**ASSIGNMENT 34.3**

1. **Oozie Action and Decision Nodes**

**Action nodes:**

Action nodes are the mechanism by which a workflow triggers the execution of a computation/processing task.

An action node represents a workflow task,

e.g., moving files into HDFS, running a MapReduce, Pig or[Hive](http://www.guru99.com/hive-tutorials.html)jobs, importing data using Sqoop or running a shell script of a program written in Java.

**Decision nodes:**

A decision node enables a workflow to make a selection on the execution path to follow.

The behavior of a decision node can be seen as a switch-case statement.

A decision node consists of a list of predicates-transition pairs plus a default transition. Predicates are evaluated in order or appearance until one of them evaluates to true and the corresponding transition is taken. If none of the predicates evaluates to true the default transition is taken.

**EXAMPLE:**

<workflow-app name="foo-wf" xmlns="uri:oozie:workflow:0.1">

...

<decision name="mydecision">

<switch>

<case to="reconsolidatejob">

${fs:fileSize(secondjobOutputDir) gt 10 \* GB}

</case> <case to="rexpandjob">

${fs:fileSize(secondjobOutputDir) lt 100 \* MB}

</case>

<case to="recomputejob">

${ hadoop:counters('secondjob')[RECORDS][REDUCE\_OUT] lt 1000000 }

</case>

<default to="end"/>

</switch>

</decision>

...

</workflow-app>

**2) Oozie Workflow Nodes:**

Workflow nodes are classified in control flow nodes and action nodes:

* **Control flow nodes:** nodes that control the start and end of the workflow and workflow job execution path.
* **Action nodes:** nodes that trigger the execution of a computation/processing task.

**Control Flow Nodes:**

Control flow nodes define the beginning and the end of a workflow (the start , end and kill nodes) and provide a mechanism to control the workflow execution path (the decision , fork and join nodes).

##### **Start Control Node**

The start node is the entry point for a workflow job, it indicates the first workflow node the workflow job must transition to.When a workflow is started, it automatically transitions to the node specified in the start .A workflow definition must have one start node.

##### **End Control Node**

The end node is the end for a workflow job, it indicates that the workflow job has completed successfully.When a workflow job reaches the end it finishes successfully (SUCCEEDED).If one or more actions started by the workflow job are executing when the end node is reached, the actions will be killed. In this scenario the workflow job is still considered as successfully run.A workflow definition must have one end node.

##### **Kill Control Node**

The kill node allows a workflow job to kill itself.When a workflow job reaches the kill it finishes in error (KILLED).If one or more actions started by the workflow job are executing when the kill node is reached, the actions will be killed.A workflow definition may have zero or more kill nodes.

##### **Decision Control Node**

A decision node enables a workflow to make a selection on the execution path to follow.The behavior of a decision node can be seen as a switch-case statement.A decision node consists of a list of predicates-transition pairs plus a default transition. Predicates are evaluated in order or appearance until one of them evaluates to true and the corresponding transition is taken. If none of the predicates evaluates to true the default transition is taken.

##### **Fork and Join Control Nodes**

A fork node splits one path of execution into multiple concurrent paths of execution.A join node waits until every concurrent execution path of a previous fork node arrives to it.The fork and join nodes must be used in pairs. The join node assumes concurrent execution paths are children of the same fork node.

* **Workflow Action Nodes**

Action nodes are the mechanism by which a workflow triggers the execution of a computation/processing task.

An action node represents a workflow task,

e.g., moving files into HDFS, running a MapReduce, Pig or[Hive](http://www.guru99.com/hive-tutorials.html)jobs, importing data using Sqoop or running a shell script of a program written in Java.

## 3) **Fork and Join Control Node in Workflow**

In scenarios where we want to run multiple jobs parallel to each other, we can use Fork. When fork is used we have to use Join as an end node to fork.

Basically Fork and Join work together. A fork node splits one path of execution into multiple concurrent paths of execution.

A join node waits until every concurrent execution path of a previous fork node arrives to it.For each fork there should be a join.

As Join assumes all the node are a child of a single fork.

We also use fork and join for running multiple independent jobs for proper utilization of cluster.

**4)** **OOZIE WEB CONSOLE:**

Oozie provides a read-only Web based console that allows to allow to monitor Oozie system status, workflow applications status and workflow jobs status.The Web base console is implemented as a client of the Web Services API.