

Map Coloring – Decision Modeling Challenge

Solution by Thierry Guerin

This is a solution to the DMN Community Challenge from May 2019.

It can be executed using the Drools open source engine (www.drools.org) and it can be imported into Trisotech's DMN Modeler for editing/authoring (www.trisotech.com).

Problem statement

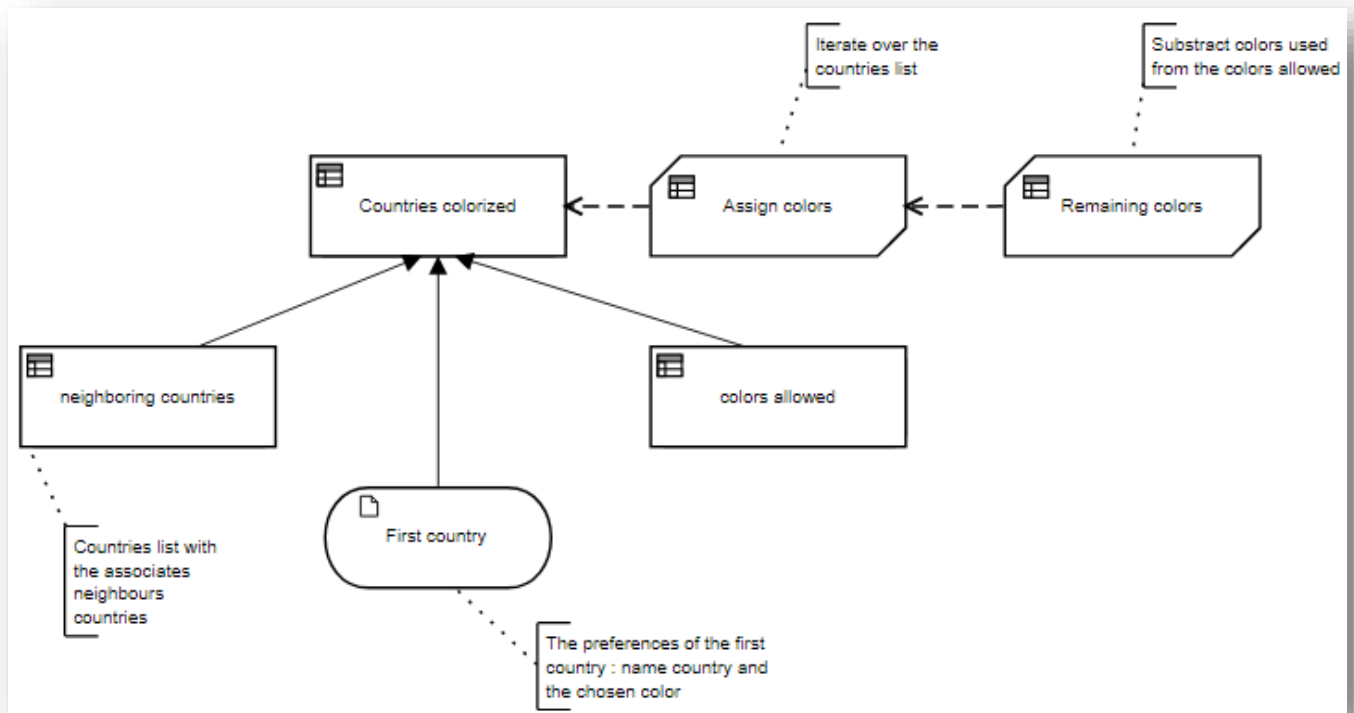


This challenge deals with map coloring. You need to use no more than 4 colors (blue, red, green, or yellow) to color six European countries: Belgium, Denmark, France, Germany, Luxembourg, and the Netherlands in such a way that no neighboring countries use the same color. Click on the map to enlarge it and to see which countries have common borders. Please [submit](#) your solution using your favorite BR&DM tool or no tools at all.

Solution

In accordance with DMN, the high-level solution is modeled in a Decision Requirements Diagram (DRD) shown below. Each element of the diagram is described in the following pages.

Decision Requirements Diagram (DRD)



The two zero-input decisions modeled as relations are static data embedded in the model. "*Colors allowed*" is a list of colors (red, blue, green and yellow). "*Neighboring countries*" is a list of countries with their associates neighbours countries.

The "*First country*" input node is a country associated with a color. This allows you to force a color for one specific country.

Countries colorized (Decision)

It compute all the colored countries by invoking the Business Knowledge Model (BKM) “Assign colors”. The list of the colorized countries is initialized with the “First country” input node.

Countries colorized <i>tCountriesList</i>		
Assign colors		
1	countries List <i>tNeighbourList</i>	<code>neighboring countries[name != First country.name]</code>
2	countriesColorised <i>tCountriesList</i>	<code>[First country]</code>
3	colorsAllowed <i>tColorsAllowed</i>	<code>colors allowed</code>

Assign colors Business Knowledge Model

It iterate over the countries list and assign color to each country.

Assign colors tCountriesList		
F	(countries List tNeighbourList , countriesColorised tCountriesList , colorsAllowed tColorsAllowed)	
1	next country tNeighbours	countries List[1]
2	assigned color tCountry	1 name Texte next country.name
		2 color Texte Remaining colors(colorsAllowed, next country.neighbours, countriesColorised)[1]
+ Résultat		
3	remaining countries tNeighbourList	remove(countries List, 1)
4	updated Countries List tCountriesList	append(countriesColorised, assigned color)
<pre>if count(remaining countries) > 0 then Assign colors(remaining countries, updated Countries List, colorsAllowed) else updated Countries List</pre>		

Remaining colors Business Knowledge Model

It compute the colors used by the neighbours countries of the current country and set the list of the colors allowed.

Remaining colors <i>tColorsAllowed</i>		
F	(colors countriesList countriesColored) <i>tColorsAllowed</i> , <i>tNeighbourList</i> , <i>tCountriesList</i>	
1	colorsUsed <i>tColorsAllowed</i>	distinct values(countriesColored[name in countriesList].color)
2	colorsResult <i>tColorsAllowed</i>	flatten(for color in colors return if color in colorsused then "" else color)
colorsResult[item != ""]		

Neighboring countries Decision

“*Neighboring countries*” is a list of countries with their associates neighbours countries and these are static data embedded in the model.

neighboring countries <i>tNeighbours</i>		
	name	neighbours
	<i>Texte</i>	<i>tNeighbourList</i>
1	"France"	["Belgium","Germany","Luxembourg"]
2	"Belgium"	["France","Germany","Netherlands"]
3	"Denmark"	"Germany"
4	"Germany"	["France","Belgium","Netherlands","Denmark"]
5	"Luxembourg"	["France","Belgium","Germany"]
6	"Netherlands"	["Belgium","Germany"]

Colors allowed Decision

“Colors allowed” is a list of colors (red, blue, green and yellow) and these are static data embedded in the model.

colors allowed <i>tColorsAllowed</i>	
	
1	"blue"
2	"red"
3	"green"
4	"yellow"

Data types

tNeighbours

name	Texte
neighbours	III Texte

tCountry

name	Texte
color	Texte

tColorsAllowed

III Texte

tCountriesList

III tCountry

tNeighbourList

III Texte
