

The package `decision-table`*

S. Vandeveldel, F. Pantigny
s.vandeveldel@kuleuven.be
fpantigny@wanadoo.fr

December 8, 2020

Abstract

The LaTeX package `decision-table` provides a command `\dmntable`, which allows for an easy way to generate decision tables in the Decision Model and Notation (DMN) format. (See Fig. 1) This package ensures consistency in the tables (i.e. fontsize), and is thus a better alternative to inserting tables via images.

1 Description

The `decision-table` package allows for an easy way to generate decision tables in the Decision Model and Notation (DMN) format. (See Fig. 1) This package ensures consistency in the tables (i.e. fontsize), and is thus a better alternative to inserting tables via images.

Calculate BMI			
U	Weight(kgs)	Length(m)	BMI
1	—	—	weight/length*length

Figure 1: Example of a DMN table

2 How to use

The `decision-table` package adds the `dmntable` command, with which tables can be created. This command expands into a `tabular`, so it can be used within a `table` or `figure` environment. Furthermore, this allows labels and captions to be added seamlessly. It is also possible to place multiple DMN tables in one `table/figure` environment.

The `dmntable` command has the following inputs:

- title
- hit policy
- input column headers
- output column headers
- the table values

The command is used as follows:

```
% \dmntable{title}{hitpolicy}{input}{output}{values}  
%
```

*This document corresponds to the version 0.0.3 of `decision-table`, at the date of 2020/12/08.

The input, output and cell values are split by a comma. It is not necessary to include the row numbers for the cell values. For example, 1 is generated by the following code:

```
% \begin{figure}[H]
%   \centering
%   \dmntable{Calculate BMI}{U}           % Title and HP
%           {Weight(kgs), Length(m)}{BMI} % Input, output
%           {---,---,weight/length*length} % Cell values
%   \caption{Example of a DMN table}
%   \label{ex1}
% \end{figure}
%
```

If a cell value contains multiple values (e.g. multiple string values), then accolades should be written around them. See the example 4.

3 Examples

This section contains some example code and their resulting tables.

```
% \begin{figure}[H]
%   \centering
%   \dmntable{Calculate BMI}{U}           % Title and HP
%           {Weight(kgs), Length(m)}{BMI} % Input, output
%           {---,---,weight/length*length} % Cell values
%   \caption{Example of a DMN table}
% \end{figure}
%
```

Calculate BMI			
U	Weight(kgs)	Length(m)	BMI
1	—	—	weight/length*length

Figure 2: Example of a DMN table

```
% \begin{figure}[H]
%   \centering
%   \dmntable{Decide BMI Level}{U}
%           {BMI}{BMI Level, Risk Level}
%           {$ < 18.5$, Underweight, Increased,
%           $[18.5..24.9]$, Normal, Low,
%           $[25..29.9]$, overweight, Increased,
%           $[30..34.9]$, Obese I, High,
%           $[35..39.9]$, Obese II, Very High,
%           $ > 39.9$, Extreme Obesity, Extremely High}
%   \caption{Example of a DMN table}
% \end{figure}
%
```

Decide BMI Level			
U	BMI	BMI Level	Risk Level
1	< 18.5	Underweight	Increased
2	[18.5..24.9]	Normal	Low
3	[25..29.9]	overweight	Increased
4	[30..34.9]	Obese I	High
5	[35..39.9]	Obese II	Very High
6	> 39.9	Extreme Obesity	Extremely High

Figure 3: Example of a larger DMN table

```

% \begin{figure}[H]
% \centering
% \dmntable{Decide Risk Level}{U}
% {BMI level, Waist(cm)}{Risk Level}
% {Overweight, $<=88$, Increased,
% Overweight, $> 88$, High,
% Obese I, $<= 88$, High,
% Obese II, ---, Very High,
% Extreme Obesity, ---, Extremely High,
% {not(Overweight, Obese I, Obese II, Extreme Obesity)}, ---, Low
% }
% \caption{Example of cell with multiple values}
% \label{ex2}
% \end{figure}
%

