7

More BPEL Activities

Objectives

After completing this lesson, you should be able to:

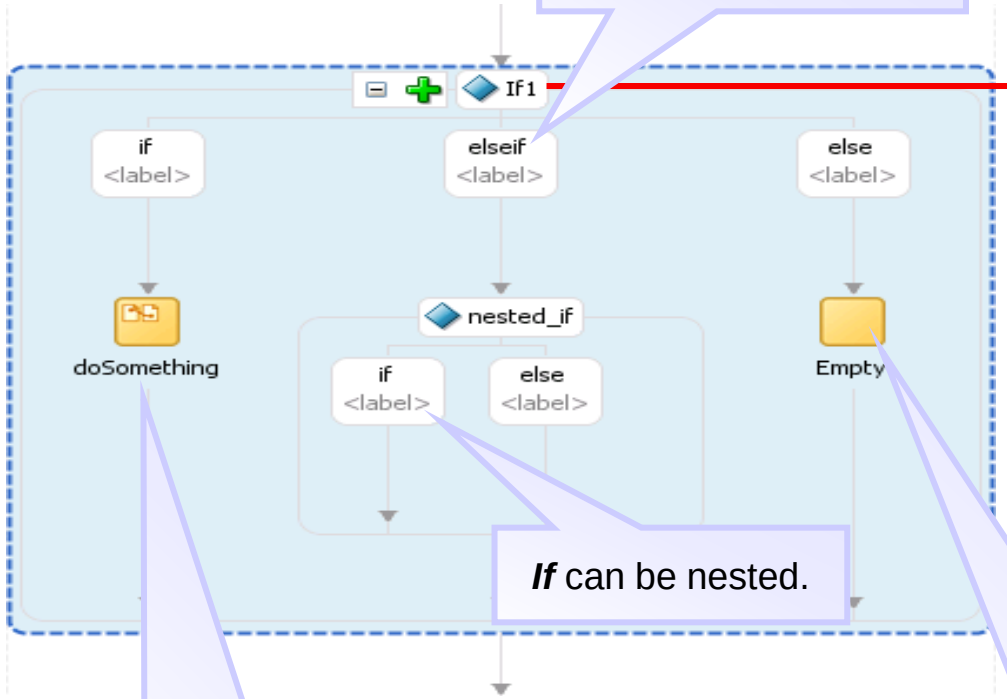
- Perform conditional branching by using an If activity
- Implement parallel processing by using a Flow activity
- Implement non-blocking invocation with a Flow
- Create parallel branches dynamically with a forEach activity
- Implement a Pick activity with an alarm and a timeout
- Execute activities repetitively with a While activity

Agenda

- More Activity Types
- Interaction Patterns

Conditional Branching with the If Activity

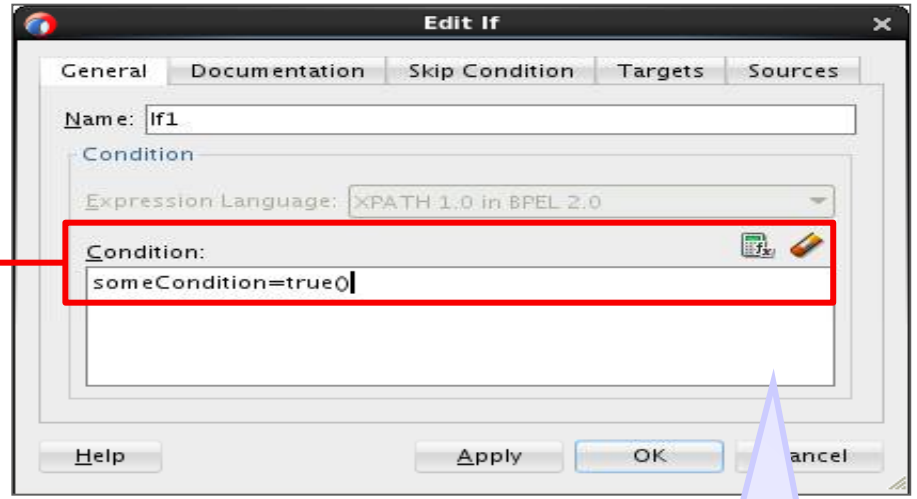
Use `elseif` to test alternative conditions.



If can be nested.

