Decision Management Community's Challenge Dec-2024 "Pricing Policy"

OpenRules-based Solution

This challenge is described at https://dmcommunity.org/challenge/challenge-dec-2024/:

Challenge Dec-2024

Pricing Policy Solutions

A company sells Standard and Premium products manufactured in different countries. It uses a mixed bracket volume and cost policy for orders with multiple units:

Cost Accounting

- **1-99 units:** $10 per unit (Standard), $15 per unit (Premium)
- **100-499 units:** $9 per unit (Standard), $14 per unit (Premium)
- **500+ units:** $8 per unit (Standard), $13 per unit (Premium)

For products that originated not in the US or Canada, there is a handling fee of $50 per order. If the order includes hazardous material, the total price should be increased by 30%.

Here is an example of the order:

- 200 Premium products originated in Germany with hazardous materials
- 50 Standard products originated in the US without hazardous materials
- 200 Premium products originated in the Canada without hazardous materials
- 150 Standard products originated in China with hazardous materials.

Use your favorite tool to create a decision model capable of addressing this or more complex sales policies. Send your solutions to DecisionManagementCommunity@gmail.com.

This challenge already has two solutions created with ChatGPT. The first one produced the incorrect total cost of these 4 orders:

Total cost = $3650 + $500 + $2800 + $1810 = **$9760**

The second Python-based solution gave a total cost of $8,825 which we assumed is correct:

200 x 14 + 50 handling fee + 30% hazardous = $3705.00
50 x 10 + 0 handling fee = $500.00
200 x 14 + 0 handling fee = $2800.00
150 x 9 + 50 handling fee + 30% hazardous = $1820.00
Total order price: $8825.00

OpenRules Decision Model

As usual with OpenRules, we start with the preparation of the proper test data:

DecisionData Order orders			
Product Type	**Number of Units**	**Origin Country**	**Hazardous Material**
Premium	200	Germany	Yes
Standard	50	US	No
Premium	200	Canada	No
Standard	150	China	Yes

Here is one test case with the expected Total Cost of $8,825:

DecisionTest testCases		
#	ActionDefine	ActionExpect
Test	**Orders**	**Total Cost**
A	orders	$8,825.00

Here is the glossary for our decision model:

Glossary glossary			
Variable Name	**Business Concept**	**Attribute**	**Type**
Product Type	Order	productType	String
Number of Units		numberOfUnits	int
Origin Country		originCountry	String
Hazardous Material		hazardousMaterial	boolean
Handling Fee		handlingFee	double
Surcharge		surcharge	double
Unit Cost		unitCost	double
Orders	InputOutput	orders	Order[]
Total Cost		totalCost	double

To calculate the Total Cost, we need to iterate over all Orders using the following table:

Decision DetermineTotalCost [for Order in Orders]
ActionExecute
Rules
CalculateHandlingFee
CalculateSurcharge
AddUnitCost

For each Order of our 4 test Orders, it will execute the following decision tables:

Decision CalculateHandlingFee		
Condition		Action
Origin Country		**Handling Fee**
		$50
Is One Of	**US, Canada**	0

DecisionTable CalculateSurcharge	
Condition	Action
Hazardous Material	**Surcharge**
TRUE	1.3
FALSE	1

DecisionTable AddUnitCost					
Condition	Condition	Action	Action		Message
Number of Units	**Product Type**	**Unit Cost**	**Total Cost**		**Error**
1..99	**Standard**	$10	+	(Number of Units * Unit Cost + Handling Fee) * Surcharge	
1..99	**Premium**	$15			
100..499	**Standard**	$9			
100..499	**Premium**	$14			
500+	**Standard**	$8			
500+	**Premium**	$13			
					INVALID DATA

The last table calculates Unit Cost that is being used to calculate the cost of each order:

(Number of Units * Unit Cost + Handling Fee) * Surcharge

Then it adds the result to the Total Cost. If the input data does not have the correct combination of Number of Units and Product Type, the message “INVALID DATA” will be produced.

When we executed this decision model, we received the expected Total Cost = **$8,825.00**.

With one click on “deployLambda.bat” we deployed this decision model as AWS Lambda function. Then we tested it using POSTMAN: