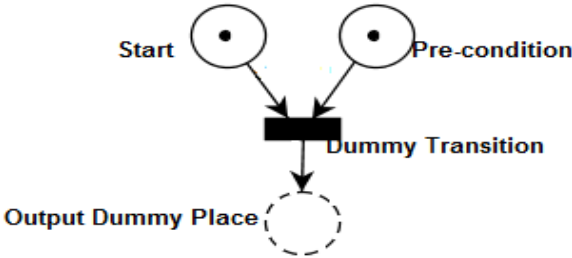
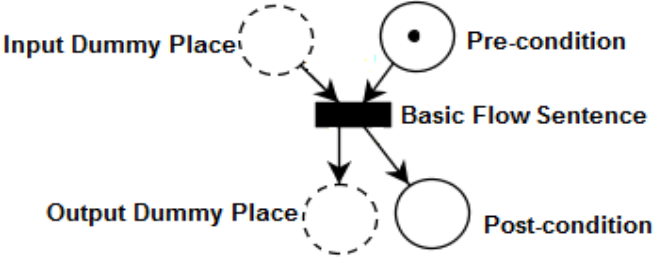
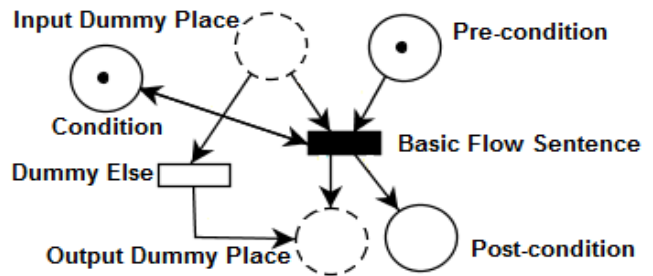
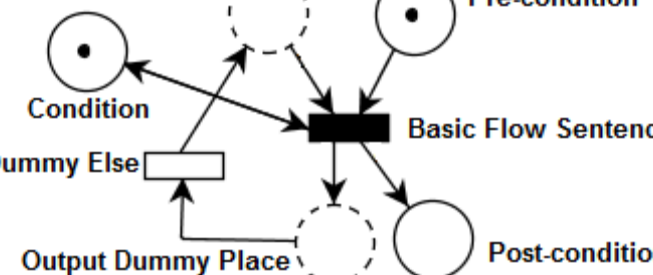
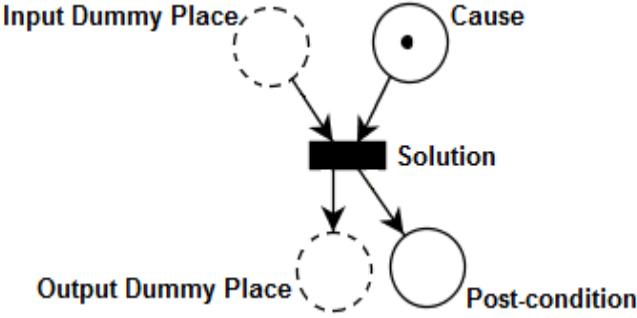
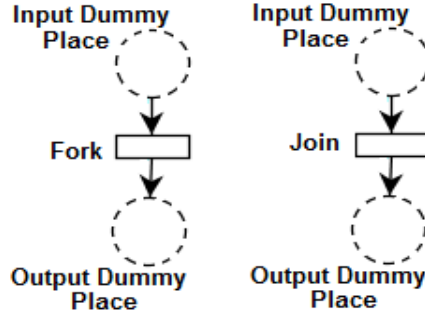


RUCM Element	Petri-Net Element
Initial state Title, Resource, Initial Context: -{Pre-condition}	 <p>The diagram for the Initial state shows a Petri net with two solid circles at the top, each containing a single black dot. The left circle is labeled 'Start' and the right circle is labeled 'Pre-condition'. Arrows from both circles point to a solid black rectangle in the center labeled 'Dummy Transition'. An arrow from the 'Dummy Transition' points down to a dashed circle labeled 'Output Dummy Place'.</p>
Simple Basic Flow <Id><Basic Flow Sentence> -{Pre-condition} -{Post-condition}	 <p>The diagram for the Simple Basic Flow shows a Petri net with three places and one transition. At the top left is a dashed circle labeled 'Input Dummy Place'. At the top right is a solid circle labeled 'Pre-condition' containing a single black dot. Arrows from both circles point to a solid black rectangle in the center labeled 'Basic Flow Sentence'. From the 'Basic Flow Sentence', two arrows point down to two circles: a dashed circle on the left labeled 'Output Dummy Place' and a solid circle on the right labeled 'Post-condition'.</p>

RUCM Element	Petri-Net Element
<p>Conditional Basic Flow <Id> IF {<Condition>} THEN <Basic Flow Sentence> -{Pre-condition} -{Post-condition}</p>	 <p>The diagram illustrates the Petri-Net structure for a conditional basic flow. It features a central black rectangle labeled 'Basic Flow Sentence'. To its left is a circle labeled 'Condition' containing a single token. Above the 'Condition' circle is a dashed circle labeled 'Input Dummy Place'. Below the 'Condition' circle is a rectangle labeled 'Dummy Else'. To the right of the 'Basic Flow Sentence' is a circle labeled 'Post-condition'. Above the 'Post-condition' circle is a circle labeled 'Pre-condition' containing a single token. Below the 'Post-condition' circle is a dashed circle labeled 'Output Dummy Place'. Arrows indicate the flow: from 'Pre-condition' to 'Basic Flow Sentence', from 'Condition' to 'Basic Flow Sentence', from 'Basic Flow Sentence' to 'Post-condition', and from 'Basic Flow Sentence' to 'Output Dummy Place'. A feedback arrow points from 'Output Dummy Place' to 'Input Dummy Place'.</p>
<p>Loop Basic Flow <Id> DO {<Condition>} WHILE <Basic Flow Sentence> -{Pre-condition} -{Post-condition}</p>	 <p>The diagram illustrates the Petri-Net structure for a loop basic flow. It features a central black rectangle labeled 'Basic Flow Sentence'. To its left is a circle labeled 'Condition' containing a single token. Above the 'Condition' circle is a dashed circle labeled 'Input Dummy Place'. Below the 'Condition' circle is a rectangle labeled 'Dummy Else'. To the right of the 'Basic Flow Sentence' is a circle labeled 'Post-condition'. Above the 'Post-condition' circle is a circle labeled 'Pre-condition' containing a single token. Below the 'Post-condition' circle is a dashed circle labeled 'Output Dummy Place'. Arrows indicate the flow: from 'Pre-condition' to 'Basic Flow Sentence', from 'Condition' to 'Basic Flow Sentence', from 'Basic Flow Sentence' to 'Post-condition', and from 'Basic Flow Sentence' to 'Output Dummy Place'. A feedback arrow points from 'Output Dummy Place' to 'Input Dummy Place'.</p>

RUCM Element	Petri-Net Element
<p>Specific Alternative Flow <Id>IF {<Cause>} THEN <Solution> -{Post-condition}</p>	
<p>Concurrency Construct #{Basic Flow Series}#</p>	
<p>Final State Context: -{Post-condition}</p>	