July 7-8, 2016,
Stony Brook University, New York, USA

*DMN: how to satisfy multiple objectives?

Prof. dr. Jan Vanthienen, KU Leuven, FEB

*Who?

Jan Vanthienen
KU Leuven
Faculty of Economics and Business
Business Information Systems Group

LIRIS (Leuven Institute for Research in Information Systems)

C.B.L KATHOLIEKE UNIVERSITEIT LEUVEN

Intelligent Business

OMG WE SET THE STANDARD

bbc BUILDING BUSINESS CAPABILITY
October 31 - November 4, 2016, Las Vegas, NV

WORLD CONGRESS ON DECISION TABLES.

Research and teaching:

- Business rules, processes and information systems
- Information & Knowledge Management
- Decision models & tables
- Business intelligence & Analytics

IBM Faculty Award
Belgian Francqui Chair 2009 at FUNDP
RuleML best paper award 2007

- Bpost bank Research Chair Actionable Analytics
- Colruyt-Symeta Research Chair Smart Data and Decisions in Marketing
- IBM Fund Intelligent Business Decision Making
- Microsoft Research Chair on Intelligent Environments
- PricewaterhouseCoopers Chair on E-Business

Email: jan.vanthienen@kuleuven.be

*Abstract

* Decision modeling with DMN allows to model decisions and offers a standard notation and expression for decision requirements and decision logic.
* Is there only one best model for a (set of) decision(s) or can you have multiple models?
* Based on a recent Decision Management Community challenge, this presentation investigates various objectives of decision modeling:
  - providing an overview for business,
  - verification of business logic,
  - traceability to knowledge sources,
  - maintainability,
  - model-driven execution.

*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.

DECISION MANAGEMENT COMMUNITY

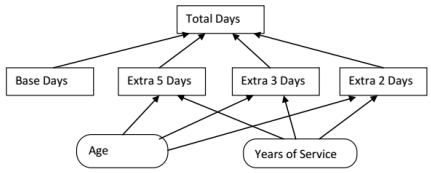
Challenge Jan-2016

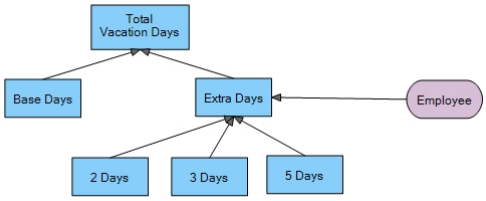
Notes
- Additional days (5, 3, 2) are only added once (that is an assumption indeed), but some of them can be combined.
- The 2 extra days cannot be combined with the 5 extra days.
- You can not have more years of service than your age – the legal minimum age (child labour is illegal).

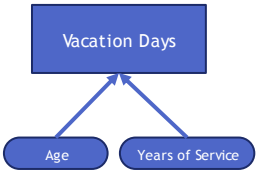
Standardization concerns

- There was a large variety of tools and formats. It would be good for exchange and readability if the common notation is used:
  - [45..60) or [45..60[are well-defined notations (*at least one of them*) and much more compact than "(Age >= 45) and (Age < 60)"
  - Decision tables can return the outcome of one rule (the first, the only one, whatever) or the outcomes of all matching rules. It was not always clear from the submissions what was meant. That is exactly the purpose of the hit indicator: tell us how to read the table. If the hit indicator is not used, only the outcome(s) of one rule will be returned.
  - An irrelevant input entry is indicated with "-", not '*' or blank.

Solutions: DRD







*Typical solution 1 (Multiple Hit)

DecisionTableMultiHit DefineVacationDays					
Condition		Condition		Conclusion	
Age in Years		**Years of Service**		**Vacation Days**	
				=	22
<	18			+=	5
>=	18	>=	30	+=	5
>=	60	<	30	+=	5
		>=	30	+=	3
>=	60	<	30	+=	3
Within	[45..60)	<	30	+=	2
<	45	Within	[15..30)	+=	2

the age of the employee	the nb of year of service		the regular holiday amount is	the holidays bonus is
≥ 15	> 0		22	
< 18				5
< 45	[15	30[		2
≥ 45	[15	30[		4
< 60	≥ 30			5
≥ 60	≥ 30			8

What?