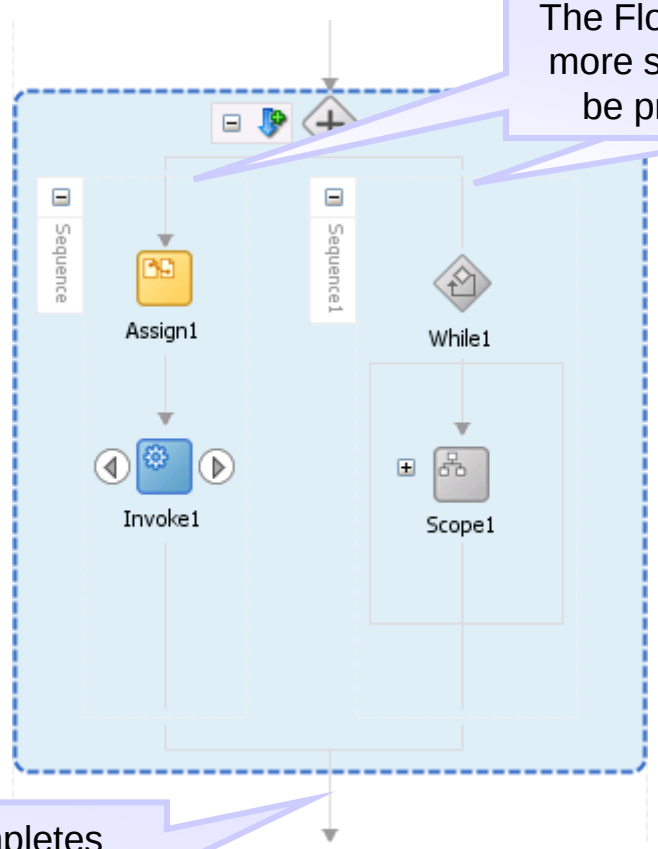
Activities are executed if condition is true.

Use the Empty activity to do nothing.



Use Expression Builder to define the test condition for the ***If***.

Processing with the Flow Activity

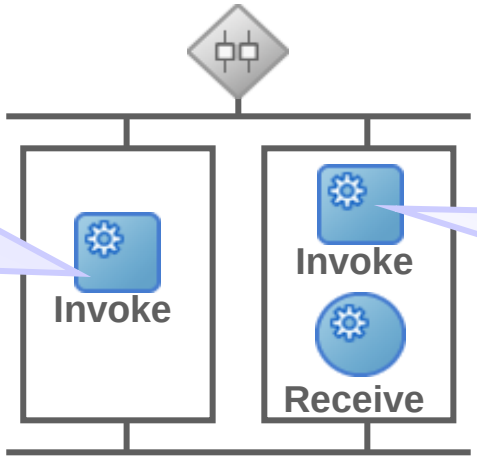


The Flow activity enables two or more sequences of activities to be processed concurrently.

The Flow completes when all activity sequences are finished.

