

```

<?xml version="1.0" encoding="UTF-8" ?>
<!-- edited with XML Spy v4.3 U (http://www.xmlspy.com) by Stephen White
(private) -->
- <process name="E-Mail Voting Process">
- <context>
  - <property name="NumVWGM">
    <value>13</value>
  </property>
  - <property name="AllItemsCompleted">
    <value>>false</value>
  </property>
  - <property name="NoMajority">
    <value>>true</value>
  </property>
</context>
- <sequence>
  <Documentation>This is the (complex) activity for the E-Mail Voting
  Process.</Documentation>
  <action name="Announce Issues for Discussion" portType="tns:emailPort"
  operation="sendDiscussionAnnouncement" />
  <delay type="duration:6 day" name="6 Day" />
  <action name="E-Mail Discussion Deadline Warning"
  portType="tns:emailPort" operation="sendDiscussionWarning" />
  <delay type="duration:1 day" name="1 Day" />
  <action name="Announce Issues for Vote" portType="tns:emailPort"
  operation="sendVoteAnnouncement" />
- <while condition="NoMajority">
  <Documentation>This while loop is indicated by the Issues 'w/o
  Majority?' Decision in the notation. But since the is an action that
  follows the decision before there is an upstream connection, the
  decision controls both the switch and the while at the same
  time.</Documentation>
  <call ref="Derived Sub-Process 1" />
  - <switch name="Issues w/o Majority?">
    - <case name="Yes" condition="NoMajority">
      - <action name="Reduce to Two Solutions"
        portType="tns:internalPort" operation="sendReceiveSolutions">
        <locate locator="Workflow Manager" />
      </action>
    </case>
    - <default name="No">
      <empty />
    </default>
  </switch>
</while>
</sequence>
- <process name="Derived Sub-Process 1">
  <Documentation>This is a nested process for the E-Mail Voting Process. This
  sub-process is generated because an all box was put in the middle of the
  process. Although there is a technical difference between an all and a
  sub-process, from the notation point of view, they will be the
  same.</Documentation>
  - <all>

```

```

- <sequence>
  <Documentation>The spawn within the foreach and the join following
  is the mechanism for handling a multi-instance situation. These
  are spawned since we are expecting to receive an email from each
  of the voting members. However, this multi-instance is likely to be
  interrupted before all the votes are in if all the issues receive a
  majority for a solution.</Documentation>
- <foreach select="range(1,NumVWGM)">
  <spawn ref="Derived Sub-Process 2" />
</foreach>
  <join ref="Derived Sub-Process 2" />
</sequence>
- <sequence>
  <delay type="duration:6 day" name="6 Day" />
  <action name="E-Mail Vote Deadline Warning"
    portType="tns:emailPort" operation="sendVoteWarning" />
</sequence>
- <context>
  - <onTimeout type="duration:7 Day">
    <call ref="Derived Sub-Process 3" />
  </onTimeout>
  - <onFault code="Voting Complete">
    <call ref="Derived Sub-Process 3" />
  </onFault>
</context>
</all>
</process>
- <process name="Derived Sub-Process 2">
  <Documentation>This is a nested process for the E-Mail Voting Process. This
  sub-process is generated through the mechanism of translating a multi-
  instance from the notation.</Documentation>
- <sequence>
  <action name="Receive E-Mail Response" portType="tns:emailPort"
    operation="ReceiveVotingResults" />
- <action name="Increment Tally" portType="tns:internalPort"
  operation="sendReceiveTotal">
  <locate locator="Workflow Manager" />
  <output select="AllItemsCompleted=True | False" />
  <output select="NoMajority=True | False" />
</action>
- <switch name="Results of Tally?">
  - <case name="Majority has voted 'Yes' or 'No' for all Issues"
    condition="AllItemsCompleted">
    <fault code="Voting Complete" />
  </case>
  - <default name="Majority Decision Not Yet Reached">
    <empty />
  </default>
</switch>
</sequence>
</process>
- <process name="Derived Sub-Process 3">
  <Documentation>This is a nested process for the E-Mail Voting Process. This

```

sub-process is generated since there are 2 ways that the following actions can be accessed through the timeout and fault handling mechanisms. Instead of duplicating the actions, a sub-process is generated and then called from each location that uses the sequence of actions.

```
</Documentation>
- <sequence>
  - <action name="Prepare Results" portType="tns:internalPort"
    operation="sendReceiveResults">
    <locate locator="Workflow Manager" />
  </action>
  <action name="E-Mail Results of Vote" portType="tns:emailPort"
    operation="sendVotingResults" />
</sequence>
</process>
</process>
```