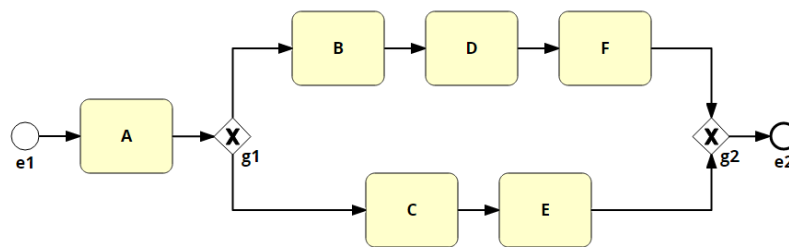


Mapping between BPMN to BPEL

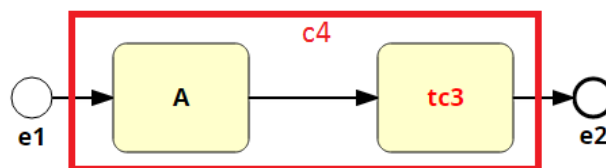
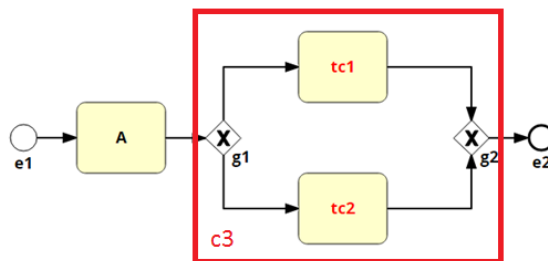
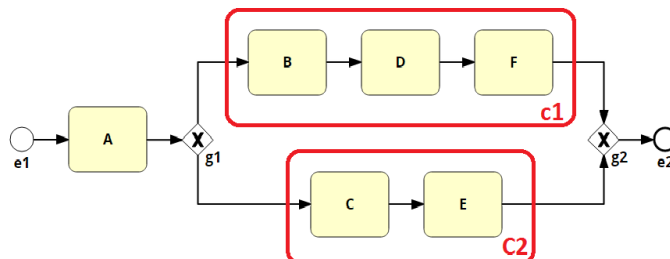
Mapping

It is composed of 2 steps: (1) component folding, and (2) gateway mappings.

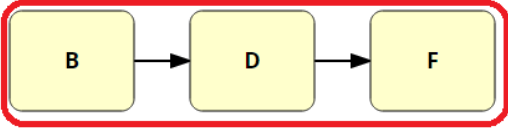
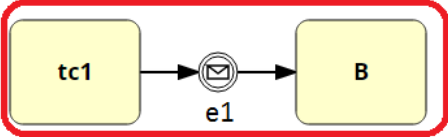
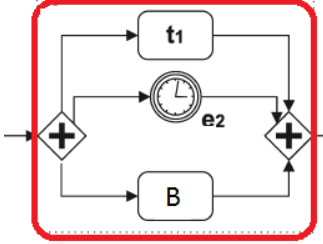
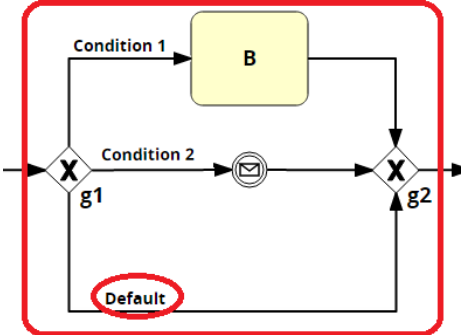
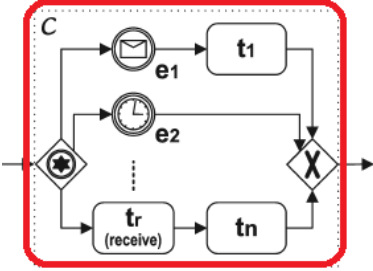
1) Component Folding

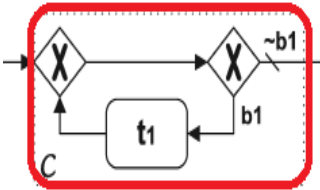
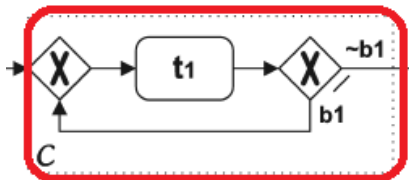
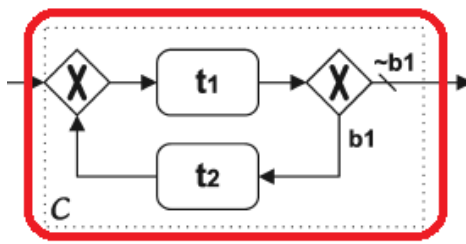


Step 1:

