**=== Decision Management Community ===**

# Challenge Sep 2024 "Rental Boats"

### Solution with OpenRules Rule Solver

by Ian Detinich, iandetinich@gmail.com

## Problem Statement

The problem was defined at https://dmcommunity.org/challenge-sep-2024/:

## Challenge Sep-2024

**Rental Boats** Solutions


Floataway tours has $420,000 that may be used to purchase new rental boats for hire during the summer. The boats can be purchased from two different manufacturers. Floataway tours would like to purchase at least 50 boats and would like to purchase the same number from Sleekboat as from Racer to maintain goodwill. Also, Floataway Tours wishes to have a total capacity of at least 200. Data about the boats is summarized below:

| Boat | Manufacturer | Cost | Seating | Expected Daily Profit |
|---|---|---|---|---|
| Speedhawk | Sleekboat | $6,000 | 3 | $70 |
| Silverbird | Sleekboat | $7,000 | 5 | $80 |
| Catman | Racer | $5,000 | 2 | $50 |
| Classy | Racer | $9,000 | 6 | $110 |

Your task is to create a decision model that will help Floataway to purchase boats while satisfying the above requirements and maximizing profit. Send your solutions to DecisionManagementCommunity@gmail.com.

**Problem Definition**

I used OpenRules Rule Solver to define and solve this problem in one Excel file "RentalBoats.xls". I started with a Glossary:

| Glossary glossary | | | |
|---|---|---|---|
| **Variables** | **Business Concept** | **Attribute** | **Type** |
| **Total Amount to Spend** | Rental | totalAmountToSpend | int |
| **Boats** | | boats | Boat[] |
| **Min Number Of Boats** | | minNumberOfBoats | int |
| **Boat Types** | | boatTypes | String[] |
| **Manufacturers** | | manufacturers | String[] |
| **Boat Costs** | | boatCost | int[] |
| **Boat Capacities** | | boatCapacities | int[] |
| **Min Total Capacity** | | minTotalCapacity | int |
| **Expected Daily Profits** | | expectedDailyProfits | int[] |
| **Total Cost** | | totalCost | int |
| **Total Profit** | | totalProfit | int |
| **Type** | Boat | type | String |
| **Manufacturer** | | manufacturer | String |
| **Cost** | | cost | int |
| **Capacity** | | capacity | int |
| **Expected Daily Profit** | | expectedDailyProfit | int |
| **Smallest Boat Cost** | Temp | smallestBoatCost | int |
| **Max Needed Boats** | | macNeededBoats | int |

Here I introduced the business concept "Rental" which refers to our input data such as "Total Amount to Spend", "Min Number Of Boats", and other decision variables in blue. The output variables (in red) are "Total Cost" and "Total Profit".

Having this Glossary defined I create a test case from the Challenge in the table "DecisionTest":

| **DecisionTest testCases** | | | | |
|---|---|---|---|---|
| # | ActionDefine | ActionDefine | ActionDefine | ActionDefine |
| **Test ID** | **Total Amount to Spend** | **Min Number Of Boats** | **Min Total Capacity** | **Boats** |
| Rental Boats | $420,000 | 50 | 200 | boats |

All boats are defined in a separate table "DecisionData":

| DecisionData Boat boats | | | | |
|---|---|---|---|---|
| Type | Manufacturer | Cost | Capacity | Expected Daily Profit |
| Speedhawk | Sleekboat | $6,000 | 3 | $70 |
| Silverbird | Sleekboat | $7,000 | 5 | $80 |
| Catman | Racer | $5,000 | 2 | $50 |
| Classy | Racer | $9,000 | 6 | $110 |

**Defining Decision Variables**

For this problem, key constrained (unknown) variables represent how many boats we want to purchase. Their values vary from 0 to "Max Needed Boats". To express problem's constraints I needed also to calculate several more decision variables in the following decision table:

| Decision CalculateBusinessVariables | |
|---|---|
| ActionAssign | |
| Variable | Value |
| **Boat Types** | Array of Type of Boats |
| **Manufacturers** | DistinctArray of Manufacturer of Boats |
| **Boat Costs** | Array of Cost of Boats |
| **Boat Capacities** | Array of Capacity of Boats |
| **Expected Daily Profits** | Array of Expected Daily Profit of Boats |
| **Smallest Boat Cost** | Min of Cost of Boats |
| **Max Needed Boats** | Total Amount to Spend / Smallest Boat Cost + 1 |

