Learn Drools:Part IV-(Inference)

As per dictionary, Inference means **An assumption or conclusion that is rationally and logically made, based on the given facts or circumstances**.

An inference is based off of facts, so the reasoning for the conclusion is often logical.

**Example** : According to the technical interview we can make an assumption a candidate is good,bad or moderate. The thing to notice here we don’t know How good he is in coding as we do not work with him/her but according to his/her answer(Facts) we can logically judge and come to a conclusion.

**So what is the relation between Inference and Drools?**

Let try to understand it by a problem statement.

**We want to give a laptop to those people who are in IT department and designation is the manager?**

When we got this problem statement we will try to solve it by a drools rule , in when part we put the logic ,**If *depertment==”IT” and isManager=true*** and in **then** part we print **give Laptop**

As simple as that.

Solution:

rule "give Laptop"

when

$dept: Department(name=="IT");

$emp: Employee(dept == $dept,manager==true);

then

$emp.setMessage("Give Laptop");

System.out.println($emp.getName()+ ": "+$emp.getDept().getName()+ ":"+$emp.getMessage());

end

**Is this an elegant way to solve this problem?**

The answer depends, If this IT Manager condition check (***depertment==”IT” and isManager=true***) is used in several rules then it is not an elegant way to implement the same we can do it better and it is the potential area where we can use Inference

On other hand if this condition used only one or in few rules and no future changes then make it simple, above rule is just fine as I am a big follower of YAGNI(You aren't gonna need it).

**Why above is not a judicious way to define the rules where the same condition is used in several rules?**

Let say IT manager condition is used in 100 different rules , and based on that condition we will take different actions. So as far the condition is not changed our design is good. But think about if the condition requirements are changed now the client wants ***(depertment==”IT” and isManager=true and experience>15)*** , whose experience is greater than 15 can perform aforesaid activities.

So we need to change the condition in 100 rules , a simple change is then hard to implement because of the condition has been duplicated over 100 rules.

It would be nice if we design in a such a way, we can define the condition in one place and use that reference in Other places like Soft link in Unix or Environment variables in Windows.

By doing that , we can use the reference to the condition in every rule , like we set

**JAVA\_HOME=”C;//java”**; and use JAVA\_HOME anywhere to declare java path so in case if we change the java path to c://javaone that will reflect in everywhere.

So, We can do the same by Inference in Drools.

Let see How to do it step by step

Step 1:

Create a rule where we define the condition/when part and create a reference to that in then part, so we can use this reference in other rules.

rule "IT Manager Inference"

when

$dept: Department(name=="IT");

$emp: Employee(dept == $dept,manager==true);

then

insert(new ITManager($emp));

end

Please note the then part of the rule here we insert a Java Object called ITManager which takes the employee reference. By this statement we instruct drools to insert an ITManager Object into KnowledgeFactory which holds the When parts condition, It acts as a reference, like JAVA\_HOME key.

Step 2: Create Other rules where we use this IT Manager reference

rule "give Manager Laptop"

when

$emp: Employee();

$itManager: ITManager(emp == $emp);

then

$emp.setMessage("Give Laptop");

System.out.println($emp.getName()+ ": "+$emp.getDept().getName()+ ":"+$emp.getMessage());

end

Pay attention to the When part here we check current employee is matching with ITManager condition ,

Here drools replace ITManger with it actual when condition stated in ***IT Manager Inference*** rule.

Step 3: Complete Drools Rule File

package com.rules

import com.example.droolsExample.pojo.Employee

import com.example.droolsExample.pojo.Department

import com.example.droolsExample.pojo.ITManager

rule "IT Manager Inference"

when

$dept: Department(name=="IT");

$emp: Employee(dept == $dept,manager==true);

then

insert(new ITManager($emp));

end

rule "give Manager Laptop"

when

$emp: Employee();

$itManager: ITManager(emp == $emp);

then

$emp.setMessage("Give Laptop");

System.out.println($emp.getName()+ ": "+$emp.getDept().getName()+ ":"+$emp.getMessage());

end

By doing so If any changes happen in the Condition we just change the IT Manager Inference when the condition and it will be automatically reflected in ***give Manager Laptop*** rule.

Step 4: Java Objects

**Employee.java**

package com.example.droolsExample.pojo;

public class Employee {

String name;

boolean manager;

String message;

Department dept;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public boolean isManager() {

return manager;

}

public void setManager(boolean manager) {

this.manager = manager;

}

public String getMessage() {

return message;

}

public void setMessage(String message) {

this.message = message;

}

public Department getDept() {

return dept;

}

public void setDept(Department dept) {

this.dept = dept;

}

}

**ITManager.java**

package com.example.droolsExample.pojo;

public class ITManager {

private Employee emp;

public ITManager(Employee emp)

{

this.emp=emp;

}

public Employee getEmp() {

return emp;

}

public void setEmp(Employee emp) {

this.emp = emp;

}

}

**Department.java**

package com.example.droolsExample.pojo;

public class Department {

String name;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

**DroolTest.java**

package com.example.droolsExample;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.io.Reader;

import org.drools.compiler.compiler.DroolsParserException;

import org.drools.compiler.compiler.PackageBuilder;

import org.drools.core.RuleBase;

import org.drools.core.RuleBaseFactory;

import org.drools.core.WorkingMemory;

import com.example.droolsExample.pojo.Department;

import com.example.droolsExample.pojo.Employee;

public class DroolsTest {

public static void main(String[] args) throws DroolsParserException,

IOException {

DroolsTest droolsTest = new DroolsTest();

droolsTest.executeDroolsEmployee("/com/rules/employeeInfernce.drl");

}

public void executeDroolsEmployee(String ruleFile) throws DroolsParserException, IOException {

PackageBuilder packageBuilder = new PackageBuilder();

InputStream resourceAsStream = getClass().getResourceAsStream(ruleFile);

Reader reader = new InputStreamReader(resourceAsStream);

packageBuilder.addPackageFromDrl(reader);

org.drools.core.rule.Package rulesPackage = packageBuilder.getPackage();

RuleBase ruleBase = RuleBaseFactory.newRuleBase();

ruleBase.addPackage(rulesPackage);

WorkingMemory workingMemory = ruleBase.newStatefulSession();

Department dep = new Department();

dep.setName("Civil");

Department dep1 = new Department();

dep1.setName("IT");

Employee emp = new Employee();

emp.setName("Shamik Mitra");

emp.setManager(true);

emp.setDept(dep1);

Employee emp1 = new Employee();

emp1.setName("Samir Mitra");

emp1.setManager(true);

emp1.setDept(dep);

workingMemory.insert(dep);

workingMemory.insert(dep1);

workingMemory.insert(emp);

workingMemory.insert(emp1);

workingMemory.fireAllRules();

}

}

Step 5: **Output**

Shamik Mitra: IT:Give Laptop