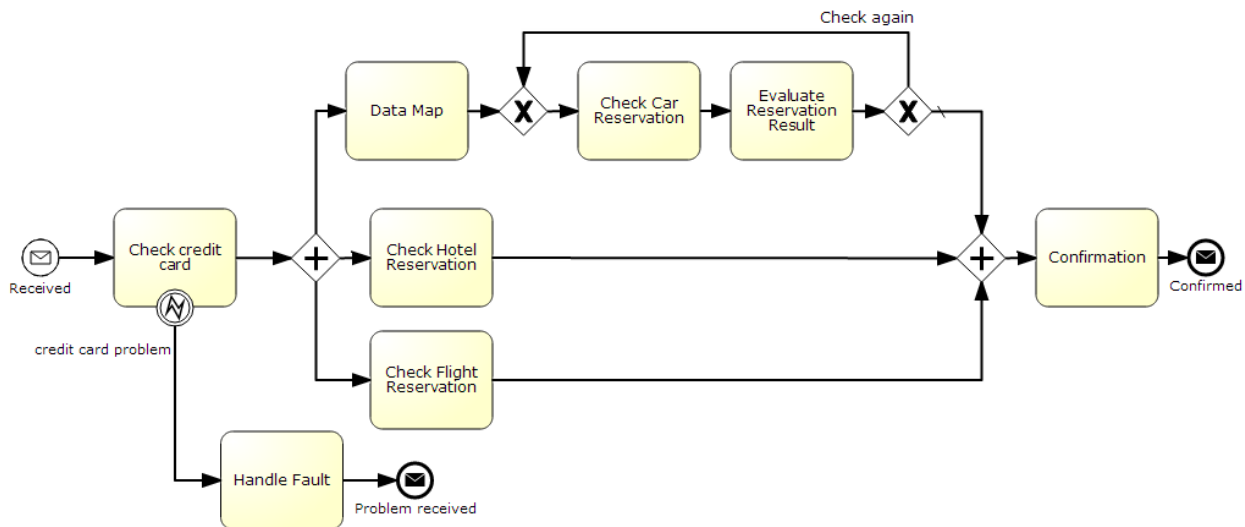


2) Getways and mapping rules

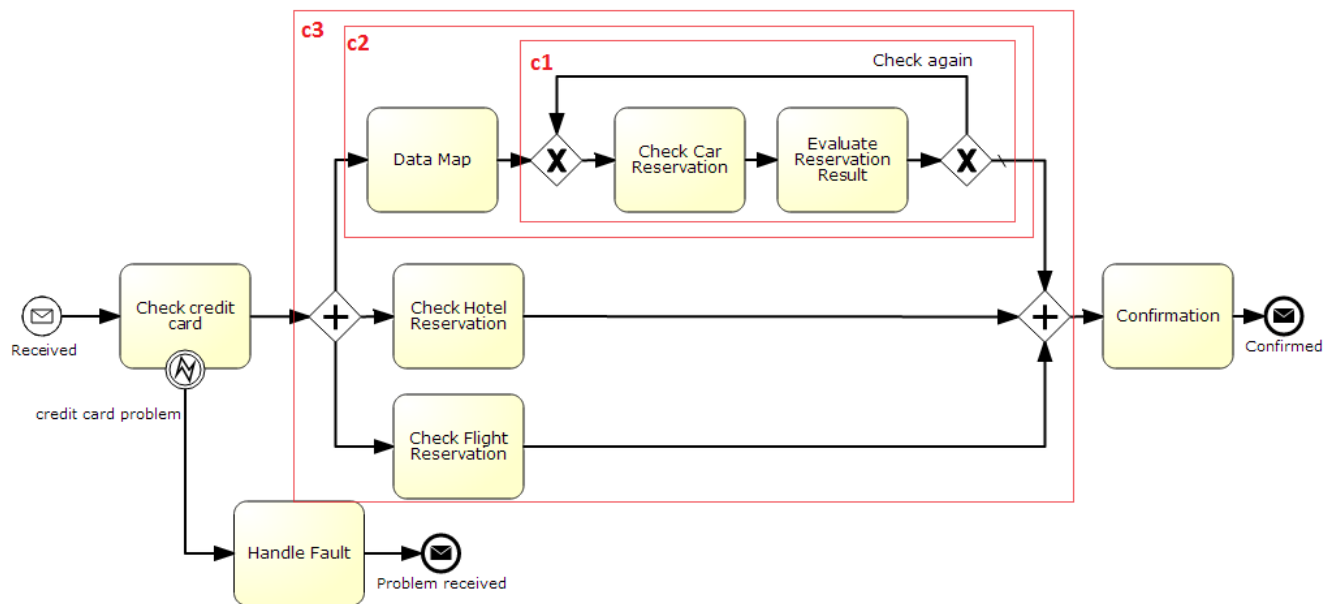
<p style="text-align: center;">Sequence Pattern</p> 	<pre> <sequence> <invoke name="B" /> <invoke name="D" /> <invoke name="F" /> </sequence> </pre>
<p style="text-align: center;">Sequence Pattern</p> 	<pre> <sequence> Mapping tc1 <receive name="e1" /> <invoke name="B"> </sequence> </pre>
<p style="text-align: center;">Flow Pattern</p> 	<pre> <flow> Mapping t1 <wait name="e2"/> <invoke name="B" /> </flow> </pre>
<p style="text-align: center;">Switch Pattern</p> 	<pre> <switch> <case condition ="condition1"> <invoke name="B" /> </case> <case condition ="condition2"> <receive name="e2" /> </case> <otherwise> <empty/> </otherwise> </switch> </pre>
<p style="text-align: center;">Pick Pattern</p> 	<pre> <pick> <onMessage name="e1"> Mapping t1 </onMessage> <onAlarm name="e2"> <empty/> </onAlarm> < onMessage name ="tr"> Mapping tn </ onMessage> </pick> </pre>

<p>While Pattern</p> 	<pre><while condition= "b1"> Mapping(t1) </while></pre>
<p>Repeat Pattern</p> 	<pre><sequence name= "tc" > Mapping (t1) <while condition= "b1"> Mapping(t1) </while> </sequence></pre>
<p>Repeat +While Pattern</p> 	<pre><sequence name= "tc" > Mapping (t1) <while condition= "b1"> <sequence> Mapping(t2) Mapping(t1) </sequence> </while> </sequence></pre>

