Decision Table for Vacation Days Calculation

By Maarten P.D. Schadd
Product Consultant at Blueriq B.V.

Contents

1 Introduction 3

2 Problem definition 3

3 Decision Tables 3

3.1 A simple decision table 3

3.2 A complete decision table 4

3.3 A decision table that handles unknowns 4

3.4 Only deriving additional days 5

3.5 Default value of 30 5

3.6 A decision table with complex conditions 6

3.7 The smallest decision table 6

4 Conclusions 6

1 Introduction

The decision management community [1] is a new initiative that started in 2014 to facilitate the sharing of news and knowledge concerning Decision Management (DM). Next to a product catalog, decision model prototypes and case studies, the decision management community also provides a monthly challenge. Every challenge consists of a problem that should be solved using any business rules and decisions management system or none at all.

As Blueriq is a vendor with an integrated rule engine and decision management capabilities in its BPM suite, we accept this challenge. This article describes how Blueriq solves the January 2016 challenge.

2 Problem definition

This challenge concerns the design of a decision table for logic that was offered by Prof. Jan Vanthienen. The problem statement reads as follows:

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.

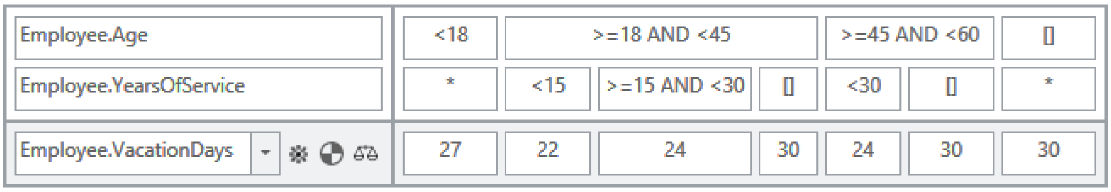
1. Only employees younger than 18 or at least 60 years, or employees with at least 30 years of service will receive 5 extra days.
2. Employees with at least 30 years of service and also employees of age 60 or more, receive 3 extra days, on top of possible additional days already given.
3. If an employee 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.

3 Decision Tables

Blueriq does not prescribe one manner in which the Business Engineer has to create decision tables. It has many options out of which the Business Engineer can choose the most convenient solution for the problem at hand, taking into account personal taste as well as standards that are sustained on each project. Therefore we are proposing multiple decision tables and discuss advantages and disadvantages for each table in the following sections. Each of these decision tables are single-hit decision tables, as Blueriq does currently not support multiple-hit decision tables. All depicted decision tables are screenshots of working decision tables from within the modeling environment.

3.1 A simple decision table

The first decision table that we want to discuss uses standard Blueriq features, and could be used as an example during a training course. The decision table is shown in Figure 1. Two special characters are used within the table. The '*' indicates that the value of the attribute on the left does not matter at this point. The '[]' indicate all other alternatives that have not been mentioned so far for the condition attribute.

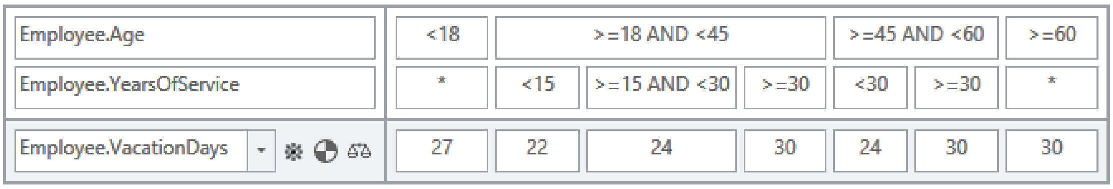


Employee.Age	<18	>=18 AND <45			>=45 AND <60		[]
Employee.YearsOfService	*	<15	>=15 AND <30	[]	<30	[]	*
Employee.VacationDays	27	22	24	30	24	30	30

Figure 1: A simple decision table

3.2 A complete decision table

Looking at the decision table shown in Figure 1, one might argue that the use of '[]' can be unclear, and that it is better to completely specify each alternative. This is also possible, and results in the decision table depicted in Figure 2.



