

Institut für Architektur von Anwendungssystemen (IAAS)

Universität Stuttgart
Universitätsstraße 38
D – 70569 Stuttgart

Studienarbeit Nr. 2052

**Pattern-based Evaluation of IBM WebSphere
BPEL**

Volker Kramberg

Studiengang: Informatik

Prüfer: Prof. Dr. Frank Leymann

Betreuer: Thorsten Scheibler

Begonnen am: 13.01.2006

Beendet am: 15.07.2006

CR-Klassifikation: H.4.1

Pattern-based Evaluation of IBM WebSphere BPEL

Table of contents

1	Introduction	1
2	Evaluation of control-flow patterns in IBM WebSphere Integration Developer	2
2.1	Basic Control Flow Patterns.....	2
	Pattern 1 (Sequence)	2
	Pattern 2 (Parallel Split)	2
	Pattern 3 (Synchronization)	3
	Pattern 4 (Exclusive Choice)	4
	Pattern 5 (Simple Merge)	6
2.2	Advanced Branching and Synchronization Patterns.....	7
	Pattern 6 (Multi-choice)	7
	Pattern 7 (Synchronizing Merge)	8
	Pattern 8 (Multi-merge)	8
	Pattern 9 (Discriminator)	9
2.3	Structural Patterns.....	10
	Pattern 10 (Arbitrary Cycles)	10
	Pattern 11 (Implicit Termination)	11
2.4	Patterns involving Multiple Instances.....	11
	Pattern 12 (Multiple Instances without Synchronization)	11
	Pattern 13 (Multiple Instances with a Priori Design Time Knowledge)	12
	Pattern 14 (Multiple Instances with a Priori Runtime Knowledge)	13
	Pattern 15 (Multiple Instances without a Priori Runtime Knowledge)	14
2.5	State-based Patterns.....	16
	Pattern 16 (Deferred Choice)	16
	Pattern 17 (Interleaved Parallel Routing)	17
	Pattern 18 (Milestone)	19
2.6	Cancellation Patterns.....	20
	Pattern 19 (Cancel Activity)	20
	Pattern 20 (Cancel Case)	21
3	Summary	23
	References	26

List of figures

Figure 1:	Sequence	2
Figure 2:	Parallel Split	3
Figure 3:	Synchronization	4
Figure 4:	Exclusive Choice	5
Figure 5:	Exclusive Choice with otherwise condition	5
Figure 6:	Simple Merge	6
Figure 7:	Multi-choice	7
Figure 8:	Multi-merge	9
Figure 9:	Discriminator	10
Figure 10:	Multiple Instances without Synchronization	12
Figure 11:	Multiple Instances with a Priori Design Time Knowledge	13
Figure 12:	Multiple Instances with a Priori Runtime Knowledge	14
Figure 13:	Multiple Instances without a Priori Runtime Knowledge	15
Figure 14:	Deferred Choice	17
Figure 15:	Interleaved Parallel Routing	18
Figure 16:	Milestone	20
Figure 17:	Cancel Activity	21
Figure 18:	Cancel Case	22

List of tables

Table 1:	Support of control-flow patterns in IBM WebSphere Integration Developer	24
Table 2:	Mapping of symbols and terms used in IBM WebSphere Integration Developer to employed BPEL constructs.....	25

Appendix: BPEL Code

Pattern 1 (Sequence)	27
Pattern 2 (Parallel Split)	28
Pattern 3 (Synchronization)	30
Pattern 4 (Exclusive Choice)	32
Pattern 5 (Simple Merge)	35
Pattern 6 (Multi-choice)	37
Pattern 7 (Synchronizing Merge)	39
Pattern 8 (Multi-merge)	42
Pattern 9 (Discriminator)	46
Pattern 12 (Multiple Instances without Synchronization)	48
Pattern 13 (Multiple Instances with a Priori Design Time Knowledge).....	50
Pattern 14 (Multiple Instances with a Priori Runtime Knowledge)	53
Pattern 15 (Multiple Instances without a Priori Runtime Knowledge)	55
Pattern 16 (Deferred Choice)	59
Pattern 17 (Interleaved Parallel Routing)	62
Pattern 18 (Milestone)	65
Pattern 19 (Cancel Activity)	68
Pattern 20 (Cancel Case)	70

1 Introduction

There are numerous Business Process Execution Language (BPEL) modeling tools available on the market today that differ in their power and ability to transform patterns into executable BPEL code. Examples for commercially available tools are ActiveBPEL [5], Oracle-BPEL [12] and IBM WebSphere Integration Developer [3]. Patterns describe business requirements and thus define the needs in workflow languages and their related modeling tools. Patterns are used as a basis to compare these tools.

In this student research paper should be examined to which extend the control-flow patterns presented in [1] are supported by IBM WebSphere Integration Developer V6.0 [3] on IBM WebSphere Process Server for Multiplatforms V6.0 [4]. IBM WebSphere Integration Developer uses the Business Process Execution Language for Web Services version 1.1 (BPEL4WS) [2, 7] as the basis but already implements functionality of WS-BPEL version 2.0. Control-flow patterns include basic control patterns, patterns involving multiple instances, state-based patterns and cancellation patterns. Extensive surveys of control-flow patterns have been made in [10] and [11].

The BPEL and the Web Service Description Language (WSDL) source code of the implementations are listed in the appendix. <Assign> activities are mostly used as placeholders for comprehensive business logic and if this is not the case, they are explained in detail.

2 Evaluation of control-flow patterns in IBM WebSphere Integration Developer

2.1 Basic Control Flow Patterns

Pattern 1 (Sequence)

Description: An activity in a workflow process is enabled after the completion of another activity in the same process [1].

Synonyms: Sequential routing, serial routing.

IBM WebSphere Integration Developer supports this pattern directly because two activities can be connected in succession. A typical implementation involves linking two activities with an unconditional control flow arrow as shown in figure 1.

IBM WebSphere Integration Developer uses the *sequence* activity provided by BPEL [2]. A *sequence* activity contains one or more activities that are performed sequentially, in the order in which they are listed within the sequence element, that is, in lexical order. The *sequence* activity completes when the final activity in the sequence has completed [2].

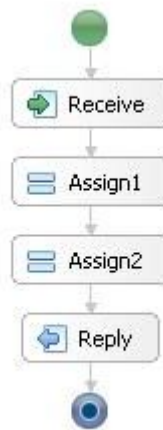


Figure 1: Sequence

Pattern 2 (Parallel Split)

Description: A point in the workflow process where a single thread of control splits into multiple threads of control which can be executed in parallel, thus allowing activities to be executed simultaneously or in any order [1].

Synonyms: AND-split, parallel routing, fork.

IBM WebSphere Integration Developer supports this pattern directly by means of the <parallel activities> construct which uses the `flow` construct provided by BPEL that contains a couple of branches that execute concurrently [2].

An example of implementing this pattern is shown in figure 2. In this example three <assign> activities inside a <parallel activities> construct are started at the same time.

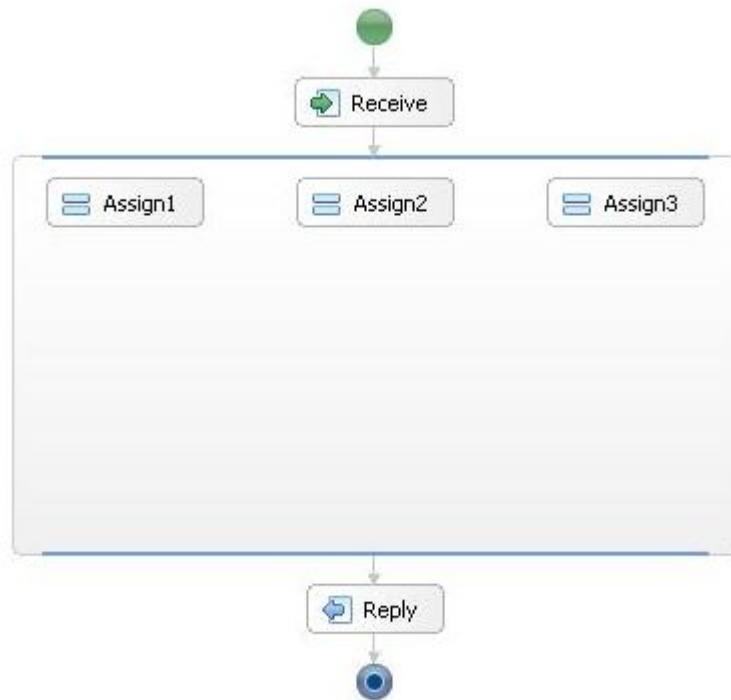


Figure 2: Parallel Split

Pattern 3 (Synchronization)

Description: A point in the workflow process where multiple parallel subprocesses/activities converge into one single thread of control, thus synchronizing multiple threads. It is an assumption of this pattern that each incoming branch of a synchronizer is executed only once [1].

Synonyms: AND-join, rendezvous, synchronizer.

IBM WebSphere Integration Developer supports this pattern directly by means of the <parallel activities> construct which uses the `flow` construct provided by BPEL [2]. The `flow` construct provides synchronization and contains a couple of

branches that execute concurrently [2]. The branches inside the <parallel activities> construct get synchronized as soon as the <parallel activities> construct has finished. The <parallel activities> construct completes when every branch has been skipped or processed.

An example of implementing this pattern is shown in figure 3. In this example three <assign> activities inside a <parallel activities> construct are executed in parallel and each of them may be executed only once. The <reply> activity gets started when all branches have completed. The point of synchronization is located where the control flow arrow starts which points to the <reply> activity.

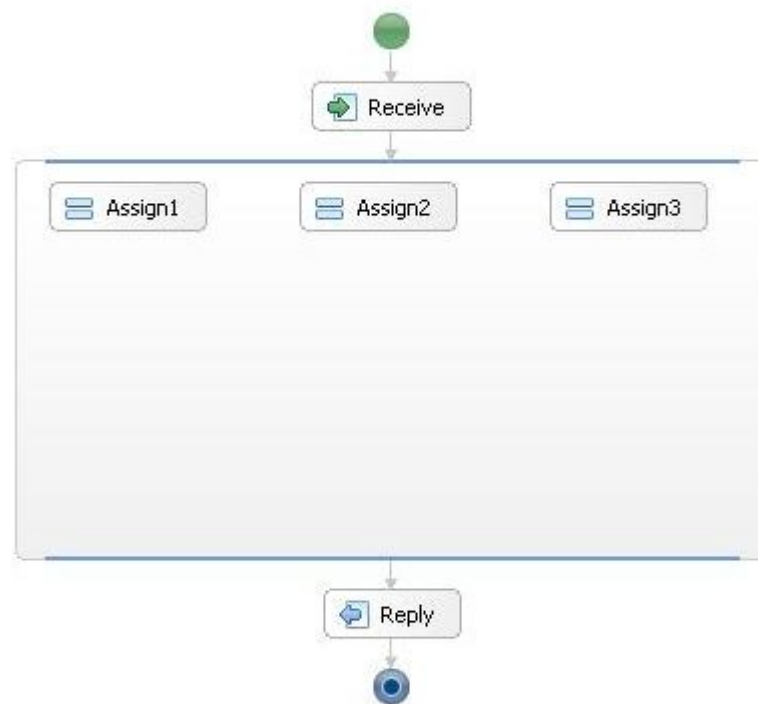


Figure 3: Synchronization

Pattern 4 (Exclusive Choice)

Description: A point in the workflow process where, based on a decision or workflow control data, one of several branches is chosen [1].

Synonyms: XOR-split, conditional routing, switch, decision.

IBM WebSphere Integration Developer supports this pattern directly by means of the <choice> construct which uses the `switch` activity available in BPEL [2] and implements the same semantics. The `switch` construct allows only one branch to be activated and none of the other branches is ever processed in parallel. It uses

an ordered list of one or more conditional branches that are configured by `case` elements and an optional `otherwise` element. The expressions of the conditions are evaluated in the order in which they appear in the BPEL source code and accordingly from left to right in the visual representation as shown in figure 4 and 5. If more than one condition is fulfilled, only the first one is executed. When the chosen branch has finished, the `<choice>` activity completes [2].

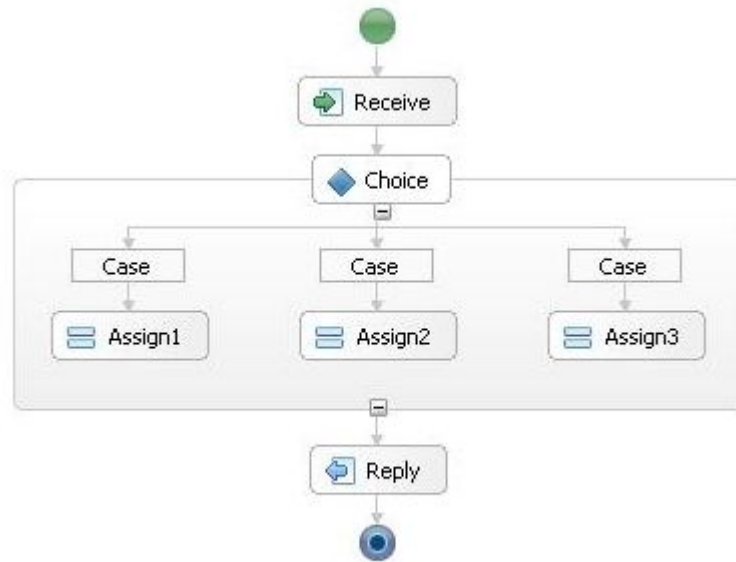


Figure 4: Exclusive Choice

In order to ensure that an activity is executed even if all conditions of available `<case>` branches evaluate to false, you can add an `<otherwise>` branch as shown in figure 5.

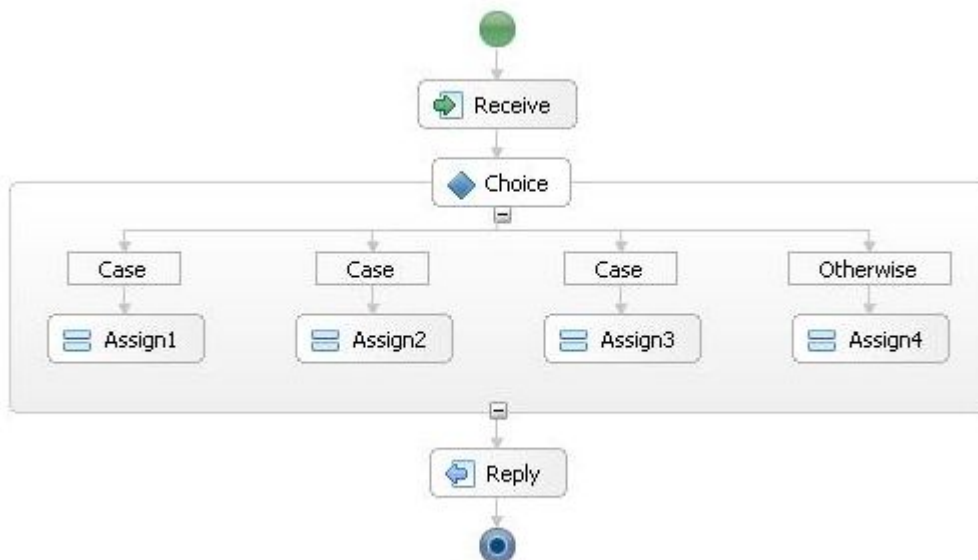


Figure 5: Exclusive Choice with explicit <otherwise> condition

The Exclusive Choice pattern can also be implemented by using links with transition conditions for every branch inside a <parallel activities> construct. Note that all transition conditions have to be disjunctive to ensure that only one branch at maximum is executed. When the chosen branch has finished, the <parallel activities> construct completes. All activities are skipped if none of the conditions evaluate to true and the <parallel activities> construct finishes too.

Pattern 5 (Simple Merge)

Description: A point in the workflow process where two or more alternative branches come together without synchronization. It is an assumption of this pattern that none of the alternative branches is ever executed in parallel [1].

Synonyms: XOR-join, asynchronous join, merge.

According to pattern 4 (Exclusive Choice) IBM WebSphere Integration Developer supports this pattern directly by means of the <choice> construct and alternatively by using links with disjunctive transition conditions inside a <parallel activities> construct.

As shown in figure 6, either <Assign1> or <Assign2> gets activated but they will never be executed concurrently. When the chosen branch has finished or both have been skipped, the <choice> construct completes and the two alternative branches come together. The point of synchronization is located where the control flow arrow starts which points to the <reply> activity. The point of synchronization is located where the control flow arrow starts which points to the <reply> activity.

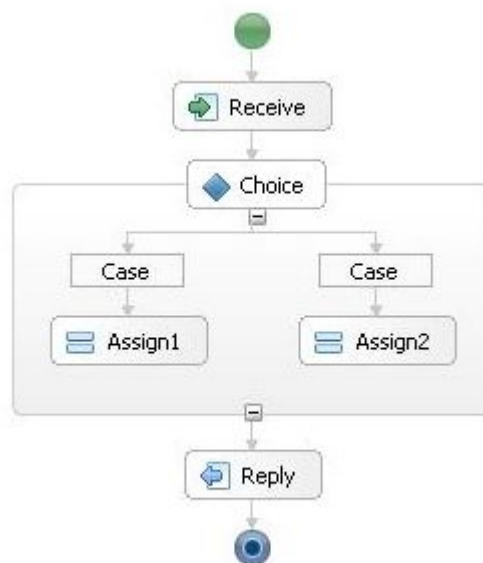


Figure 6: Simple Merge

2.2 Advanced Branching and Synchronization Patterns

Pattern 6 (Multi-choice)

Description: A point in the workflow process where, based on a decision or workflow control data, a number of branches are chosen [1].

Synonyms: Conditional routing, selection, OR-split.

This pattern is supported in IBM WebSphere Integration Developer by using links with transition conditions in every branch inside a <parallel activities> construct. The transition conditions associated with the links are evaluated based on the input provided by the client and do not have to be disjunctive. As a result more than one branch may be chosen at the same time and be executed concurrently. When all chosen branches have completed, the <parallel activities> construct finishes. An example is shown in figure 7 below. If the input data can not be used with any transition condition or every condition evaluates to false, none of the branches is executed.

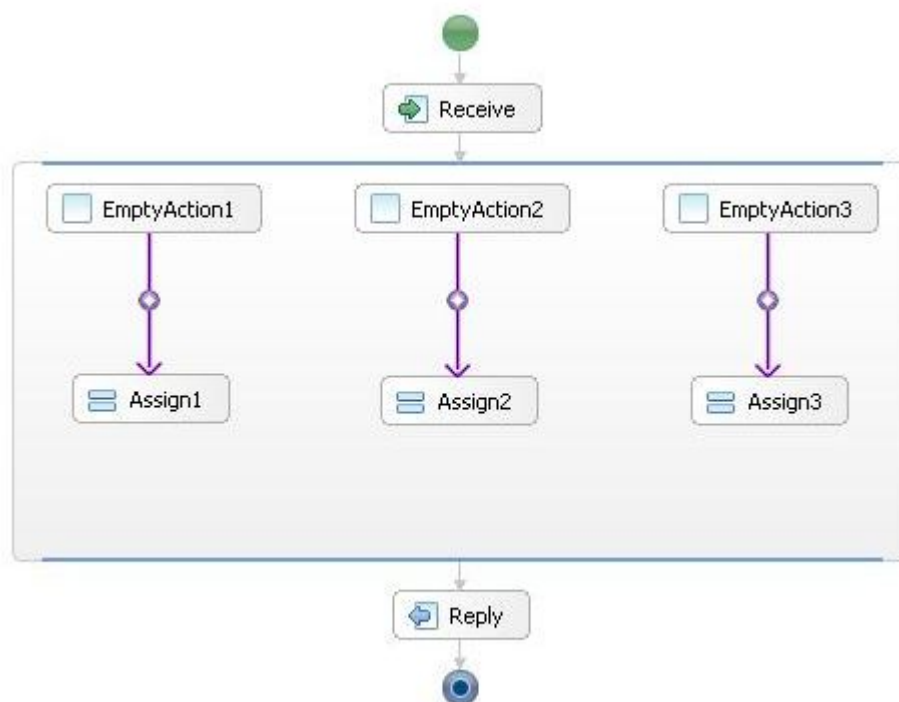


Figure 7: Multi-choice by links with transition conditions and Synchronizing Merge with transition conditions

Pattern 7 (Synchronizing Merge)

Description: A point in the workflow process where multiple paths converge into one single thread. If more than one path is taken, synchronization of the active threads needs to take place. If only one path is taken, the alternative branches should reconverge without synchronization. It is an assumption of this pattern that a branch that has already been activated cannot be activated again while the merge is still waiting for other branches to complete [1].