Having this variables calculated allowed me to define our unknown variables using the Rule Solver's column "SolverDefineVariables":

| Decision DefineVariables | | | | |
|---|---|---|---|---|
| SolverDefineVariables | | | | |
| Variable Name | Method Name | Par 1 | Par 2 | Par 3 |
| "Boat Variables" | "New Variables" | "Boat Types" | "0" | "Max Needed Boats" |
| "Total Number Of Boats" | "Sum" | "Boat Variables" | | |
| "Total Capacity" | "Scalar Product" | "Boat Variables" | "Boat Capacities" | |
| "Total Cost" | "Scalar Product" | "Boat Variables" | "Boat Costs" | |
| "Total Profit" | "Scalar Product" | "Boat Variables" | "Expected Daily Profits" | |

Then for each boat I needed to define “Sleekboat Variables” and “Racer Variables”:

| Decision **DefineSleekboatAndRacerVariables** [for each Boat in Boats] | | | |
|---|---|---|---|
| Condition | **SolverDefineVariables** | | |
| **Manufacturer** | **Variable Name** | **Method** | **Par 1** |
| Sleekboat | "Sleekboat Variables" | "Add Variable" | Type |
| Racer | "Racer Variables" | "Add Variable" | Type |

To express the “goodwill” constraint I needed to know their sums:

| Decision DefineSums | | |
|---|---|---|
| **SolverDefineVariables** | | |
| **Variable Name** | **Method Name** | **Par 1** |
| "Sum of Sleekboats" | "Sum" | "Sleekboat Variables" |
| "Sum of Racers" | "Sum" | "Racer Variables" |

Now I was ready to post all problem constraints in this table:

| Decision PostConstraints | | | | |
|---|---|---|---|---|
| **SolverPostConstraints** | | | | |
| **Constraint Name** | **Constraint Type** | **Par 1** | **Par 2** | **Par 3** |
| "Limit Total Number Of Boats" | Variable Operator Value | "Total Number Of Boats" | ">=" | "Min Number Of Boats" |
| "Limit Total Capacity" | Variable Operator Value | "Total Capacity" | ">=" | "Min Total Capacity" |
| "Limit Total Cost" | Variable Operator Value | "Total Cost" | "<=" | "Total Amount to Spend" |
| "Sleekboats=Racers" | Variable Operator Variable | "Sum of Sleekboats" | "=" | "Sum of Racers" |

And finally, I added all the above decision tables to the standard Rule Solver’s table “Define”:

| **Decision Define** |
|---|
| ActionExecute |
| **Decision Tables** |
| CalculateBusinessVariables |
| DefineVariables |
| DefineSleekboatAndRacerVariables |
| DefineSums |
| PostConstraints |

This completed my Problem Definition.

**Problem Resolution**

To solve the problem, I needed to define the objective as “Total Profit” and use the standard table “Solve”. In did it using the following tables:

| Decision SetObjective |
|---|
| SolverSetObjective |
| **Objective** |
| "Total Profit" |