Employee.Age	<18	>=18 AND <45			>=45 AND <60		>=60
Employee.YearsOfService	*	<15	>=15 AND <30	>=30	<30	>=30	*
Employee.VacationDays	27	22	24	30	24	30	30

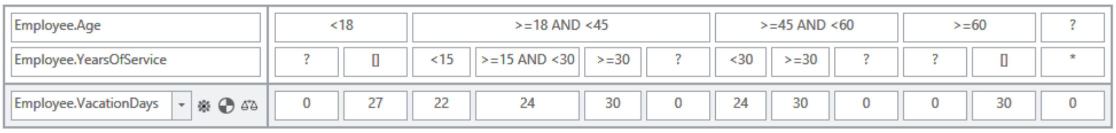
Figure 2: A complete decision table

3.3 A decision table that handles unknowns

Blueriq recognizes the concept of unknown. A value of an attribute remains unknown as long as the end user did not fill in a value on a page, or if no logic exists that can derive the value. Looking back at the decision tables shown in Figure 1 and 2, what will happen if either Employee.Age or Employee.YearsOfService is unknown when trying to derive a value for Employee.VacationDays? In most cases these decision tables are not able to derive a value.[1] Please note here that '[]' does not include an unknown value, but that '*' does include unknown. If the decision table can not derive a value, other logic is consulted for a value. If no logic is able to derive a value, Employee.VacationDays remains unknown, which often is exactly what you want.

Knowing this, one might argue that the decision table shown in Figure 2 is not complete at all, as the case of unknown values is not handled. We can add these cases to the table, and determine that in case of an unknown value, the resulting number of vacation days should be 0 or 22 as the mentioned in the problem description. We chose 0 and the result is shown in Figure 3.

[1] There exist cases in which the value can be determined. For example if the Employee.Age is below 18 and Employee.YearsOfService is unknown.

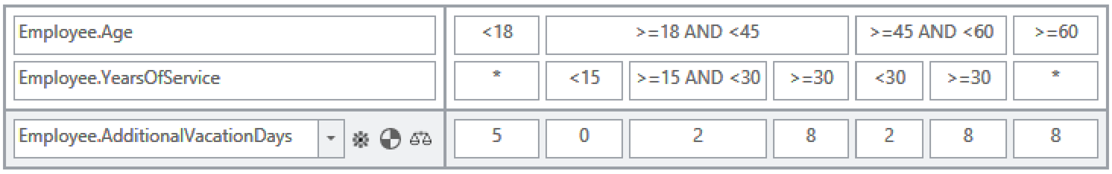


Employee.Age	<18		>=18 AND <45				>=45 AND <60			>=60		?
Employee.YearsOfService	?	[]	<15	>=15 AND <30	>=30	?	<30	>=30	?	?	[]	*
Employee.VacationDays	0	27	22	24	30	0	24	30	0	0	30	0

Figure 3: A decision table that handles unknowns

3.4 Only deriving additional days

When reading the specifications, one might also choose to only derive the additional days instead of the total number of days. In this solution three attributes are needed for Employee, (1) VacationDays, (2) BaseVacationDays and (3) AdditionalVacationDays. BaseVacationDays should have default value 22 and VacationDays has a default expression with the addition of the other two attributes. The decision table to derive the additional vacation days is shown in Figure 4.



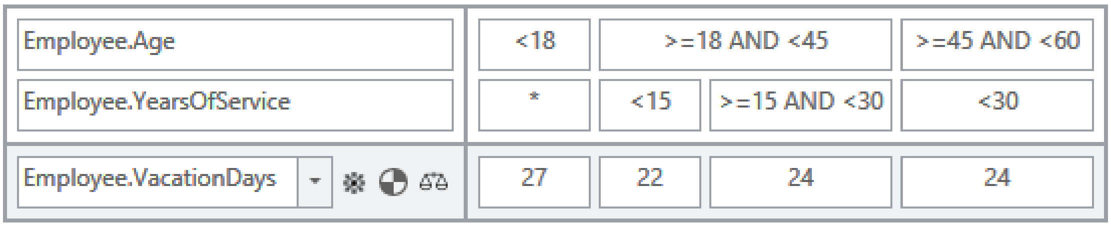
Employee.Age	<18	>=18 AND <45			>=45 AND <60		>=60
Employee.YearsOfService	*	<15	>=15 AND <30	>=30	<30	>=30	*
Employee.AdditionalVacationDays	5	0	2	8	2	8	8

Figure 4: Only deriving additional days

3.5 Default value of 30

In addition to deriving the value of an attribute in a decision table, it is also possible to set a default value on an attribute. Any path that leads to the default value can then be discarded from the table. When the table can not determine the value in these cases, the default value is used. An advantage of this method is that the decision table can become smaller. A disadvantage is that not all possible values of a variable are visible on the same screen.

A natural default value for this problem would be 22, as that is the lower minimum. This will however only eliminate one outcome of the decision table shown in Figure 2. Therefore, we choose the default value of 30 in this example, which is less intuitive, but removes more cases from the decision table. The resulting table is shown in Figure 5.

