Vacation Days - Decision Modeling Challenge

Solution by Edson Tirelli

Table of Contents

Introduction

This is a solution to the DMN Community challenge from January/2016. This solution is strictly based on the **DMN specification**, compliance level 3. It can be executed using the **Drools open source engine (www.drools.org)**.

Problem statement

The number of vacation days depends on age and years of service.

Every employee receives at least 22 days.
Additional days are provided according to the following criteria:

1) Only employees younger than 18 or at least 60 years, or employees
   with at least 30 years of service will receive 5 extra days.

2) Employees with at least 30 years of service and also employees of age 60 or more, receive 3
   extra days, on top of possible additional days already given.

3) If an employee has at least 15 but less than 30 years of service, 2 extra days
   are given. These 2 days are also provided for employees of age 45 or more.
   These 2 extra days can not be combined with the 5 extra days.

Solution

Following the DMN standard, the high-level solution is modelled in a DRD (Decision Requirements
Diagram) that is presented below. Each node of the diagram is then explained in the following pages.
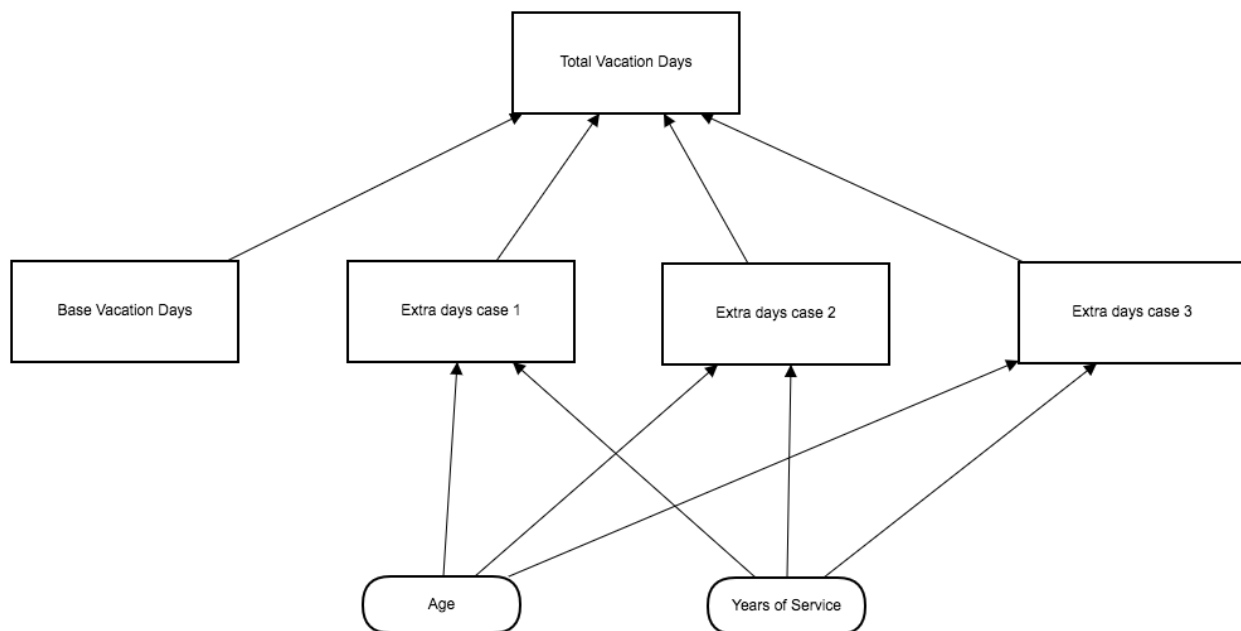
Input Nodes

This problem statement defines two inputs: the age and the number of years of service of the employee.

Base Vacation Days

The base vacation days is modelled as a decision in the model.

Base Vacation Days
22

Extra Days

The problem statement presents 3 different rules/cases for extra vacation days. Although the cases can be combined into a single expression or single decision table, in order to improve the maintainability of the solution, a separate decision table was defined for each case. They are presented here:

Extra days case 1

C>	Age	Years of Service	Extra days
			0
1	<18, >=60	–	5
2	–	>=30	5

Extra days case 2

C>	Age	Years of Service	Extra days
			0
1	–	>=30	3
2	>=60	–	3

Extra days case 3

C>	Age	Years of Service	Extra days
			0
1	–	[15..30)	2
2	>=45	–	2

It is important to note that for simple decision tables like this, there are several hit policies that could be used with the same result (for instance, "Any", "Priority", etc). The choice here to use "Collect Max" was

made to facilitate future changes. In the future, if any of the rules change to allow for different number of vacation days, this hit policy will likely attend, without the need for more extensive changes. It also demonstrates the flexibility of the DMN specification in cases like this.

Total Vacation Days

Calculating the total vacation days is then a trivial summation of all the component decisions, with a single caveat: the problem statement indicates that the extra vacation days from cases 1 and 3 are not cumulative. This can be easily implemented using the "max()" function to only consider the higher number of extra vacation days between the two cases.

Total Vacation Days

```
Base Vacation Days +
max( Extra days case 1, Extra days case 3 ) +
Extra days case 2
```