A Tool for the Uniqueification of DMN Decision Tables

Kimon Batoulis and Mathias Weske

Hasso Plattner Institute, University of Potsdam, Potsdam, Germany
{Kimon.Batoulis,Mathias.Weske}@hpi.de

Abstract. The Decision Model and Notation (DMN) prescribes decision tables as the standardized way of expressing decision logic. The rules of the decision table may be overlapping, meaning that given some input, multiple rules of the table match and some conflict resolution strategy has to be applied to determine the final output. Therefore, the input-output behavior of tables with overlapping rules is not immediately clear, which makes them hard to understand and unsuitable for analysis tasks. In this paper, we present a tool that transforms DMN decision tables with overlapping rules into equivalent ones that only contain exclusive rules, a process we call uniqueification.

Keywords: DMN, Decision Table Analysis, Uniqueification, Camunda

1 Introduction

The Decision Model and Notation (DMN) [7] is an OMG standard for the design of decisions. In DMN, a decision is modeled on two levels. The decision requirements level represents the dependencies between decisions and what input data is necessary. The decision logic level, in turn, defines the logic of each individual decision. DMN specifies decision tables as the default option for that and prescribes a standardized design. An important attribute of a DMN decision table is its hit policy. This attribute becomes important when tables with overlapping rules are designed.

Consider the table in Fig. 1. This table assigns a credit rating score to a person

R	Income (k) *Number ≥ 0*	Assets (k) *Number ≥ 0*	Credit Rating *A, B, C, D, E*
1	≤ 30	≤ 30	A
2	∈ [10..60]	∈ [10..25]	B
3	∈ [20..95]	∈ [40..90]	C
4	≥ 80	-	D
5	≥ 40	≥ 85	E

Fig. 1. Example DMN decision table containing overlapping rules with a rule order hit policy

based on their income and assets. However, for certain inputs more than one rule matches. For example, for the input $(15, 10)$ both rules 1 and 2 are eligible. This is a conflict that requires further guidance, because it is not clear which rule's output should be chosen in this case. Therefore, a table with overlapping rules has to indicate a hit policy that should be applied in such cases. This policy is represented by a single letter in the upper left corner of the table. For instance, the table in Fig. 1 specifies a *rule order* policy, represented by the letter R. With this policy the outputs of all matching rules are collected in a list which is sorted according to the order of the matching rules in the table. Therefore, the table would return $[A, B]$ as the decision result given the input $(15, 10)$.

Apparently, the set of possible outputs of a decision table can be different from the outputs of the individual rules, and depending on how many overlapping rules there are, this set may not be obvious. For example, the set of possible outputs of the example table is:

$$A, B, C, D, E, [A, B], [C, D], [C, E], [D, E], [C, D, E].$$

This set is not easily identifiable by looking at the table. The same holds for the other way round. Which inputs are responsible for which outputs? Phrased more generally, what is the input-output behavior of the table?

Knowing the input-output behavior of a table eases understanding its logic. Also, it facilitates conducting certain analysis tasks that are based on the input-output behavior, such as checking the sound integration of a decision table with a process model [1, 2, 5]. Therefore, we developed and implemented an algorithm that translates a decision table with any number of overlapping rules and any hit policy into an equivalent one (i.e., one having the same input-output behavior) containing only exclusive rules—a process we call uniqueification, which was already described in [4]. In such a table the input-output behavior is clearly visible from the individual rules.

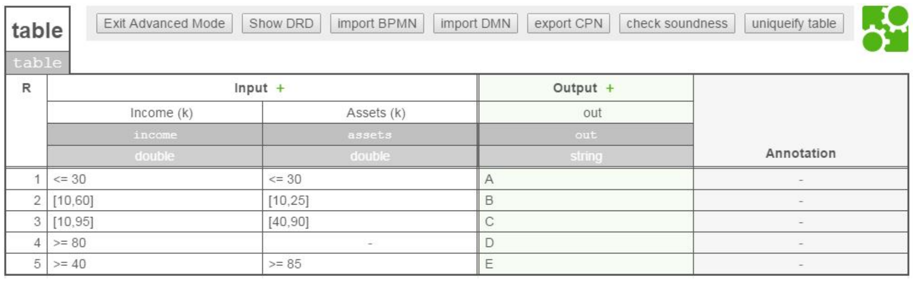
2 Tool

The tool to uniqueify DMN decision tables is implemented in *dmn-js*, a DMN decision table editor developed by Camunda[1]. Our implementation builds upon functionality provided in a tool to verify DMN decision tables [6] and also a tool to check the soundness of decision-aware business processes [3]. In fact, it is an extension of [3], with added functionality for table uniqueification.

Our tool is available for download together with exemplary models, the pseudocode of the algorithm, and a screencast at `https://bpt.hpi.uni-potsdam.de/Public/TableUniqueification`.