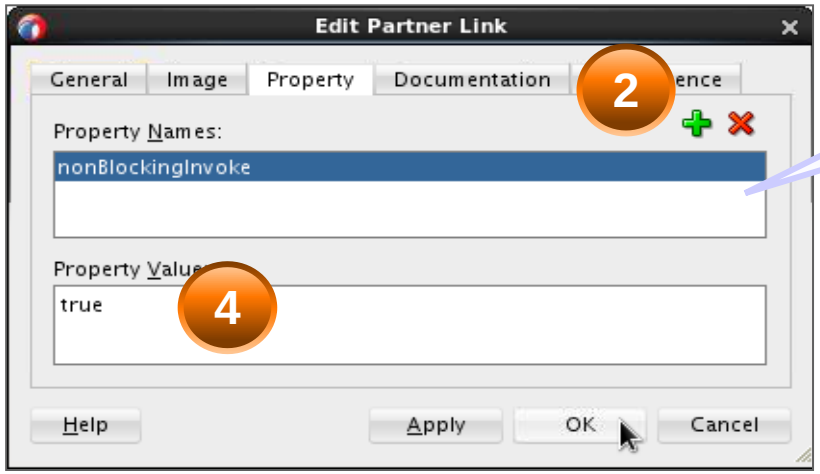
Using the `nonBlockingInvoke` Property

If a synchronous Invoke blocks execution until it completes…

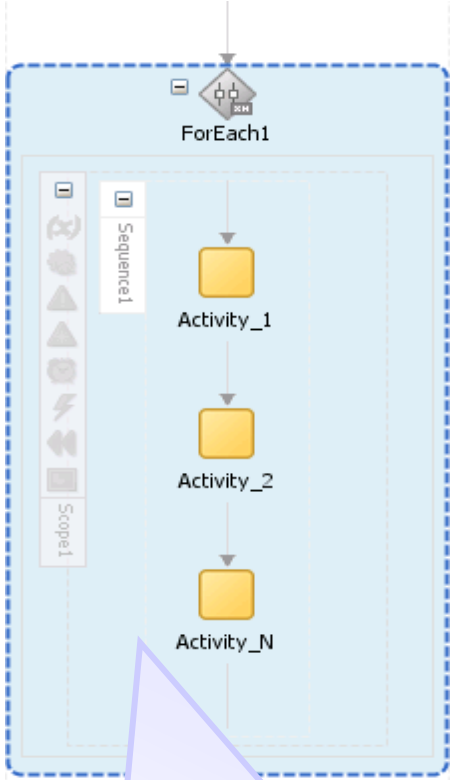


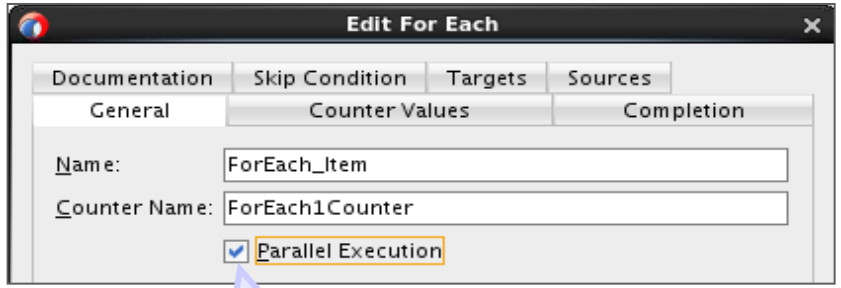
…the asynchronous Invoke waits until the synchronous Invoke completes.

Solution: Edit the partner link for the synchronous service. Set the `nonBlockingInvoke` property to `true`.

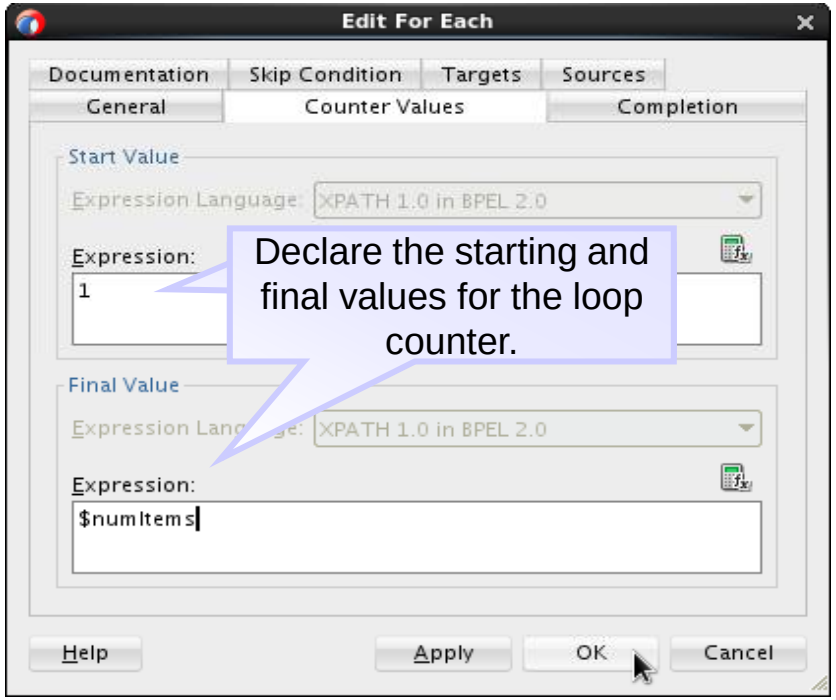


Parallel and Sequential Looping with the forEach Activity





Both *sequential* and *parallel* looping are supported.

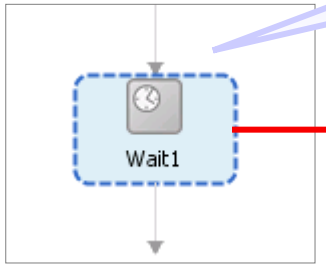


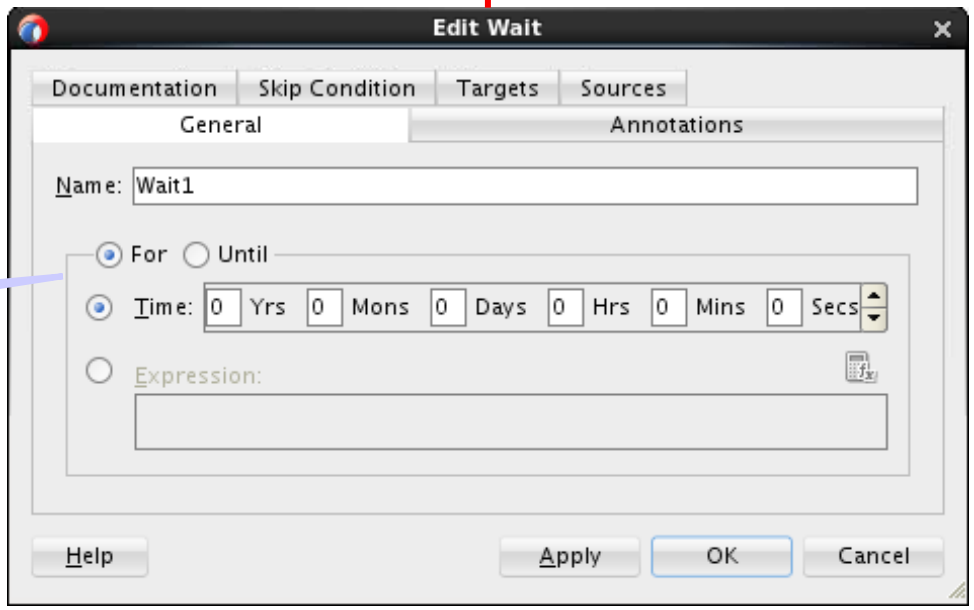
Declare the starting and final values for the loop counter.

In BPEL 2.0, the `forEach` activity performs looping iterations sequentially *N* times over a given set of activities.

Suspending a Process with the Wait Activity

You can suspend a BPEL process by using the Wait activity.



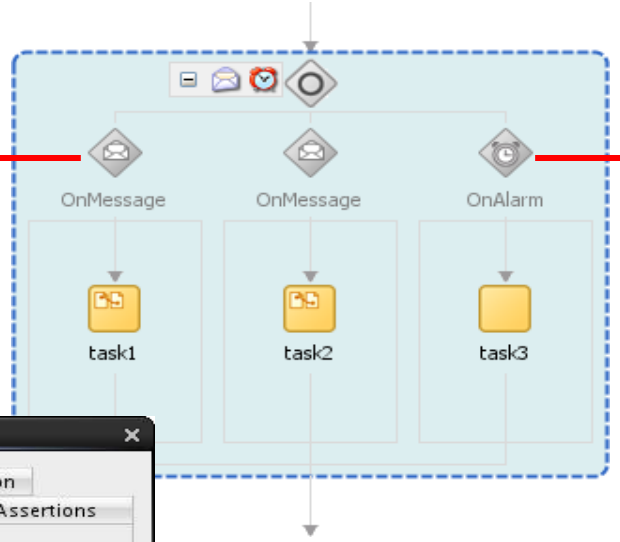


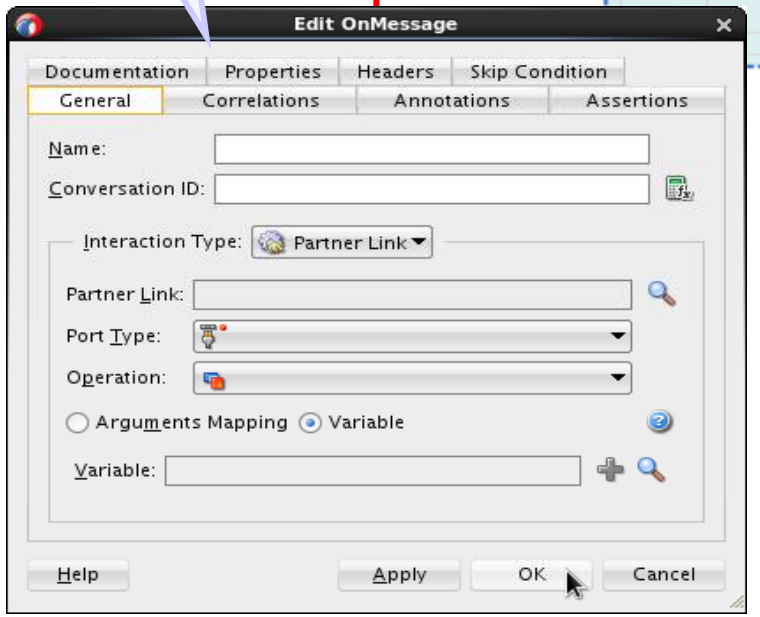
The Wait activity pauses for a specified amount of elapsed time or until a specified date-time.

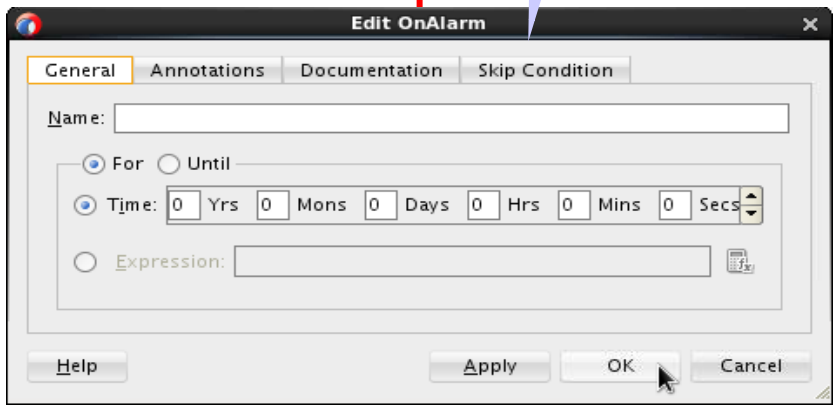
Waiting for a Message with the Pick Activity

The Pick activity is used to wait for a message, just as a Receive activity does.