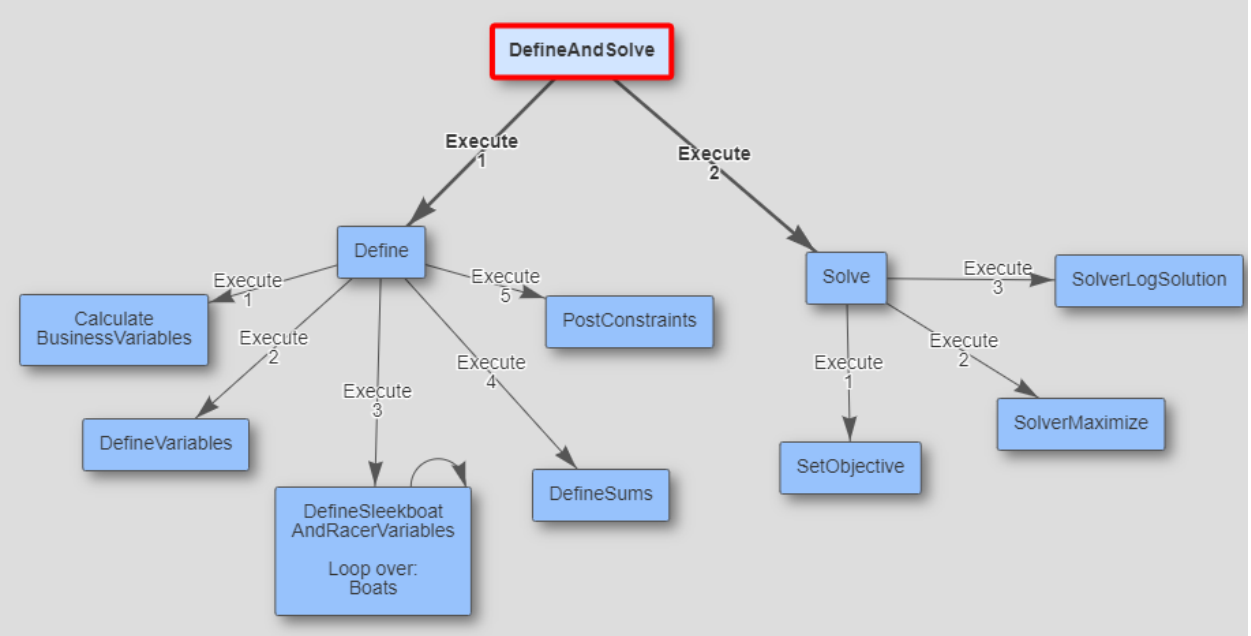
| Decision Solve |
|---|
| ActionExecute |
| **Actions** |
| SetObjective |
| SolverMaximize |
| SolverLogSolution |

**Execution Results**

Then I simply executed this decision model against our test data using the standard file “test.bat”. It produced the expected results:

Found solution #1 with objective -4750. Mon Sep 09 11:38:32 EDT 2024
Found solution #2 with objective -4940. Mon Sep 09 11:38:32 EDT 2024
Found solution #3 with objective -4950. Mon Sep 09 11:38:32 EDT 2024
Found solution #4 with objective -4970. Mon Sep 09 11:38:32 EDT 2024
Found solution #5 with objective -4990. Mon Sep 09 11:38:32 EDT 2024
Found solution #6 with objective -5010. Mon Sep 09 11:38:32 EDT 2024
Found solution #7 with objective -5020. Mon Sep 09 11:38:32 EDT 2024
Found solution #8 with objective -5040. Mon Sep 09 11:38:32 EDT 2024
Optimal solution is found. Objective: **5040**
Solution #8:
**Speedhawk[28]** Silverbird[0] Catman[0] **Classy[28]** Total Number Of Boats[56] Total Capacity[252] **Total Cost[420000] Total Profit[5040]** Sum of Sleekboats[28] Sum of Racers[28]
*** Execution Profile ***
Number of Choice Points: 2771
Number of Failures: 2754
Execution time: 176 msec

I opened my decision model using OpenRules IDE. It automatically built a graphical diagram of my decision model:




I was able to use OpenRules Debugger to execute all the rules of my decision model one by one and analyze the values of all decision variables.