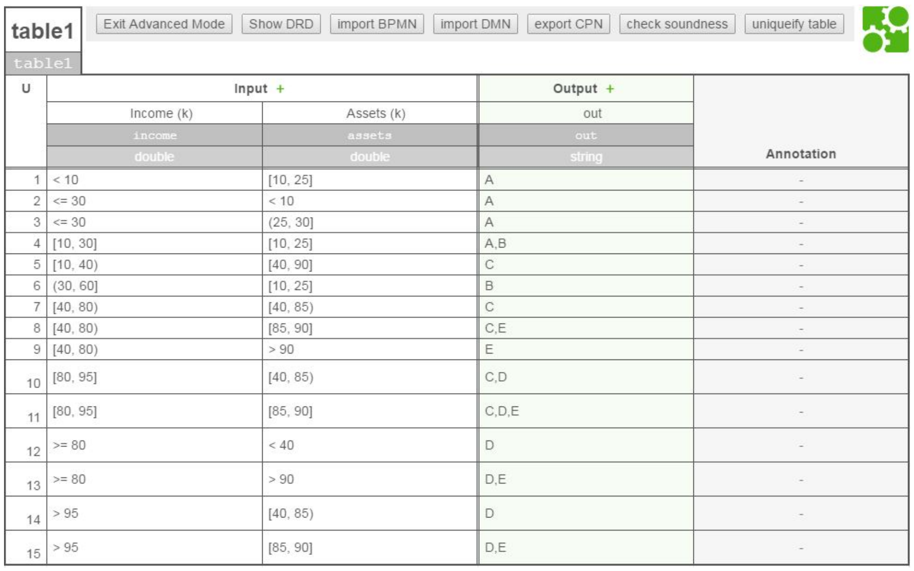
In the following, we will apply the tool to the example from Section 1. Fig. 2 shows the interface of the tool after the table was opened. Since this is an extension of an existing tool, there are several buttons at the top. The rightmost one triggers the algorithm for uniqueifying the table. This will translate the table into an equivalent one (in terms of input-output behavior) that now consists only of exclusive rules. The view of the tool displaying this translated table is shown in Fig. 3. The table has 15 rules, all of them being unique (i.e., exclusive).

[1] `https://camunda.org`



R	Input +		Output +	Annotation
	Income (k)	Assets (k)	out	
	income	assets	out	
	double	double	string	
1	<= 30	<= 30	A	-
2	[10,60]	[10,25]	B	-
3	[10,95]	[40,90]	C	-
4	>= 80	-	D	-
5	>= 40	>= 85	E	-

Fig. 2. View of the tool after opening a DMN decision table



U	Input +		Output +	Annotation
	Income (k)	Assets (k)	out	
	income	assets	out	
	double	double	string	
1	< 10	[10, 25]	A	-
2	<= 30	< 10	A	-
3	<= 30	(25, 30]	A	-
4	[10, 30]	[10, 25]	A,B	-
5	[10, 40)	[40, 90]	C	-
6	(30, 60]	[10, 25]	B	-
7	[40, 80)	[40, 85)	C	-
8	[40, 80)	[85, 90]	C,E	-
9	[40, 80)	> 90	E	-
10	[80, 95]	[40, 85)	C,D	-
11	[80, 95]	[85, 90]	C,D,E	-
12	>= 80	< 40	D	-
13	>= 80	> 90	D,E	-
14	> 95	[40, 85)	D	-
15	> 95	[85, 90]	D,E	-

Fig. 3. View of the tool after pressing the *uniqueify table* button for the table in Fig. 2

Therefore, the table's hit policy is now changed to *unique*, denoted by the letter *U* in the upper left corner.

3 Maturity

In this section, we discuss two aspects regarding our tool. First, we report the results of our performance analysis of the uniqueification algorithm. Second, we discuss the tool's maturity.

For the performance analysis, we generated 1000 synthetic decision tables with up to 50 rows and 30 columns. The results showed that there are two factors that the runtime of the algorithm depends on: On the one hand, it is influenced by the number of overlapping rules of the table. The more overlapping rules there are, the more exclusive rules there can potentially be in the translated table. For example, the maximum execution time of 151 seconds was reached for a 50×6-table with 1138 overlapping rules. On the other hand, the runtime is highly affected by the number of rows of the table because the more rows there are, the more potential overlaps need to be checked. The average execution time for the 1000 decision tables with up to 50 rows and 30 columns was 18 seconds.

Our tool is able to uniqueify any DMN decision table that is based on the S-FEEL language. This language allows standard data types such as strings, numbers and booleans to be used for the input values of the table. Moreover, disjunctions of input values are permissible as well as multiple output variables. The full FEEL language, however, that also supports, for example, formulating output values as functions of input values (such as $output = (input1 + input2) \times 2$) is not supported.

References

1. Batoulis, K., Haarmann, S., Weske, M.: Various notions of soundness for decision-aware business processes. In: Mayr, H.C., Guizzardi, G., Ma, H., Pastor, O. (eds.) Conceptual Modeling. pp. 403–418. Springer International Publishing, Cham (2017)
2. Batoulis, K., Weske, M.: Soundness of decision-aware business processes. In: Carmona, J., Engels, G., Kumar, A. (eds.) Business Process Management Forum. pp. 106–124. Springer International Publishing, Cham (2017)
3. Batoulis, K., Weske, M.: A tool for checking soundness of decision-aware business processes. In: Proceedings of the BPM Demo Track and BPM Dissertation Award co-located with 15th International Conference on Business Process Modeling (BPM 2017), Barcelona, Spain, September 13, 2017. (2017), http://ceur-ws.org/Vol-1920/BPM_2017_paper_184.pdf
4. Batoulis, K., Weske, M.: Disambiguation of dmn decision tables. In: Abramowicz, W., Paschke, A. (eds.) Business Information Systems. pp. 236–249. Springer International Publishing, Cham (2018)
5. Hasić, F., Smedt, J.D., Vanthienen, J.: Augmenting processes with decision intelligence: Principles for integrated modelling. Decision Support Systems (2017)
6. Laurson, Ü., Maggi, F.M.: A tool for the analysis of DMN decision tables. In: Proceedings of the BPM Demo Track 2016 Co-located with the 14th International Conference on Business Process Management (BPM 2016), Rio de Janeiro, Brazil, September 21, 2016. pp. 56–60 (2016), http://ceur-ws.org/Vol-1789/bpm-demo-2016-paper11.pdf
7. OMG: Decision Model and Notation, Version 1.1 (May 2016)