Synonyms: Synchronizing join.

Corresponding to pattern 6 (Multi-choice) IBM WebSphere Integration Developer supports this pattern by means of the <parallel activities> construct that contains a couple of branches as shown in figure 7. Transition conditions are used to determine which branches are to be activated.

The <parallel activities> construct avoids the problem to decide in which case to synchronize and in which one to merge. The <parallel activities> construct ensures executing or skipping of each branch and synchronizes all of them. It will only complete if every branch has been skipped or has been completed and none of the branches can be executed multiple times.

Pattern 8 (Multi-merge)

Description: A point in a workflow process where two or more branches reconverge without synchronization. If more than one branch gets activated, possibly concurrently, the activity following the merge is started for every activation of every incoming branch [1].

Synonyms: -

IBM WebSphere Integration Developer does not support this pattern directly. A possible solution would look like the one shown in figure 8. The first <assign> activity sets a counter to zero and there are multiple branches inside the <parallel activities> construct. Every branch contains an <empty action> construct which is needed to provide the following <assign> activity with a transition condition. The transition conditions do not have to be disjunctive and several activities may be executed concurrently. The <assign> activity that represents the workflow is linked with another <assign> activity which increases the counter. Finally <Assign4> represents the activity that follows the merge and it is executed as many times as defined in the counter or rather as many times a branch in the <parallel activities> construct was activated.

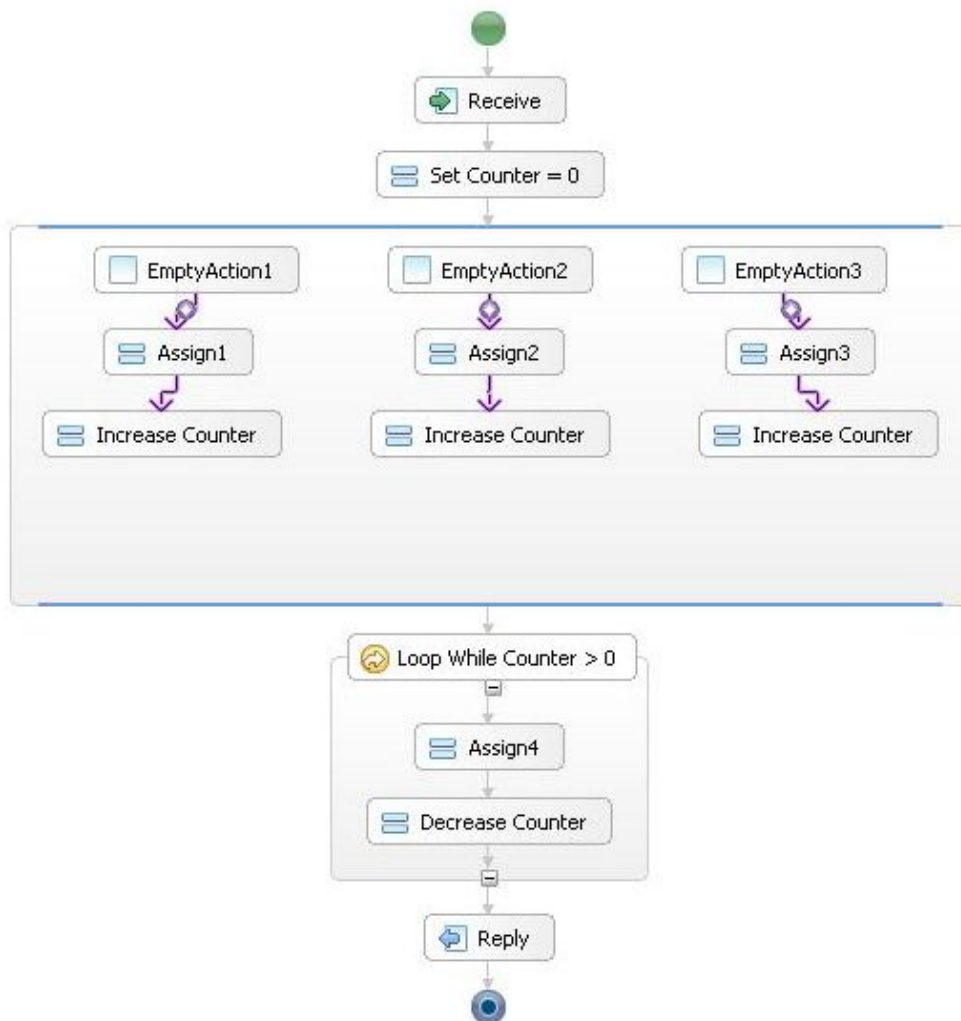


Figure 8: Multi-merge

Pattern 9 (Discriminator)

Description: The discriminator is a point in a workflow process that waits for one of the incoming branches to complete before activating the subsequent activity. From that moment on it waits for all remaining branches to complete and ignores them. Once all incoming branches have been triggered, it resets itself so that it can be triggered again (which is important otherwise it could not really be used in the context of a loop) [1].

Synonyms: -

IBM WebSphere Integration Developer supports this pattern. Figure 9 shows an example of a possible implementation. The <parallel activities> construct contains multiple <assign> activities that are all linked to an <empty action> construct. The join condition of the merging link is not specified whereby it evaluates to the logical OR of all incoming links in IBM WebSphere Integration Developer. Thus <EmptyAction> is processed for one incoming link but nonrecurring as the workflow engine marks the <EmptyAction> activity as processed. The <parallel activities> construct completes as soon as the <empty action> activity has finished.

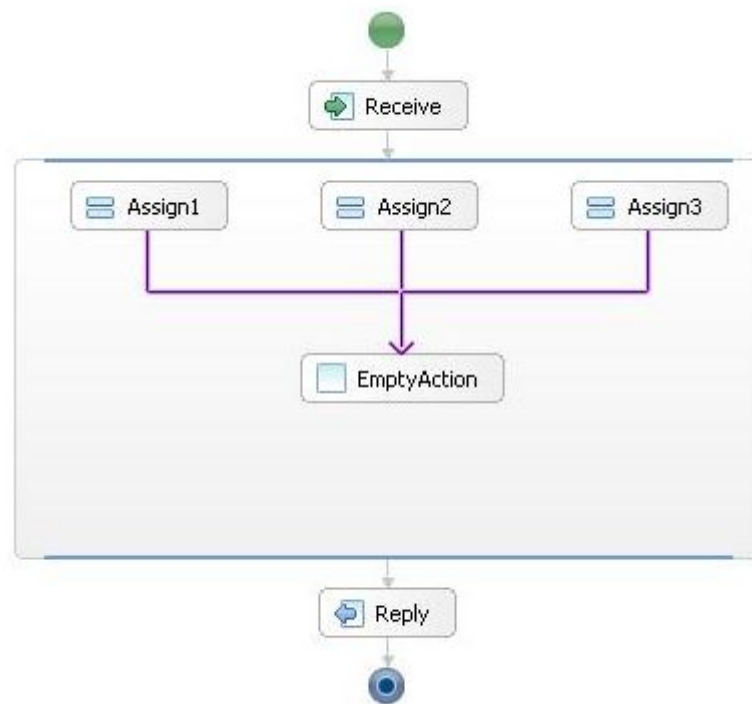


Figure 9: Discriminator

2.3 Structural Patterns

Pattern 10 (Arbitrary Cycles)

Description: A point in a workflow process where one or more activities can be done repeatedly [1].

Synonyms: Loop, iteration, cycle.

IBM WebSphere Integration Developer does not support this pattern. It is feasible to create structured cycles but is not possible to jump into another <while loop> construct and build overlapping and nested cycles. The only possibility to create loops is by using the <while loop> construct which allows only one starting and one ending point and does not allow random jumps into a loop. A link must not cross the boundary of a while activity [2].

Pattern 11 (Implicit Termination)

Description: A given subprocess should be terminated when there is nothing else to be done. In other words, there are no active activities in the workflow and no other activity can be made active (and at the same time the workflow is not in deadlock) [1].

Synonyms: -

IBM WebSphere Integration Developer supports this pattern in the way that a process is automatically finished when all activities in the workflow are processed and there are no activities left. The process as a whole completes normally. In cases where the join condition for an activity turns out to be false, the activity and its following activities are not performed (Dead-Path-Elimination). In IBM WebSphere Integration Developer you can use the attribute `suppressJoinFailure` provided by BPEL [2]. The default value of `suppressJoinFailure` is “no” in order to avoid unexpected behavior and you can add it to a specific activity or to the whole process.

2.4 Patterns involving Multiple Instances

Pattern 12 (Multiple Instances Without Synchronization)

Description: Within the context of a single case (i.e., workflow instance) multiple instances of an activity can be created, i.e., there is a facility to spawn off new threads of control. Each of these threads of control is independent of other threads. Moreover, there is no need to synchronize these threads [1].

Synonyms: Multi threading without synchronization, Spawn off facility.

IBM WebSphere Integration Developer supports this pattern in the way that an

<invoke> construct can be placed inside a <while loop> construct as shown in figure 10 and asynchronously call new instances of a service which are independent of each other and do not respond to the call. Although all instances are created asynchronously, they are all but sequential as they are created in succession by the <while loop> construct.

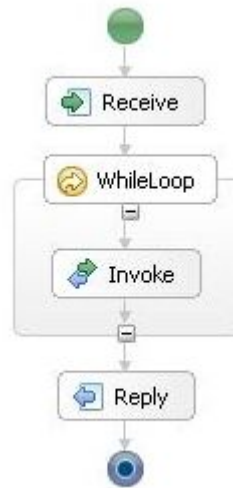


Figure 10: Multiple Instances Without Synchronization

Pattern 13 (Multiple Instances With a Priori Design Time Knowledge)

Description: For one process instance an activity is enabled multiple times. The number of instances of a given activity for a given process instance is known at design time. Once all instances are completed some other activity needs to be started [1].

Synonyms: -

IBM WebSphere Integration Developer supports this pattern by adding as much <invoke> constructs as needed into the body of a <parallel activities> construct. Figure 11 shows an example. Each <invoke> construct enables the same activity. In this way, the number of activated process instances is fixed.

All <invoke> activities should be a synchronous (request/response) operation in order to ensure that the <assign> activity is executed after all created instances have completed. Once all <invoke> constructs have been processed, the <parallel activities> construct finishes and the <assign> activity is started.

There is a <for-each> construct available in WS-BPEL that can be used in future releases of IBM WebSphere Integration Developer to implement this pattern.



Figure 11: Multiple Instances With a Priori Design Time Knowledge

Pattern 14 (Multiple Instances With a Priori Runtime Knowledge)

Description: For one case an activity is enabled multiple times. The number of instances of a given activity for a given case varies and may depend on characteristics of the case or availability of resources, but is known at some stage during runtime, before the instances of that activity have to be created. Once all instances are completed some other activity needs to be started [1].

Synonyms: -

IBM WebSphere Integration Developer does not support this pattern. The problem is that there is no function available which supports a variable quantity of branches in the body of a <parallel activity> construct.

A possible workaround could be to place an <invoke> construct within the body of a <while loop> as shown in figure 12 which corresponds to a serial execution and consequently is not exactly an implementation of the pattern. In this solution it is not possible to create multiple instances in parallel. The <invoke> activity has to be a synchronous (request/response) operation in order to ensure that the <assign> activity is executed after all created instances have completed.

There is a <for-each> construct available in WS-BPEL that can be used in future releases of IBM WebSphere Integration Developer to implement this pattern.

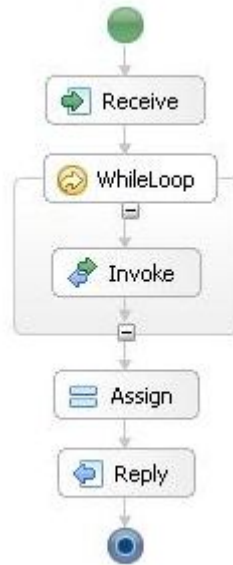


Figure 12: Multiple Instances With a Priori Runtime Knowledge

Pattern 15 (Multiple Instances Without a Priori Runtime Knowledge)

Description: For one case an activity is enabled multiple times. The number of instances of a given activity for a given case is not known during design time, nor is it known at any stage during runtime, before the instances of that activity have to be created. Once all instances are completed some other activity needs to be started. The difference with pattern 14 (Multiple Instances With a Priori Runtime Knowledge) is that even while some of the instances are being executed or already completed, new ones can be created [1].

Synonyms: -

IBM WebSphere Integration Developer offers no direct support for this pattern but a possible workaround can be implemented by using the <receive choice> construct and <while loop> activities as shown in figure 13. <Receive choice> uses the `pick` construct provided by BPEL together with a message handler (`onMessage`) that waits for messages from a particular partner [2]. <Receive choice> will complete as soon as the message handler's activity has finished and it is included in a <while loop> construct which repeats the <receive choice> activity until a the variable 'execute' evaluates to false. This <while loop> ensures that even while some of the instances have already been started, new messages can arrive and yet other instances can be invoked later on.

The operation in <ReceiveChoice> contains another <while loop> which repeats the <invoke> activity as many times as needed for the given case. The <invoke> activity has to be a synchronous (request/response) operation in order to ensure that <Assign> is executed after all created instances have completed.

As the <invoke> construct is included in a <while loop>, this implementation corresponds to a serial execution and the services get invoked in succession. There is a <for-each> construct available in WS-BPEL that can be used in future releases of IBM WebSphere Integration Developer to implement this pattern.

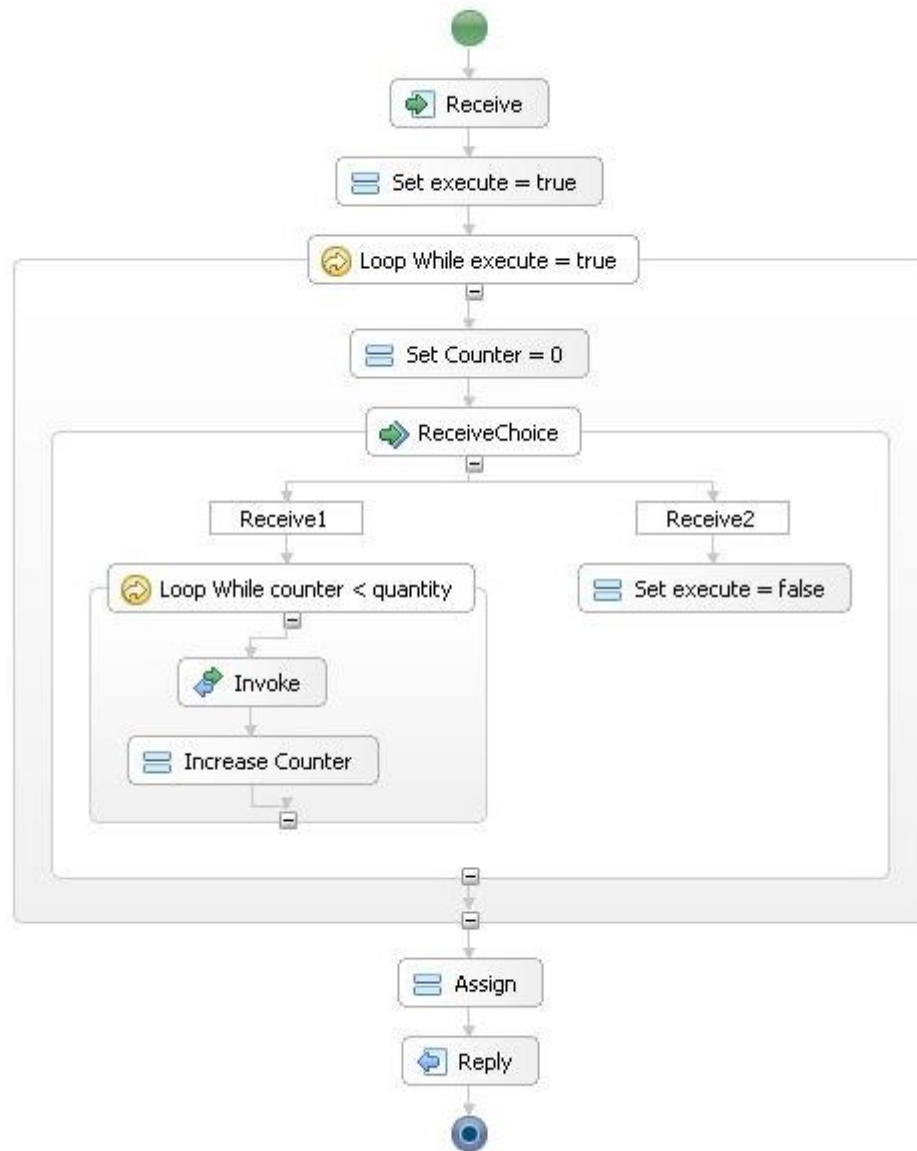


Figure 13: Multiple Instances Without a Priori Runtime Knowledge

2.5 State-based Patterns

Pattern 16 (Deferred Choice)

Description: A point in the workflow process where one of several branches is chosen. In contrast to the XOR-split, the choice is not made explicitly (e.g. based on data or a decision) but several alternatives are offered to the environment. However, in contrast to the AND-split, only one of the alternatives is executed. This means that once the environment activates one of the branches the other alternative branches are withdrawn. It is important to note that the choice is delayed until the processing in one of the alternative branches is actually started, i.e. the moment of choice is as late as possible [1].

Synonyms: -

IBM WebSphere Integration Developer supports this pattern by using the <receive choice> construct as shown in figure 14. <Receive choice> uses the `pick` construct provided by BPEL together with a message handler (`onMessage`) that waits for messages from a particular partner [2]. <Receive choice> derives a message and starts the operation <Receive1> which handles the message. <Receive1> contains a <choice> construct with a couple of cases whose branches are executed based on the content of the message received by <ReceiveChoice>. By using the <choice> construct it is guaranteed that only one of the alternatives gets processed. An additional <choice> construct may be needed and is used in this example because the <receive choice> construct distinguishes between the types of the message or rather different variables and is not able to process the content of a message and the value of a variable respectively.

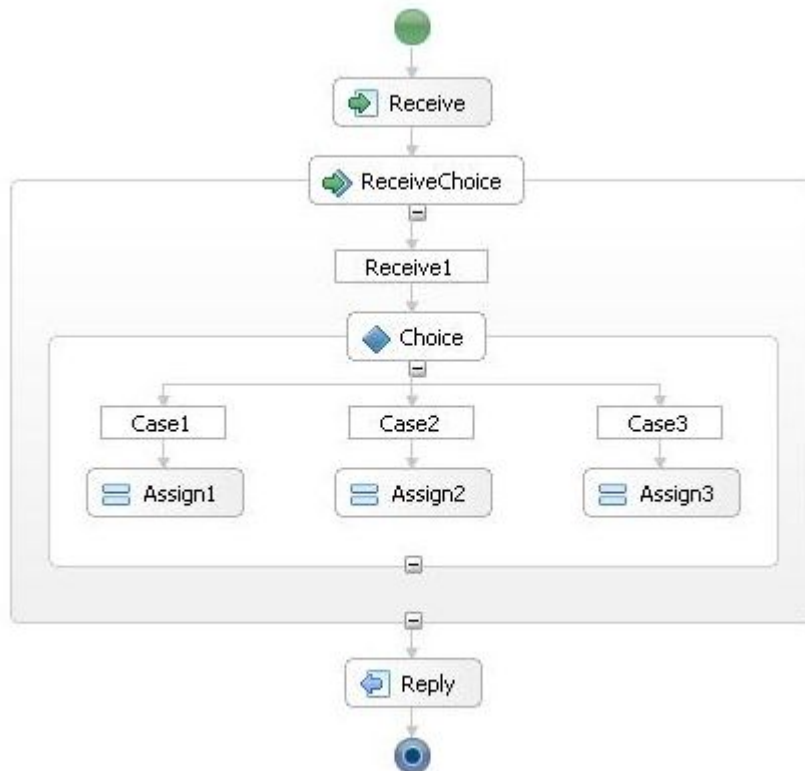


Figure 14: Deferred Choice

Pattern 17 (Interleaved Parallel Routing)

Description: A set of activities is executed in an arbitrary order: Each activity in the set is executed, the order is decided at run-time, and no two activities are executed at the same moment (i.e. no two activities are active for the same workflow instance at the same time) [1].