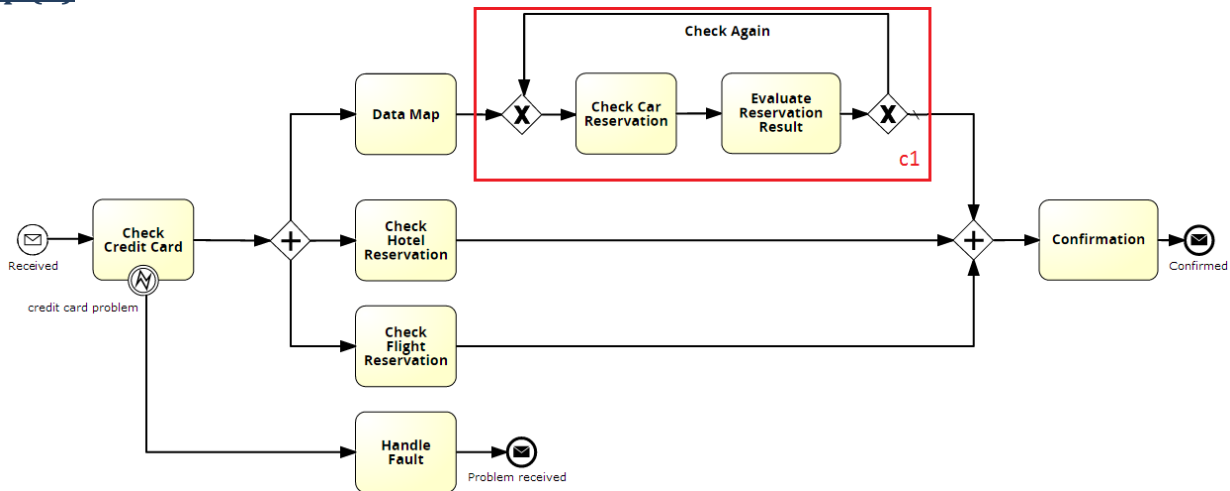
BPMN to BPEL Example1:



Solution:



Step (1)



<!-- Mapping c1-->

<sequence name="t_c¹">

<invoke name= "Check Car Reservation"/>

<invoke name= "Evaluate Reservation Result"/>

<while condition = "Check again">

<sequence>

<invoke name= "Check Car Reservation"/>

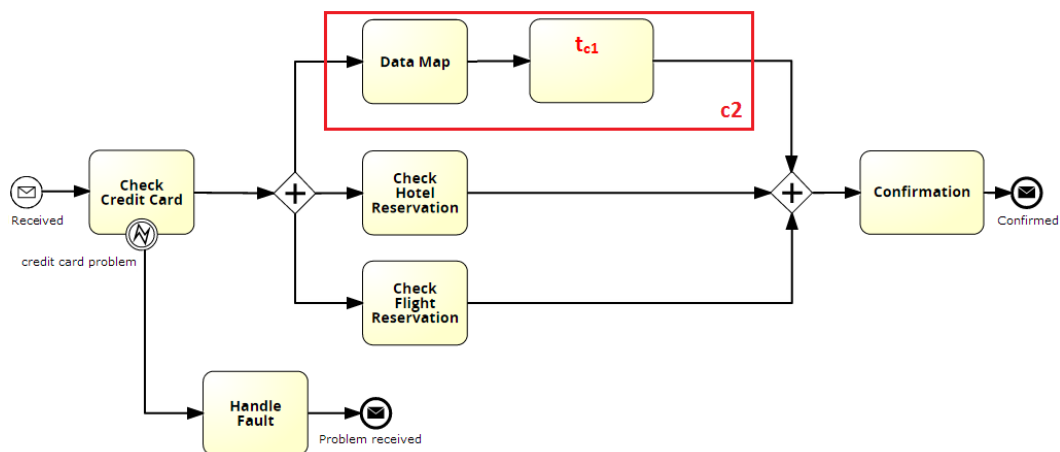
<invoke name= "Evaluate Reservation Result"/>

</sequence>

</while>

</sequence>

Step (2)



<!-- Mapping c2-->

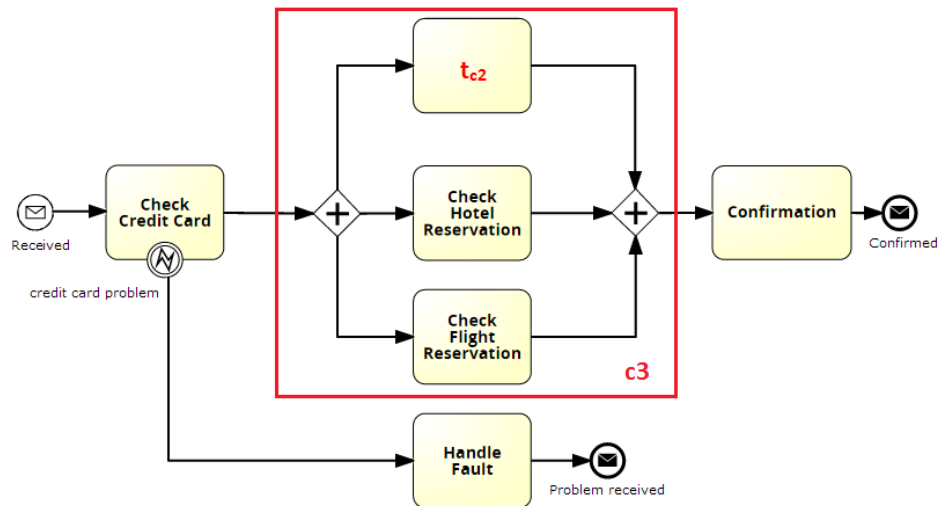
<sequence name="t_c²">

<invoke name= "Data Map"/>

Mapping (C1)

</sequence>

Step (3)



<!-- Mapping c3-->

<flow name="t_c³">

Mapping (c2)

<invoke name= "Check Hotel Reservation"/>

<invoke name= "Check Flight Reservation"/>

</flow>

Step (4)