* Collect/add all rules that apply

*Typical solution 1 (Multiple Hit)

Conditions	0	1	2	3	4	5	6	7
Age		< 18	>= 60	18..59	>= 60	18..59	45..59	18..44
Service		-	-	>= 30	>= 30	15..29	< 30	< 15
Actions								
Post Message(s)								
Base Days	22							
Extra Days		5	5	5	3	2	2	0

Rule Statements

Ref	Text
0	Base: Every employee receives at least 22 days
	Extra days:
1	1A. Only employees younger than 18 receive 5 extra days
2	1B. Only employees at least 60 receive 5 extra days
3	1C. Only employees with at least 30 years of service receive 5 extra days (if they didn't already get 5 days because of their age)
4	2. Employees with at least 30 years of service and [also employees of] age 60 or more receive 3 extra days on top of other days
5	3A. If an employee has at least 15 but less than 30 years of service 2 extra days are given. They cannot be combined with the 5 extra days.
6	3B. These 2 days are also provided for employees age 45 or more. They cannot be combined with the 5 extra days.
7	Employees do not get extra days when years of service is less than 15 and age is 18 - 44

What?

* Collect/add all rules that apply

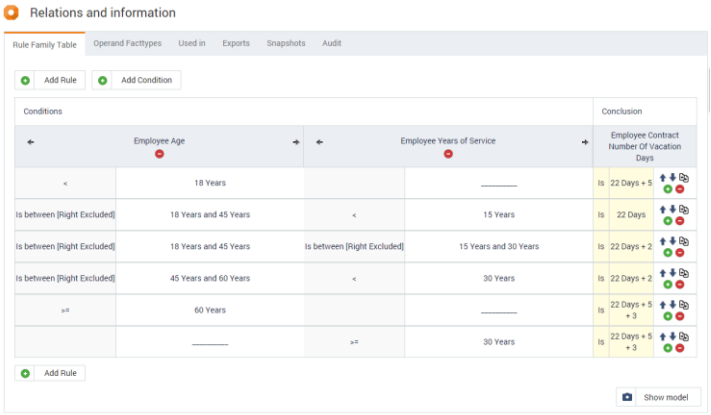
Pro

* Traceability +/-

Con

* Avoiding 5+2 is work
* V&V: Who gets 5+3?
* Where is 'my case'?

*Typical solution 2 (Any)



What?

* Take any rule that applies (result =)

Pro

* Traceability +/-

Con

* Avoiding 5+2 is work
* Completeness?
* Overlap/Redundancy

*Typical solution 3 (First)

	inputs		outputs	
F	Age	Years of Service	Vacation Days	Description
	Number [14,)	Number [0,)	Number [22,)	
1	<18	-	22+5	Basic days + rule #1 People younger than 18 cannot have 30 years of experience and this is the only case that would have changed their holidays count so this information is irrelevant
2	>=60	-	22+8	Basic days + rule #1 + rule #2
3	-	>=30		
4	>=45	-	22+2	Basic days + rule #3
5	-	>=15		
6	-	-	22	Basic days only

Initial Action	RULES					
Days_leave <- 22						
Condition	1	2	3	4	5	ELSE
AGE < 18	Y	N	N	N	N	ELSE
AGE >= 60	N	Y	-	N	N	-
SERVICE >= 30	N	-	Y	N	N	-
SERVICE >= 15	N	-	-	Y	N	-
AGE >= 45	N	-	-	-	Y	-
Action						
Days_leave + 5	X	X	X	.	.	.
Days_leave + 2	.	.	.	X	X	.
Days_leave + 3	.	X	X	.	.	.

What?

* Hit & run

Pro

* Traceability +/-

Con

* Who gets 22+2?
* Hard to maintain
* Completeness?

*Typical solution 4 (Unique)

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

What?

* Take the one rule that applies

DecisionTable DefineVacationDays		
If	If	Then
Age in Years	**Years of Service**	**Vacation Days**
<18		::= (22 + 5)
[18..45)	<15	::= 22
[18..45)	[15..30)	::= (22 + 2)
[18..45)	>=30	::= (22 + 5 + 3)
[45..60)	<15	::= (22 + 2)
[45..60)	[15..30)	::= (22 + 2)
[45..60)	>=30	::= (22 + 5 + 3)
60+		::= (22 + 5 +3)

Initial Action									
Days_leave MV 22		RULES							
Condition	1	2	3	4	5	6	7	8	ELSE
AGE >= ?	-	18	18	18	45	45	45	60	ELSE
AGE < ?	18	45	45	45	60	60	60	-	-
SERVICE >= 15	-	N	Y	-	N	Y	-	-	-
SERVICE >= 30	-	N	N	Y	N	N	Y	-	-
Action									
Days_leave + 5	X	.	.	X	.	.	X	X	.
Days_leave + 2	.	.	X	.	X	X	.	.	.
Days_leave + 3	.	.	.	X	.	.	X	X	.