Synonyms: Unordered sequence.

IBM WebSphere Integration Developer does not support this pattern. The problem is to process activities in a random order.

A possible workaround to this pattern is to define the order of execution and to create a sequence of activities (Pattern 1). As the activities should be executed in an arbitrary order and may not be executed in parallel, it might be an adoptable solution to use only one of the possible sequences. However this solution is not an accurate implementation of the Interleaved Parallel Routing Pattern.

A possible but very complex solution is to use a sequence for every possible order of the activities as cases in a <choice> construct. Before an execution of a branch one sequence has to be selected due to the value of a variable. This is a combination of the sequence and the exclusive choice pattern.

Alternatively you can use a sequence for every possible order of the activities inside a <parallel activities> construct. Every branch must have a transition condition that is disjunct to the other ones to ensure that only one branch is executed. The problem is to select the appropriate sequence and to choose the values of the variables used in the transition conditions respectively. In addition to that, the implementation might become very complex because every possible sequence of actions has to be available and must be created at design time. An adequate solution is to pass an array that contains the order of execution and to use a <while loop> which successively processes the activities as shown in figure 15. The <while loop> continues as many times as there are <assign> activities in the <choice> construct. In every cycle of the loop, the <Assign> activity picks a value out of the array and copies it into a variable. After this the corresponding <assign> activity in the <choice> construct is executed.

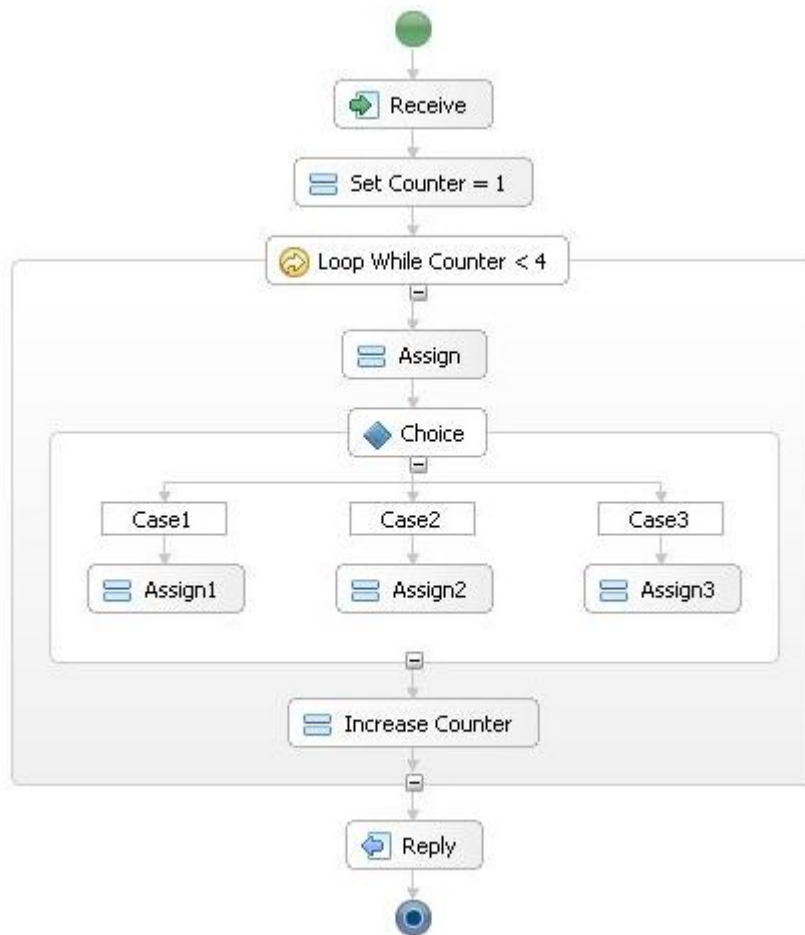


Figure 15: Interleaved Parallel Routing

Pattern 18 (Milestone)

Description: The enabling of an activity depends on the case being in a specified state, i.e. the activity is only enabled if a certain milestone has been reached which did not expire yet. Consider three activities named A, B, and C. Activity A is only enabled if activity B has been executed and C has not been executed yet, i.e. A is not enabled before the execution of B and A is not enabled after the execution of C. The state in between B and C is modeled by place m. This place is a milestone for A [1].

Synonyms: Test arc, deadline, state condition, withdraw message.

IBM WebSphere Integration Developer supports this pattern by using the <receive choice> construct inside a <while loop> activity as shown in figure 16. The first <assign> activity corresponds to activity B and sets the value of a boolean variable named 'milestone' to true. The <while loop> construct repeats the activities inside of it until the variable 'milestone' evaluates to false. In an analogous manner as in pattern 16 (Deferred Choice), <ReceiveChoice> derives a message and starts the operation <Receive> which contains a <choice> construct with two cases whose branches are executed based on the content of the message received by <ReceiveChoice>. If <Case1> is the chosen branch, the <empty action> construct that correspond to activity A is executed and the <while loop> activity continues. If <Case2> is the chosen one, the <assign> activity is executed which sets the value of the variable 'milestone' to false and through this stops the loop. After the <while loop> activity has completed, the <assign> activity which corresponds to activity C is executed.

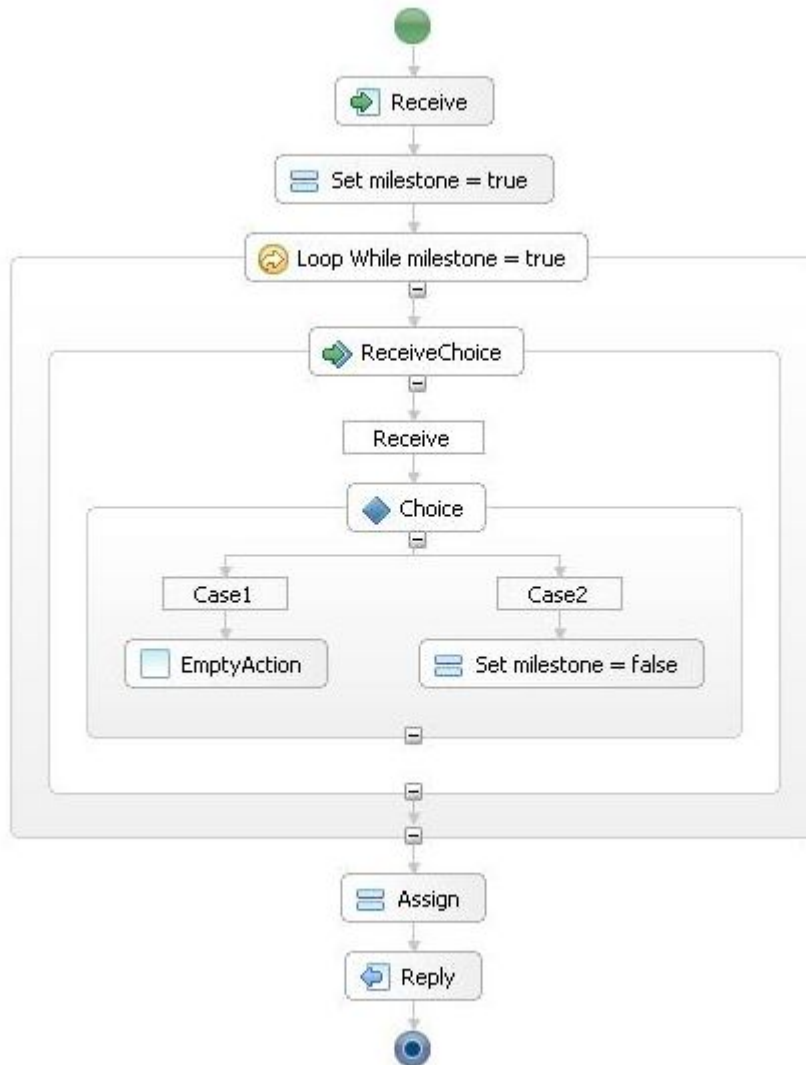


Figure 16: Milestone

2.6 Cancellation Patterns

Pattern 19 (Cancel Activity)

Description: An enabled activity is disabled, i.e. a thread waiting for the execution of an activity is removed [1].

Synonyms: Withdraw activity.

IBM WebSphere Integration Developer supports this pattern by means of the

<terminate> construct in an event handler attached to the activity that should be terminated. The event handler is executed through an external message and the <terminate> activity instantly stops the activity. The process receives the message concurrently with the normal activity of the scope to which the event handler is attached whereby the events can arise at any time.

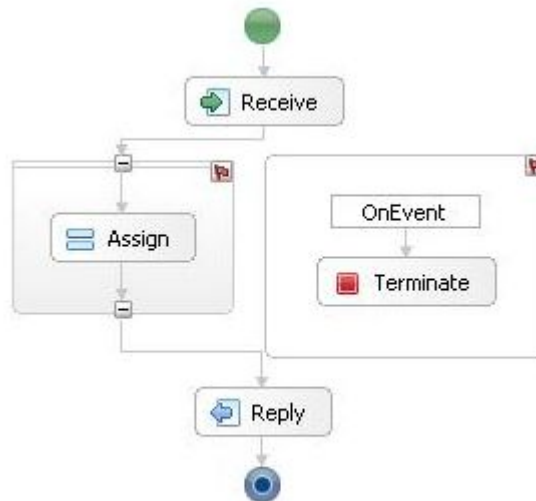


Figure 17: Cancel Activity

Pattern 20 (Cancel Case)

Description: A case, i.e. workflow instance, is removed completely (i.e., even if parts of the process are instantiated multiple times, all descendants are removed) [1].

Synonyms: Withdraw case.

IBM WebSphere Integration Developer supports this pattern by means of the <terminate> construct in an event handler attached to the whole process. The event handler is executed through an external message and the <terminate> activity instantly stops the business process instance. The process receives the message concurrently with the normal activity of the scope to which the event handler is attached whereby the events can arise at any time.

The process instance is explicitly terminated by the <terminate> activity. In this case the termination is abnormal in comparison to pattern 11 (Implicit Termination).

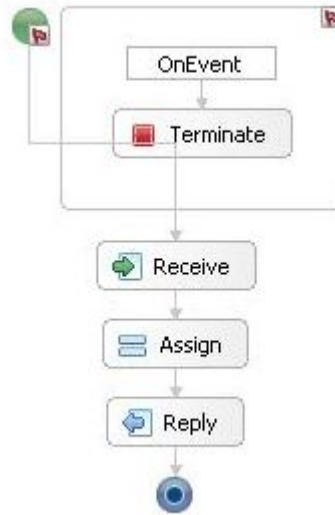


Figure 18: Cancel Case

3 Summary

The evaluation of IBM WebSphere Integration Developer from the control-flow perspective is summarized in Table 1. The evaluation shows that IBM WebSphere Integration Developer supports the majority of control patterns. In many cases where a direct support by a specific construct is not available, a workaround is possible by combining different modules. Merely pattern 10 (Arbitrary Cycles) and pattern 15 (Multiple Instances without a Priori Runtime Knowledge) are not supported and are not possible to implement. In addition to the availability of BPEL implementation, IBM WebSphere Integration Developer offers additive features that allow more patterns to be supported than the original BPEL4WS specification [2] does.

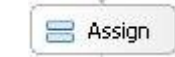

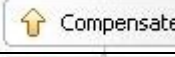
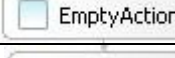
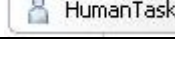
In future releases of IBM WebSphere Integration Developer, an additional <for each> construct should be available to support patterns 13 to 15. In addition to that, another construct might be advisable that allows the creation of multiple instances of a task at run-time for the implementation of pattern 14 like the <flowN> construct provided by Oracle BPEL PM [12].

	Pattern	Score	Annotation
1	Sequence	+	Directly supported by link or sequence.
2	Parallel Split	+	Directly supported by the <parallel activities> construct.
3	Synchronization	+	Directly supported by the <parallel activities> construct.
4	Exclusive Choice	+	Directly supported by the <choice> construct.
5	Simple Merge	+	Directly supported by the <choice> construct.
6	Multi-choice	+	Directly supported by links with transition conditions inside a <parallel activities> construct.
7	Synchronizing Merge	+	Directly supported by links with transition conditions inside a <parallel activities> construct.
8	Multi-merge	+ / -	A workaround is needed by a <while loop> activity and links with transition conditions inside a <parallel activities> construct.
9	Discriminator	+	Directly supported by merging link with an unspecified join condition.
10	Arbitrary Cycles	-	Not supported. There are only structured loops by means of the <while loop> construct possible.
11	Implicit Termination	+	Directly supported by default.
12	Multiple Instances Without Synchronization	+	Directly supported by placing an <invoke> construct inside the body of a <while loop> activity.

	Pattern	Score	Annotation
13	Multiple Instances With a Priori Design Time Knowledge	+	Directly supported by replication of an activity inside a <parallel activity> construct.
14	Multiple Instances With a Priori Runtime Knowledge	-	Not supported. Only serial execution is possible by placing an <invoke> activity inside the body of a <while loop> construct.
15	Multiple Instances Without a Priori Runtime Knowledge	+ / -	A workaround is needed by placing a <receive choice> construct inside the body of a <while loop> activity.
16	Deferred Choice	+	Directly supported by the <receive choice> construct.
17	Interleaved Parallel Routing	+ / -	A workaround is needed by passing an array with the order of execution and using a <choice> construct inside a <while loop> activity.
18	Milestone	+ / -	A workaround is needed by placing a <receive choice> construct inside the body of a <while loop> activity.
19	Cancel Activity	+	Directly supported by placing a <terminate> construct in an event handler of an activity.
20	Cancel Case	+	Directly supported by placing a <terminate> construct in an event handler of the process.

Table 1: Support of control-flow patterns in IBM WebSphere Integration Developer

A list of available activities in IBM WebSphere Integration Developer and the BPEL constructs they use is presented in Table 2. Note that the BPEL elements `switch`, `flow` and `pick` are used by constructs with a different name in IBM WebSphere Integration Developer and that <human task>, <invoke> and <snippet> internally use the `invoke` BPEL construct.

Symbol	Name in WebSphere Integration Developer	Name of corresponding BPEL construct
 Assign	<Assign>	assign
 Choice Case	<Choice>	switch
 Compensate	<Compensate>	compensate
 EmptyAction	<Empty Action>	empty
 HumanTask	<Human Task>	invoke

Symbol	Name in WebSphere Integration Developer	Name of corresponding BPEL construct
	<Invoke>	invoke
	<Parallel Activities>	flow
	<Receive>	receive
	<Receive Choice>	pick
	<Reply>	reply
	<Rethrow>	rethrow
	<Scope>	scope
	<Sequence>	sequence
	<Snippet>	invoke
	<Terminate>	terminate
	<Throw>	throw
	<Wait>	wait
	<While Loop>	while

Table 2: Mapping of symbols and terms used in IBM WebSphere Integration Developer to employed BPEL constructs

As the scope of this evaluation is restricted to control-flow patterns, another evaluation should be performed in future to examine the data patterns and the resource patterns available in [8] and [9].

References

- [1] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Workflow Patterns. *Distributed and Parallel Databases*, 14(3), pages 5-51, July 2003.
- [2] BPEL4WS Specification, Business Process Execution Language for Web Services Version 1.1, 5. May 2003
<http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>
- [3] IBM WebSphere Integration Developer.
<http://www-306.ibm.com/software/integration/wid/>
- [4] IBM WebSphere Process Server.
<http://www-306.ibm.com/software/integration/wps/>
- [5] Active BPELEngine. An Introduction to the ActiveBPEL Engine.
<http://www.activebpel.org/info/intro.html>
- [6] Workflow Patterns. A repository for workflow modeling patterns.
<http://www.workflowpatterns.com>
- [7] P. Wohed, W.M.P. van der Aalst, M. Dumas, and A.H.M. ter Hofstede. Pattern-Based Analysis of BPEL4WS. QUT Technical report, FIT-TR-2002-04, Queensland University of Technology, Brisbane, 2002.
- [8] N. Russell, A.H.M. ter Hofstede, D. Edmond and W.M.P. van der Aalst. Workflow Data Patterns. QUT Technical report, FIT-TR-2004-01, Queensland University of Technology, Brisbane, 2004.
- [9] N. Russell, A.H.M. ter Hofstede, D. Edmond, and W.M.P. van der Aalst. Workflow Resource Patterns. BETA Working Paper Series, WP 127, Eindhoven University of Technology, Eindhoven, 2004.
- [10] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Advanced Workflow Patterns. In O. Etzion and P. Scheuermann, editors, 7th International Conference on Cooperative Information Systems (CoopIS 2000), volume 1901 of Lecture Notes in Computer Science, pages 18-29. Springer-Verlag, Berlin, 2000.
- [11] B. Kiepuszewski, A.H.M. ter Hofstede, and W.M.P. van der Aalst. Fundamentals of Control Flow in Workflows. QUT Technical report, FIT-TR-2002-03, Queensland University of Technology, Brisbane, 2002.
- [12] N. Mulyar. Pattern-based Evaluation of Oracle-BPEL. Department of Technology Management, Eindhoven University of Technology. BPM Center Report BPM-05-24, BPMcenter.org, 2005.

