Tackling the DMN Challenges with cDMN: A Tight Integration of DMN and Constraint Reasoning

RuleML + RR 2020

Bram Aerts, **Simon Vandevelde**
Joost Vennekens
KU Leuven — EAVISE
29 June 2020

KU LEUVEN eavise

Outline

1. DMN
2. cDMN & Examples
3. Comparison
4. Conclusion

1 DMN

What is DMN?

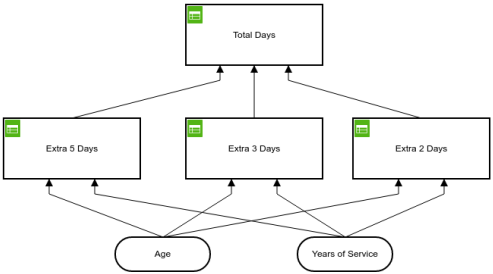
Decision Model and Notation

- Notation standard for decision logic
- Published by OMG Group
- Table-based representation of decisions
- Decision Requirements Diagram
- Main focus: readability, traceability

What is DMN?

Decision Requirements Diagram

- Represents structure of decision model
- Inputs, decision tables, knowledge sources, . . .



What is DMN?

Decision Tables

- Decide outputs (blue) based on inputs (green)
- Rows represent rules
- Columns represent values
- Behaviour defined by Hit Policy
- (S-)Feel in inputs

Eligible for 5 extra days			
U	**Age**	**Years of Service**	**Eligible5**
1	< 18	—	Yes
2	$[18, 60]$	≤ 10	No
3	$[18, 60]$	≥ 10	Yes
4	≥ 60	—	Yes

What is DMN?

(Simple) Friendly Enough Expression Language

- Simple values
- Lists of values
- Numerical comparisons
- Ranges of values
- Calculations

DM Challenges

- Monthly challenges, posted by DMCommunity
- Varying degree of difficulty
- Exploring the boundaries of DMN

Challenges

Ponder Decision Management Challenges

The links below describe decision modeling problems that we post on a regular basis. We invite visitors of this website to submit their solutions. You may use any business rules and decisions management system or none at all. The submitted solutions are posted on this website along with authors names and affiliations. The earlier submitted solutions are published first. Send your solutions to DecisionManagementCommunity@gmail.com.

Challenges used in presentation: “Who Killed Agatha?”, “Make a Good Burger”, “Monkey Business”, “Balanced Assignment”

2 cDMN & Examples

cDMN

cDMN: constraint DMN

- Extends DMN with constraint reasoning
- Quantification, types, functions, relations, optimization, ..
- Model complex decisions
- Introduces new tables
  - Glossary
  - Constraint tables
  - Data tables
  - Execute tables

Glossary

- DMN: constants
- cDMN: types, relations, booleans, constants and functions
- Define these in glossary tables

Glossary

- DMN: constants
- cDMN: types, relations, booleans, constants and functions
- Define these in glossary tables

Type		
Name	**Type**	**Values**
Person	string	Agatha, Butler, Charles
Number	int	[0..100]

Glossary

- DMN: constants
- cDMN: types, relations, booleans, constants and functions
- Define these in glossary tables

Type		
Name	**Type**	**Values**
Person	string	Agatha, Butler, Charles
Number	int	[0..100]

Relation
Name
Person hates Person
Person is richer than Person

Glossary

- DMN: constants
- cDMN: types, relations, booleans, constants and functions
- Define these in glossary tables

Type		
Name	**Type**	**Values**
Person	string	Agatha, Butler, Charles
Number	int	[0..100]

Relation
Name
Person hates Person
Person is richer than Person

Boolean
Name
Suicide

Glossary

- DMN: constants
- cDMN: types, relations, booleans, constants and functions
- Define these in glossary tables

Type		
Name	**Type**	**Values**
Person	string	Agatha, Butler, Charles
Number	int	[0..100]

Relation
Name
Person hates Person
Person is richer than Person

Boolean
Name
Suicide

Constant	
Name	**Type**
Killer	Person

Glossary

- DMN: constants
- cDMN: types, relations, booleans, constants and functions
- Define these in glossary tables

Type		
Name	**Type**	**Values**
Person	string	Agatha, Butler, Charles
Number	int	[0..100]

Relation
Name
Person hates Person
Person is richer than Person

Boolean
Name
Suicide

Constant	
Name	**Type**
Killer	Person

Function	
Name	**Type**
Age of Person	Number
Hatees of Person	Number

Decision and Constraint Tables

DMN Decision table:

- Uniquely define output
- Only single values as output
- Default to Null

Constraint table:

- Range of values
- S-FEEL in output
- No default
- Represent implications

Decision and Constraint Tables

DMN Decision table:

- Uniquely define output
- Only single values as output
- Default to Null

Constraint table:

- Range of values
- S-FEEL in output
- No default
- Represent implications

Nutritional values				
E*	**RequireHealthy**	**Total Sodium**	**Total Fat**	**Total Calories**
1	Yes	< 3000	< 150	< 3000

$$RequireHealthy \Rightarrow TotalSodium < 3000 \wedge TotalFat < 150 \wedge TotalCalories < 3000$$