```

Decide Risk Level			
U	BMI level	Waist(cm)	Risk Level
1	Overweight	<= 88	Increased
2	Overweight	> 88	High
3	Obese I	<= 88	High
4	Obese II	—	Very High
5	Extreme Obesity	—	Extremely High
6	not(Overweight, Obese I, Obese II, Extreme Obesity)		Low

Figure 4: Example of cell with multiple values

```

% \begin{figure}[H]
% \centering
% \dmntable{Rule 1}{U}
% {Age, Service Years}{Eligible1}
% {$\leq 18$, ---, Yes,
% $\geq 60$, ---, Yes,
% {$[18, 60]$, $\geq 30$, Yes
% }
% \dmntable{Rule 2}{U}
% {Age, Service Years}{Eligible2}
% {---, $\geq 30$, Yes,
% $\geq 60$, $< 30$, Yes
% }
% \dmntable{Rule 3}{U}
% {Age, Service Years}{Eligible3}
% {$< 45$, {$[15, 30]$, Yes,
% $> 45$, ---, Yes
% }
% \dmntable{Vacation days}{C+}
% {Eligible1, Eligible2, Eligible3}{Vacation Days}
% {---, ---, ---, 22,
% Yes, ---, ---, 5,
% ---, Yes, ---, 3,
% No, ---, Yes, 2
% }
% \caption{dmn table example for the Vacation Days}
% \end{figure}
%

```

Rule 1			
U	Age	Service Years	Eligible1
1	< 18	—	Yes
2	>= 60	—	Yes
3	[18, 60]	>= 30	Yes

Rule 2			
U	Age	Service Years	Eligible2
1	—	>= 30	Yes
2	>= 60	< 30	Yes

Rule 3			
U	Age	Service Years	Eligible3
1	< 45	[15, 30)	Yes
2	> 45	—	Yes

Vacation days				
C+	Eligible1	Eligible2	Eligible3	Vacation Days
1	—	—	—	22
2	Yes	—	—	5
3	—	Yes	—	3
4	No	—	Yes	2

Figure 5: DMN table example for the Vacation Days

4 Contributing

Contributions are always welcome. The project is hosted at <https://gitlab.com/Vadevesi/dmn-tex>

5 Implementation

<@@=dmn>

We give the traditional declaration of a package written with `expl3`:

```
1 \RequirePackage{l3keys2e}
2 \ProvidesExplPackage
3   {decision-table}
4   {\dmnfiledate}
5   {\dmnfileversion}
6   {Table of decision}
7 \RequirePackage { nicematrix }
```

We define the command `\dmntable` with the tools of `xparse` (in 2020 October, `xparse` will be included in the LaTeX kernel).

```
8 \NewDocumentCommand \dmntable { m m m m m }
9   {
```

The clist (*commad separated list*) `\l_@@_input_clist` is for the list of the names of the input fields.

```
10   \clist_clear_new:N \l_@@_input_clist
11   \clist_set:Nn \l_@@_input_clist { #3 }
```

The clist `\l_@@_output_clist` is for the list of the names of the output fields.

```
12   \clist_clear_new:N \l_@@_output_clist
13   \clist_set:Nn \l_@@_output_clist { #4 }
```

The integer `\l_@@_input_int` is the number of the input fields.

```
14   \int_zero_new:N \l_@@_input_int
15   \int_set:Nn \l_@@_input_int { \clist_count:N \l_@@_input_clist }
```

The integer `\l_@@_output_int` is the number of the output fields.

```
16   \int_zero_new:N \l_@@_output_int
17   \int_set:Nn \l_@@_output_int { \clist_count:N \l_@@_output_clist }
```

The sequence `\l_@@_cells_seq` is the sequence of all the cells of the “body” of the tabular.

```
18   \seq_clear_new:N \l_@@_cells_seq
19   \seq_set_split:Nnn \l_@@_cells_seq { , } { #5 }
```

Now, we will begin the construction of the tabular (a `{NiceTabular}` of `nicematrix`).