<!-- Mapping Exception-->

<faultHandlers>

<catch faultName="exception" faultVariable= "Credit card problem">

<sequence>

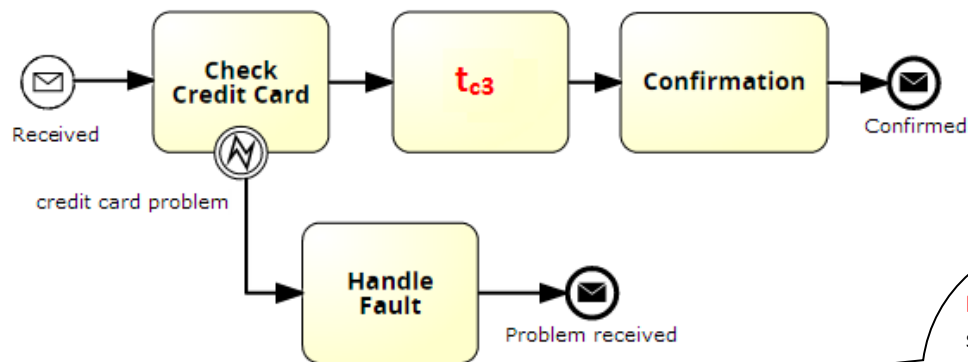
<invoke name= "Handle Fault"/>

< reply name= "Problem received"/>

</sequence>

</catch>

</faultHandlers>



<!-- Mapping the rest-->

<scope name="t_c⁴">

<faultHandlers>

<catch faultName="exception" name="Credit card problem">

<sequence>

<invoke name="Handle Fault"/>

<reply name="Problem received"/>

</sequence>

</catch>

</faultHandlers>

</scope>

<receive name="Received" create_instance="yes"/>

<sequence>

<invoke name="Check Credit Card"/>

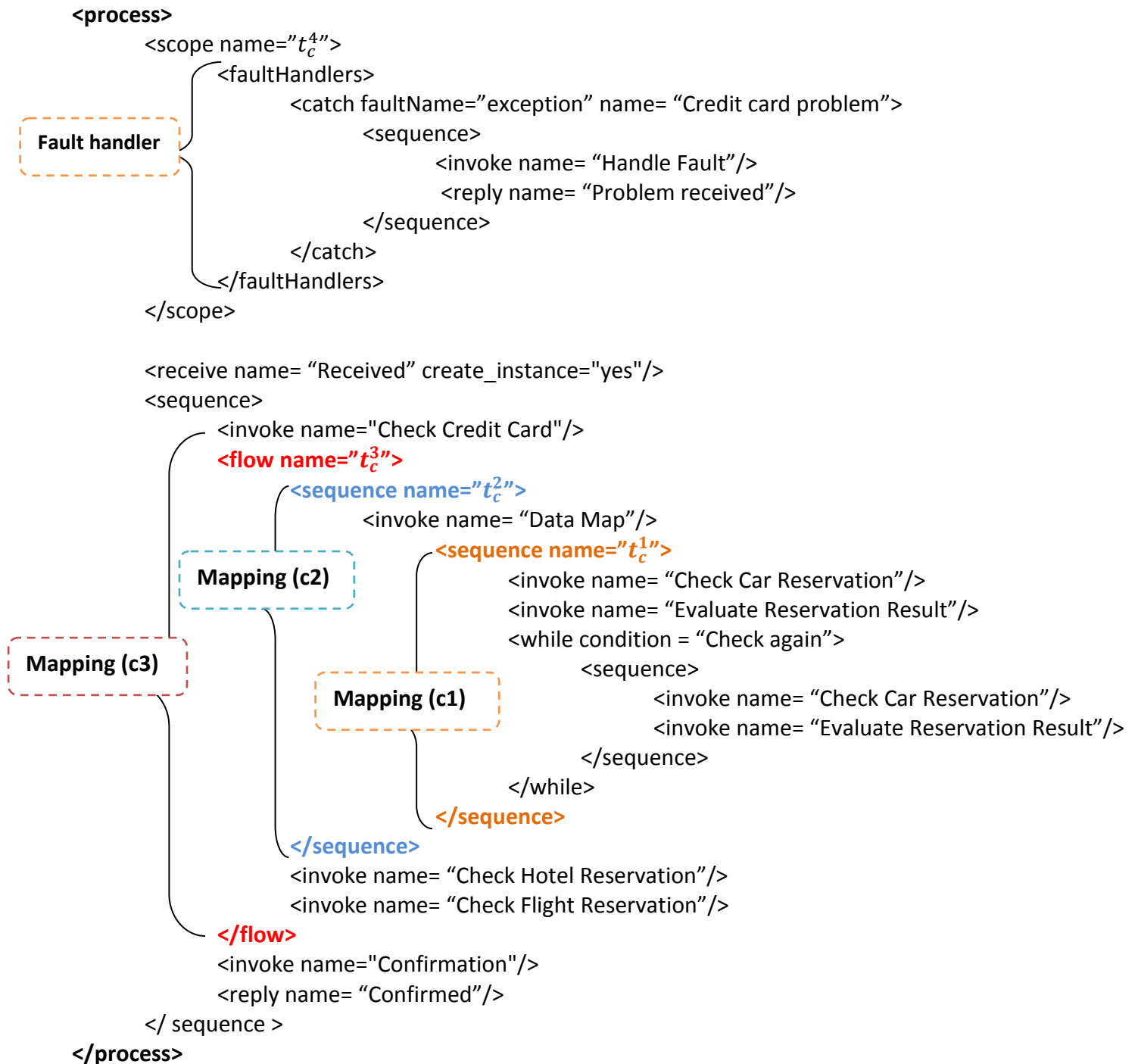
Mapping (c3)