**Deploying the Decision Model as a RESTful Decision Service**

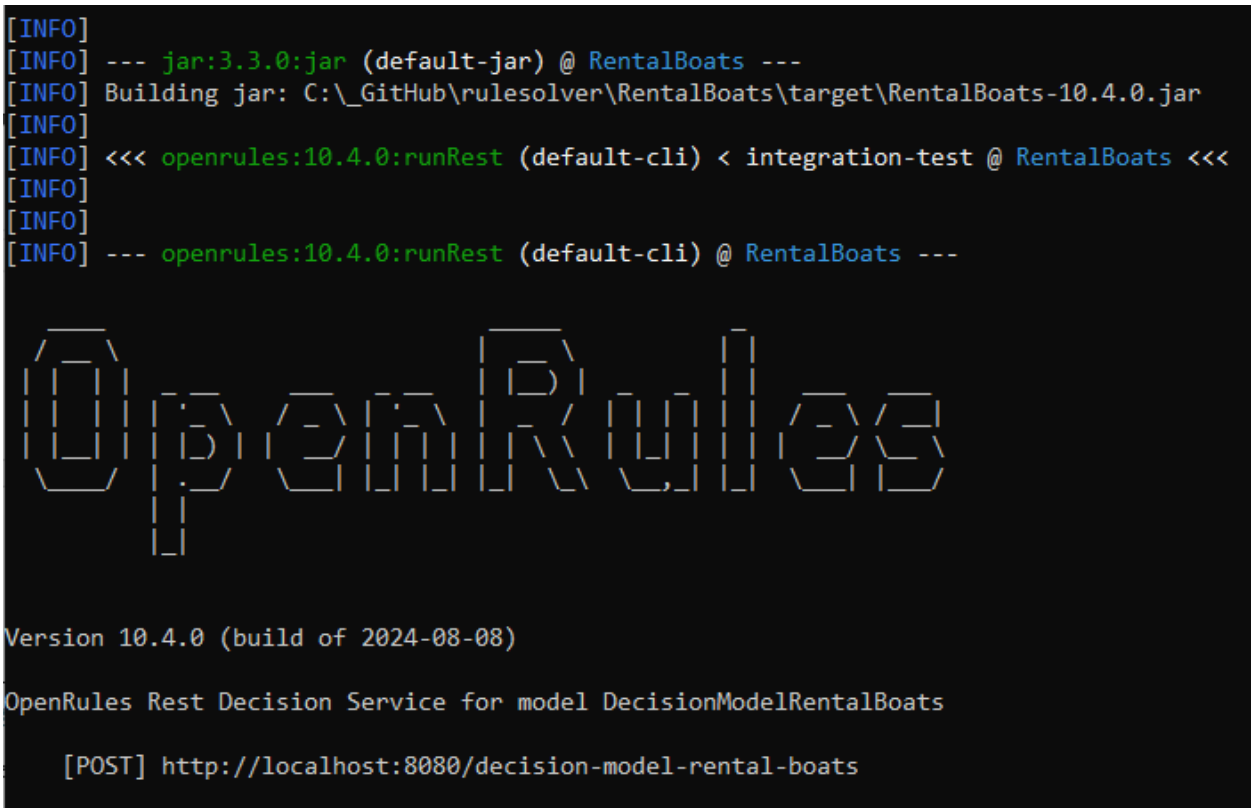
I tried to deploy my decision model as a RESTful decision service available via JSON interface already generated when I ran my model:


```
{
  "rental" : {
    "totalAmountToSpend" : 420000,
    "boats" : [ {
      "type" : "Speedhawk",
      "manufacturer" : "Sleekboat",
      "cost" : 6000,
      "capacity" : 3,
      "expectedDailyProfit" : 70
    }, {
      "type" : "Silverbird",
      "manufacturer" : "Sleekboat",
      "cost" : 7000,
      "capacity" : 5,
      "expectedDailyProfit" : 80
    }, {
      "type" : "Catman",
      "manufacturer" : "Racer",
      "cost" : 5000,
      "capacity" : 2,
      "expectedDailyProfit" : 50
    }, {
      "type" : "Classy",
      "manufacturer" : "Racer",
      "cost" : 9000,
      "capacity" : 6,
      "expectedDailyProfit" : 110
    } ],
    "minNumberOfBoats" : 50,
    "minTotalCapacity" : 200,
    "totalCost" : 0,
    "totalProfit" : 0
  }
}
```


I added the deployment property “rest” to the default OpenRules configuration file “project.properties”:

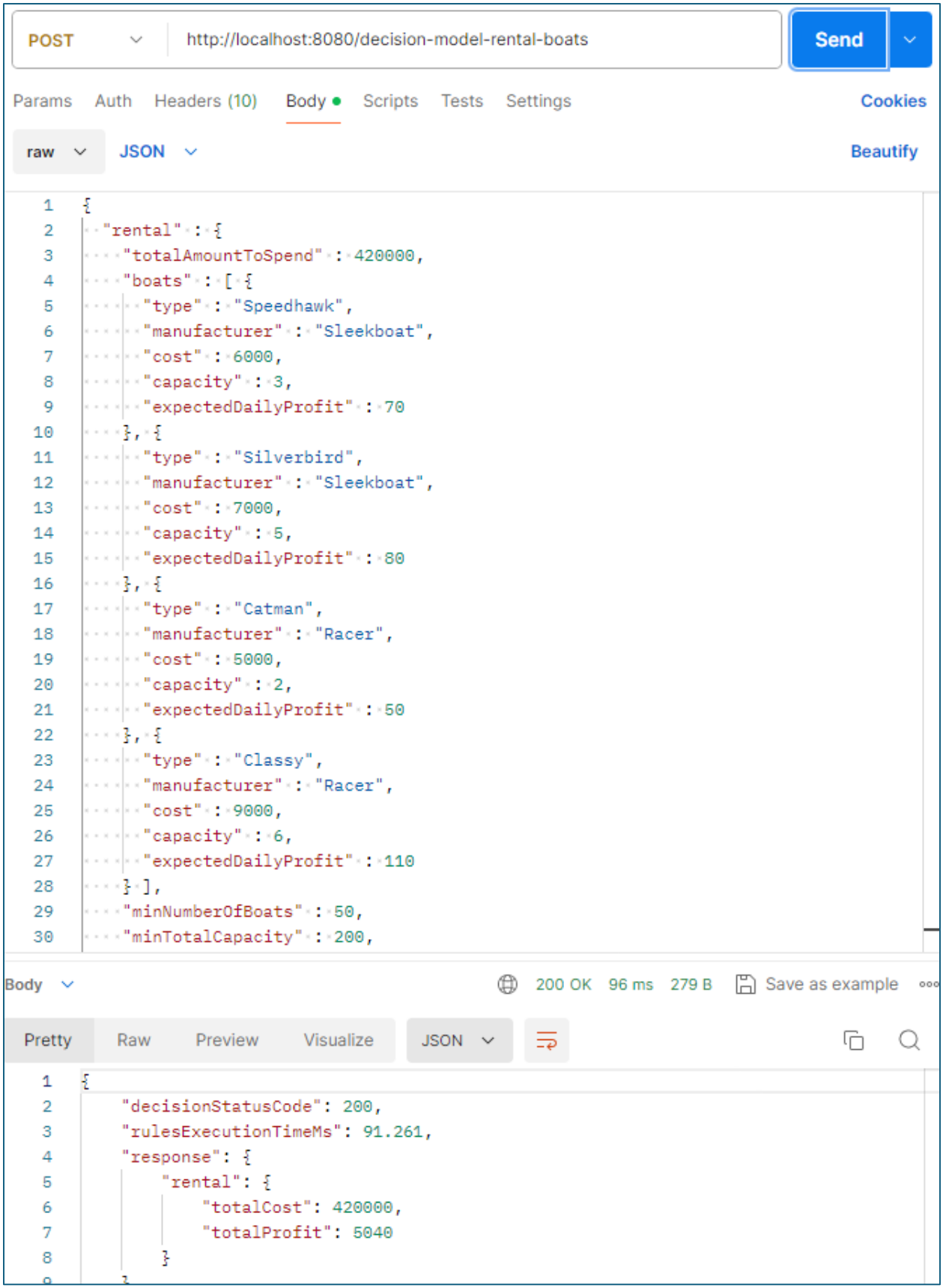
```
model.file="rules/RentalBoats.xls"
report=On
trace=On
deployment=rest
```

And executed the standard OpenRules file “RunLocalServer.bat”. It produced:




Then I ran POSTMAN using the above URL http://localhost:8080/decision-model-rental-boats. As a result, I got a decision service accessible through a standard REST interface. I used POSTMAN to test it. This service solved the problem with a total profit of 5040 and a total cost of 420000. It took only 91 milliseconds from start to finish.

Below is my POSTMAN view:


POST
http://localhost:8080/decision-model-rental-boats
Send
Params
Auth
Headers (10)
Body
Scripts
Tests
Settings
Cookies
raw
JSON
Beautify
{
"rental" : {
"totalAmountToSpend" : 420000,
"boats" : [ {
"type" : "Speedhawk",
"manufacturer" : "Sleekboat",
"cost" : 6000,
"capacity" : 3,
"expectedDailyProfit" : 70
}, {
"type" : "Silverbird",
"manufacturer" : "Sleekboat",
"cost" : 7000,
"capacity" : 5,
"expectedDailyProfit" : 80
}, {
"type" : "Catman",
"manufacturer" : "Racer",
"cost" : 5000,
"capacity" : 2,
"expectedDailyProfit" : 50
}, {
"type" : "Classy",
"manufacturer" : "Racer",
"cost" : 9000,
"capacity" : 6,
"expectedDailyProfit" : 110
} ],
"minNumberOfBoats" : 50,
"minTotalCapacity" : 200,
Body
200 OK
96 ms
279 B
Save as example
Pretty
Raw
Preview
Visualize
JSON
{
"decisionStatusCode": 200,
"rulesExecutionTimeMs": 91.261,
"response": {
"rental": {
"totalCost": 420000,
"totalProfit": 5040
}
}