Appendix: BPEL Code

Pattern 1 (Sequence)

The content of the BPEL file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<bpws:process xmlns:bpws="http://schemas.xmlsoap.org/ws/2004/03/business-
process/" xmlns:ns="http://control-flow/sequenceArtifacts"
xmlns:ns0="http://control-flow/sequenceInterface"
xmlns:wpc="http://www.ibm.com/xmlns/prod/websphere/business-
process/6.0.0/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.ibm.com/xmlns/prod/websphere/business-
process/expression-lang/java/6.0.0/" name="sequence"
suppressJoinFailure="yes" targetNamespace="http://control-flow"
wpc:displayName="sequence" wpc:executionMode="microflow" wpc:id="1">
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="sequence.wsdl" namespace="http://control-
flow/sequenceInterface"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="sequenceArtifacts.wsdl" namespace="http://control-
flow/sequenceArtifacts"/>
  <bpws:partnerLinks>
    <bpws:partnerLink myRole="SequenceRole" name="Sequence"
partnerLinkType="ns:SequencePLT"/>
  </bpws:partnerLinks>
  <bpws:variables>
    <bpws:variable name="Input" type="xsd:string" wpc:id="2"/>
    <bpws:variable name="copy1" type="xsd:string" wpc:id="11"/>
    <bpws:variable name="output" type="xsd:string" wpc:id="12"/>
  </bpws:variables>
  <bpws:sequence name="HiddenSequence" wpc:id="1073741827">
    <bpws:receive createInstance="yes" name="Receive"
operation="operation1" partnerLink="Sequence" portType="ns0:Sequence"
wpc:displayName="Receive" wpc:id="4">
      <wpc:output>
        <wpc:parameter name="input1" variable="Input"/>
      </wpc:output>
    </bpws:receive>
    <bpws:assign name="Assign1" wpc:displayName="Assign1" wpc:id="13">
      <bpws:copy>
        <bpws:from variable="Input"/>
        <bpws:to variable="copy1"/>
      </bpws:copy>
    </bpws:assign>
    <bpws:assign name="Assign2" wpc:displayName="Assign2" wpc:id="14">
      <bpws:copy>
        <bpws:from variable="copy1"/>
        <bpws:to variable="output"/>
      </bpws:copy>
    </bpws:assign>
    <bpws:reply name="Reply" operation="operation1"
partnerLink="Sequence" portType="ns0:Sequence" wpc:displayName="Reply"
wpc:id="5">
      <wpc:input>
        <wpc:parameter name="output1" variable="output"/>
      </wpc:input>
    </bpws:reply>
  </bpws:sequence>
</bpws:process>
```

The content of the WSDL file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:tns="http://control-flow/sequenceInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/" name="sequence"
targetNamespace="http://control-flow/sequenceInterface">
  <types>
    <xsd:schema targetNamespace="http://control-flow/sequenceInterface"
xmlns:tns="http://control-flow/sequenceInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:element name="operation1">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="input1" nillable="true"
type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="operation1Response">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="output1" nillable="true"
type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </types>
  <message name="operation1Request">
    <part element="tns:operation1" name="operation1Parameters"/>
  </message>
  <message name="operation1Response">
    <part element="tns:operation1Response" name="operation1Result"/>
  </message>
  <portType name="Sequence">
    <operation name="operation1">
      <input message="tns:operation1Request" name="operation1Request"/>
      <output message="tns:operation1Response"
name="operation1Response"/>
    </operation>
  </portType>
</definitions>
```

Pattern 2 (Parallel Split)

The content of the BPEL file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<bpws:process xmlns:bpws="http://schemas.xmlsoap.org/ws/2004/03/business-
process/" xmlns:ns="http://control-flow/parallel_splitArtifacts"
xmlns:ns0="http://control-flow/parallel_splitInterface"
xmlns:wpc="http://www.ibm.com/xmlns/prod/websphere/business-
process/6.0.0/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.ibm.com/xmlns/prod/websphere/business-
process/expression-lang/java/6.0.0/" name="parallel_split"
suppressJoinFailure="yes" targetNamespace="http://control-flow"
wpc:displayName="parallel_split" wpc:executionMode="microflow"
wpc:id="1">
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
```



```

location="parallel_split.wsdl" namespace="http://control-
flow/parallel_splitInterface"/>
<bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="parallel_splitArtifacts.wsdl" namespace="http://control-
flow/parallel_splitArtifacts"/>
<bpws:partnerLinks>
  <bpws:partnerLink myRole="Parallel_splitRole" name="Parallel_split"
partnerLinkType="ns:Parallel_splitPLT"/>
</bpws:partnerLinks>
<bpws:variables>
  <bpws:variable name="Input" type="xsd:string" wpc:id="2"/>
  <bpws:variable name="copy1" type="xsd:string" wpc:id="10"/>
  <bpws:variable name="copy2" type="xsd:string" wpc:id="11"/>
  <bpws:variable name="copy3" type="xsd:string" wpc:id="12"/>
  <bpws:variable name="output" type="xsd:string" wpc:id="13"/>
</bpws:variables>
<bpws:sequence name="HiddenSequence" wpc:id="1073741827">
  <bpws:receive createInstance="yes" name="Receive"
operation="operation1" partnerLink="Parallel_split"
portType="ns0:Parallel_split" wpc:displayName="Receive" wpc:id="4">
  <wpc:output>
    <wpc:parameter name="input1" variable="Input"/>
  </wpc:output>
</bpws:receive>
<bpws:flow name="ParallelActivities"
wpc:displayName="ParallelActivities" wpc:id="6">
  <bpws:assign name="Assign1" wpc:displayName="Assign1" wpc:id="7">
    <bpws:copy>
      <bpws:from variable="Input"/>
      <bpws:to variable="copy1"/>
    </bpws:copy>
  </bpws:assign>
  <bpws:assign name="Assign2" wpc:displayName="Assign2" wpc:id="8">
    <bpws:copy>
      <bpws:from variable="Input"/>
      <bpws:to variable="copy2"/>
    </bpws:copy>
  </bpws:assign>
  <bpws:assign name="Assign3" wpc:displayName="Assign3" wpc:id="9">
    <bpws:copy>
      <bpws:from variable="Input"/>
      <bpws:to variable="copy3"/>
    </bpws:copy>
  </bpws:assign>
</bpws:flow>
<bpws:reply name="Reply" operation="operation1"
partnerLink="Parallel_split" portType="ns0:Parallel_split"
wpc:displayName="Reply" wpc:id="5">
  <wpc:input>
    <wpc:parameter name="output1" variable="output"/>
  </wpc:input>
</bpws:reply>
</bpws:sequence>
</bpws:process>

```

The content of the WSDL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:tns="http://control-flow/parallel_splitInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/" name="parallel_split"
targetNamespace="http://control-flow/parallel_splitInterface">
  <types>
    <xsd:schema targetNamespace="http://control-
flow/parallel_splitInterface" xmlns:tns="http://control-

```

```

flow/parallel_splitInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="operation1">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="input1" nillable="true"
type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="operation1Response">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="output1" nillable="true"
type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
</types>
<message name="operation1Request">
  <part element="tns:operation1" name="operation1Parameters"/>
</message>
<message name="operation1Response">
  <part element="tns:operation1Response" name="operation1Result"/>
</message>
<portType name="Parallel_split">
  <operation name="operation1">
    <input message="tns:operation1Request" name="operation1Request"/>
    <output message="tns:operation1Response"
name="operation1Response"/>
  </operation>
</portType>
</definitions>

```

Pattern 3 (Synchronization)

The content of the BPEL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<bpws:process xmlns:bpws="http://schemas.xmlsoap.org/ws/2004/03/business-
process/" xmlns:ns="http://control-flow/synchronizationArtifacts"
xmlns:ns0="http://control-flow/synchronizationInterface"
xmlns:wpc="http://www.ibm.com/xmlns/prod/websphere/business-
process/6.0.0/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.ibm.com/xmlns/prod/websphere/business-
process/expression-lang/java/6.0.0/" name="synchronization"
suppressJoinFailure="yes" targetNamespace="http://control-flow"
wpc:displayName="synchronization" wpc:executionMode="microflow"
wpc:id="1">
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="synchronization.wsdl" namespace="http://control-
flow/synchronizationInterface"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="synchronizationArtifacts.wsdl" namespace="http://control-
flow/synchronizationArtifacts"/>
  <bpws:partnerLinks>
    <bpws:partnerLink myRole="SynchronizationRole" name="Synchronization"
partnerLinkType="ns:SynchronizationPLT"/>
  </bpws:partnerLinks>
  <bpws:variables>
    <bpws:variable name="Input" type="xsd:string" wpc:id="2"/>

```

```

    <bpws:variable name="copy1" type="xsd:string" wpc:id="10"/>
    <bpws:variable name="copy2" type="xsd:string" wpc:id="11"/>
    <bpws:variable name="copy3" type="xsd:string" wpc:id="12"/>
    <bpws:variable name="output" type="xsd:string" wpc:id="13"/>
  </bpws:variables>
  <bpws:sequence name="HiddenSequence" wpc:id="1073741827">
    <bpws:receive createInstance="yes" name="Receive"
operation="operation1" partnerLink="Synchronization"
portType="ns0:Synchronization" wpc:displayName="Receive" wpc:id="4">
      <wpc:output>
        <wpc:parameter name="input1" variable="Input"/>
      </wpc:output>
    </bpws:receive>
    <bpws:flow name="ParallelActivities"
wpc:displayName="ParallelActivities" wpc:id="6">
      <bpws:assign name="Assign1" wpc:displayName="Assign1" wpc:id="7">
        <bpws:copy>
          <bpws:from variable="Input"/>
          <bpws:to variable="copy1"/>
        </bpws:copy>
      </bpws:assign>
      <bpws:assign name="Assign2" wpc:displayName="Assign2" wpc:id="8">
        <bpws:copy>
          <bpws:from variable="Input"/>
          <bpws:to variable="copy2"/>
        </bpws:copy>
      </bpws:assign>
      <bpws:assign name="Assign3" wpc:displayName="Assign3" wpc:id="9">
        <bpws:copy>
          <bpws:from variable="Input"/>
          <bpws:to variable="copy3"/>
        </bpws:copy>
      </bpws:assign>
    </bpws:flow>
    <bpws:reply name="Reply" operation="operation1"
partnerLink="Synchronization" portType="ns0:Synchronization"
wpc:displayName="Reply" wpc:id="5">
      <wpc:input>
        <wpc:parameter name="output1" variable="output"/>
      </wpc:input>
    </bpws:reply>
  </bpws:sequence>
</bpws:process>

```

The content of the WSDL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:tns="http://control-flow/synchronizationInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/" name="synchronization"
targetNamespace="http://control-flow/synchronizationInterface">
  <types>
    <xsd:schema targetNamespace="http://control-
flow/synchronizationInterface" xmlns:tns="http://control-
flow/synchronizationInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:element name="operation1">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="input1" nillable="true"
type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="operation1Response">

```

```

        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="output1" nillable="true"
type="xsd:string"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
</types>
<message name="operation1Request">
    <part element="tns:operation1" name="operation1Parameters"/>
</message>
<message name="operation1Response">
    <part element="tns:operation1Response" name="operation1Result"/>
</message>
<portType name="Synchronization">
    <operation name="operation1">
        <input message="tns:operation1Request" name="operation1Request"/>
        <output message="tns:operation1Response"
name="operation1Response"/>
    </operation>
</portType>
</definitions>

```

Pattern 4 (Exclusive Choice) - Exclusive choice with otherwise condition

The content of the BPEL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<bpws:process xmlns:bpws="http://schemas.xmlsoap.org/ws/2004/03/business-
process/" xmlns:ns="http://control-
flow/exclusive_choice_with_otherwise_conditionArtifacts"
xmlns:ns0="http://control-
flow/exclusive_choice_with_otherwise_conditionInterface"
xmlns:wpc="http://www.ibm.com/xmlns/prod/websphere/business-
process/6.0.0/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.ibm.com/xmlns/prod/websphere/business-
process/expression-lang/java/6.0.0/"
name="exclusive_choice_with_otherwise_condition"
suppressJoinFailure="yes" targetNamespace="http://control-flow"
wpc:displayName="exclusive_choice_with_otherwise_condition"
wpc:executionMode="microflow" wpc:id="1">
    <bpws:import importType="http://schemas.xmlsoap.org/wsd1/"
location="exclusive_choice_with_otherwise_condition.wsd1"
namespace="http://control-
flow/exclusive_choice_with_otherwise_conditionInterface"/>
    <bpws:import importType="http://schemas.xmlsoap.org/wsd1/"
location="exclusive_choice_with_otherwise_conditionArtifacts.wsd1"
namespace="http://control-
flow/exclusive_choice_with_otherwise_conditionArtifacts"/>
    <bpws:partnerLinks>
        <bpws:partnerLink
myRole="Exclusive_choice_with_otherwise_conditionRole"
name="Exclusive_choice_with_otherwise_condition"
partnerLinkType="ns:Exclusive_choice_with_otherwise_conditionPLT"/>
    </bpws:partnerLinks>
    <bpws:variables>
        <bpws:variable name="Input" type="xsd:string" wpc:id="2"/>
        <bpws:variable name="copy1" type="xsd:string" wpc:id="16"/>
        <bpws:variable name="copy2" type="xsd:string" wpc:id="17"/>
        <bpws:variable name="copy3" type="xsd:string" wpc:id="18"/>
        <bpws:variable name="copy4" type="xsd:string" wpc:id="19"/>

```

```

    <bpws:variable name="case1" type="xsd:string" wpc:id="20"/>
    <bpws:variable name="case2" type="xsd:string" wpc:id="21"/>
    <bpws:variable name="case3" type="xsd:string" wpc:id="22"/>
  </bpws:variables>
  <bpws:sequence name="HiddenSequence" wpc:id="1073741827">
    <bpws:receive createInstance="yes" name="Receive"
operation="operation1"
partnerLink="Exclusive_choice_with_otherwise_condition"
portType="ns0:Exclusive_choice_with_otherwise_condition"
wpc:displayName="Receive" wpc:id="4">
      <wpc:output>
        <wpc:parameter name="input1" variable="Input"/>
      </wpc:output>
    </bpws:receive>
    <bpws:switch name="Choice" wpc:displayName="Choice" wpc:id="6">
      <bpws:case wpc:id="7">
        <bpws:condition><![CDATA[ java.lang.String __result__2 = "case1";
boolean __result__3;
{// text equal to (ignore case)
  __result__3 = Input.equalsIgnoreCase(__result__2);
}
return __result__3;
        <bpws:assign name="Assign1" wpc:displayName="Assign1"
wpc:id="12">
          <bpws:copy>
            <bpws:from variable="Input"/>
            <bpws:to variable="copy1"/>
          </bpws:copy>
        </bpws:assign>
      </bpws:case>
      <bpws:case wpc:id="8">
        <bpws:condition><![CDATA[ java.lang.String __result__2 = "case2";
boolean __result__3;
{// text equal to (ignore case)
  __result__3 = Input.equalsIgnoreCase(__result__2);
}
return __result__3;
        <bpws:sequence name="HiddenSequence1" wpc:id="1073741833">
          <bpws:assign name="Assign2" wpc:displayName="Assign2"
wpc:id="13">
            <bpws:copy>
              <bpws:from variable="Input"/>
              <bpws:to variable="copy2"/>
            </bpws:copy>
          </bpws:assign>
        </bpws:sequence>
      </bpws:case>
      <bpws:case wpc:id="10">
        <bpws:condition><![CDATA[ java.lang.String __result__2 = "case3";
boolean __result__3;
{// text equal to (ignore case)
  __result__3 = Input.equalsIgnoreCase(__result__2);
}
return __result__3;
        <bpws:sequence name="HiddenSequence2" wpc:id="1073741835">
          <bpws:assign name="Assign3" wpc:displayName="Assign3"
wpc:id="14">
            <bpws:copy>
              <bpws:from variable="Input"/>
              <bpws:to variable="copy3"/>
            </bpws:copy>
          </bpws:assign>
        </bpws:sequence>
      </bpws:case>
      <bpws:otherwise>
        <bpws:assign name="Assign4" wpc:displayName="Assign4"
wpc:id="15">

```

```

        <bpws:copy>
            <bpws:from variable="Input"/>
            <bpws:to variable="copy4"/>
        </bpws:copy>
    </bpws:assign>
    </bpws:otherwise>
</bpws:switch>
    <bpws:reply name="Reply" operation="operation1"
partnerLink="Exclusive_choice_with_otherwise_condition"
portType="ns0:Exclusive_choice_with_otherwise_condition"
wpc:displayName="Reply" wpc:id="5">
        <wpc:input>
            <wpc:parameter name="output1" variable="Input"/>
        </wpc:input>
    </bpws:reply>
</bpws:sequence>
</bpws:process>

```

The content of the WSDL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:tns="http://control-
flow/exclusive_choice_with_otherwise_conditionInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/"
name="exclusive_choice_with_otherwise_condition"
targetNamespace="http://control-
flow/exclusive_choice_with_otherwise_conditionInterface">
    <types>
        <xsd:schema targetNamespace="http://control-
flow/exclusive_choice_with_otherwise_conditionInterface"
xmlns:tns="http://control-
flow/exclusive_choice_with_otherwise_conditionInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
            <xsd:element name="operation1">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="input1" nillable="true"
type="xsd:string"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="operation1Response">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="output1" nillable="true"
type="xsd:string"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:schema>
    </types>
    <message name="operation1Request">
        <part element="tns:operation1" name="operation1Parameters"/>
    </message>
    <message name="operation1Response">
        <part element="tns:operation1Response" name="operation1Result"/>
    </message>
    <portType name="Exclusive_choice_with_otherwise_condition">
        <operation name="operation1">
            <input message="tns:operation1Request" name="operation1Request"/>
            <output message="tns:operation1Response"
name="operation1Response"/>
        </operation>
    </portType>

```

```
</definitions>
```

Pattern 5 (Simple Merge)

The content of the BPEL file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<bpws:process xmlns:bpws="http://schemas.xmlsoap.org/ws/2004/03/business-
process/" xmlns:ns="http://control-flow/simple_mergeArtifacts"
xmlns:ns0="http://control-flow/simple_mergeInterface"
xmlns:wpc="http://www.ibm.com/xmlns/prod/websphere/business-
process/6.0.0/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.ibm.com/xmlns/prod/websphere/business-
process/expression-lang/java/6.0.0/" name="simple_merge"
suppressJoinFailure="yes" targetNamespace="http://control-flow"
wpc:displayName="simple_merge" wpc:executionMode="microflow" wpc:id="1">
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="simple_merge.wsdl" namespace="http://control-
flow/simple_mergeInterface"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="simple_mergeArtifacts.wsdl" namespace="http://control-
flow/simple_mergeArtifacts"/>
  <bpws:partnerLinks>
    <bpws:partnerLink myRole="Simple_mergeRole" name="Simple_merge"
partnerLinkType="ns:Simple_mergePLT"/>
  </bpws:partnerLinks>
  <bpws:variables>
    <bpws:variable name="Input" type="xsd:string" wpc:id="2"/>
    <bpws:variable name="copy1" type="xsd:string" wpc:id="12"/>
    <bpws:variable name="copy2" type="xsd:string" wpc:id="13"/>
    <bpws:variable name="case1" type="xsd:string" wpc:id="14"/>
    <bpws:variable name="case2" type="xsd:string" wpc:id="15"/>
    <bpws:variable name="output" type="xsd:string" wpc:id="16"/>
  </bpws:variables>
  <bpws:sequence name="HiddenSequence" wpc:id="1073741827">
    <bpws:receive createInstance="yes" name="Receive"
operation="operation1" partnerLink="Simple_merge"
portType="ns0:Simple_merge" wpc:displayName="Receive" wpc:id="4">
      <wpc:output>
        <wpc:parameter name="input1" variable="Input"/>
      </wpc:output>
    </bpws:receive>
    <bpws:switch name="Choice" wpc:displayName="Choice" wpc:id="6">
      <bpws:case wpc:id="7">
        <bpws:condition><![CDATA[ java.lang.String __result__2 = "Case1";
boolean __result__3;
{// text equal to (ignore case)
  __result__3 = Input.equalsIgnoreCase(__result__2);
}
return __result__3;
<bpws:assign name="Assign1" wpc:displayName="Assign1"
wpc:id="10">
      <bpws:copy>
        <bpws:from variable="Input"/>
        <bpws:to variable="copy1"/>
      </bpws:copy>
    </bpws:assign>
  </bpws:case>
  <bpws:case wpc:id="8">
    <bpws:condition><![CDATA[ java.lang.String __result__2 = "Case2";
boolean __result__3;
{// text equal to (ignore case)
```

```

        __result__3 = Input.equalsIgnoreCase(__result__2);
    }
    return __result__3;
    <bpws:sequence name="HiddenSequence1" wpc:id="1073741833">
        <bpws:assign name="Assign2" wpc:displayName="Assign2"
wpc:id="11">
            <bpws:copy>
                <bpws:from variable="Input"/>
                <bpws:to variable="copy2"/>
            </bpws:copy>
        </bpws:assign>
    </bpws:sequence>
</bpws:case>
</bpws:switch>
<bpws:reply name="Reply" operation="operation1"
partnerLink="Simple_merge" portType="ns0:Simple_merge"
wpc:displayName="Reply" wpc:id="5">
    <wpc:input>
        <wpc:parameter name="output1" variable="output"/>
    </wpc:input>
</bpws:reply>
</bpws:sequence>
</bpws:process>

```

The content of the WSDL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:tns="http://control-flow/simple_mergeInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/" name="simple_merge"
targetNamespace="http://control-flow/simple_mergeInterface">
    <types>
        <xsd:schema targetNamespace="http://control-
flow/simple_mergeInterface" xmlns:tns="http://control-
flow/simple_mergeInterface" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
            <xsd:element name="operation1">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="input1" nillable="true"
type="xsd:string"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="operation1Response">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="output1" nillable="true"
type="xsd:string"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:schema>
    </types>
    <message name="operation1Request">
        <part element="tns:operation1" name="operation1Parameters"/>
    </message>
    <message name="operation1Response">
        <part element="tns:operation1Response" name="operation1Result"/>
    </message>
    <portType name="Simple_merge">
        <operation name="operation1">
            <input message="tns:operation1Request" name="operation1Request"/>
            <output message="tns:operation1Response"
name="operation1Response"/>
        </operation>

```



```

    </portType>
</definitions>

```

Pattern 6 (Multi-choice)