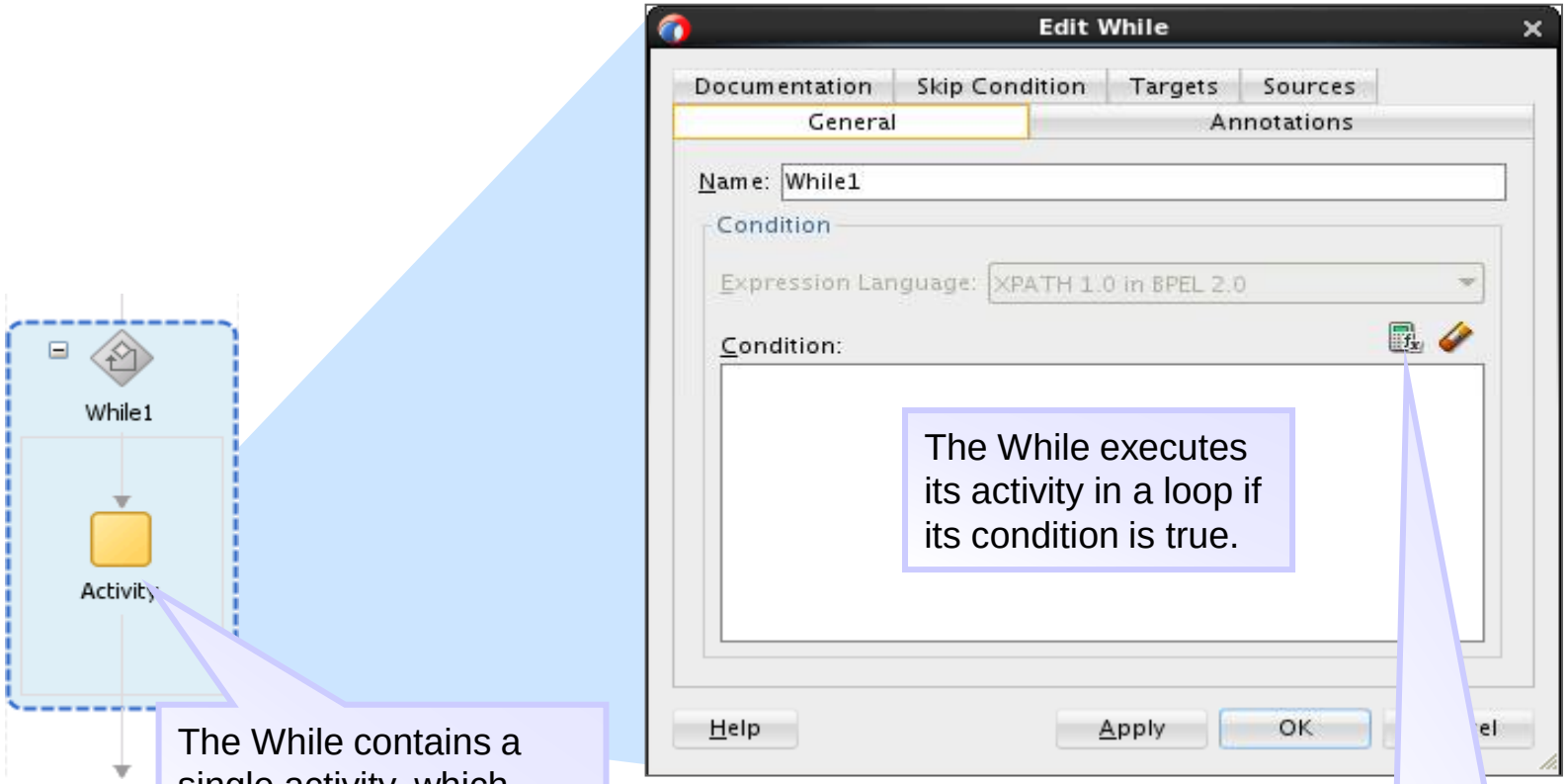
An `onAlarm` branch implements a wait time, and provides an optional timeout.







Looping with the While Activity



The While executes its activity in a loop if its condition is true.

The While contains a single activity, which may be a Sequence or a Scope.

The condition is defined with Expression Builder.

Indexing XML Arrays Dynamically

To access the *n*th element of an XML array, use the XPath expression syntax: `/parent/array[n]/child`

```
<while name="While1">
 <condition>$i &lt;= $numItems and $inStock=true()</condition>
     <sequence name="Sequence1">
     <assign name="Assign_prodid">
    <copy>
     <from>$inputVariable.payload/ns2:product[position()=$i]/ns2:prodId</from>
     <to>$Invoke_check_getInternalStoreView1_InputVariable.parameters/ns4:prodId</to>
          </copy>
     </assign>
```

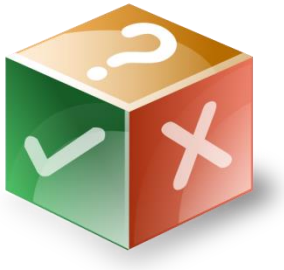
For dynamic index expression, use either the XPath syntax `[position()=num]` or simply the value `[num]`.