Decision and Constraint Tables

Quantification

- Apply rules over all elements of type
- More flexible and compact tables

Example: Map Coloring

Bordering countries can not share colors				
E*	**Country called c1**	**Country called c2**	**c1 and c2 are Bordering**	**Color of c1**
1	-	-	Yes	Not(Color of c2)

$$\forall c1[Country]c2[Country] : Bordering(c1, c2) \Rightarrow Color(c1) \neq Color(c2)$$

Data Tables

Typically, problems can be split up in two parts:

1. General logic of the problem
2. Specific problem instance to solve

Data Tables

Typically, problems can be split up in two parts:

1. General logic of the problem
2. Specific problem instance to solve

Map coloring problem:

1. Two bordering countries cannot share a color.
2. The specific map to color (e.g. Western Europe).

Data Tables

Logic: decision and constraint tables
Problem instance: data tables

Data Tables

Logic: decision and constraint tables
Problem instance: data tables

- No hit policy, but "Data Table" in name
- Only basic values
- Quantification possible

Data Table: Declaring which countries border			
	Country called c1	**Country called c2**	**c1 and c2 are Bordering**
1	Belgium	France, Luxembourg, Netherlands, Germany	Yes
2	Germany	France, Denmark, Luxembourg, Belgium, Netherlands	Yes
...	...	...	Yes

Execute table

- DMN model: unique solution
- cDMN model: solution space

→ Execute table defines solution

Execute table

- DMN model: unique solution
- cDMN model: solution space

→ Execute table defines solution

Execute
Get 1 model

Execute table

- DMN model: unique solution
- cDMN model: solution space

→ Execute table defines solution

Execute
Get 1 model

Execute
Get all models

Execute table

- DMN model: unique solution
- cDMN model: solution space

→ Execute table defines solution

Execute
Get 1 model

Execute
Get all models

Execute
Maximize Score

cDMN: SUMMARY

constraint DMN

- Extends DMN with constraint reasoning and more
- Adds four new tables:
  - Glossary
  - Constraint tables
  - Data tables
  - Execute tables
- No single solution, but solution space

3 Comparison

Constraint Tables

Map coloring:

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]
3	remaining countries *tNeighbourList*	remove(countries List, 1)
4	updated Countries List *tCountriesList*	append(countriesColorised, assigned color)

```
if count( remaining countries ) > 0  then
  Assign colors( remaining countries, updated Countries List, colorsAllowed  )
else
  updated Countries List
```

Constraint Tables

Map coloring:

Bordering countries can not share colors				
E*	**Country called c1**	**Country called c2**	**c1 and c2 are Bordering**	**Color of c1**
1	—	—	Yes	Not(Color of c2)

Constraint Tables

Who Killed Agatha? (OpenRules)

Rule

The killer hates Agatha, and is not richer than her.

DecisionTable KillerHatesAndNoRicherHisVictim						
ConditionXoperY			ActionXoperY			
IF			THEN			
BUTLER KILLED AGATHA	=	1	Butler Hates Agatha	=	1
BUTLER KILLED AGATHA	=	1	Butler Richer Than Agatha	=	0
CHARLES KILLED AGATHA	=	1	Charles Hates Agatha	=	1
CHARLES KILLED AGATHA	=	1	Charles Richer Than Agatha	=	0
AGATHA KILLED AGATHA	=	1	Agatha Hates Agatha	=	1
AGATHA KILLED AGATHA	=	1	Agatha Richer Than Agatha	=	0

Constraint Tables

Who Killed Agatha? (OpenRules)

Killer constraints		
E*	**Killer hates Agatha**	**Killer richer than Agatha**
1	Yes	No

DecisionTable KillerHatesAndNoRicherHisVictim					
ConditionXoperY			ActionXoperY		
IF			**THEN**		
BUTLER KILLED AGATHA	=	1	Butler Hates Agatha	=	1
BUTLER KILLED AGATHA	=	1	Butler Richer Than Agatha	=	0
CHARLES KILLED AGATHA	=	1	Charles Hates Agatha	=	1
CHARLES KILLED AGATHA	=	1	Charles Richer Than Agatha	=	0
AGATHA KILLED AGATHA	=	1	Agatha Hates Agatha	=	1
AGATHA KILLED AGATHA	=	1	Agatha Richer Than Agatha	=	0

Quantification

Monkey Business (Corticon)

Rule

Monkeys do not share fruit or their place

Mike.fruit<>Sam.fruit	T
Mike.fruit<>Harriet.fruit	T
Mike.fruit<>Anna.fruit	T
Sam.fruit<>Harriet.fruit	T
Sam.fruit<>Anna.fruit	T
Harriet.fruit<>Anna.fruit	T

Mike.place<>Sam.place	T
Mike.place<>Harriet.place	T
Mike.place<>Anna.place	T
Sam.place<>Harriet.place	T
Sam.place<>Anna.place	T
Harriet.place<>Anna.place	T

