

Assignment-2 Report

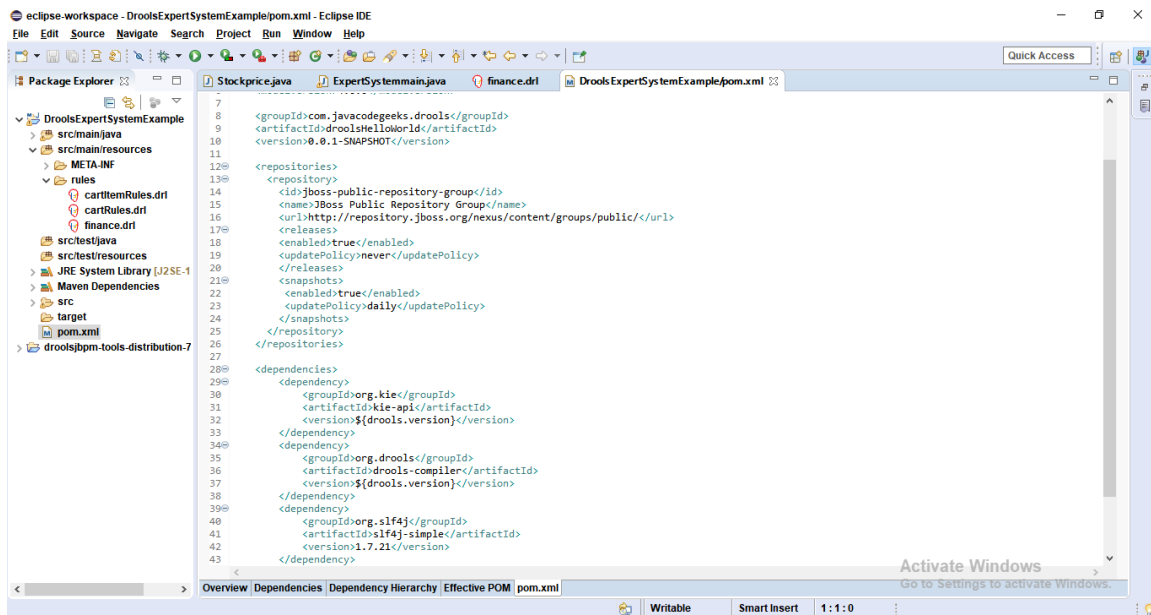
Expert System in Finance domain

Introduction

The requirement is to develop an Expert System in any of the domains such as finance, biology, chemistry etc. I have chosen Finance as the domain. Various data has been gathered from the internet regarding the stock market. Various finance parameters that impact the trading price of a share are gathered. All the data is expressed as rules using DROOLS in Eclipse. With the user entering the input data, the Expert system will execute a set of rules and gives the investing strategy.

Design and Implementation

Step-1: Check the pom.xml file where we keep all the DROOLS dependency here.



Drools dependency is under 'org.kie' group id

```

<dependency>
  <groupId>org.kie</groupId>
  <artifactId>kie-api</artifactId>
  <version>${drools.version}</version>
</dependency>

```

Step-2: Java main file is used to call the drools. Develop a main program that has main method. 'Kieservices' is used to set up the knowledge session of which the dependency is mentioned in 'pom' file as 'org.kie' as mentioned in step-1.

```

try {
    // load up the knowledge base
    KieServices ks = KieServices.Factory.get(); //Knowledge Services
    KieContainer kContainer = ks.getKieClasspathContainer(); //Knowledge container
    KieSession kSession = kContainer.newKieSession("ksession-rules"); //whole rule engine is embedded into kSession
}

```

Create the object of Kieservices, KieContainer and KieSession. Classpath is set using Kiecontainer and kSession is the place where whole rules section runs. The keyword "ksession-rules" comes from 'kmodule.xml' shown below

Node	Content
?? xml	version="1.0" encoding="UTF-8"
▼ [e] kmodule	
ⓐ xmlns	http://jboss.org/kie/6.0.0/kmodule
▼ [e] kbase	
ⓐ name	rules
ⓐ packages	rules
▼ [e] ksession	
ⓐ name	ksession-rules

Step -3: To get values of facts a method is called to getStockpricevalues()

```

Stockprice stockprice = getStockpriceValues();