<invoke name="Confirmation"/>

<reply name="Confirmed"/>

</sequence>

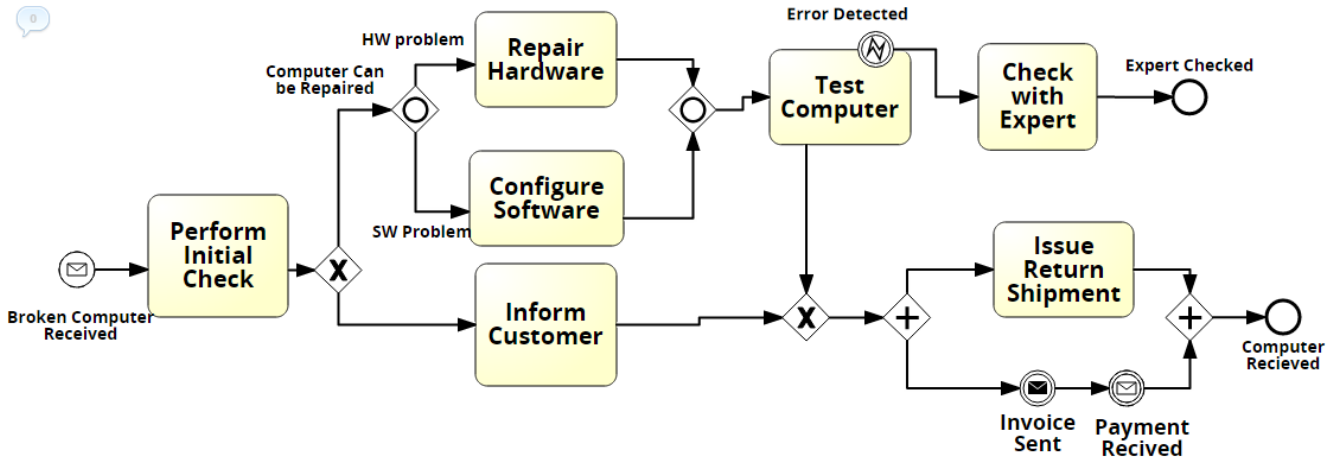
Note: The scope surrounding the FaultHandler is optional (all the examples)



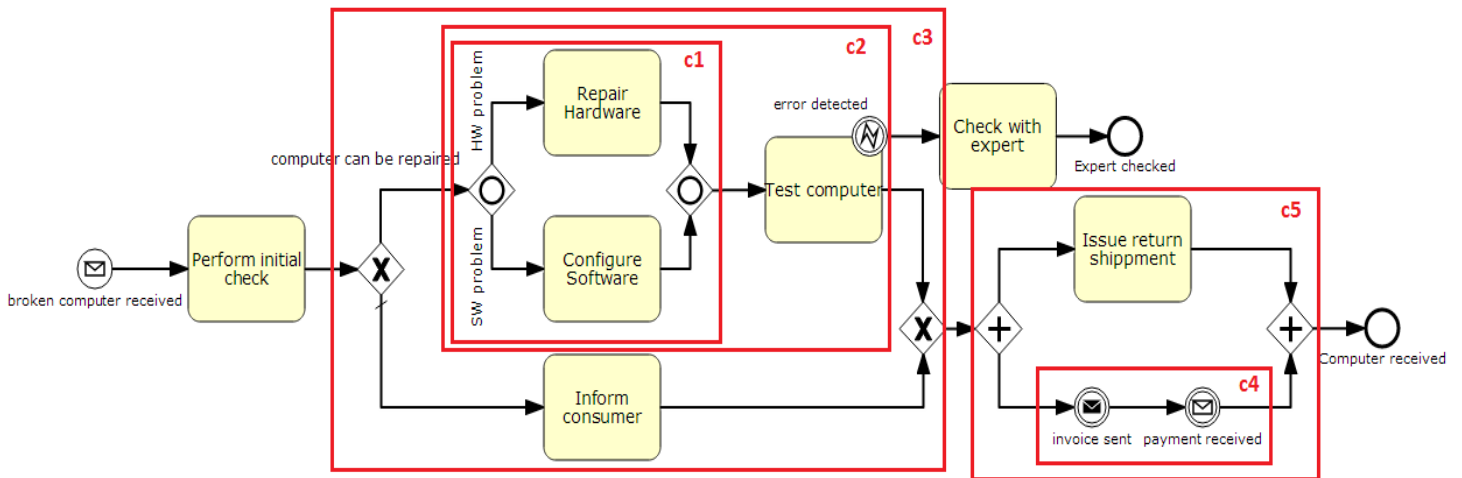
BPMN to BPEL Example2:

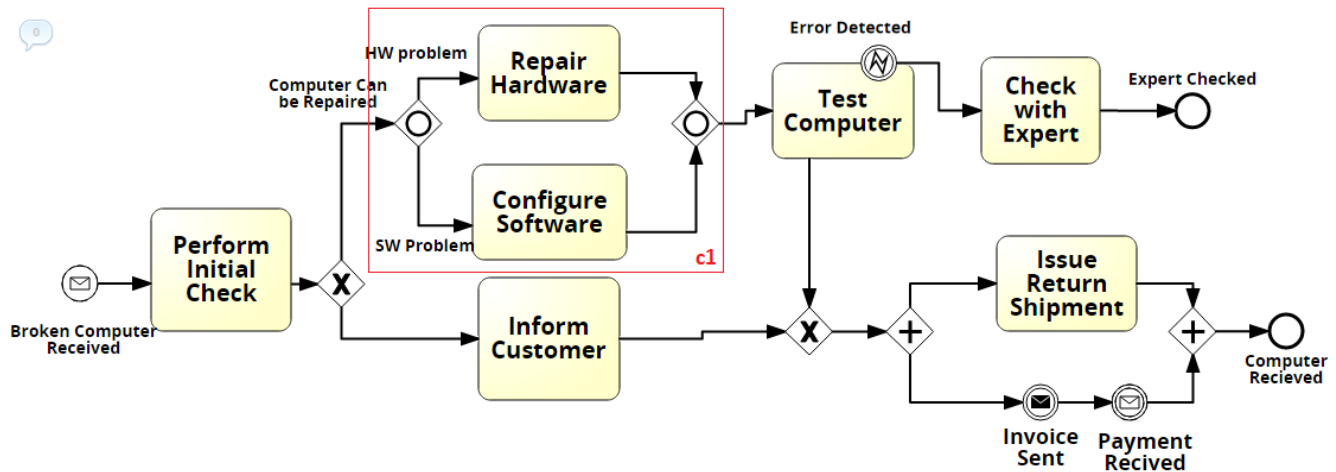
Computer Repair

Visualize this business process using BPMN.



Solution





Step (1)

<!--Mapping c1-->

<!--Mapping OR gateway to other known gateways for easier mapping to BPEL--!>

<switch>

<case condition = "HW problem">

<invoke name="Repair hardware"/>

</case>

<case condition = "SW problem">

<invoke name="Configure software"/>

</case>

<case condition = "HW problem" and "SW problem">

<invoke name="Repair hardware"/>

<invoke name="Configure software"/>

</case>

</switch>

<!--Mapping OR gateway to other known gateways -- Another way--!>

<flow>

<switch>

<case condition = "HW problem">

<invoke name="Repair hardware"/>

</case>

<otherwise>

<empty/>

</otherwise>

</switch>

<switch>

<case condition = "SW problem">

<invoke name="Configure software"/>

</case>

<otherwise>

<empty/>

</otherwise>

</switch>

</flow>

Step (2)

```
<!-- Mapping c2-->
<sequence>
    Mapping (c1)
    <invoke name="Test Computer"/>
    <throw faultVariable= "error detected"/>
</ sequence >
```

Step (3)

```
<!-- Mapping c3-->
<flow>
    <switch>
        <case condition = "computer can be repaired">
            Mapping(c2)
            </case>
            <otherwise>
                <invoke name= "Inform customer"/>
            </otherwise >
        </switch>
    </flow>
```

Step (4)

```
<!-- Mapping c4-->
<sequence>
    <reply name= "Invoice Sent"/>
    <receive name= "Payment received"/>
</sequence>
```

Step (5)

```
<!-- Mapping c5-->
<flow>
    <invoke name= "Issue return shipment"/>
    Mapping (c4)
</flow>
```

Step (6)

```
<!-- Mapping Exception-->
<FaultHandlers>
    <catch faultName= "exception" faultVariable= "error detected">
        <invoke name= "Check with Expert"/>
        <reply name= "Expert Checked"/>
    </catch>
</FaultHandlers>
```

Step (7)

<!-- Mapping the rest-->

```
<sequence>
  <receive name="Broken Computer Received"/>
  <invoke name="Perform Initial Check">
    Mapping (c3)
    Mapping (c5)
    <reply name= "computer received"/>
  </sequence>
```

Step (8) Process:

```
<process>
  <FaultHandlers>
    <catch faultName= "exception" faultVariable= "error detected">
      <invoke name= "Check with Expert"/>
      <reply name= "Expert Checked"/>
    </catch>
  </FaultHandlers>
  <receive name="Broken Computer Received"/>
  <sequence>
    <receive name="Broken Computer Received"/>
    <invoke name= "Perform Initial check"/>
    Mapping(c3)
    Mapping(c5)
    <reply name= "computer received"/>
  </sequence>
</process>
```

Complete Process:

<!--The whole Mapping -->

```

<process>
  <FaultHandlers>
    <catch faultName= "exception" faultVariable= "error detected">
      <invoke name= "Check with Expert"/>
      <reply name= "Expert Checked"/>
    </catch>
  </FaultHandlers>
  <sequence>
    <receive name="Broken Computer Received"/>
    <invoke name="Perform Initial Check">
      <flow>
        <switch>
          <case condition = "computer can be repaired">
            <sequence>
              <switch>
                <case condition = "HW problem">
                  <invoke name="Repair hardware"/>
                </case>
                <case condition = "SW problem">
                  <invoke name="Configure software"/>
                </case>
                <case condition = "HW problem" and "SW problem">
                  <invoke name="Repair hardware"/>
                  <invoke name="Configure software"/>
                </case>
              </switch>
              <invoke name="Test Computer"/>
              <throw faultVariable= "error detected"/>
            </ sequence >
          </case>
          <otherwise>
            <invoke name= "Inform customer"/>
          </otherwise >
        </switch>
      </flow>
      <flow>
        <invoke name= "Issue return shipment"/>
        <sequence>
          <reply name= "Invoice Sent"/>
          <receive name= "Payment received"/>
        </sequence>
      </flow>
    </sequence>
  </process>

```

Mapping (c1) (orange dashed box):

- <switch>
 <case condition = "HW problem">
 <invoke name="Repair hardware"/>
 </case>
 <case condition = "SW problem">
 <invoke name="Configure software"/>
 </case>
 <case condition = "HW problem" and "SW problem">
 <invoke name="Repair hardware"/>
 <invoke name="Configure software"/>
 </case>
 </switch>
 <invoke name="Test Computer"/>
 <throw faultVariable= "error detected"/>

Mapping (c2) (blue dashed box):

- <sequence>
 <switch>
 <case condition = "computer can be repaired">
 <sequence>
 <switch>
 <case condition = "HW problem">
 <invoke name="Repair hardware"/>
 </case>
 <case condition = "SW problem">
 <invoke name="Configure software"/>
 </case>
 <case condition = "HW problem" and "SW problem">
 <invoke name="Repair hardware"/>
 <invoke name="Configure software"/>
 </case>
 </switch>
 <invoke name="Test Computer"/>
 <throw faultVariable= "error detected"/>
 </ sequence >
 </case>
 <otherwise>
 <invoke name= "Inform customer"/>
 </otherwise >
 </switch>
 </sequence>

Mapping (c3) (red dashed box):

- <sequence>
 <switch>
 <case condition = "computer can be repaired">
 <sequence>
 <switch>
 <case condition = "HW problem">
 <invoke name="Repair hardware"/>
 </case>
 <case condition = "SW problem">
 <invoke name="Configure software"/>
 </case>
 <case condition = "HW problem" and "SW problem">
 <invoke name="Repair hardware"/>
 <invoke name="Configure software"/>
 </case>
 </switch>
 <invoke name="Test Computer"/>
 <throw faultVariable= "error detected"/>
 </ sequence >
 </case>
 <otherwise>
 <invoke name= "Inform customer"/>
 </otherwise >
 </switch>
 </sequence>

Mapping (c4) (orange dashed box):

- <sequence>
 <reply name= "Invoice Sent"/>
 <receive name= "Payment received"/>
 </sequence>

Mapping (c5) (orange dashed box):

- <flow>
 <invoke name= "Issue return shipment"/>
 </flow>

BPMN to BPEL Example3:

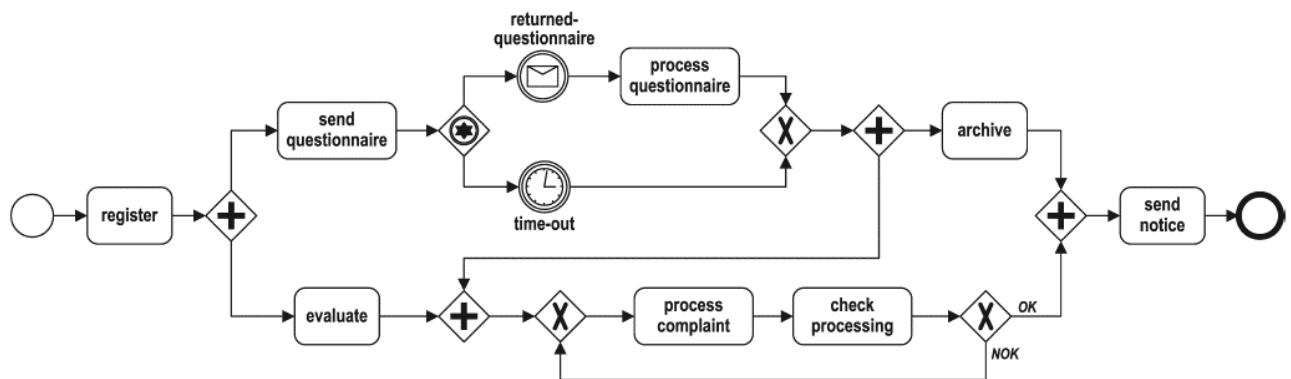
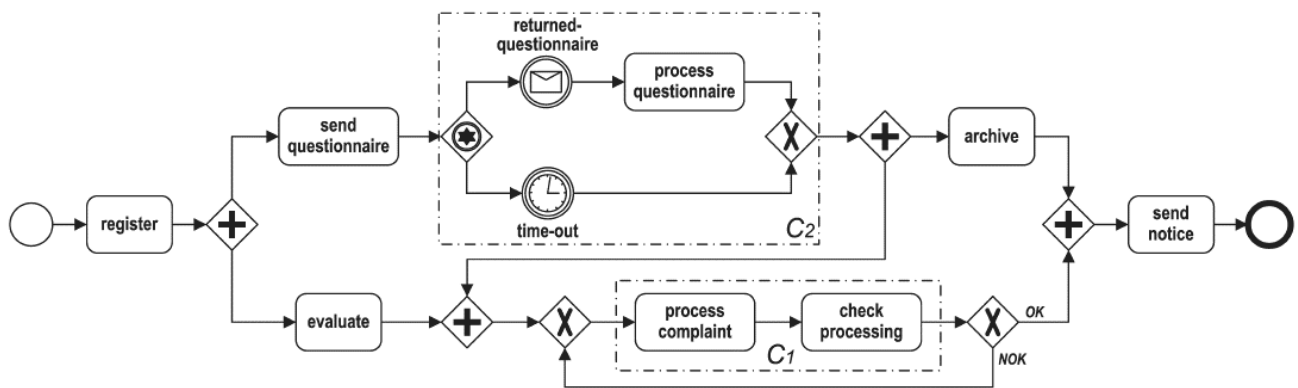


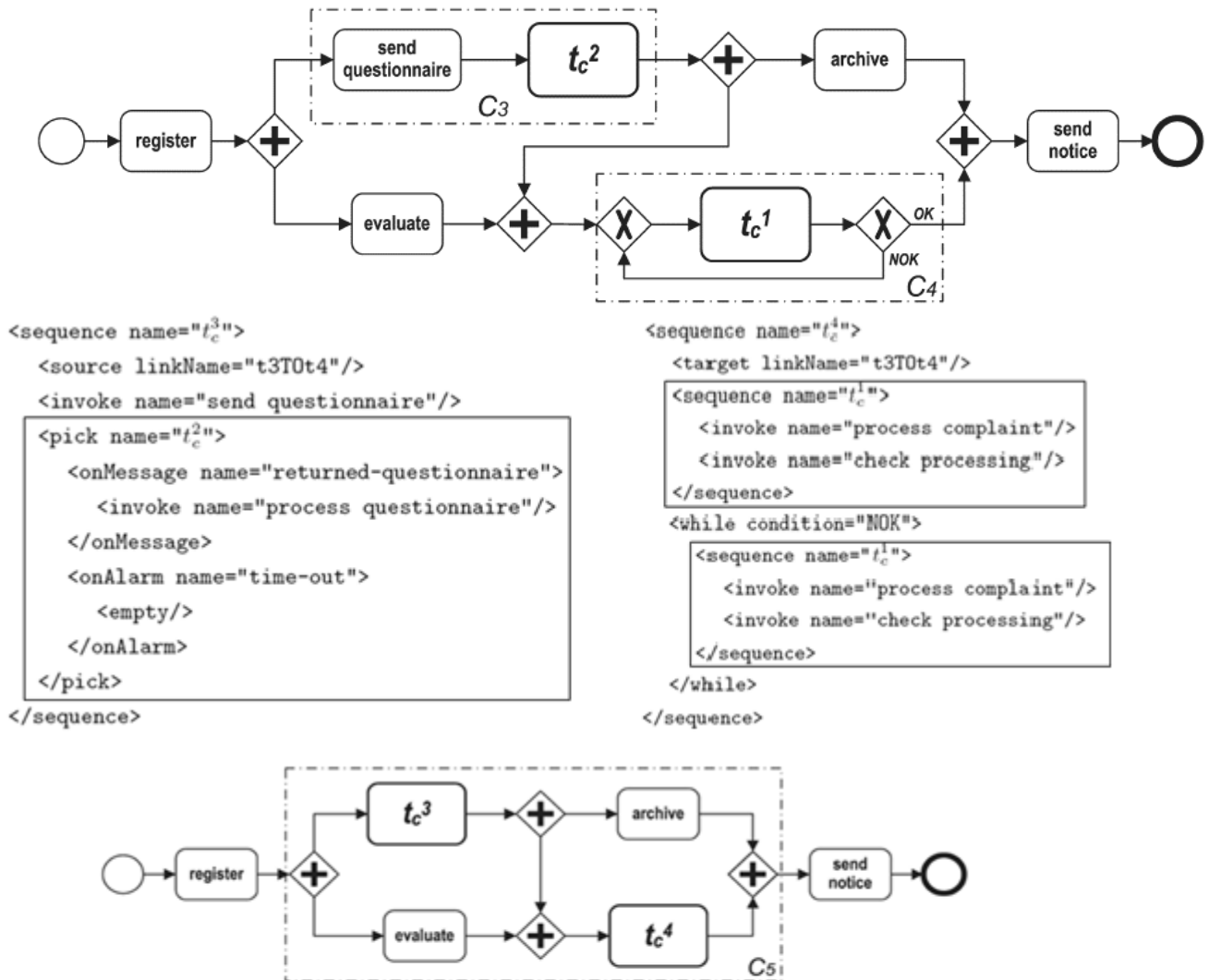
Figure 6. A complaint handling process model.