Quiz

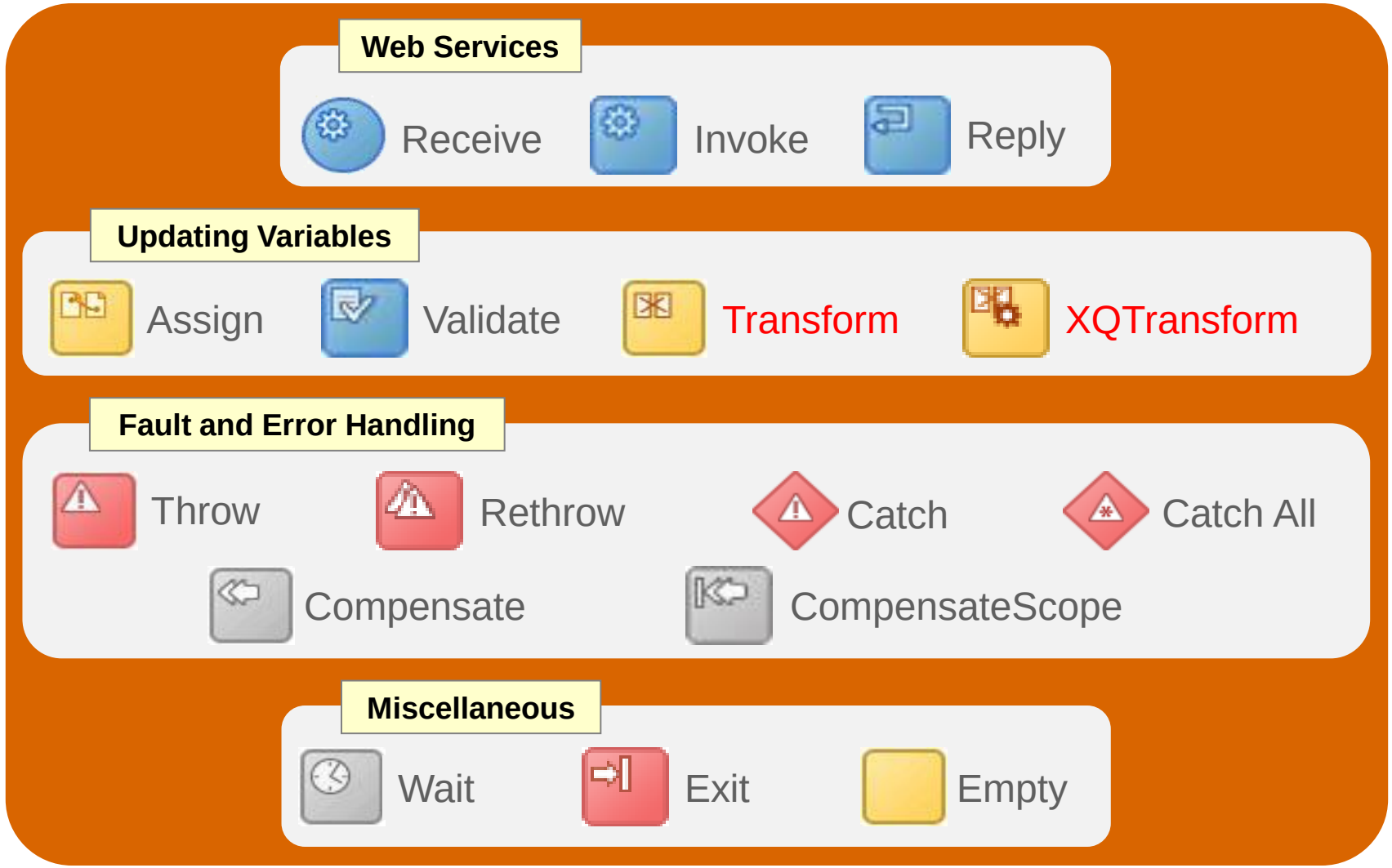
The Flow and forEach activities are the two looping constructs that are available in BPEL.

