**Example of Scenario Running**

This document provides a detailed example scenario to demonstrate the functionality of the Fuzzy Logic and Bayesian Network Integration Project. The scenario includes setting up input attributes for valves and pumps, processing these inputs through Fuzzy Logic, and updating the Bayesian Network with the results.

**Step 1: Set Up the Input Attributes**

First, create a HashMap to store the attributes of valves and pumps. These attributes will be used to evaluate the reliability of the nodes.

import java.util.HashMap;  
  
public class ExampleScenario {  
 public static void main(String[] args) {  
 // Setting up valve attributes  
 HashMap<String, Double> valveAttributes = new HashMap<>();  
 valveAttributes.put("valvePressure", 5.0);  
 valveAttributes.put("valveTemperature", 75.0);  
  
 // Setting up pump attributes  
 HashMap<String, Double> pumpAttributes = new HashMap<>();  
 pumpAttributes.put("pumpFlowRate", 10.0);  
 pumpAttributes.put("pumpEfficiency", 85.0);  
  
 // Process input through Fuzzy Logic  
 FuzzyLogicProcessor fuzzyProcessor = new FuzzyLogicProcessor();  
 boolean isNodeReliable = fuzzyProcessor.evaluateReliability(valveAttributes, pumpAttributes);  
  
 // Set the values in the Bayesian Network  
 BayesianNetworkProcessor bnProcessor = new BayesianNetworkProcessor();  
 bnProcessor.setNodeValues(isNodeReliable);  
  
 // Print the BN Inference Result  
 System.out.println("Bayesian Network Inference Result: " + bnProcessor.getInferenceResult());  
 }  
}

**Step 2: Process the Input through Fuzzy Logic**

Next, use the FuzzyLogicProcessor class to evaluate the reliability of the nodes based on the input attributes.

public class FuzzyLogicProcessor {  
 public boolean evaluateReliability(HashMap<String, Double> valveAttributes, HashMap<String, Double> pumpAttributes) {  
 // Load the Fuzzy Logic Controller from the FCL file  
 String fileName = "solution/final.fcl";  
 FIS fis = FIS.load(fileName, true);  
  
 if (fis == null) {  
 System.err.println("Cannot load file: " + fileName);  
 return false;  
 }  
  
 // Set inputs for the Fuzzy Logic system  
 fis.setVariable("valvePressure", valveAttributes.get("valvePressure"));  
 fis.setVariable("valveTemperature", valveAttributes.get("valveTemperature"));  
 fis.setVariable("pumpFlowRate", pumpAttributes.get("pumpFlowRate"));  
 fis.setVariable("pumpEfficiency", pumpAttributes.get("pumpEfficiency"));  
  
 // Evaluate the Fuzzy Logic system  
 fis.evaluate();  
  
 // Retrieve the output variable  
 double reliability = fis.getVariable("reliability").getValue();  
  
 // Determine if the node is reliable  
 return reliability > 0.5; // Assuming reliability > 0.5 means the node is reliable  
 }  
}

**Step 3: Set the Values in the Bayesian Network**

After evaluating the reliability using Fuzzy Logic, use the BayesianNetworkProcessor class to update the Bayesian Network with the results.

import unbbayes.io.NetIO;  
import unbbayes.prs.bn.ProbabilisticNetwork;  
import unbbayes.prs.bn.TreeVariable;  
  
public class BayesianNetworkProcessor {  
 private ProbabilisticNetwork bn;  
  
 public BayesianNetworkProcessor() {  
 try {  
 // Load the Bayesian Network from a file  
 NetIO netIO = new NetIO();  
 bn = (ProbabilisticNetwork) netIO.load("path/to/your/bn/file.net");  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 public void setNodeValues(boolean isNodeReliable) {  
 // Assuming the node is named "ReliabilityNode"  
 TreeVariable reliabilityNode = (TreeVariable) bn.getNode("ReliabilityNode");  
  
 // Set the probability values based on the Fuzzy Logic output  
 if (isNodeReliable) {  
 reliabilityNode.addFinding(0); // Reliable  
 } else {  
 reliabilityNode.addFinding(1); // Not reliable  
 }  
  
 // Propagate the findings in the Bayesian Network  
 bn.updateEvidences();  
 }  
  
 public String getInferenceResult() {  
 // Get the updated probabilities after propagation  
 TreeVariable reliabilityNode = (TreeVariable) bn.getNode("ReliabilityNode");  
 return reliabilityNode.getMarginalAt(0) + " (Reliable), " + reliabilityNode.getMarginalAt(1) + " (Not reliable)";  
 }  
}

**Step 4: Print the Bayesian Network Inference Result**

Finally, print the result of the Bayesian Network inference to verify the integration.

public class ExampleScenario {  
 public static void main(String[] args) {  
 // Setting up valve attributes  
 HashMap<String, Double> valveAttributes = new HashMap<>();  
 valveAttributes.put("valvePressure", 5.0);  
 valveAttributes.put("valveTemperature", 75.0);  
  
 // Setting up pump attributes  
 HashMap<String, Double> pumpAttributes = new HashMap<>();  
 pumpAttributes.put("pumpFlowRate", 10.0);  
 pumpAttributes.put("pumpEfficiency", 85.0);  
  
 // Process input through Fuzzy Logic  
 FuzzyLogicProcessor fuzzyProcessor = new FuzzyLogicProcessor();  
 boolean isNodeReliable = fuzzyProcessor.evaluateReliability(valveAttributes, pumpAttributes);  
  
 // Set the values in the Bayesian Network  
 BayesianNetworkProcessor bnProcessor = new BayesianNetworkProcessor();  
 bnProcessor.setNodeValues(isNodeReliable);  
  
 // Print the BN Inference Result  
 System.out.println("Bayesian Network Inference Result: " + bnProcessor.getInferenceResult());  
 }  
}

**Conclusion**

By following these steps, we successfully ran the example scenario to demonstrate the integration of Fuzzy Logic and Bayesian Networks. This scenario illustrates how input attributes are processed through Fuzzy Logic to determine node reliability and how the results are reflected in the Bayesian Network using the UNBBayes API.