Solution:

Step (1)

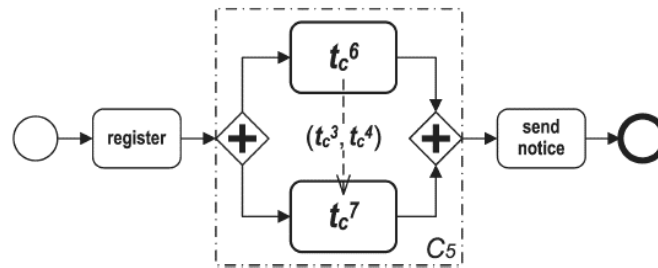


```
<pick name="tc2">
  <onMessage name="returned-questionnaire">
    <invoke name="process questionnaire"/>
  </onMessage>
  <onAlarm name="time-out">
    <empty/>
  </onAlarm>
</pick>
<sequence name="tc1">
  <invoke name="process complaint"/>
  <invoke name="check processing"/>
</sequence>
```



- Problem
 - AND synchronization between concurrent activities
 - Component C4 can begin only when evaluate and C3 complete
- Solution
 - Link between these components

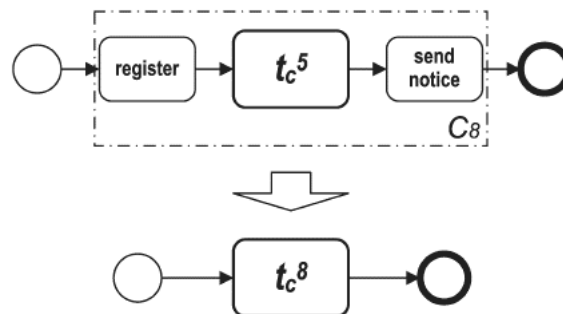
Step (3)



```
<sequence name=" $t_c^6$ ">
  <sequence name=" $t_c^3$ ">
    <source linkName="t3T0t4"/>
    <invoke name="send questionnaire"/>
    <pick name=" $t_c^2$ ">
      <onMessage name="returned-questionnaire">
        <invoke name="process questionnaire"/>
      </onMessage>
      <onAlarm name="time-out">
        <empty/>
      </onAlarm>
    </pick>
  </sequence>
  <invoke name="archive"/>
</sequence>
```

```
<sequence name=" $t_c^7$ ">
  <invoke name="evaluate"/>
  <sequence name=" $t_c^4$ ">
    <target linkName="t3T0t4"/>
    <sequence name=" $t_c^1$ ">
      <invoke name="process complaint"/>
      <invoke name="check processing"/>
    </sequence>
    <while condition="NOK">
      <sequence name=" $t_c^1$ ">
        <invoke name="process complaint"/>
        <invoke name="check processing"/>
      </sequence>
    </while>
  </sequence>
</sequence>
```

Step (4)




```

<process>
  <links>
    <link name="t3T0t4"/>
  </links>
  <sequence name="tc8">
    <invoke name="register"/>
    <flow name="tc5">
      <sequence name="tc6">
        <sequence name="tc3">
          <source linkName="t3T0t4"/>
          <invoke name="send questionnaire"/>
          <pick name="tc2">
            <onMessage name="returned-questionnaire">
              <invoke name="process questionnaire"/>
            </onMessage>
            <onAlarm name="time-out">
              <empty/>
            </onAlarm>
          </pick>
        </sequence>
        <invoke name="archive"/>
      </sequence>
      <sequence name="tc7">
        <invoke name="evaluate"/>
        <sequence name="tc4">
          <target linkName="t3T0t4"/>
          <sequence name="tc1">
            <invoke name="process complaint"/>
            <invoke name="check processing"/>
          </sequence>
          <while condition="NOK">
            <sequence name="tc1">
              <invoke name="process complaint"/>
              <invoke name="check processing"/>
            </sequence>
          </while>
        </sequence>
      </sequence>
    </flow>
    <invoke name="send notice"/>
  </sequence>
</process>

```

Resources:

1. A tool for translating BPMN models into BPEL processes <https://code.google.com/p/bpmn2bpel/>
2. Using BPMN to Model a BPEL Process
http://www.omg.org/bpmn/Documents/Mapping_BPMN_to_BPEL_Example.pdf
3. Transforming BPMN into BPEL: Why and How <http://www.oracle.com/technetwork/articles/dikmans-bpm-101437.html>
4. On Visualizing and Modelling BPEL with BPMN <http://www.iaas.uni-stuttgart.de/RUS-data/INPROC-2009-27%20-%20On%20Visualizing%20and%20Modelling%20BPEL%20with%20BPMN.pdf>
5. From Business Process Models to Process-oriented Software Systems: The BPMN to BPEL Way
<http://eprints.qut.edu.au/5266/1/5266.pdf>