a. True

b. False



Basic BPEL 2.0 Activities



Structured and Extension BPEL 2.0 Activities

Execution Context

Scope

Sequential and Parallel Processing

Sequence

Flow

Looping Execution

While

ForEach

RepeatUntil

Conditional Branching

Pick

onMessage

onAlarm

If

Else If

Else

SOA Components

Human Task

Business Rules

BPEL Extensions

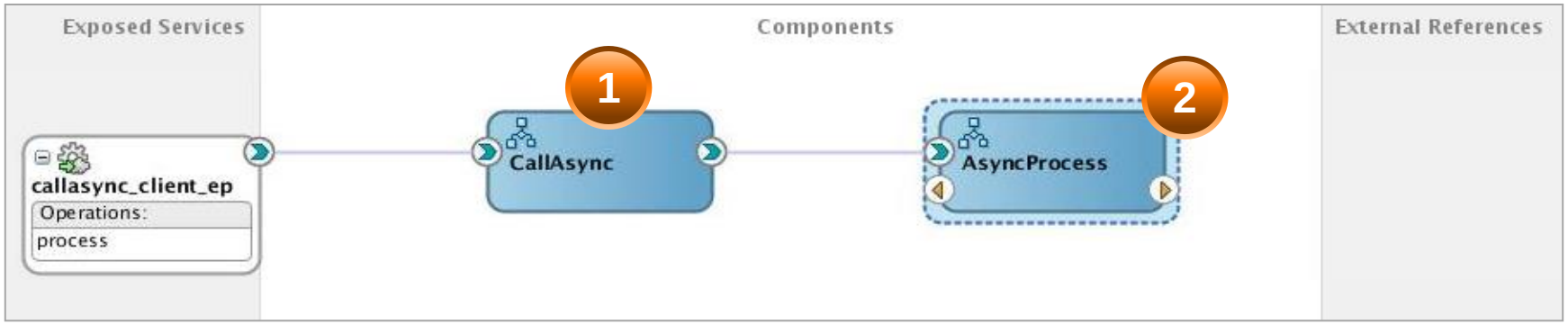
Many others!

Summary

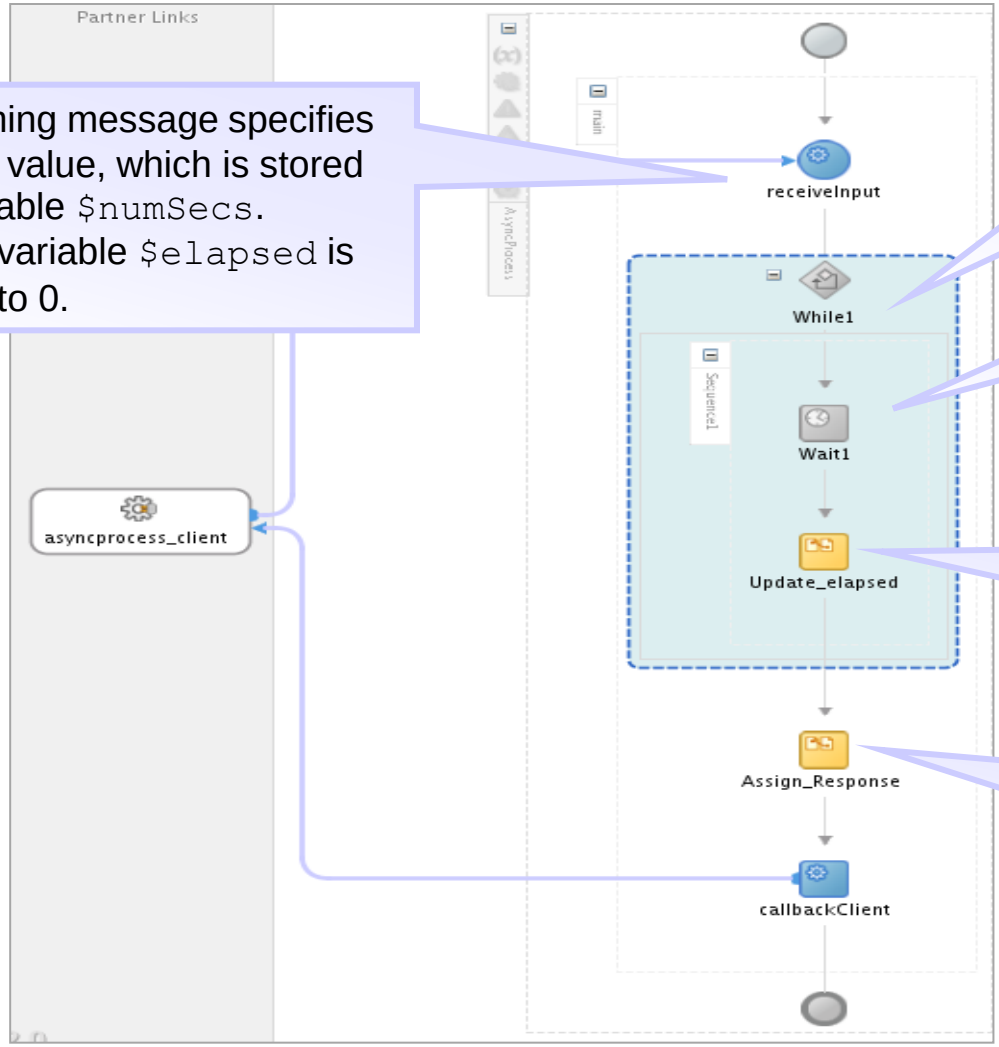
In this lesson, you should have learned how to:

- Perform conditional branching by using an If activity
- Implement parallel processing by using a Flow activity
- Implement non-blocking invocation with a Flow
- Create parallel branches dynamically with a forEach activity
- Implement a Pick activity with an alarm and a timeout
- Execute activities repetitively with a While activity
- Explore various SOA component interaction patterns
- Explore a dynamic service invocation pattern

Practice 7 Overview



Practice 7 Overview



The incoming message specifies a numeric value, which is stored in the variable `$numSecs`. A second variable `$elapsed` is initialized to 0.