*Typical solution 4 (Unique)

	inputs		outputs	
U	Age	Years of Service	Vacation Days	Description
	Number (14..)	Number (R..)	Number (22..)	
1	<18	-	22+5	Basic days + rule #1 People younger than 18 cannot have 30 years of experience and this is the only case that would have changed their holidays count so this information is irrelevant
2	[18..45)	<15	22	Basic days
3		[15..30)	22+2	Basic days + rule #3
4		>=30	22+5+3	Basic days + rule #1 + rule #2
5	[45..60)	<30	22+2	Basic days + rule #3
6		>=30	22+5+3	Basic days + rule #1 + rule #2
7	>=60	-	22+5+3	Basic days + rule #1

What?

* Take the one rule that applies

Pro

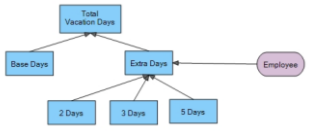
* V&V
* Completeness

Con

* Traceability +/-

Conditions	1	2	3	4	5	6	7
Age	< 18	[18..45)	[18..45)	[18..45)	[45..60)	[45..60)	>= 60
Service	-	< 15	[15..30)	>= 30	< 30	>= 30	-
Actions							
Post Message(s)							
Allocate 22 days	☑	☑	☑	☑	☑	☑	☑
Add 5 days	☑			☑		☑	☑
Add 3 days				☑		☑	☑
Add 2 days			☑		☑		

*Solutions with Decomposition



Base Days Decision

BaseDays.xml

Negate | Expression Builder | Properties | Import

Age

	A	B	C
1	Age	Years of Service	Is Eligible
4			true

Extra 2 Days Decision

2Days.xml

Negate | Expression Builder | Properties | Import

Age

	A	B	C
1	Age	Years of Service	Is Eligible
4	>=45		true
5		[15;30)	true
6			false

Total Vacation Days

TotalVacationDays.xml

Negate | Expression Builder | Properties | Import

Base Days

	A	B	C	D	E
1	Base Days	Eligible for Extra 5 Days	Eligible for Extra 3 Days	Eligible for Extra 2 Days	Vacation Days
4	true				22
5		true			5
6			true		3
7		false		true	2

What?

* Model decision dependencies (how detailed?)
* Tables still have hit policy

DecisionTableMultiHit DefineVacationDays

If	If	If	Conclusion	
Eligible to Extra 5 Days	Eligible to Extra 3 Days	Eligible to Extra 2 Days	Vacation Days	
			=	22
TRUE			+=	5
	TRUE		+=	3
FALSE		TRUE	+=	2

DecisionTable SetEligibleToExtra5Days

If	If	Then
Age in Years	Years of Service	Eligible to Extra 5 Days
< 18		TRUE
>= 60		TRUE
	>= 30	TRUE
		FALSE

DecisionTable SetEligibleToExtra3Days

If	If	Then
Age in Years	Years of Service	Eligible to Extra 3 Days
	>= 30	TRUE
>= 60		TRUE

*Solutions with Decomposition

Total Days

Base Days + (if Extra 5 Days then 5 else 0) + (if Extra 3 Days then 3 else 0) + (if Extra 2 Days and not(Extra 5 Days) then 2 else 0)

Base Days

22

Extra 5 Days

A	Age	Years of Service	
			false, true
1	<18, >=60	-	true
2	-	>= 30	true

Extra 3 Days

A	Age	Years of Service	
			false, true
1	>=60	-	true
2	-	>= 30	true

Extra 2 Days

A	Age	Years of Service	
			false, true
1	>=45	-	true
2	-	[15..30)	true

What?

* Model decision dependencies (how detailed?)
* Tables still have hit policy

Pro

* Traceability
* Simple

Con

* Overview

Some DMN features

Have a look at some DMN conventions:

- A boxed expression to set the base days.
- The A (Any) hit-indicator to list rules connected by “or”, with the same outcome. This allows to stay close to he original text (and automatically avoids giving the 5 days more than once).
- The underlined value false to state the default outcome.
- The use of “,” in ‘<18, >=60’ to indicate ‘or’.
- The expression to calculate the total days (it could also be a table, for people who do not understand ‘if then else’).

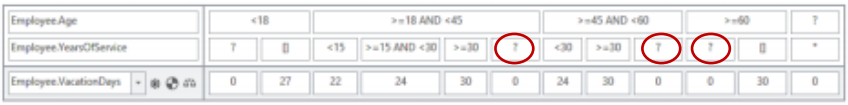
Separating eligibility - exclusion

Conditions	1	2	3	4	5
Age	< 18	>= 60	-	-	>= 45
Years of Service	-	-	>= 30	15..29	-
Actions					
Post Message(s)					
Eligible for A	✓	✓	✓		
Eligible for B		✓	✓		
Eligible for C				✓	✓

Conditions	0	1	2	3	4
Eligible for plan A?		T	-	F	F
Eligible for plan B?		-	T	-	F
Eligible for plan C?		-	-	T	F
Actions					
Post Message(s)		✉	✉	✉	
Base	22				
Extra		5	3	2	0

Ref	Text
0	22 base days for everyone
1	5 extra days for the A entitlement
2	3 extra days for the B entitlement
3	2 extra days for the C entitlement but only if they are not entitled to A
4	0 extra days

*Solution with unknown



Model or execution?