The content of the BPEL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<bpws:process xmlns:bpws="http://schemas.xmlsoap.org/ws/2004/03/business-
process/" xmlns:ns="http://control-flow/multi-choiceArtifacts"
xmlns:ns0="http://control-flow/multi-choiceInterface"
xmlns:wpc="http://www.ibm.com/xmlns/prod/websphere/business-
process/6.0.0/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.ibm.com/xmlns/prod/websphere/business-
process/expression-lang/java/6.0.0/" name="multi-choice"
suppressJoinFailure="yes" targetNamespace="http://control-flow"
wpc:displayName="multi-choice" wpc:executionMode="microflow" wpc:id="1">
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="multi-choice.wsdl" namespace="http://control-flow/multi-
choiceInterface"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="multi-choiceArtifacts.wsdl" namespace="http://control-
flow/multi-choiceArtifacts"/>
  <bpws:partnerLinks>
    <bpws:partnerLink myRole="Multi-choiceRole" name="Multichoice"
partnerLinkType="ns:MultichoicePLT"/>
  </bpws:partnerLinks>
  <bpws:variables>
    <bpws:variable name="Input" type="xsd:string" wpc:id="2"/>
    <bpws:variable name="copy1" type="xsd:string" wpc:id="7"/>
    <bpws:variable name="copy2" type="xsd:string" wpc:id="8"/>
    <bpws:variable name="copy3" type="xsd:string" wpc:id="9"/>
    <bpws:variable name="cond1" type="xsd:string" wpc:id="10"/>
    <bpws:variable name="cond2" type="xsd:string" wpc:id="11"/>
    <bpws:variable name="cond3" type="xsd:string" wpc:id="12"/>
    <bpws:variable name="output" type="xsd:string" wpc:id="13"/>
  </bpws:variables>
  <bpws:sequence name="HiddenSequence" wpc:id="1073741827">
    <bpws:receive createInstance="yes" name="Receive"
operation="operation1" partnerLink="Multichoice" portType="ns0:Multi-
choice" wpc:displayName="Receive" wpc:id="4">
      <wpc:output>
        <wpc:parameter name="input1" variable="Input"/>
      </wpc:output>
    </bpws:receive>
    <bpws:flow name="ParallelActivities"
wpc:displayName="ParallelActivities" wpc:id="14">
      <bpws:links>
        <bpws:link name="Link1" wpc:id="21"/>
        <bpws:link name="Link2" wpc:id="22"/>
        <bpws:link name="Link3" wpc:id="23"/>
      </bpws:links>
      <bpws:empty name="EmptyAction1" wpc:displayName="EmptyAction1"
wpc:id="15">
        <bpws:sources>
          <bpws:source linkName="Link1">
            <bpws:transitionCondition><![CDATA[boolean __result__3;
{// text equal to (ignore case)
  __result__3 = Input.equalsIgnoreCase(cond1);
}
return __result__3;
</bpws:source>

```

```

        </bpws:sources>
    </bpws:empty>
    <bpws:empty name="EmptyAction2" wpc:displayName="EmptyAction2"
wpc:id="16">
        <bpws:sources>
            <bpws:source linkName="Link2">
                <bpws:transitionCondition><![CDATA[boolean __result__3;
{// text equal to (ignore case)
    __result__3 = Input.equalsIgnoreCase(cond2);
}
return __result__3;
        </bpws:source>
    </bpws:sources>
    </bpws:empty>
    <bpws:empty name="EmptyAction3" wpc:displayName="EmptyAction3"
wpc:id="17">
        <bpws:sources>
            <bpws:source linkName="Link3">
                <bpws:transitionCondition><![CDATA[boolean __result__3;
{// text equal to (ignore case)
    __result__3 = Input.equalsIgnoreCase(cond3);
}
return __result__3;
        </bpws:source>
    </bpws:sources>
    </bpws:empty>
    <bpws:assign name="Assign1" wpc:displayName="Assign1" wpc:id="18">
        <bpws:targets>
            <bpws:target linkName="Link1"/>
        </bpws:targets>
        <bpws:copy>
            <bpws:from variable="Input"/>
            <bpws:to variable="copy1"/>
        </bpws:copy>
    </bpws:assign>
    <bpws:assign name="Assign2" wpc:displayName="Assign2" wpc:id="19">
        <bpws:targets>
            <bpws:target linkName="Link2"/>
        </bpws:targets>
        <bpws:copy>
            <bpws:from variable="Input"/>
            <bpws:to variable="copy2"/>
        </bpws:copy>
    </bpws:assign>
    <bpws:assign name="Assign3" wpc:displayName="Assign3" wpc:id="20">
        <bpws:targets>
            <bpws:target linkName="Link3"/>
        </bpws:targets>
        <bpws:copy>
            <bpws:from variable="Input"/>
            <bpws:to variable="copy3"/>
        </bpws:copy>
    </bpws:assign>
    </bpws:flow>
    <bpws:reply name="Reply" operation="operation1"
partnerLink="Multichoice" portType="ns0:Multi-choice"
wpc:displayName="Reply" wpc:id="5">
        <wpc:input>
            <wpc:parameter name="output1" variable="output"/>
        </wpc:input>
    </bpws:reply>
    </bpws:sequence>
</bpws:process>

```

The content of the WSDL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:tns="http://control-flow/multi-choiceInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/" name="multi-choice"
targetNamespace="http://control-flow/multi-choiceInterface">
  <types>
    <xsd:schema targetNamespace="http://control-flow/multi-
choiceInterface" xmlns:tns="http://control-flow/multi-choiceInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:element name="operation1">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="input1" nillable="true"
type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="operation1Response">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="output1" nillable="true"
type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </types>
  <message name="operation1Request">
    <part element="tns:operation1" name="operation1Parameters"/>
  </message>
  <message name="operation1Response">
    <part element="tns:operation1Response" name="operation1Result"/>
  </message>
  <portType name="Multi-choice">
    <operation name="operation1">
      <input message="tns:operation1Request" name="operation1Request"/>
      <output message="tns:operation1Response"
name="operation1Response"/>
    </operation>
  </portType>
</definitions>

```

Pattern 7 (Synchronizing Merge)

The content of the BPEL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<bpws:process xmlns:bpws="http://schemas.xmlsoap.org/ws/2004/03/business-
process/" xmlns:ns="http://control-flow/synchronizing_mergeArtifacts"
xmlns:ns0="http://control-flow/synchronizing_mergeInterface"
xmlns:wpc="http://www.ibm.com/xmlns/prod/websphere/business-
process/6.0.0/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.ibm.com/xmlns/prod/websphere/business-
process/expression-lang/java/6.0.0/" name="synchronizing_merge"
suppressJoinFailure="yes" targetNamespace="http://control-flow"
wpc:displayName="synchronizing_merge" wpc:executionMode="microflow"
wpc:id="1">
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="synchronizing_merge.wsdl" namespace="http://control-
flow/synchronizing_mergeInterface"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="synchronizing_mergeArtifacts.wsdl" namespace="http://control-

```

```

flow/synchronizing_mergeArtifacts"/>
  <bpws:partnerLinks>
    <bpws:partnerLink myRole="Synchronizing_mergeRole"
name="Synchronizing_merge" partnerLinkType="ns:Synchronizing_mergePLT"/>
  </bpws:partnerLinks>
  <bpws:variables>
    <bpws:variable name="Input" type="xsd:string" wpc:id="2"/>
    <bpws:variable name="copy1" type="xsd:string" wpc:id="16"/>
    <bpws:variable name="copy2" type="xsd:string" wpc:id="17"/>
    <bpws:variable name="copy3" type="xsd:string" wpc:id="18"/>
    <bpws:variable name="cond1" type="xsd:string" wpc:id="19"/>
    <bpws:variable name="cond2" type="xsd:string" wpc:id="20"/>
    <bpws:variable name="cond3" type="xsd:string" wpc:id="21"/>
    <bpws:variable name="Output" type="xsd:string" wpc:id="22"/>
  </bpws:variables>
  <bpws:sequence name="HiddenSequence" wpc:id="1073741827">
    <bpws:receive createInstance="yes" name="Receive"
operation="operation1" partnerLink="Synchronizing_merge"
portType="ns0:Synchronizing_merge" wpc:displayName="Receive" wpc:id="4">
      <wpc:output>
        <wpc:parameter name="input1" variable="Input"/>
      </wpc:output>
    </bpws:receive>
    <bpws:flow name="ParallelActivities"
wpc:displayName="ParallelActivities" wpc:id="6">
      <bpws:links>
        <bpws:link name="Link1" wpc:id="13"/>
        <bpws:link name="Link2" wpc:id="14"/>
        <bpws:link name="Link3" wpc:id="15"/>
      </bpws:links>
      <bpws:empty name="EmptyAction1" wpc:displayName="EmptyAction1"
wpc:id="7">
        <bpws:sources>
          <bpws:source linkName="Link1">
            <bpws:transitionCondition><![CDATA[boolean __result__3;
{// text equal to (ignore case)
__result__3 = Input.equalsIgnoreCase(cond1);
}
return __result__3;
          </bpws:source>
        </bpws:sources>
      </bpws:empty>
      <bpws:empty name="EmptyAction2" wpc:displayName="EmptyAction2"
wpc:id="8">
        <bpws:sources>
          <bpws:source linkName="Link2">
            <bpws:transitionCondition><![CDATA[boolean __result__3;
{// text equal to (ignore case)
__result__3 = Input.equalsIgnoreCase(cond2);
}
return __result__3;
          </bpws:source>
        </bpws:sources>
      </bpws:empty>
      <bpws:empty name="EmptyAction3" wpc:displayName="EmptyAction3"
wpc:id="9">
        <bpws:sources>
          <bpws:source linkName="Link3">
            <bpws:transitionCondition><![CDATA[boolean __result__3;
{// text equal to (ignore case)
__result__3 = Input.equalsIgnoreCase(cond3);
}
return __result__3;
          </bpws:source>
        </bpws:sources>
      </bpws:empty>
      <bpws:assign name="Assign1" wpc:displayName="Assign1" wpc:id="10">

```

```

        <bpws:targets>
            <bpws:target linkName="Link1" />
        </bpws:targets>
        <bpws:copy>
            <bpws:from variable="Input" />
            <bpws:to variable="copy1" />
        </bpws:copy>
    </bpws:assign>
    <bpws:assign name="Assign2" wpc:displayName="Assign2" wpc:id="11">
        <bpws:targets>
            <bpws:target linkName="Link2" />
        </bpws:targets>
        <bpws:copy>
            <bpws:from variable="Input" />
            <bpws:to variable="copy2" />
        </bpws:copy>
    </bpws:assign>
    <bpws:assign name="Assign3" wpc:displayName="Assign3" wpc:id="12">
        <bpws:targets>
            <bpws:target linkName="Link3" />
        </bpws:targets>
        <bpws:copy>
            <bpws:from variable="Input" />
            <bpws:to variable="copy3" />
        </bpws:copy>
    </bpws:assign>
</bpws:flow>
    <bpws:reply name="Reply" operation="operation1"
partnerLink="Synchronizing_merge" portType="ns0:Synchronizing_merge"
wpc:displayName="Reply" wpc:id="5">
        <wpc:input>
            <wpc:parameter name="output1" variable="Output" />
        </wpc:input>
    </bpws:reply>
</bpws:sequence>
</bpws:process>

```

The content of the WSDL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:tns="http://control-flow/synchronizing_mergeInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/" name="synchronizing_merge"
targetNamespace="http://control-flow/synchronizing_mergeInterface">
    <types>
        <xsd:schema targetNamespace="http://control-
flow/synchronizing_mergeInterface" xmlns:tns="http://control-
flow/synchronizing_mergeInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
            <xsd:element name="operation1">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="input1" nillable="true"
type="xsd:string" />
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="operation1Response">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="output1" nillable="true"
type="xsd:string" />
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:schema>
    </types>

```

```

    </xsd:schema>
</types>
<message name="operation1Request">
  <part element="tns:operation1" name="operation1Parameters"/>
</message>
<message name="operation1Response">
  <part element="tns:operation1Response" name="operation1Result"/>
</message>
<portType name="Synchronizing_merge">
  <operation name="operation1">
    <input message="tns:operation1Request" name="operation1Request"/>
    <output message="tns:operation1Response"
name="operation1Response"/>
  </operation>
</portType>
</definitions>

```

Pattern 8 (Multi-merge)

The content of the BPEL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<bpws:process xmlns:bpws="http://schemas.xmlsoap.org/ws/2004/03/business-
process/" xmlns:ns="http://control-flow/multi-mergeArtifacts"
xmlns:ns0="http://control-flow/multi-mergeInterface"
xmlns:wpc="http://www.ibm.com/xmlns/prod/websphere/business-
process/6.0.0/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.ibm.com/xmlns/prod/websphere/business-
process/expression-lang/java/6.0.0/" name="multi-merge"
suppressJoinFailure="yes" targetNamespace="http://control-flow"
wpc:autonomy="peer" wpc:displayName="multi-merge" wpc:id="1">
  <bpws:import importType="http://schemas.xmlsoap.org/wSDL/"
location="multi-merge.wsdl" namespace="http://control-flow/multi-
mergeInterface"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wSDL/"
location="multi-mergeArtifacts.wsdl" namespace="http://control-
flow/multi-mergeArtifacts"/>
  <bpws:partnerLinks>
    <bpws:partnerLink myRole="Multi-mergeRole" name="Multimerge"
partnerLinkType="ns:MultimergePLT"/>
  </bpws:partnerLinks>
  <bpws:variables>
    <bpws:variable name="Input" type="xsd:string" wpc:id="2"/>
    <bpws:variable name="output" type="xsd:string" wpc:id="6"/>
    <bpws:variable name="copy1" type="xsd:string" wpc:id="12"/>
    <bpws:variable name="copy2" type="xsd:string" wpc:id="16"/>
    <bpws:variable name="copy3" type="xsd:string" wpc:id="17"/>
    <bpws:variable name="cond1" type="xsd:boolean" wpc:id="25"/>
    <bpws:variable name="cond2" type="xsd:boolean" wpc:id="26"/>
    <bpws:variable name="cond3" type="xsd:boolean" wpc:id="27"/>
    <bpws:variable name="counter" type="xsd:int" wpc:id="30"/>
  </bpws:variables>
  <bpws:sequence name="HiddenSequence" wpc:id="1073741827">
    <bpws:receive createInstance="yes" name="Receive"
operation="operation1" partnerLink="Multimerge" portType="ns0:Multi-
merge" wpc:displayName="Receive" wpc:id="4">
      <wpc:output>
        <wpc:parameter name="input1" variable="Input"/>
      </wpc:output>
    </bpws:receive>
    <bpws:assign name="SetCounter0" wpc:displayName="Set Counter = 0"
wpc:id="29">

```

```

    <bpws:copy>
      <bpws:from>0</bpws:from>
      <bpws:to variable="counter"/>
    </bpws:copy>
  </bpws:assign>
  <bpws:flow name="ParallelActivities"
wpc:displayName="ParallelActivities" wpc:id="10">
    <bpws:links>
      <bpws:link name="Link1" wpc:id="22"/>
      <bpws:link name="Link2" wpc:id="23"/>
      <bpws:link name="Link3" wpc:id="28"/>
      <bpws:link name="Link4" wpc:id="35"/>
      <bpws:link name="Link5" wpc:id="36"/>
      <bpws:link name="Link6" wpc:id="37"/>
    </bpws:links>
    <bpws:assign name="Assign1" wpc:displayName="Assign1" wpc:id="13">
      <bpws:targets>
        <bpws:target linkName="Link1"/>
      </bpws:targets>
      <bpws:sources>
        <bpws:source linkName="Link4"/>
      </bpws:sources>
      <bpws:copy>
        <bpws:from variable="Input"/>
        <bpws:to variable="copy1"/>
      </bpws:copy>
    </bpws:assign>
    <bpws:assign name="Assign2" wpc:displayName="Assign2" wpc:id="14">
      <bpws:targets>
        <bpws:target linkName="Link2"/>
      </bpws:targets>
      <bpws:sources>
        <bpws:source linkName="Link5"/>
      </bpws:sources>
      <bpws:copy>
        <bpws:from variable="Input"/>
        <bpws:to variable="copy2"/>
      </bpws:copy>
    </bpws:assign>
    <bpws:assign name="Assign3" wpc:displayName="Assign3" wpc:id="15">
      <bpws:targets>
        <bpws:target linkName="Link3"/>
      </bpws:targets>
      <bpws:sources>
        <bpws:source linkName="Link6"/>
      </bpws:sources>
      <bpws:copy>
        <bpws:from variable="Input"/>
        <bpws:to variable="copy3"/>
      </bpws:copy>
    </bpws:assign>
    <bpws:empty name="EmptyAction1" wpc:displayName="EmptyAction1"
wpc:id="19">
      <bpws:sources>
        <bpws:source linkName="Link1">
          <bpws:transitionCondition><![CDATA[return
cond1.booleanValue();

          </bpws:source>
        </bpws:sources>
      </bpws:empty>
    <bpws:empty name="EmptyAction2" wpc:displayName="EmptyAction2"
wpc:id="20">
      <bpws:sources>
        <bpws:source linkName="Link2">
          <bpws:transitionCondition><![CDATA[return
cond2.booleanValue();

```

```

        </bpws:source>
    </bpws:sources>
</bpws:empty>
    <bpws:empty name="EmptyAction3" wpc:displayName="EmptyAction3"
wpc:id="21">
    <bpws:sources>
        <bpws:source linkName="Link3">
            <bpws:transitionCondition><![CDATA[return
cond3.booleanValue();
            </bpws:source>
        </bpws:sources>
    </bpws:empty>
    <bpws:assign name="IncreaseCounter" wpc:displayName="Increase
Counter" wpc:id="32">
    <bpws:targets>
        <bpws:target linkName="Link4"/>
    </bpws:targets>
    <bpws:copy>
        <bpws:from>
            <bpws:expression
expressionLanguage="http://www.w3.org/TR/1999/REC-xpath-
19991116"><![CDATA["bpws:getVariableData('counter')+1"
variable="counter"]]></bpws:expression>
            </bpws:from>
            <bpws:to variable="counter">
                <bpws:query queryLanguage="http://www.w3.org/TR/1999/REC-
xpath-19991116"><![CDATA["bpws:getVariableData('counter')"
variable="counter"]]></bpws:query>
            </bpws:to>
        </bpws:copy>
    </bpws:assign>
    <bpws:assign name="IncreaseCounter1" wpc:displayName="Increase
Counter" wpc:id="33">
    <bpws:targets>
        <bpws:target linkName="Link5"/>
    </bpws:targets>
    <bpws:copy>
        <bpws:from>
            <bpws:expression
expressionLanguage="http://www.w3.org/TR/1999/REC-xpath-
19991116"><![CDATA["bpws:getVariableData('counter')+1"
variable="counter"]]></bpws:expression>
            </bpws:from>
            <bpws:to variable="counter">
                <bpws:query queryLanguage="http://www.w3.org/TR/1999/REC-
xpath-19991116"><![CDATA["bpws:getVariableData('counter')"
variable="counter"]]></bpws:query>
            </bpws:to>
        </bpws:copy>
    </bpws:assign>
    <bpws:assign name="IncreaseCounter2" wpc:displayName="Increase
Counter" wpc:id="34">
    <bpws:targets>
        <bpws:target linkName="Link6"/>
    </bpws:targets>
    <bpws:copy>
        <bpws:from>
            <bpws:expression
expressionLanguage="http://www.w3.org/TR/1999/REC-xpath-
19991116"><![CDATA["bpws:getVariableData('counter')+1"
variable="counter"]]></bpws:expression>
            </bpws:from>
            <bpws:to variable="counter">
                <bpws:query queryLanguage="http://www.w3.org/TR/1999/REC-
xpath-19991116"><![CDATA["bpws:getVariableData('counter')"
variable="counter"]]></bpws:query>

```



```

        </bpws:to>
    </bpws:copy>
</bpws:assign>
</bpws:flow>
    <bpws:while name="LoopWhileCounter0" wpc:displayName="Loop While
Counter &gt; 0" wpc:id="31">
    <bpws:condition><![CDATA[boolean __result__1 =
com.ibm.wbiservers.brules.core.codegen.NillableUtility.getInt(counter,
"counter") > 0;
return __result__1;