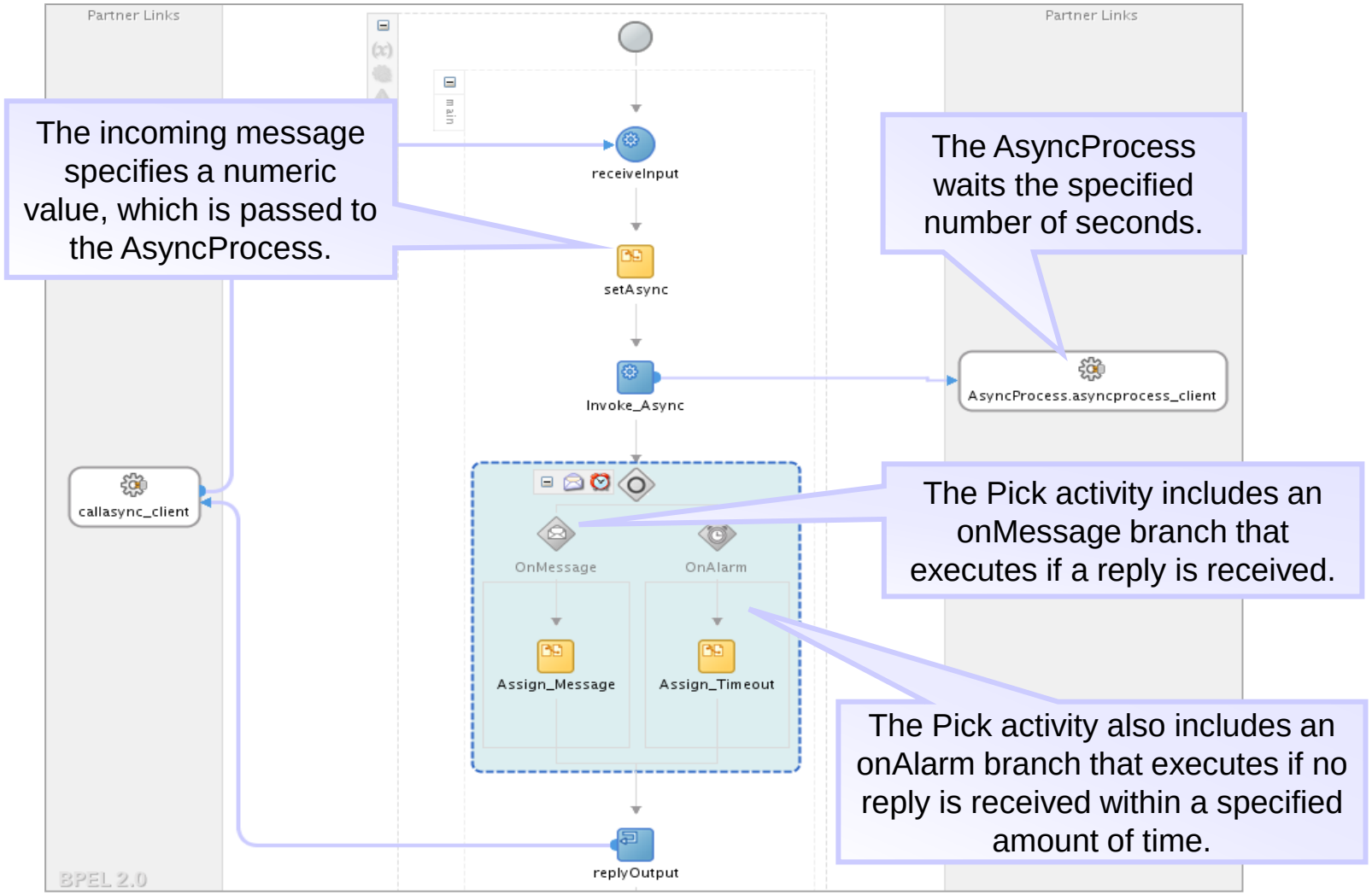
The While activity will loop as long as `$elapsed` < `$numSecs`.

A Wait activity pauses for 3 seconds.

An Assign activity adds 3 to `$elapsed`. At this point, the While activity loops.

The value of `$elapsed` is copied to the reply and returned to the client.

Practice 7 Overview



Before You Begin

The instructions for this practice and those that follow include path references that include XML namespace prefixes, for example:

- "Use Expression Builder to define the expression `string($inputVariable.payload/ns4:shipMethod)`."

If you have built each of the components in the order that was prescribed in the instructions, Expression Builder generates an expression that includes namespaces that match.

If you have built components in a different order, or deleted, and then re-created components, *your namespace prefixes may not match what is listed in these instructions*.

- In that case, *use the namespace references in your project*.
- Do **NOT** change those references to match the book.