The command `\use:x` will expand its argument. Indeed, the preamble of the `{NiceTabular}` (which has the same format as a preamble of `{tabular}`) must be computed before the execution of the `\begin{NiceTabular}`.

```
20   \use:x
21   {
22     \exp_not:N \begin { NiceTabular }
```

Here is the preamble of the tabular. The command `\prg_replicate:nn` is expandable and hence will be expanded by the `use:x`.

```
23      { r \prg_replicate:nn { \l_@@_input_int + \l_@@_output_int } 1 }
```

Here is the list of options of the `{NiceTabular}` (a standard tabular of `{array}` don't have such list of options. Once again, we have to compute some quantities in this list of options before the execution of `\begin{NiceTabular}`).

```
24      [
```

The key `hvlines-except-corners` will draw all the rules of the tabular, excepted in the (upper right) corner.

```
25      hvlines-except-corners ,
```

The key `code-before` of `{NiceTabular}` contains instructions to color the cells *before* the rules (doing so, the resulting PDF gives better results in the PDF viewers).

```
26      code-before =
```

First, a `\rectanglecolor` for the labels of the “input” fields. The command `\int_eval:n` is expandable and, hence, will be expanded by the `\use:x`. On the other side, we have to prevent the expansion of `\rectanglecolor` which, in fact, at that point is not defined (it will be defined by `nicematrix` after the construction of the array).

```
27      \exp_not:N \rectanglecolor
28      { blue!10!green!60!black!30 }
29      { 2 - 2 }
30      { 2 - \int_eval:n { \l_@@_input_int + 1 } }
```

A `\rectanglecolor` for the labels of the “output” fields.

```
31      \exp_not:N \rectanglecolor
32      { green!30!blue!15 }
33      { 2 - \int_eval:n { \l_@@_input_int + 2 } }
34      { 2 - \int_eval:n { \l_@@_input_int + \l_@@_output_int + 1 } }
35      ]
36      }
```

Now, we begin the body of the tabular (the environment `{NiceTabular}`).

The body begins by a `\multicolumn` for the title. However, we have to compute the number of cells of that `\multicolumn`. That's why we have to expand the first argument of the `\multicolumn` before executing the `\multicolumn`. However, we have to do that in an expandable way in order to prevent the functionality of the `\multicolumn` (which internally give a `\omit` of TeX). That's why we have to use `\exp_args:Ne` (`\exp_args:Nx` would not do the job).

```
37      \exp_args:Ne \multicolumn
38      { \int_eval:n { \l_@@_input_int + 1 } }
39      { 1 }
40      { #1 } \\\
```

`#2` is the *hit policy*.

```
41      #2 &
```

Now, the fields (“input fields” and “output fields”). By using `\clist_use:Nn`, we replace the commas by ampersands (`&`).

```
42      \clist_use:Nn \l_@@_input_clist { & } &
43      \clist_use:Nn \l_@@_output_clist { & } \\\
```

Now, all the rows corresponding to the rules. We begin a loop over all the cells with `\seq_map_inline:Nn`.

```
44     1 &
45     \seq_map_inline:Nn \l_@@_cells_seq
46     {
```

`\c@jCol` and `\c@iRow` are counters provided by `{NiceTabular}` for the current column and the current row. If you are in the first column, we insert the number of rule.

```
47         \int_compare:nT { \c@jCol = 0 } { \int_eval:n { \c@iRow - 1 } & }
```

Now, we add one composante of `\l_@@_cells_seq`.

```
48         ##1
```

Before the following cell, we have, of course, to add `\\` (if we are at the end of the row) or `&` (elsewhere).

```
49         \int_compare:nTF { \c@jCol = \l_@@_input_int + \l_@@_output_int + 1 }
50             { \\ }
51             { & }
52     }
53     \end { NiceTabular }
54 }
```

Contents

1	Description	1
2	How to use	1
3	Examples	2
4	Contributing	3
5	Implementation	4