    <bpws:sequence name="HiddenSequence1" wpc:id="1073741863">
    <bpws:assign name="Assign4" wpc:displayName="Assign4"
wpc:id="40">
    <bpws:copy>
    <bpws:from variable="Input"/>
    <bpws:to variable="output"/>
    </bpws:copy>
    </bpws:assign>
    <bpws:assign name="DecreaseCounter" wpc:displayName="Decrease
Counter" wpc:id="38">
    <bpws:copy>
    <bpws:from>
    <bpws:expression
expressionLanguage="http://www.w3.org/TR/1999/REC-xpath-
19991116"><![CDATA["bpws:getVariableData('counter')-1"
variable="counter" ]]></bpws:expression>
    </bpws:from>
    <bpws:to variable="counter">
    <bpws:query queryLanguage="http://www.w3.org/TR/1999/REC-
xpath-19991116"><![CDATA["bpws:getVariableData('counter')"
variable="counter" ]]></bpws:query>
    </bpws:to>
    </bpws:copy>
    </bpws:assign>
    </bpws:sequence>
    </bpws:while>
    <bpws:reply name="Reply" operation="operation1"
partnerLink="Multimerge" portType="ns0:Multi-merge"
wpc:displayName="Reply" wpc:id="5">
    <wpc:input>
    <wpc:parameter name="output1" variable="output"/>
    </wpc:input>
    </bpws:reply>
    </bpws:sequence>
</bpws:process>

```

The content of the WSDL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:tns="http://control-flow/multi-mergeInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/" name="multi-merge"
targetNamespace="http://control-flow/multi-mergeInterface">
    <types>
    <xsd:schema targetNamespace="http://control-flow/multi-
mergeInterface" xmlns:tns="http://control-flow/multi-mergeInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:element name="operation1">
    <xsd:complexType>
    <xsd:sequence>
    <xsd:element name="input1" nillable="true"
type="xsd:string"/>
    </xsd:sequence>
    </xsd:complexType>

```

```

        </xsd:element>
        <xsd:element name="operation1Response">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="output1" nillable="true"
type="xsd:string" />
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:schema>
</types>
<message name="operation1Request">
    <part element="tns:operation1" name="operation1Parameters" />
</message>
<message name="operation1Response">
    <part element="tns:operation1Response" name="operation1Result" />
</message>
<portType name="Multi-merge">
    <operation name="operation1">
        <input message="tns:operation1Request" name="operation1Request" />
        <output message="tns:operation1Response"
name="operation1Response" />
    </operation>
</portType>
</definitions>

```

Pattern 9 (Discriminator)

The content of the BPEL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<bpws:process xmlns:bpws="http://schemas.xmlsoap.org/ws/2004/03/business-
process/" xmlns:ns="http://control-flow/discriminatorArtifacts"
xmlns:ns0="http://control-flow/discriminatorInterface"
xmlns:wpc="http://www.ibm.com/xmlns/prod/websphere/business-
process/6.0.0/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.ibm.com/xmlns/prod/websphere/business-
process/expression-lang/java/6.0.0/" name="discriminator"
suppressJoinFailure="yes" targetNamespace="http://control-flow"
wpc:displayName="discriminator" wpc:executionMode="microflow" wpc:id="1">
    <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="discriminator.wsdl" namespace="http://control-
flow/discriminatorInterface" />
    <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="discriminatorArtifacts.wsdl" namespace="http://control-
flow/discriminatorArtifacts" />
    <bpws:partnerLinks>
        <bpws:partnerLink myRole="DiscriminatorRole" name="Discriminator"
partnerLinkType="ns:DiscriminatorPLT" />
    </bpws:partnerLinks>
    <bpws:variables>
        <bpws:variable name="Input" type="xsd:string" wpc:id="2" />
        <bpws:variable name="copy1" type="xsd:string" wpc:id="15" />
        <bpws:variable name="copy2" type="xsd:string" wpc:id="16" />
        <bpws:variable name="copy3" type="xsd:string" wpc:id="23" />
        <bpws:variable name="output" type="xsd:string" wpc:id="24" />
    </bpws:variables>
    <bpws:sequence name="HiddenSequence" wpc:id="1073741827">
        <bpws:receive createInstance="yes" name="Receive"
operation="operation1" partnerLink="Discriminator"
portType="ns0:Discriminator" wpc:displayName="Receive" wpc:id="4">
            <wpc:output>

```

```

        <wpc:parameter name="input1" variable="Input"/>
    </wpc:output>
</bpws:receive>
<bpws:flow name="ParallelActivities"
wpc:displayName="ParallelActivities" wpc:id="6">
    <bpws:links>
        <bpws:link name="Link1" wpc:id="26"/>
        <bpws:link name="Link2" wpc:id="27"/>
        <bpws:link name="Link3" wpc:id="28"/>
    </bpws:links>
    <bpws:assign name="Assign1" wpc:displayName="Assign1" wpc:id="7">
        <bpws:sources>
            <bpws:source linkName="Link1"/>
        </bpws:sources>
        <bpws:copy>
            <bpws:from variable="Input"/>
            <bpws:to variable="copy1"/>
        </bpws:copy>
    </bpws:assign>
    <bpws:assign name="Assign2" wpc:displayName="Assign2" wpc:id="8">
        <bpws:sources>
            <bpws:source linkName="Link2"/>
        </bpws:sources>
        <bpws:copy>
            <bpws:from variable="Input"/>
            <bpws:to variable="copy2"/>
        </bpws:copy>
    </bpws:assign>
    <bpws:assign name="Assign3" wpc:displayName="Assign3" wpc:id="18">
        <bpws:sources>
            <bpws:source linkName="Link3"/>
        </bpws:sources>
        <bpws:copy>
            <bpws:from variable="Input"/>
            <bpws:to variable="copy3"/>
        </bpws:copy>
    </bpws:assign>
    <bpws:empty name="EmptyAction" wpc:displayName="EmptyAction"
wpc:id="25">
        <bpws:targets>
            <bpws:target linkName="Link1"/>
            <bpws:target linkName="Link2"/>
            <bpws:target linkName="Link3"/>
        </bpws:targets>
    </bpws:empty>
</bpws:flow>
    <bpws:reply name="Reply" operation="operation1"
partnerLink="Discriminator" portType="ns0:Discriminator"
wpc:displayName="Reply" wpc:id="5">
        <wpc:input>
            <wpc:parameter name="output1" variable="output"/>
        </wpc:input>
    </bpws:reply>
</bpws:sequence>
</bpws:process>

```

The content of the WSDL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:tns="http://control-flow/discriminatorInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/" name="discriminator"

```

```

targetNamespace="http://control-flow/discriminatorInterface">
  <types>
    <xsd:schema targetNamespace="http://control-
flow/discriminatorInterface" xmlns:tns="http://control-
flow/discriminatorInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:element name="operation1">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="input1" nillable="true"
type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="operation1Response">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="output1" nillable="true"
type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </types>
  <message name="operation1Request">
    <part element="tns:operation1" name="operation1Parameters"/>
  </message>
  <message name="operation1Response">
    <part element="tns:operation1Response" name="operation1Result"/>
  </message>
  <portType name="Discriminator">
    <operation name="operation1">
      <input message="tns:operation1Request" name="operation1Request"/>
      <output message="tns:operation1Response"
name="operation1Response"/>
    </operation>
  </portType>
</definitions>

```

Pattern 12 (Multiple Instances Without Synchronization)

The content of the BPEL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<bpws:process xmlns:bpws="http://schemas.xmlsoap.org/ws/2004/03/business-
process/" xmlns:ns="http://control-
flow/multiple_instances_without_synchronizationArtifacts"
xmlns:ns0="http://control-
flow/multiple_instances_without_synchronizationInterface"
xmlns:ns1="http://control-flow/serviceInterface"
xmlns:ns2="http://control-flow/Service_OneWayInterface"
xmlns:wpc="http://www.ibm.com/xmlns/prod/websphere/business-
process/6.0.0/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.ibm.com/xmlns/prod/websphere/business-
process/expression-lang/java/6.0.0/"
name="multiple_instances_without_synchronization"
suppressJoinFailure="yes" targetNamespace="http://control-flow"
wpc:displayName="multiple_instances_without_synchronization"
wpc:executionMode="microflow" wpc:id="1">
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="multiple_instances_without_synchronization.wsdl"
namespace="http://control-

```

```

flow/multiple_instances_without_synchronizationInterface"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="multiple_instances_without_synchronizationArtifacts.wsdl"
namespace="http://control-
flow/multiple_instances_without_synchronizationArtifacts"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="service.wsdl" namespace="http://control-
flow/serviceInterface"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="Service_OneWay.wsdl" namespace="http://control-
flow/Service_OneWayInterface"/>
  <bpws:partnerLinks>
    <bpws:partnerLink
myRole="Multiple_instances_without_synchronizationRole"
name="Multiple_instances_without_synchronization"
partnerLinkType="ns:Multiple_instances_without_synchronizationPLT"/>
    <bpws:partnerLink name="Partner" partnerLinkType="ns:PartnerPLT1"
partnerRole="partnerRole"/>
  </bpws:partnerLinks>
  <bpws:variables>
    <bpws:variable name="Input1" type="xsd:string" wpc:id="2"/>
    <bpws:variable name="Invoke_Input" type="xsd:string" wpc:id="10"/>
    <bpws:variable name="stop" type="xsd:boolean" wpc:id="11"/>
    <bpws:variable name="Invoke_Output" type="xsd:string" wpc:id="12"/>
  </bpws:variables>
  <bpws:sequence name="HiddenSequence" wpc:id="1073741827">
    <bpws:receive createInstance="yes" name="Receive"
operation="operation1"
partnerLink="Multiple_instances_without_synchronization"
portType="ns0:Multiple_instances_without_synchronization"
wpc:displayName="Receive" wpc:id="4">
      <wpc:output>
        <wpc:parameter name="input1" variable="Input1"/>
      </wpc:output>
    </bpws:receive>
    <bpws:while name="WhileLoop" wpc:displayName="WhileLoop" wpc:id="6">
      <bpws:condition><![CDATA[boolean __result__1 =
com.ibm.wbiservers.brules.core.codegen.NillableUtility.getBoolean(stop,
"stop") == false;
return __result__1;

      <bpws:invoke name="Invoke" operation="operation1"
partnerLink="Partner" portType="ns2:Service_OneWay"
wpc:displayName="Invoke" wpc:id="7">
        <wpc:input>
          <wpc:parameter name="input1" variable="Invoke_Input"/>
        </wpc:input>
      </bpws:invoke>
    </bpws:while>
    <bpws:reply name="Reply" operation="operation1"
partnerLink="Multiple_instances_without_synchronization"
portType="ns0:Multiple_instances_without_synchronization"
wpc:displayName="Reply" wpc:id="5">
      <wpc:input>
        <wpc:parameter name="output1" variable="Input1"/>
      </wpc:input>
    </bpws:reply>
  </bpws:sequence>
</bpws:process>

```

The content of the WSDL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:tns="http://control-
flow/multiple_instances_without_synchronizationInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

```

```

xmlns="http://schemas.xmlsoap.org/wsdl/"
name="multiple_instances_without_synchronization"
targetNamespace="http://control-
flow/multiple_instances_without_synchronizationInterface">
  <types>
    <xsd:schema targetNamespace="http://control-
flow/multiple_instances_without_synchronizationInterface"
xmlns:tns="http://control-
flow/multiple_instances_without_synchronizationInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:element name="operation1">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="input1" nillable="true"
type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="operation1Response">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="output1" nillable="true"
type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </types>
  <message name="operation1Request">
    <part element="tns:operation1" name="operation1Parameters"/>
  </message>
  <message name="operation1Response">
    <part element="tns:operation1Response" name="operation1Result"/>
  </message>
  <portType name="Multiple_instances_without_synchronization">
    <operation name="operation1">
      <input message="tns:operation1Request" name="operation1Request"/>
      <output message="tns:operation1Response"
name="operation1Response"/>
    </operation>
  </portType>
</definitions>

```

Pattern 13 (Multiple Instances With a Priori Design Time Knowledge)

The content of the BPEL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<bpws:process xmlns:bpws="http://schemas.xmlsoap.org/ws/2004/03/business-
process/" xmlns:ns="http://control-
flow/multiple_instances_with_a_priori_design_time_knowledgeArtifacts"
xmlns:ns0="http://control-
flow/multiple_instances_with_a_priori_design_time_knowledgeInterface"
xmlns:ns1="http://control-flow/serviceInterface"
xmlns:ns2="http://control-flow/Service_TwoWayInterface"
xmlns:wpc="http://www.ibm.com/xmlns/prod/websphere/business-
process/6.0.0/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.ibm.com/xmlns/prod/websphere/business-
process/expression-lang/java/6.0.0/"
name="multiple_instances_with_a_priori_design_time_knowledge"
suppressJoinFailure="yes" targetNamespace="http://control-flow"
wpc:displayName="multiple_instances_with_a_priori_design_time_knowledge"

```

```

wpc:executionMode="microflow" wpc:id="1">
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="multiple_instances_with_a_priori_design_time_knowledge.wsdl"
namespace="http://control-
flow/multiple_instances_with_a_priori_design_time_knowledgeInterface"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="multiple_instances_with_a_priori_design_time_knowledgeArtifacts
.wsdl" namespace="http://control-
flow/multiple_instances_with_a_priori_design_time_knowledgeArtifacts"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="service.wsdl" namespace="http://control-
flow/serviceInterface"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="Service_TwoWay.wsdl" namespace="http://control-
flow/Service_TwoWayInterface"/>
  <bpws:partnerLinks>
    <bpws:partnerLink
myRole="Multiple_instances_with_a_priori_design_time_knowledgeRole"
name="Multiple_instances_with_a_priori_design_time_knowledge"
partnerLinkType="ns:Multiple_instances_with_a_priori_design_time_knowledg
ePLT"/>
    <bpws:partnerLink name="Partner" partnerLinkType="ns:PartnerPLT"
partnerRole="partnerRole"/>
  </bpws:partnerLinks>
  <bpws:variables>
    <bpws:variable name="Input" type="xsd:string" wpc:id="2"/>
    <bpws:variable name="Output" type="xsd:string" wpc:id="12"/>
    <bpws:variable name="Invoke_Input" type="xsd:string" wpc:id="13"/>
    <bpws:variable name="Invoke_Output" type="xsd:string" wpc:id="14"/>
  </bpws:variables>
  <bpws:sequence name="HiddenSequence" wpc:id="1073741827">
    <bpws:receive createInstance="yes" name="Receive"
operation="operation1"
partnerLink="Multiple_instances_with_a_priori_design_time_knowledge"
portType="ns0:Multiple_instances_with_a_priori_design_time_knowledge"
wpc:displayName="Receive" wpc:id="4">
      <wpc:output>
        <wpc:parameter name="input1" variable="Input"/>
      </wpc:output>
    </bpws:receive>
    <bpws:flow name="ParallelActivities"
wpc:displayName="ParallelActivities" wpc:id="6">
      <bpws:invoke name="Invoke1" operation="operation1"
partnerLink="Partner" portType="ns2:Service_TwoWay"
wpc:displayName="Invoke" wpc:id="7">
        <wpc:input>
          <wpc:parameter name="input1" variable="Invoke_Input"/>
        </wpc:input>
        <wpc:output>
          <wpc:parameter name="output1" variable="Invoke_Output"/>
        </wpc:output>
      </bpws:invoke>
      <bpws:invoke name="Invoke" operation="operation1"
partnerLink="Partner" portType="ns2:Service_TwoWay"
wpc:displayName="Invoke" wpc:id="8">
        <wpc:input>
          <wpc:parameter name="input1" variable="Invoke_Input"/>
        </wpc:input>
        <wpc:output>
          <wpc:parameter name="output1" variable="Invoke_Output"/>
        </wpc:output>
      </bpws:invoke>
      <bpws:invoke name="Invoke2" operation="operation1"
partnerLink="Partner" portType="ns2:Service_TwoWay"
wpc:displayName="Invoke" wpc:id="9">
        <wpc:input>
          <wpc:parameter name="input1" variable="Invoke_Input"/>

```

```

        </wpc:input>
        <wpc:output>
            <wpc:parameter name="output1" variable="Invoke_Output"/>
        </wpc:output>
    </bpws:invoke>
</bpws:flow>
<bpws:assign name="Assign" wpc:displayName="Assign" wpc:id="10">
    <bpws:copy>
        <bpws:from variable="Input"/>
        <bpws:to variable="Output"/>
    </bpws:copy>
</bpws:assign>
<bpws:reply name="Reply" operation="operation1"
partnerLink="Multiple_instances_with_a_priori_design_time_knowledge"
portType="ns0:Multiple_instances_with_a_priori_design_time_knowledge"
wpc:displayName="Reply" wpc:id="5">
    <wpc:input>
        <wpc:parameter name="output1" variable="Output"/>
    </wpc:input>
</bpws:reply>
</bpws:sequence>
</bpws:process>

```

The content of the WSDL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:tns="http://control-
flow/multiple_instances_with_a_priori_design_time_knowledgeInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/"
name="multiple_instances_with_a_priori_design_time_knowledge"
targetNamespace="http://control-
flow/multiple_instances_with_a_priori_design_time_knowledgeInterface">
    <types>
        <xsd:schema targetNamespace="http://control-
flow/multiple_instances_with_a_priori_design_time_knowledgeInterface"
xmlns:tns="http://control-
flow/multiple_instances_with_a_priori_design_time_knowledgeInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
            <xsd:element name="operation1">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="input1" nillable="true"
type="xsd:string"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="operation1Response">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="output1" nillable="true"
type="xsd:string"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:schema>
    </types>
    <message name="operation1Request">
        <part element="tns:operation1" name="operation1Parameters"/>
    </message>
    <message name="operation1Response">
        <part element="tns:operation1Response" name="operation1Result"/>
    </message>
    <portType
name="Multiple_instances_with_a_priori_design_time_knowledge">

```



```

        <operation name="operation1">
            <input message="tns:operation1Request" name="operation1Request"/>
            <output message="tns:operation1Response"
name="operation1Response"/>
        </operation>
    </portType>
</definitions>

```

Pattern 14 (Multiple Instances With a Priori Runtime Knowledge)

The content of the BPEL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<bpws:process xmlns:bpws="http://schemas.xmlsoap.org/ws/2004/03/business-
process/" xmlns:ns="http://control-
flow/multiple_instances_with_runtime_knowledgeArtifacts"
xmlns:ns0="http://control-
flow/multiple_instances_with_runtime_knowledgeInterface"
xmlns:ns1="http://control-flow/serviceInterface"
xmlns:ns2="http://control-flow/Service_TwoWayInterface"
xmlns:wpc="http://www.ibm.com/xmlns/prod/websphere/business-
process/6.0.0/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.ibm.com/xmlns/prod/websphere/business-
process/expression-lang/java/6.0.0/"
name="multiple_instances_with_runtime_knowledge"
suppressJoinFailure="yes" targetNamespace="http://control-flow"
wpc:displayName="multiple_instances_with_runtime_knowledge"
wpc:executionMode="microflow" wpc:id="1">
    <bpws:import importType="http://schemas.xmlsoap.org/wSDL/"
location="multiple_instances_with_runtime_knowledge.wsdl"
namespace="http://control-
flow/multiple_instances_with_runtime_knowledgeInterface"/>
    <bpws:import importType="http://schemas.xmlsoap.org/wSDL/"
location="multiple_instances_with_runtime_knowledgeArtifacts.wsdl"
namespace="http://control-
flow/multiple_instances_with_runtime_knowledgeArtifacts"/>
    <bpws:import importType="http://schemas.xmlsoap.org/wSDL/"
location="service.wsdl" namespace="http://control-
flow/serviceInterface"/>
    <bpws:import importType="http://schemas.xmlsoap.org/wSDL/"
location="Service_TwoWay.wsdl" namespace="http://control-
flow/Service_TwoWayInterface"/>
    <bpws:partnerLinks>
        <bpws:partnerLink
myRole="Multiple_instances_with_runtime_knowledgeRole"
name="Multiple_instances_with_runtime_knowledge"
partnerLinkType="ns:Multiple_instances_with_runtime_knowledgePLT"/>
        <bpws:partnerLink name="Partner" partnerLinkType="ns:PartnerPLT"
partnerRole="partnerRole"/>
    </bpws:partnerLinks>
    <bpws:variables>
        <bpws:variable name="Input" type="xsd:string" wpc:id="2"/>
        <bpws:variable name="copy" type="xsd:string" wpc:id="10"/>
        <bpws:variable name="Output" type="xsd:string" wpc:id="11"/>
        <bpws:variable name="Invoke_Input" type="xsd:string" wpc:id="12"/>
        <bpws:variable name="Invoke_Output" type="xsd:string" wpc:id="13"/>
        <bpws:variable name="stop" type="xsd:boolean" wpc:id="16"/>
    </bpws:variables>
    <bpws:sequence name="HiddenSequence" wpc:id="1073741827">
        <bpws:receive createInstance="yes" name="Receive"
operation="operation1"
partnerLink="Multiple_instances_with_runtime_knowledge"

```