*Incomplete Solution with default

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

Default is 30

Compact (!) solution (?)

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

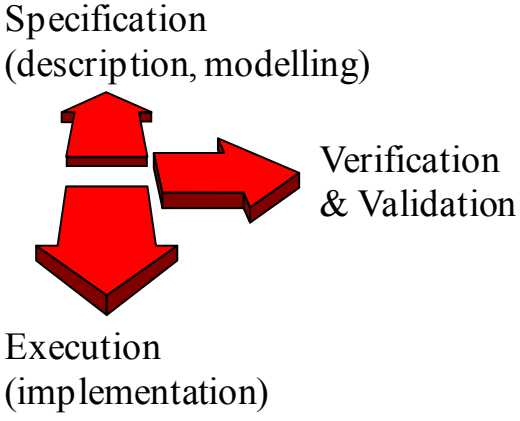
Business questions and criteria

- Decision making: John Doe is 50 and has 30 years of service, how much does he get (30 days)?
- Overview: Who finally gets the 2 extra days? Where is my case?
- Analysis: Are there some strange assignments in giving additional days? Yes, if you turn 18, e.g., you lose 5 bonus days.
- Maintenance: What if the combination policy for 2 and 5 bonus days changes?
- Traceability: How easy is it to trace back to the original text (rules)?

Solution Types

- **Building a model which clearly captures the original specification and can easily be traced back to it.** This is flexible and easy to maintain. If the bonus rules or the combination rules change, the model easily follows.
- **Buiding a model which easily shows the overview, allowing to analyze and validate the business concerns.** These models allow to compare the bonus days over age categories, spot strange outcomes and show the result in a blink of the eye.
- **Building a model (and tables) that tries to combine both views: show the final outcomes and ensure traceability to some extent.** That is often a compromise and challenging.

Modeling objectives



Solutions

Table type?	DRD?	Traceable	Maintainable	Overview	DMN conformant	Score
Unique		medium	medium	high	high	*very good*
Multiple		medium	medium	medium	high	*good*
First	**Decomposed**	**high**	**high**	**medium**	**medium**	***excellent***
Any	**Decomposed**	**high**	**high**	**medium**	**high**	***excellent***
Multiple		medium	low	medium	medium	*good*
Multiple	**Separating Exclusion**	**high**	**high**	**medium**	**high**	***excellent***
Unique		medium	medium	high	high	*very good*
Unique		**high**	**medium**	**high**	**high**	***excellent***
First		high	medium	medium	high	*very good*
Any		high	medium	medium	medium	*good*
Unique		medium	medium	high	medium	*good*
Unique with default		medium	medium	high	medium	*good*
Incomplete with default		medium	medium	medium	medium	*good*
Compact		medium	low	medium	low	*low*
More Compact		medium	low	low	low	*low*
First		high	medium	medium	medium	*good*
Unique		medium	medium	high	medium	*very good*
Multiple		medium	medium	medium	medium	*good*
First	Decomposed	high	high	medium	low	*good*

Note 1: DMN Power

- All this is possible in DMN, in a standardized way, and immediately executable from the model.
- With all these possibilities, it is important to understand and exchange each other's models. So, follow the DMN conventions.
- Methodology should not be blind for objectives.

Model transformations and optimizations

Lots of conversions/transformations/optimizations already exist:

* From tables to optimal code (Codasyl 1982)
* Decision table mining (1998, 2003)
* Lifecycle of hit policies (1988)
* Normalization of decision tables (1993)
* Decision models and business processes (2007)
* Factoring/Defactoring of decision models (1996)
* Decision model mining (2015)
* Mixed-paradigm process modeling (2016)
* Verification of tables (1998)
* Verification between tables (1998)
* From rules to tables (1982, 1993)
* From tables to minimal rules (1986)
* From logs to decision requirements (2010)
* Mixed-paradigm mining (2015)
* Construction methodologies (1986)
* Decision model dependencies (2012)
* Rule set consistency (2007)
* From Knowledge discovery to decision tables (2001, 2005)

Note 2: Full DMN usage

There is more than the basic big decision : *John Doe is 50 and has 30 years of service, how much does he get?*

The model should also allow to answer:

* What if I turn 45?
* Why do I get 27 days?
* What do I have to change to obtain x days?
* How do I get the maximum? (optimization)
* I have incomplete data, what can I already decide/exclude
* …

*Thank you

and thanks to all who submitted their model