```

The getStockpricevalues() method instantiates the object of Stockprices class and setting the values according to user input as below.

```

Stockprice stockprice = new Stockprice();

Scanner myObj = new Scanner(System.in);
System.out.println("Enter Dividend announced to the shareholders: ");
div = myObj.nextFloat();
stockprice.setDividend(div);

System.out.println("Rate the Global cues on a scale of 5, 5 being Extremely good for business to 1 being very poor: ");
glbl = myObj.nextInt();
stockprice.setGlobalcues(glbl);

System.out.println("Enter net revenue growth(YOY) in percentage: ");
//YOY denotes comparison of revenues of a particular quarter in the present year with the same quarter previous year
rev = myObj.nextFloat();
stockprice.setRevenuegrowth(rev);

System.out.println("Enter the net number of clients change in integers: ");
cli = myObj.nextInt();
stockprice.setNetclientschange(cli);

```

Step -4: Insert the fact into ksession using ‘insert’ method that takes object argument which is a fact for ‘finance.drl’ file. The next method fireAllRules() activates all rules in the drl file.

```

kSession.insert(stockprice);
kSession.fireAllRules();

```

Step-5: ‘Finance.drl’ file description. Below is the syntax of the rules. Write the package of the rules, import ‘Stockprice’ class that has to set and get the values accordingly.

rule – to declare name of the rule

when – condition like an if statement

then – part to be executed if when condition is met

end – rules end with ‘end’

The next rule follows the same syntax.

```

package com.javacodegeeks.drools;
import com.javacodegeeks.drools.Stockprice;

rule "Rule-1: All parameters together in positive trend"
when
    stockprice : Stockprice(dividend > 10 && globalcues >= 4 && revenuegrowth > 15 && netclientschange > 10 && taxbenefits >= 4 && newproduct:
then
    System.out.println("***** Rule-1 is fired *****");
    System.out.println("Stock price of the company is in very high bullish range. Very safe to invest on the company.");
and

```

Step-6: In the above image, under the when condition ‘Stockprice’ is the class on which the when condition gets executed.

The values inserting in the 'finance.drl' file are from 'Stockprice.java' file. We declare all the variables in the latter file and set all setters and getters for all the variables. These variables are then referred by drl file.

```
package com.javacodegeeks.drools;

public class Stockprice {
    public float dividend;
    public int globalcues;
    public float revenuegrowth;
    public int netclientschange;
    public float randd;
    public int taxbenefits;
    public int newproducts;

    public int getNewproducts() {
        return newproducts;
    }

    public void setNewproducts(int np) {
        this.newproducts = np;
    }

    public int getTaxbenefits() {
        return taxbenefits;
    }

    public void setTaxbenefits(int tax) {
        this.taxbenefits = tax;
    }
}
```

The above setter method is used by the getStockpricevalues() method in the main file to take user input and set values to the variables. These values are referred by drl file to execute the rules.

Test cases

Test Case -1:

Input values : Take the values to get the best output that shows 5/5 for investing.

dividend = \$12

global cues = 5(very good for business)

revenue growth year-over-year = 12%

net clients change = 12

Research and Development expenditure in percentage of revenue generated = 12%

Tax benefits from government = 5(very good)

Success of new products released = 5(High)

Expected Result: 5/5 Excellent time to invest on the company shares

Actual Result : 5/5 in majority of the rules

Enter Dividend announced to the shareholders:

12

Rate the Global cues on a scale of 5, 5 being Extremely good for business to 1 being very poor:

5

Enter net revenue growth(YOY) in percentage:

12

Enter the net number of clients change in integers:

12

Enter the percentage of revenue allotted to Research and Development:

12

Rate the expected Tax benefits from the government on a scale of 5, 5 being Extremely good to 1 being very poor:

5

Rate the success of new products released on a scale of 5, 5 being Extremely good to 1 being very poor:

5

|***** Rule-2 is fired *****

Stock price of the company is expected to be stable. Can invest safely!