```

portType="ns0:Multiple_instances_with_runtime_knowledge"
wpc:displayName="Receive" wpc:id="4">
  <wpc:output>
    <wpc:parameter name="input1" variable="Input"/>
  </wpc:output>
</bpws:receive>
<bpws:while name="WhileLoop" wpc:displayName="WhileLoop" wpc:id="6">
  <bpws:condition><![CDATA[boolean __result__1 =
com.ibm.wbiservers.brules.core.codegen.NillableUtility.getBoolean(stop,
"stop") == false;
return __result__1;

  <bpws:invoke name="Invoke" operation="operation1"
partnerLink="Partner" portType="ns2:Service_TwoWay"
wpc:displayName="Invoke" wpc:id="15">
  <wpc:input>
    <wpc:parameter name="input1" variable="Invoke_Input"/>
  </wpc:input>
  <wpc:output>
    <wpc:parameter name="output1" variable="Invoke_Output"/>
  </wpc:output>
</bpws:invoke>
</bpws:while>
<bpws:assign name="Assign" wpc:displayName="Assign" wpc:id="9">
  <bpws:copy>
    <bpws:from variable="Input"/>
    <bpws:to variable="Output"/>
  </bpws:copy>
</bpws:assign>
  <bpws:reply name="Reply" operation="operation1"
partnerLink="Multiple_instances_with_runtime_knowledge"
portType="ns0:Multiple_instances_with_runtime_knowledge"
wpc:displayName="Reply" wpc:id="5">
  <wpc:input>
    <wpc:parameter name="output1" variable="Output"/>
  </wpc:input>
</bpws:reply>
</bpws:sequence>
</bpws:process>

```

The content of the WSDL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:tns="http://control-
flow/multiple_instances_with_runtime_knowledgeInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/"
name="multiple_instances_with_runtime_knowledge"
targetNamespace="http://control-
flow/multiple_instances_with_runtime_knowledgeInterface">
  <types>
    <xsd:schema targetNamespace="http://control-
flow/multiple_instances_with_runtime_knowledgeInterface"
xmlns:tns="http://control-
flow/multiple_instances_with_runtime_knowledgeInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:element name="operation1">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="input1" nillable="true"
type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="operation1Response">

```

```

        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="output1" nillable="true"
type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </types>
  <message name="operation1Request">
    <part element="tns:operation1" name="operation1Parameters"/>
  </message>
  <message name="operation1Response">
    <part element="tns:operation1Response" name="operation1Result"/>
  </message>
  <portType name="Multiple_instances_without_runtime_knowledge">
    <operation name="operation1">
      <input message="tns:operation1Request" name="operation1Request"/>
      <output message="tns:operation1Response"
name="operation1Response"/>
    </operation>
  </portType>
</definitions>

```

Pattern 15 (Multiple Instances Without a Priori Runtime Knowledge)

The content of the BPEL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<bpws:process xmlns:bpws="http://schemas.xmlsoap.org/ws/2004/03/business-
process/" xmlns:ns="http://control-
flow/multiple_instances_without_runtime_knowledgeArtifacts"
xmlns:ns0="http://control-
flow/multiple_instances_without_runtime_knowledgeInterface"
xmlns:ns1="http://control-flow/serviceInterface"
xmlns:ns2="http://control-flow/Service_TwoWayInterface"
xmlns:wpc="http://www.ibm.com/xmlns/prod/websphere/business-
process/6.0.0/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.ibm.com/xmlns/prod/websphere/business-
process/expression-lang/java/6.0.0/"
name="multiple_instances_without_runtime_knowledge"
suppressJoinFailure="yes" targetNamespace="http://control-flow"
wpc:autonomy="peer"
wpc:displayName="multiple_instances_without_runtime_knowledge"
wpc:id="1">
  <bpws:import importType="http://schemas.xmlsoap.org/wsd1/"
location="multiple_instances_without_runtime_knowledge.wsdl"
namespace="http://control-
flow/multiple_instances_without_runtime_knowledgeInterface"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wsd1/"
location="multiple_instances_without_runtime_knowledgeArtifacts.wsdl"
namespace="http://control-
flow/multiple_instances_without_runtime_knowledgeArtifacts"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wsd1/"
location="service.wsdl" namespace="http://control-
flow/serviceInterface"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wsd1/"
location="Service_TwoWay.wsdl" namespace="http://control-
flow/Service_TwoWayInterface"/>
  <bpws:partnerLinks>
    <bpws:partnerLink
myRole="Multiple_instances_without_runtime_knowledgeRole"

```

```

name="Multiple_instances_without_runtime_knowledge"
partnerLinkType="ns:Multiple_instances_without_runtime_knowledgePLT"/>
  <bpws:partnerLink name="Partner" partnerLinkType="ns:PartnerPLT"
partnerRole="partnerRole"/>
</bpws:partnerLinks>
<bpws:variables>
  <bpws:variable name="Input" type="xsd:string" wpc:id="2"/>
  <bpws:variable name="Output" type="xsd:string" wpc:id="9"/>
  <bpws:variable name="counter" type="xsd:int" wpc:id="13"/>
  <bpws:variable name="quantity" type="xsd:string" wpc:id="14"/>
  <bpws:variable name="Invoke_Input" type="xsd:string" wpc:id="17"/>
  <bpws:variable name="Invoke_Output" type="xsd:string" wpc:id="18"/>
  <bpws:variable name="execute" type="xsd:boolean" wpc:id="24"/>
  <bpws:variable name="stop" type="xsd:string" wpc:id="26"/>
</bpws:variables>
<bpws:correlationSets>
  <bpws:correlationSet name="CorrelationSet" properties="ns:Property"/>
  <bpws:correlationSet name="CorrelationSet1"
properties="ns:property1"/>
</bpws:correlationSets>
  <bpws:sequence name="HiddenSequence" wpc:id="1073741827">
    <bpws:receive createInstance="yes" name="Receive"
operation="operation1"
partnerLink="Multiple_instances_without_runtime_knowledge"
portType="ns0:Multiple_instances_without_runtime_knowledge"
wpc:displayName="Receive" wpc:id="4">
      <wpc:output>
        <wpc:parameter name="input1" variable="Input"/>
      </wpc:output>
    </bpws:receive>
    <bpws:assign name="Setexecutetrue" wpc:displayName="Set execute =
true" wpc:id="25">
      <bpws:copy>
        <bpws:from>true</bpws:from>
        <bpws:to variable="execute"/>
      </bpws:copy>
    </bpws:assign>
    <bpws:while name="LoopWhileexecutetrue" wpc:displayName="Loop While
execute = true" wpc:id="7">
      <bpws:condition><![CDATA[boolean __result__1 =
com.ibm.wbiservers.brules.core.codegen.NullableUtility.getBoolean(execute
, "execute") == true;
return __result__1;

      <bpws:sequence name="HiddenSequence2" wpc:id="1073741845">
        <bpws:assign name="SetCounter0" wpc:displayName="Set Counter = 0"
wpc:id="22">
          <bpws:copy>
            <bpws:from>0</bpws:from>
            <bpws:to variable="counter"/>
          </bpws:copy>
        </bpws:assign>
        <bpws:pick createInstance="no" name="ReceiveChoice"
wpc:displayName="ReceiveChoice" wpc:id="6">
          <bpws:onMessage operation="operation1"
partnerLink="Multiple_instances_without_runtime_knowledge"
portType="ns0:Multiple_instances_without_runtime_knowledge">
            <wpc:output>
              <wpc:parameter name="input1" variable="quantity"/>
            </wpc:output>
          <bpws:correlations>
            <bpws:correlation initiate="yes" set="CorrelationSet"/>
          </bpws:correlations>
          <bpws:while name="LoopWhilecounterquantity"
wpc:displayName="Loop While counter &lt; quantity" wpc:id="12">
            <bpws:condition><![CDATA[boolean __result__3;
{// less than

```

```

        __result__3 = (counter.intValue()) < (quantity.intValue());
    }
    return __result__3;

    <bpws:sequence name="HiddenSequence1" wpc:id="1073741843">
        <bpws:invoke name="Invoke" operation="operation1"
partnerLink="Partner" portType="ns2:Service_TwoWay"
wpc:displayName="Invoke" wpc:id="8">
            <wpc:input>
                <wpc:parameter name="input1"
variable="Invoke_Input"/>
            </wpc:input>
            <wpc:output>
                <wpc:parameter name="output1"
variable="Invoke_Output"/>
            </wpc:output>
        </bpws:invoke>
        <bpws:assign name="IncreaseCounter"
wpc:displayName="Increase Counter" wpc:id="20">
            <bpws:copy>
                <bpws:from>
                    <bpws:expression
expressionLanguage="http://www.w3.org/TR/1999/REC-xpath-
19991116"><![CDATA["bpws:getVariableData('counter')+1"
variable="counter"]]></bpws:expression>
                </bpws:from>
                <bpws:to variable="counter">
                    <bpws:query
queryLanguage="http://www.w3.org/TR/1999/REC-xpath-
19991116"><![CDATA["bpws:getVariableData('counter')"
variable="counter"]]></bpws:query>
                </bpws:to>
            </bpws:copy>
        </bpws:assign>
    </bpws:sequence>
</bpws:while>
</bpws:onMessage>
    <bpws:onMessage operation="operation2"
partnerLink="Multiple_instances_without_runtime_knowledge"
portType="ns0:Multiple_instances_without_runtime_knowledge">
        <wpc:output>
            <wpc:parameter name="input2" variable="stop"/>
        </wpc:output>
        <bpws:correlations>
            <bpws:correlation initiate="yes" set="CorrelationSet1"/>
        </bpws:correlations>
        <bpws:assign name="Setexecutefalse" wpc:displayName="Set
execute = false" wpc:id="23">
            <bpws:copy>
                <bpws:from>false</bpws:from>
                <bpws:to variable="execute"/>
            </bpws:copy>
        </bpws:assign>
    </bpws:onMessage>
</bpws:pick>
</bpws:sequence>
</bpws:while>
<bpws:assign name="Assign" wpc:displayName="Assign" wpc:id="15">
    <bpws:copy>
        <bpws:from variable="Input"/>
        <bpws:to variable="Output"/>
    </bpws:copy>
</bpws:assign>
    <bpws:reply name="Reply" operation="operation1"
partnerLink="Multiple_instances_without_runtime_knowledge"
portType="ns0:Multiple_instances_without_runtime_knowledge"
wpc:displayName="Reply" wpc:id="5">

```

```

        <wpc:input>
            <wpc:parameter name="output1" variable="Output" />
        </wpc:input>
    </bpws:reply>
</bpws:sequence>
</bpws:process>

```

The content of the WSDL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="http://control-
flow/multiple_instances_without_runtime_knowledgeInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
name="multiple_instances_without_runtime_knowledge"
targetNamespace="http://control-
flow/multiple_instances_without_runtime_knowledgeInterface">
    <types>
        <xsd:schema targetNamespace="http://control-
flow/multiple_instances_without_runtime_knowledgeInterface"
xmlns:tns="http://control-
flow/multiple_instances_without_runtime_knowledgeInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
            <xsd:element name="operation1">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="input1" nillable="true"
type="xsd:string" />
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="operation1Response">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="output1" nillable="true"
type="xsd:string" />
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="operation2">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="input2" nillable="true"
type="xsd:string" />
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="operation2Response">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="output2" nillable="true"
type="xsd:string" />
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:schema>
    </types>
    <message name="operation1Request">
        <part element="tns:operation1" name="operation1Parameters" />
    </message>
    <message name="operation1Response">
        <part element="tns:operation1Response" name="operation1Result" />
    </message>
    <message name="operation2RequestMsg">
        <part element="tns:operation2" name="operation2Parameters" />

```

```

</message>
  <message name="operation2ResponseMsg">
    <part element="tns:operation2Response" name="operation2Result"/>
  </message>
  <portType name="Multiple_instances_without_runtime_knowledge">
    <operation name="operation1">
      <input message="tns:operation1Request" name="operation1Request"/>
      <output message="tns:operation1Response"
name="operation1Response"/>
    </operation>
    <operation name="operation2"><input
message="tns:operation2RequestMsg" name="operation2Request"/>
      <output message="tns:operation2ResponseMsg"
name="operation2Response"/>    </operation>
    </portType>
  </definitions>

```

Pattern 16 (Deferred Choice)

The content of the BPEL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<bpws:process xmlns:bpws="http://schemas.xmlsoap.org/ws/2004/03/business-
process/" xmlns:ns="http://control-flow/deferred_choiceArtifacts"
xmlns:ns0="http://control-flow/deferred_choiceInterface"
xmlns:wpc="http://www.ibm.com/xmlns/prod/websphere/business-
process/6.0.0/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.ibm.com/xmlns/prod/websphere/business-
process/expression-lang/java/6.0.0/" name="deferred_choice"
suppressJoinFailure="yes" targetNamespace="http://control-flow"
wpc:autonomy="peer" wpc:businessRelevant="yes"
wpc:displayName="deferred_choice" wpc:id="1">
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="deferred_choice.wsdl" namespace="http://control-
flow/deferred_choiceInterface"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="deferred_choiceArtifacts.wsdl" namespace="http://control-
flow/deferred_choiceArtifacts"/>
  <bpws:partnerLinks>
    <bpws:partnerLink myRole="Deferred_choiceRole" name="Deferred_choice"
partnerLinkType="ns:Deferred_choicePLT"/>
    <bpws:partnerLink name="Partner" partnerLinkType="ns:PartnerPLT"
partnerRole="partnerRole"/>
  </bpws:partnerLinks>
  <bpws:variables>
    <bpws:variable name="Input" type="xsd:string" wpc:id="2"/>
    <bpws:variable name="copy1" type="xsd:string" wpc:id="10"/>
    <bpws:variable name="copy2" type="xsd:string" wpc:id="11"/>
    <bpws:variable name="copy3" type="xsd:string" wpc:id="12"/>
    <bpws:variable name="Output" type="xsd:string" wpc:id="19"/>
    <bpws:variable name="ChoiceInput" type="xsd:string" wpc:id="26"/>
  </bpws:variables>
  <bpws:correlationSets>
    <bpws:correlationSet name="CorrelationSet" properties="ns:property"/>
  </bpws:correlationSets>
  <bpws:sequence name="HiddenSequence" wpc:id="1073741827">
    <bpws:receive createInstance="yes" name="Receive"
operation="operation1" partnerLink="Deferred_choice"
portType="ns0:Deferred_choice" wpc:displayName="Receive" wpc:id="4">
      <wpc:output>
        <wpc:parameter name="input1" variable="Input"/>
      </wpc:output>

```

```

        </bpws:receive>
        <bpws:pick name="ReceiveChoice" wpc:displayName="ReceiveChoice"
wpc:id="6">
            <bpws:onMessage operation="operation1"
partnerLink="Deferred_choice" portType="ns0:Deferred_choice">
                <wpc:output>
                    <wpc:parameter name="input1" variable="ChoiceInput"/>
                </wpc:output>
                <bpws:correlations>
                    <bpws:correlation initiate="yes" set="CorrelationSet"/>
                </bpws:correlations>
                <bpws:switch name="Choice" wpc:displayName="Choice" wpc:id="13">
                    <bpws:case wpc:id="14">
                        <bpws:condition><![CDATA[ java.lang.String __result__2 =
"Case1";
boolean __result__3;
{// text equal to (ignore case)
    __result__3 = ChoiceInput.equalsIgnoreCase(__result__2);
}
return __result__3;

                    <bpws:assign name="Assign1" wpc:displayName="Assign1"
wpc:id="20">
                        <bpws:copy>
                            <bpws:from variable="Input"/>
                            <bpws:to variable="copy1"/>
                        </bpws:copy>
                    </bpws:assign>
                </bpws:case>
                <bpws:case wpc:id="15">
                    <bpws:condition><![CDATA[ java.lang.String __result__2 =
"Case2";
boolean __result__3;
{// text equal to (ignore case)
    __result__3 = ChoiceInput.equalsIgnoreCase(__result__2);
}
return __result__3;

                    <bpws:sequence name="HiddenSequence1" wpc:id="1073741840">
                        <bpws:assign name="Assign2" wpc:displayName="Assign2"
wpc:id="21">
                            <bpws:copy>
                                <bpws:from variable="Input"/>
                                <bpws:to variable="copy2"/>
                            </bpws:copy>
                        </bpws:assign>
                    </bpws:sequence>
                </bpws:case>
                <bpws:case wpc:id="17">
                    <bpws:condition><![CDATA[ java.lang.String __result__2 =
"Case3";
boolean __result__3;
{// text equal to (ignore case)
    __result__3 = ChoiceInput.equalsIgnoreCase(__result__2);
}
return __result__3;

                    <bpws:sequence name="HiddenSequence2" wpc:id="1073741842">
                        <bpws:assign name="Assign3" wpc:displayName="Assign3"
wpc:id="22">
                            <bpws:copy>
                                <bpws:from variable="Input"/>
                                <bpws:to variable="copy3"/>
                            </bpws:copy>
                        </bpws:assign>
                    </bpws:sequence>
                </bpws:case>

```



```

        </bpws:switch>
    </bpws:onMessage>
</bpws:pick>
    <bpws:reply name="Reply" operation="operation1"
partnerLink="Deferred_choice" portType="ns0:Deferred_choice"
wpc:displayName="Reply" wpc:id="5">
        <wpc:input>
            <wpc:parameter name="output1" variable="Output"/>
        </wpc:input>
    </bpws:reply>
</bpws:sequence>
</bpws:process>

```

The content of the WSDL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:tns="http://control-flow/deferred_choiceInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/" name="deferred_choice"
targetNamespace="http://control-flow/deferred_choiceInterface">
    <types>
        <xsd:schema targetNamespace="http://control-
flow/deferred_choiceInterface" xmlns:tns="http://control-
flow/deferred_choiceInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
            <xsd:element name="operation1">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="input1" nillable="true"
type="xsd:string"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="operation1Response">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="output1" nillable="true"
type="xsd:string"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:schema>
    </types>
    <message name="operation1Request">
        <part element="tns:operation1" name="operation1Parameters"/>
    </message>
    <message name="operation1Response">
        <part element="tns:operation1Response" name="operation1Result"/>
    </message>
    <portType name="Deferred_choice">
        <operation name="operation1">
            <input message="tns:operation1Request" name="operation1Request"/>
            <output message="tns:operation1Response"
name="operation1Response"/>
        </operation>
    </portType>
</definitions>

```

Pattern 17 (Interleaved Parallel Routing)

The content of the BPEL file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<bpws:process xmlns:bpws="http://schemas.xmlsoap.org/ws/2004/03/business-
process/" xmlns:ns="http://control-
flow/interleaved_parallel_routingArtifacts" xmlns:ns0="http://control-
flow/interleaved_parallel_routingInterface" xmlns:ns1="http://control-
flow/control_flow"
xmlns:wpc="http://www.ibm.com/xmlns/prod/websphere/business-
process/6.0.0/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.ibm.com/xmlns/prod/websphere/business-
process/expression-lang/java/6.0.0/" name="interleaved_parallel_routing"
suppressJoinFailure="yes" targetNamespace="http://control-flow"
wpc:displayName="interleaved_parallel_routing"
wpc:executionMode="microflow" wpc:id="1">
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="interleaved_parallel_routing.wsdl" namespace="http://control-
flow/interleaved_parallel_routingInterface"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="interleaved_parallel_routingArtifacts.wsdl"
namespace="http://control-flow/interleaved_parallel_routingArtifacts"/>
  <bpws:import importType="http://www.w3.org/2001/XMLSchema"
location="control_flow/ipr.xsd" namespace="http://control-
flow/control_flow"/>
  <bpws:partnerLinks>
    <bpws:partnerLink myRole="interleaved_parallel_routingRole"
name="interleaved_parallel_routing"
partnerLinkType="ns:interleaved_parallel_routingPLT"/>
  </bpws:partnerLinks>
  <bpws:variables>
    <bpws:variable name="Input1" type="ns1:ipr" wpc:id="2"/>
    <bpws:variable name="Counter" type="xsd:int" wpc:id="20"/>
    <bpws:variable name="active" type="xsd:int" wpc:id="23"/>
    <bpws:variable name="Output" type="xsd:string" wpc:id="27"/>
    <bpws:variable name="copy" type="xsd:string" wpc:id="28"/>
    <bpws:variable name="element" type="xsd:int" wpc:id="29"/>
  </bpws:variables>
  <bpws:sequence name="HiddenSequence" wpc:id="1073741827">
    <bpws:receive createInstance="yes" name="Receive"
operation="operation1" partnerLink="interleaved_parallel_routing"
portType="ns0:interleaved_parallel_routing" wpc:displayName="Receive"
wpc:id="4">
      <wpc:output>
        <wpc:parameter name="input1" variable="Input1"/>
      </wpc:output>
    </bpws:receive>
    <bpws:assign name="SetCounter1" wpc:displayName="Set Counter = 1"
wpc:id="19">
      <bpws:copy>
        <bpws:from>0</bpws:from>
        <bpws:to variable="Counter"/>
      </bpws:copy>
    </bpws:assign>
    <bpws:while name="LoopWhileCounter4" wpc:displayName="Loop While
Counter &lt; 4" wpc:id="6">
      <bpws:condition><![CDATA[boolean __result__1 =
com.ibm.wbiservers.brules.core.codegen.NullableUtility.getInt(Counter,
"Counter") < 4;
return __result__1;
]]></bpws:condition>
      <bpws:sequence name="HiddenSequence3" wpc:id="1073741845">
        <bpws:assign name="Assign" wpc:displayName="Assign" wpc:id="22">
          <bpws:copy>
            <bpws:from variable="Input1">
              <bpws:query queryLanguage="http://www.w3.org/TR/1999/REC-
xpath-
19991116"><![CDATA[/array/item[bpws:getVariableData('Counter')]]]></bpws:

```