Monkey Business (Corticon)

Different Preferences				
E*	**Monkey called m1**	**Monkey called m2**	**Place of m1**	**Fruit of m1**
1	—	not(m1)	not(Place of m2)	not(Fruit of m2)

Mike.fruit<>Sam.fruit	T
Mike.fruit<>Harriet.fruit	T
Mike.fruit<>Anna.fruit	T
Sam.fruit<>Harriet.fruit	T
Sam.fruit<>Anna.fruit	T
Harriet.fruit<>Anna.fruit	T

Mike.place<>Sam.place	T
Mike.place<>Harriet.place	T
Mike.place<>Anna.place	T
Sam.place<>Harriet.place	T
Sam.place<>Anna.place	T
Harriet.place<>Anna.place	T

Quantification

Hamburger Challenge (Corticon)

Rule

The sum of sodium and calories of all ingredients have to be lower than 3000 and the total fat has to be lower than 150.

	Conditions	1
a	beefPatties.count * 50 + bun.count * 330 + cheese.count * 310 + onion.count + ketchup_lettuce.count * (3+160) + pickle_tomato.count * (260+3)	< 3000
b	beefPatties.count * 17 + bun.count * 9 + cheese.count * 6 + onion.count*2	< 150
c	beefPatties.count * 220 + bun.count * 260 + cheese.count * 70 + onion.count*10 + ketchup_lettuce.count * (4+20) + pickle_tomato.count * (5+9)	< 3000

Quantification

Hamburger Challenge (Corticon)

Determine Nutrition					
C+	**Item**	**Total Sodium**	**Total Fat**	**Total Calories**	**Total Cost**
1	-	Number of Item * Sodium of Item	Number of Item * Fat of Item	Number of Item * Calories of Item	Number of Item * Cost of Item

Nutrition Constraints			
E*	**Total Sodium**	**Total Fat**	**Total Calories**
1	<3000	<150	<3000

	Conditions	1
a	beefPatties.count * 50 + bun.count * 330 + cheese.count * 310 + onion.count + ketchup_lettuce.count * (3+160) + pickle_tomato.count * (260+3)	< 3000
b	beefPatties.count * 17 + bun.count * 9 + cheese.count * 6 + onion.count*2	< 150
c	beefPatties.count * 220 + bun.count * 260 + cheese.count * 70 + onion.count*10 + ketchup_lettuce.count * (4+20) + pickle_tomato.count * (5+9)	< 3000

Hamburger Challenge (Corticon)

Determine Nutrition					
C+	**Item**	**Total Sodium**	**Total Fat**	**Total Calories**	**Total Cost**
1	-	Number of Item * Sodium of Item	Number of Item * Fat of Item	Number of Item * Calories of Item	Number of Item * Cost of Item

Nutrition Constraints			
E*	**Total Sodium**	**Total Fat**	**Total Calories**
1	<3000	<150	<3000

Data Table: Nutritions					
	Item	**Sodium of Item**	**Fat of Item**	**Calories of Item**	**Cost of Item**
1	Beef Patty	50	17	220	25
2	Bun	330	9	260	15
3	Cheese	310	6	70	10
4	Onions	1	2	10	9
5	Pickles	260	0	5	3
6	Lettuce	3	0	4	4
7	Ketchup	160	0	20	2
8	Tomato	3	0	9	4

Optimization

Balanced Assignment

Diversity score								
C+	**Person called p1**	**Person called p2**	**Department of p1**	**Location of p1**	**Gender of p1**	**Title of p1**	**Group of p1**	**Score**
1	-	-	= Department of p2	-	-	-	not(Group of p2)	1
2	-	-	-	= Location of p2	-	-	not(Group of p2)	1
3	-	-	-	-	= Gender of p2	-	not(Group of p2)	1
4	-	-	-	-	-	= Title of p2	not(Group of p2)	1

Execute
Maximize Score

Comparison in number of cells

- cDMN is often most compact

	Who Killed A.?	Change Making	A Good Burger	Define Dupl.	Coll. of Cars	Monkey Business	Vacation Days	Family Riddle	Cust. Greeting	Online Dating	Class. Employees	Reindeer Order	Zoo, Buses, Kids	Balanced Assign.	Vac. Days Adv.	Map Coloring	Map Color Viol.	Crack the Code	Numerical Haiku
cDMN	53	26	35	20	26	47	38	76	88	45	36	14	24	55	124	21	48	77	41
Corticon	54	14	20	19	45	64	32	22		78	21	64							
OpenRules	176		95	21		150	31		205		70	111	43	30	97				
Others			76[1]		48[1]		14[2]				34[3]	370[4]				31[4]			

4 Conclusion

Conclusion

- Constraints increase expressivity and flexibility
- More complex models possible
- Models are compact and straightforward
- Independent of size of problem

$\rightarrow$ cDMN models are readable, maintainable and scalable

Thank you for your attention.

Any questions?

For further questions or discussion:
s.vandevelde@kuleuven.be

For more information on cDMN and the cDMN solver:
www.cdmn.be