo"
    partnerLink="poInfoProvider"
    portType="ns1:poInfoPT"
    variable="poInfoRequestMessage"/>
  <if>
    <condition>($poInfoRequestMessage.payload/ns1:par = 'doc')</condition>
    <sequence>
```

```
<assign>
  <copy>
    <from part="payload" variable="poInfoRequestMessage">
      <query>ns1:pono</query>
    </from>
    <to part="pono" variable="poDocRequestMessage"/>
  </copy>
</assign>
<invoke inputVariable="poDocRequestMessage"
  outputVariable="poDocResponseMessage"
  operation="getOrderDoc"
  partnerLink="poDocRequester"
  portType="ns2:poOrderDocServicePortType">
</invoke>
<assign>
  <copy>
    <from>$poDocResponseMessage.doc</from>
    <to>$poInfoResponseMessage.payload</to>
  </copy>
</assign>
</sequence>
<elseif>
  <condition>($poInfoRequestMessage.payload/ns1:par =
' status')</condition>
  <sequence>
    <assign>
      <copy>
        <from part="payload" variable="poInfoRequestMessage">
```

```
        <query>ns1:pono</query>
    </from>
    <to part="pono" variable="poStatusRequestMessage"/>
</copy>
</assign>
<invoke inputVariable="poStatusRequestMessage"
        outputVariable="poStatusResponseMessage"
        operation="getOrderStatus"
        partnerLink="poStatusRequester"
        portType="ns3:poOrderStatusServicePortType">
</invoke>
<assign>
    <copy>
        <from>$poStatusResponseMessage.status</from>
        <to>$poInfoResponseMessage.payload</to>
    </copy>
</assign>
</sequence>
</elseif>
<else>
    <assign>
        <copy>
            <from>'Wrong input parameter. Should be either doc or
status!'  
</from>
            <to>$poInfoResponseMessage.payload</to>
        </copy>
    </assign>
</else>
```

```
</if>  
<reply operation="getInfo"  
  partnerLink="poInfoProvider"  
  portType="ns1:poInfoPT"  
  variable="poInfoResponseMessage"/>  
</sequence>  
</process>
```

例 12

```
<sequence>

    activities

    <if>
        <condition>...</condition>
        <sequence>

            activities

        </sequence>
    <elseif>
        <condition>...</condition>

        <sequence>

            activities

        </sequence>
    </elseif>
    <else>
        activity
    </else>
</if>
```


activities

例 13

```
<?php
//File: SoapClient_poInfo.php
$client = new SoapClient("http://localhost:8081/active-
bpel/services/poInfoService?wsdl");
$xmlrpc = '<wrapper><pono>108128476</pono><par>doc</par></wrapper>';
$xmlrpc = simplexml_load_string($xmlrpc);
try {
    print($client->getInfo($xmlrpc));
}
catch (SoapFault $e) {
    print $e->getMessage();
}
?>
```