|***** Rule-3 is fired *****

5/5 Excellent time to invest on the company shares

|***** Rule-4 is fired *****

5/5 Excellent time to invest on the company shares

|***** Rule-5 is fired *****

5/5 Excellent time to invest on the company shares

|***** Rule-9 is fired *****

4/5 Good time to invest on the company shares

|***** Rule-10 is fired *****

4/5 Good time to invest on the company shares

Test Case -2:

Input values : Take the values to get the best output that shows 5/5 for investing.

dividend = \$4

global cues = 3(Good for business)

revenue growth year-over-year = 2%

net clients change = -2

Research and Development expenditure in percentage of revenue generated = 5%

Tax benefits from government = 3(very good)

Success of new products released = 3(High)

Expected Result: 3/5 Invest with caution

Actual Result: 3/5 Invest with caution

```
Enter Dividend announced to the shareholders:
4
Rate the Global cues on a scale of 5, 5 being Extremely good for business to 1 being very poor:
3
Enter net revenue growth(YOY) in percentage:
2
Enter the net number of clients change in integers:
-2
Enter the percentage of revenue allotted to Research and Development:
5
Rate the expected Tax benefits from the government on a scale of 5, 5 being Extremely good to 1 being very poor:
3
Rate the success of new products released on a scale of 5, 5 being Extremely good to 1 being very poor:
3
|***** Rule-12 is fired *****
3/5 Invest with caution
***** Rule-14 is fired *****
3/5 Invest with caution
***** Rule-18 is fired *****
3/5 Invest with caution
***** Rule-19 is fired *****
3/5 Invest with caution
```

Test Case -3:

Input values : Take the values to get the best output that shows 5/5 for investing.

dividend = \$0

global cues = 1(Good for business)

revenue growth year-over-year = -10%

net clients change = 4

Research and Development expenditure in percentage of revenue generated = 0%

Tax benefits from government = 2(very good)

Success of new products released = 2(High)

Expected Result: 1/5 Stay completely away from investing

Actual Result: 1/5 Stay completely away from investing

```

Enter Dividend announced to the shareholders:
0
Rate the Global cues on a scale of 5, 5 being Extremely good for business to 1 being very poor:
1
Enter net revenue growth(YOY) in percentage:
-10
Enter the net number of clients change in integers:
4
Enter the percentage of revenue allotted to Research and Development:
0
Rate the expected Tax benefits from the government on a scale of 5, 5 being Extremely good to 1 being very poor:
2
Rate the success of new products released on a scale of 5, 5 being Extremely good to 1 being very poor:
2
|***** Rule-22 is fired *****
2/5 Do not invest
***** Rule-26 is fired *****
1/5 Stay completely away from investing
***** Rule-28 is fired *****
1/5 Stay completely away from investing
***** Rule-29 is fired *****
1/5 Stay completely away from investing
***** Rule-30 is fired *****
1/5 Stay completely away from investing

```

Summary

An expert system that suggests the investing ideas is developed. Have used DROOLS for writing the rules, a main program namely 'ExpertSystemmain.java' and a class 'Stockprice.java' and drools file 'finance.drl'. The main program loads up the knowledge base using the inbuilt 'KieServices'. An object is created for the 'Stockprice' class and inserted into 'ksession' using which all rules are fired. A 'getStockpriceValues()' method is created to take the user input and set the values in the 'Stockprice.java' file. These values are then referred by the 'finance.drl' file to check the 'when' conditions and fire the rules.

References

1. <http://www.mastertheboss.com/jboss-jbpm/drools/jboss-drools-tutorial>
2. <http://a4academics.com/tutorials/58-drools-tutorial-and-example>
3. <https://dzone.com/articles/introduction-to-rules-engine-using-drools-expert>
4. <https://docs.jboss.org/drools/release/5.2.0.M2/drools-expert-docs/html/ch05.html>
5. <https://www.baeldung.com/drools>
6. <http://a4academics.com/tutorials/58-drools-tutorial-and-example/505-drools-example-creating-knowledgebase-and-knowledgesession>
7. <https://www.codota.com/code/java/methods/org.kie.api.KieServices/getKieClasspathContainer>
8. <https://www.programcreek.com/java-api-examples/?api=org.kie.api.KieServices>
9. https://www.javainuse.com/drools_hello
10. <https://docs.jboss.org/jbpm/v6.2/javadocs/org/kie/api/KieServices.Factory.html>