```

query>
    </bpws:from>
    <bpws:to variable="element"/>
  </bpws:copy>
</bpws:assign>
<bpws:switch name="Choice" wpc:displayName="Choice" wpc:id="7">
  <bpws:case wpc:id="8">
    <bpws:condition><![CDATA[boolean __result__1 =
com.ibm.wbiservers.brules.core.codegen.NillableUtility.getInt(element,
"element") == 1;
return __result__1;
    <bpws:assign name="Assign1" wpc:displayName="Assign1"
wpc:id="9">
      <bpws:copy>
        <bpws:from variable="copy"/>
        <bpws:to variable="Output"/>
      </bpws:copy>
    </bpws:assign>
  </bpws:case>
  <bpws:case wpc:id="10">
    <bpws:condition><![CDATA[boolean __result__1 =
com.ibm.wbiservers.brules.core.codegen.NillableUtility.getInt(element,
"element") == 2;
return __result__1;
    <bpws:sequence name="HiddenSequence1" wpc:id="1073741835">
      <bpws:assign name="Assign2" wpc:displayName="Assign2"
wpc:id="12">
        <bpws:copy>
          <bpws:from variable="copy"/>
          <bpws:to variable="Output"/>
        </bpws:copy>
      </bpws:assign>
    </bpws:sequence>
  </bpws:case>
  <bpws:case wpc:id="13">
    <bpws:condition><![CDATA[boolean __result__1 =
com.ibm.wbiservers.brules.core.codegen.NillableUtility.getInt(element,
"element") == 3;
return __result__1;
    <bpws:sequence name="HiddenSequence2" wpc:id="1073741838">
      <bpws:assign name="Assign3" wpc:displayName="Assign3"
wpc:id="15">
        <bpws:copy>
          <bpws:from variable="copy"/>
          <bpws:to variable="Output"/>
        </bpws:copy>
      </bpws:assign>
    </bpws:sequence>
  </bpws:case>
</bpws:switch>
<bpws:assign name="IncreaseCounter" wpc:displayName="Increase
Counter" wpc:id="18">
  <bpws:copy>
    <bpws:from>
      <bpws:expression
expressionLanguage="http://www.w3.org/TR/1999/REC-xpath-
19991116"><![CDATA["bpws:getVariableData('Counter')+1"
variable="Counter"]]></bpws:expression>
    </bpws:from>
    <bpws:to variable="Counter">
      <bpws:query queryLanguage="http://www.w3.org/TR/1999/REC-
xpath-19991116"><![CDATA["bpws:getVariableData('Counter')"
variable="Counter"]]></bpws:query>
    </bpws:to>
  </bpws:copy>
</bpws:assign>
</bpws:sequence>

```

```

        </bpws:while>
        <bpws:reply name="Reply" operation="operation1"
partnerLink="interleaved_parallel_routing"
portType="ns0:interleaved_parallel_routing" wpc:displayName="Reply"
wpc:id="5">
            <wpc:input>
                <wpc:parameter name="output1" variable="Input1"/>
            </wpc:input>
        </bpws:reply>
    </bpws:sequence>
</bpws:process>

```

The content of the WSDL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:bons1="http://control-flow/control_flow" xmlns:tns="http://control-
flow/interleaved_parallel_routingInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
name="interleaved_parallel_routing" targetNamespace="http://control-
flow/interleaved_parallel_routingInterface">
    <types>
        <xsd:schema targetNamespace="http://control-
flow/interleaved_parallel_routingInterface" xmlns:bons1="http://control-
flow/control_flow" xmlns:tns="http://control-
flow/interleaved_parallel_routingInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
            <xsd:import namespace="http://control-flow/control_flow"
schemaLocation="control_flow/ipr.xsd"/>
            <xsd:element name="operation1">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="input1" nillable="true"
type="bons1:ipr"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="operation1Response">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="output1" nillable="true"
type="bons1:ipr"/>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        </xsd:schema>
    </types>
    <message name="operation1Request">
        <part element="tns:operation1" name="operation1Parameters"/>
    </message>
    <message name="operation1Response">
        <part element="tns:operation1Response" name="operation1Result"/>
    </message>
    <portType name="interleaved_parallel_routing">
        <operation name="operation1">
            <input message="tns:operation1Request"
name="operation1Request"/><output message="tns:operation1Response"
name="operation1Response"/>
        </operation>
    </portType>
</definitions>

```

The content of the XSD file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://control-flow/control_flow">
  <xsd:complexType name="ipr">
    <xsd:sequence>
      <xsd:element minOccurs="0" name="array" maxOccurs="unbounded">
        <xsd:simpleType>
          <xsd:restriction base="xsd:int">
            <xsd:minInclusive value="1"/>
            <xsd:maxInclusive value="3"/>
            <xsd:enumeration value="1"/>
            <xsd:enumeration value="2"/>
            <xsd:enumeration value="3"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

Pattern 18 (Milestone)

The content of the BPEL file is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<bpws:process xmlns:bpws="http://schemas.xmlsoap.org/ws/2004/03/business-
process/" xmlns:ns="http://control-flow/milestoneArtifacts"
xmlns:ns0="http://control-flow/milestoneInterface"
xmlns:wpc="http://www.ibm.com/xmlns/prod/websphere/business-
process/6.0.0/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.ibm.com/xmlns/prod/websphere/business-
process/expression-lang/java/6.0.0/" name="milestone"
suppressJoinFailure="yes" targetNamespace="http://control-flow"
wpc:autonomy="peer" wpc:displayName="milestone" wpc:id="1">
  <bpws:import importType="http://schemas.xmlsoap.org/wsd1/"
location="milestone.wsdl" namespace="http://control-
flow/milestoneInterface"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wsd1/"
location="milestoneArtifacts.wsdl" namespace="http://control-
flow/milestoneArtifacts"/>
  <bpws:partnerLinks>
    <bpws:partnerLink myRole="MilestoneRole" name="Milestone"
partnerLinkType="ns:MilestonePLT"/>
  </bpws:partnerLinks>
  <bpws:variables>
    <bpws:variable name="Input" type="xsd:string" wpc:id="2"/>
    <bpws:variable name="Output" type="xsd:string" wpc:id="7"/>
    <bpws:variable name="milestone" type="xsd:boolean" wpc:id="8"/>
    <bpws:variable name="control" type="xsd:string" wpc:id="24"/>
  </bpws:variables>
  <bpws:correlationSets>
    <bpws:correlationSet name="CorrelationSet" properties="ns:Property"/>
  </bpws:correlationSets>
  <bpws:sequence name="HiddenSequence" wpc:id="1073741827">
    <bpws:receive createInstance="yes" name="Receive"
operation="operation1" partnerLink="Milestone" portType="ns0:Milestone"
wpc:displayName="Receive" wpc:id="4">
      <wpc:output>
        <wpc:parameter name="input1" variable="Input"/>
      </wpc:output>
    </bpws:receive>
  </bpws:sequence>
</bpws:process>
```

```

        </wpc:output>
    </bpws:receive>
    <bpws:assign name="Setmilestonetrue" wpc:displayName="Set milestone =
true" wpc:id="6">
        <bpws:copy>
            <bpws:from>true</bpws:from>
            <bpws:to variable="milestone"/>
        </bpws:copy>
    </bpws:assign>
    <bpws:while name="LoopWhilemilestonetrue" wpc:displayName="Loop While
milestone = true" wpc:id="9">
        <bpws:condition><![CDATA[return milestone.booleanValue();
        <bpws:pick name="ReceiveChoice" wpc:displayName="ReceiveChoice"
wpc:id="12">
            <bpws:onMessage operation="operation1" partnerLink="Milestone"
portType="ns0:Milestone">
                <wpc:output>
                    <wpc:parameter name="input1" variable="control"/>
                </wpc:output>
                <bpws:correlations>
                    <bpws:correlation initiate="yes" set="CorrelationSet"/>
                </bpws:correlations>
                <bpws:switch name="Choice" wpc:displayName="Choice"
wpc:id="18">
                    <bpws:case wpc:id="19">
                        <bpws:condition><![CDATA[ java.lang.String __result__2 =
"continue";
boolean __result__3;
{// text equal to (ignore case)
    __result__3 = control.equalsIgnoreCase(__result__2);
}
return __result__3;
                    <bpws:sequence name="HiddenSequence1" wpc:id="1073741844">
                        <bpws:empty name="EmptyAction1"
wpc:displayName="EmptyAction" wpc:id="21"/>
                    </bpws:sequence>
                    </bpws:case>
                    <bpws:case wpc:id="22">
                        <bpws:condition><![CDATA[ java.lang.String __result__2 =
"stop";
boolean __result__3;
{// text equal to (ignore case)
    __result__3 = control.equalsIgnoreCase(__result__2);
}
return __result__3;
                    <bpws:sequence name="HiddenSequence2" wpc:id="1073741847">
                        <bpws:assign name="Setmilestonefalse"
wpc:displayName="Set milestone = false" wpc:id="16">
                            <bpws:copy>
                                <bpws:from>false</bpws:from>
                                <bpws:to variable="milestone"/>
                            </bpws:copy>
                        </bpws:assign>
                    </bpws:sequence>
                    </bpws:case>
                </bpws:switch>
            </bpws:onMessage>
        </bpws:pick>
    </bpws:while>
    <bpws:assign name="Assign" wpc:displayName="Assign" wpc:id="25">
        <bpws:copy>
            <bpws:from variable="Input"/>
            <bpws:to variable="Output"/>
        </bpws:copy>
    </bpws:assign>
    <bpws:reply name="Reply" operation="operation1"

```

```

partnerLink="Milestone" portType="ns0:Milestone" wpc:displayName="Reply"
wpc:id="5">
  <wpc:input>
    <wpc:parameter name="output1" variable="Output"/>
  </wpc:input>
</bpws:reply>
</bpws:sequence>
</bpws:process>

```

The content of the WSDL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:tns="http://control-flow/milestoneInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/" name="milestone"
targetNamespace="http://control-flow/milestoneInterface">
  <types>
    <xsd:schema targetNamespace="http://control-flow/milestoneInterface"
xmlns:tns="http://control-flow/milestoneInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:element name="operation1">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="input1" nillable="true"
type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="operation1Response">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="output1" nillable="true"
type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </types>
  <message name="operation1Request">
    <part element="tns:operation1" name="operation1Parameters"/>
  </message>
  <message name="operation1Response">
    <part element="tns:operation1Response" name="operation1Result"/>
  </message>
  <portType name="Milestone">
    <operation name="operation1">
      <input message="tns:operation1Request" name="operation1Request"/>
      <output message="tns:operation1Response"
name="operation1Response"/>
    </operation>
  </portType>
</definitions>

```

Pattern 19 (Cancel Activity)

The content of the BPEL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<bpws:process xmlns:bpws="http://schemas.xmlsoap.org/ws/2004/03/business-

```

```

process/" xmlns:ns="http://control-flow/cancel_activityArtifacts"
xmlns:ns0="http://control-flow/cancel_activityInterface"
xmlns:wpc="http://www.ibm.com/xmlns/prod/websphere/business-
process/6.0.0/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.ibm.com/xmlns/prod/websphere/business-
process/expression-lang/java/6.0.0/" name="cancel_activity"
suppressJoinFailure="yes" targetNamespace="http://control-flow"
wpc:autonomy="peer" wpc:displayName="cancel_activity" wpc:id="1">
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="cancel_activity.wsdl" namespace="http://control-
flow/cancel_activityInterface"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="cancel_activityArtifacts.wsdl" namespace="http://control-
flow/cancel_activityArtifacts"/>
  <bpws:partnerLinks>
    <bpws:partnerLink myRole="Cancel_activityRole" name="Cancel_activity"
partnerLinkType="ns:Cancel_activityPLT"/>
    <bpws:partnerLink myRole="myRole" name="Partner"
partnerLinkType="ns:PartnerPLT"/>
    <bpws:partnerLink name="Partner1" partnerLinkType="ns:Partner1PLT"
partnerRole="partnerRole"/>
  </bpws:partnerLinks>
  <bpws:variables>
    <bpws:variable name="Input1" type="xsd:string" wpc:id="2"/>
    <bpws:variable name="Output" type="xsd:string" wpc:id="11"/>
  </bpws:variables>
  <bpws:correlationSets>
    <bpws:correlationSet name="CorrelationSet" properties="ns:property"/>
  </bpws:correlationSets>
  <bpws:sequence name="HiddenSequence" wpc:id="1073741827">
    <bpws:receive createInstance="yes" name="Receive"
operation="operation1" partnerLink="Cancel_activity"
portType="ns0:Cancel_activity" wpc:displayName="Receive" wpc:id="4">
      <wpc:output>
        <wpc:parameter name="input1" variable="Input1"/>
      </wpc:output>
    </bpws:receive>
    <bpws:scope name="Scope" wpc:displayName="Scope" wpc:id="12">
      <bpws:eventHandlers>
        <bpws:onEvent operation="operation1"
partnerLink="Cancel_activity" portType="ns0:Cancel_activity">
          <wpc:output>
            <wpc:parameter name="input1" type="xsd:string"
variable="message" wpc:variableId="15"
wpc:variableIsBusinessRelevant="no"/>
          </wpc:output>
          <bpws:correlations>
            <bpws:correlation initiate="yes" set="CorrelationSet"/>
          </bpws:correlations>
          <bpws:terminate name="Terminate" wpc:displayName="Terminate"
wpc:id="14"/>
        </bpws:onEvent>
      </bpws:eventHandlers>
      <bpws:assign name="Assign" wpc:displayName="Assign" wpc:id="13">
        <bpws:copy>
          <bpws:from variable="Input1"/>
          <bpws:to variable="Output"/>
        </bpws:copy>
      </bpws:assign>
    </bpws:scope>
    <bpws:reply name="Reply" operation="operation1"
partnerLink="Cancel_activity" portType="ns0:Cancel_activity"
wpc:displayName="Reply" wpc:id="5">
      <wpc:input>
        <wpc:parameter name="output1" variable="Output"/>
      </wpc:input>
    </bpws:reply>

```



```

    </bpws:sequence>
</bpws:process>

```

The content of the WSDL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:tns="http://control-flow/cancel_activityInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/" name="cancel_activity"
targetNamespace="http://control-flow/cancel_activityInterface">
  <types>
    <xsd:schema targetNamespace="http://control-
flow/cancel_activityInterface" xmlns:tns="http://control-
flow/cancel_activityInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:element name="operation1">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="input1" nillable="true"
type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="operation1Response">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="output1" nillable="true"
type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </types>
  <message name="operation1Request">
    <part element="tns:operation1" name="operation1Parameters"/>
  </message>
  <message name="operation1Response">
    <part element="tns:operation1Response" name="operation1Result"/>
  </message>
  <portType name="Cancel_activity">
    <operation name="operation1">
      <input message="tns:operation1Request" name="operation1Request"/>
      <output message="tns:operation1Response"
name="operation1Response"/>
    </operation>
  </portType>
</definitions>

```

Pattern 20 (Cancel Case)

The content of the BPEL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<bpws:process xmlns:bpws="http://schemas.xmlsoap.org/ws/2004/03/business-

```

```

process/" xmlns:ns="http://control-flow/cancel_caseArtifacts"
xmlns:ns0="http://control-flow/cancel_caseInterface"
xmlns:wpc="http://www.ibm.com/xmlns/prod/websphere/business-
process/6.0.0/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.ibm.com/xmlns/prod/websphere/business-
process/expression-lang/java/6.0.0/" name="cancel_case"
suppressJoinFailure="yes" targetNamespace="http://control-flow"
wpc:autonomy="peer" wpc:displayName="cancel_case" wpc:id="1">
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="cancel_case.wsdl" namespace="http://control-
flow/cancel_caseInterface"/>
  <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="cancel_caseArtifacts.wsdl" namespace="http://control-
flow/cancel_caseArtifacts"/>
  <bpws:partnerLinks>
    <bpws:partnerLink myRole="Cancel_caseRole" name="Cancel_case"
partnerLinkType="ns:Cancel_casePLT"/>
  </bpws:partnerLinks>
  <bpws:variables>
    <bpws:variable name="Input1" type="xsd:string" wpc:id="2"/>
    <bpws:variable name="Output" type="xsd:string" wpc:id="7"/>
  </bpws:variables>
  <bpws:correlationSets>
    <bpws:correlationSet name="CorrelationSet" properties="ns:property"/>
  </bpws:correlationSets>
  <bpws:eventHandlers>
    <bpws:onEvent operation="operation1" partnerLink="Cancel_case"
portType="ns0:Cancel_case">
      <wpc:output>
        <wpc:parameter name="input1" type="xsd:string" variable="message"
wpc:variableId="9" wpc:variableIsBusinessRelevant="no"/>
      </wpc:output>
      <bpws:correlations>
        <bpws:correlation initiate="yes" set="CorrelationSet"/>
      </bpws:correlations>
      <bpws:terminate name="Terminate" wpc:displayName="Terminate"
wpc:id="6"/>
    </bpws:onEvent>
  </bpws:eventHandlers>
  <bpws:sequence name="HiddenSequence" wpc:id="1073741827">
    <bpws:receive createInstance="yes" name="Receive"
operation="operation1" partnerLink="Cancel_case"
portType="ns0:Cancel_case" wpc:displayName="Receive" wpc:id="4">
      <wpc:output>
        <wpc:parameter name="input1" variable="Input1"/>
      </wpc:output>
    </bpws:receive>
    <bpws:assign name="Assign" wpc:displayName="Assign" wpc:id="8">
      <bpws:copy>
        <bpws:from variable="Input1"/>
        <bpws:to variable="Output"/>
      </bpws:copy>
    </bpws:assign>
    <bpws:reply name="Reply" operation="operation1"
partnerLink="Cancel_case" portType="ns0:Cancel_case"
wpc:displayName="Reply" wpc:id="5">
      <wpc:input>
        <wpc:parameter name="output1" variable="Output"/>
      </wpc:input>
    </bpws:reply>
  </bpws:sequence>
</bpws:process>

```

The content of the WSDL file is shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:tns="http://control-flow/cancel_caseInterface"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/" name="cancel_case"
targetNamespace="http://control-flow/cancel_caseInterface">
  <types>
    <xsd:schema targetNamespace="http://control-
flow/cancel_caseInterface" xmlns:tns="http://control-
flow/cancel_caseInterface" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:element name="operation1">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="input1" nillable="true"
type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="operation1Response">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="output1" nillable="true"
type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </types>
  <message name="operation1Request">
    <part element="tns:operation1" name="operation1Parameters"/>
  </message>
  <message name="operation1Response">
    <part element="tns:operation1Response" name="operation1Result"/>
  </message>
  <portType name="Cancel_case">
    <operation name="operation1">
      <input message="tns:operation1Request" name="operation1Request"/>
      <output message="tns:operation1Response"
name="operation1Response"/>
    </operation>
  </portType>
</definitions>

```