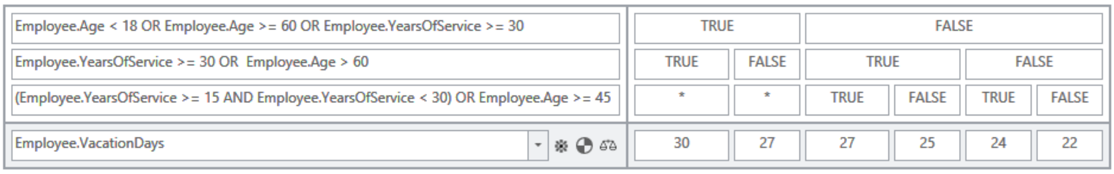


Employee.Age	<18	>=18 AND <45		>=45 AND <60
Employee.YearsOfService	*	<15	>=15 AND <30	<30
Employee.VacationDays	27	22	24	24

Figure 5: Default value of 30

3.6 A decision table with complex conditions

The decision tables that we have seen so far always had a simple attribute on the left hand side. The business engineer is not limited to simple attributes in Blueriq, and it is possible to write complex expressions on the left hand side as well. In this example we want to demonstrate that, and to create a decision table that reads exactly as the specifications given. Figure 6 shows the results of this approach. Although the left hand side reads like the specifications, this style can become difficult to understand or maintain when these expressions become complex.

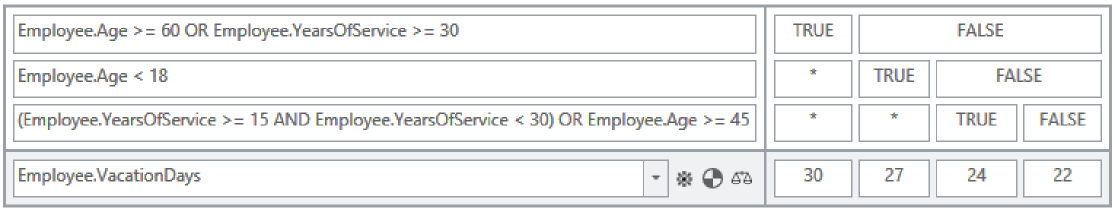


Employee.Age < 18 OR Employee.Age >= 60 OR Employee.YearsOfService >= 30	TRUE		FALSE			
Employee.YearsOfService >= 30 OR Employee.Age > 60	TRUE	FALSE	TRUE		FALSE	
(Employee.YearsOfService >= 15 AND Employee.YearsOfService < 30) OR Employee.Age >= 45	*	*	TRUE	FALSE	TRUE	FALSE
Employee.VacationDays	30	27	27	25	24	22

Figure 6: A decision table with complex conditions

3.7 The smallest decision table

When looking carefully at the specifications, you can notice that anyone that qualifies for rule #2, also qualifies for rule #1. By starting with rule #2 in the decision table, we can simplify the table, as can be seen in Figure 7. This decision table has the lowest number of cells seen so far while still being complete, which is something one may strive for when designing a decision table. However, as the rules in this table are drastically different compared to the stated rules in the problem definition, one can argue that it is difficult to maintain when the business decides to perform a change to the rules.



Employee.Age >= 60 OR Employee.YearsOfService >= 30	TRUE	FALSE		
Employee.Age < 18	*	TRUE	FALSE	
(Employee.YearsOfService >= 15 AND Employee.YearsOfService < 30) OR Employee.Age >= 45	*	*	TRUE	FALSE
Employee.VacationDays	30	27	24	22

Figure 7: The smallest decision table

4 Conclusions

We have seen seven different decision tables that are able to solve the stated problem. We leave it to the Business Engineer to decide which table is preferable. We believe that the table that is easiest to implement and maintain is shown in Figure 1 or 2.

Contact Us

If you have any questions about this article or if you would like to start a discussion, do not hesitate to contact us.

Email the author	:	m.schadd@blueriq.com
Email Blueriq	:	info@blueriq.com
Call Blueriq	:	+31 (0)73 6450467
Website Blueriq	:	http://www.blueriq.com

Blueriq BV
Veemarktkade 8
5222 AE s-Hertogenbosch
The Netherlands

About Blueriq

Blueriq is a rule-driven software platform designed to deliver dynamic business solutions for organizations with knowledge-intensive processes. It empowers organizations in fast changing environments to quickly and cost-effectively respond to changing business conditions and regulations. Blueriq provides solutions for Decision Management, Dynamic Case Management and intelligent User Experience Management across multiple channels. Solutions based on Blueriq are modelled, not programmed, giving you the opportunity to respond more quickly to your customers needs and improving your business outcomes. With Blueriq, you make your own rules!

References

[1] Decision Management Community. https://dmcommunity.wordpress.com/home/, 2014.