A Tool for Decision Logic Verification in DMN Decision Tables

Carl Corea, Jonas Blatt, and Patrick Delfmann

University of Koblenz-Landau, Germany
{ccorea,jonasblatt,delfmann}@uni-koblenz.de

Abstract. The Decision Model and Notation (DMN) is a popular standard to model company decision logic. Here, decision tables can be used to specify decision logic by the means of business rules. As these tables are modelled and maintained in an incremental and collaborative manner, this raises the need to verify the correctness of DMN decision tables. In this report, we therefore present a tool which allows to analyze the decision logic in DMN decision tables at design-time. Our tool implements all so-called verification capabilities from the recently proposed "business rule management capability framework" by Smit et al. [10], and also allows to detect errors distributed among multiple tables.

Keywords: DMN, Decision Logic Analysis, Camunda

1 Introduction

The Decision Model and Notation (DMN)[1] is an OMG standard for the representation of operational decision logic. Here, decisions can be expressed by the means of so-called decision tables. Fig. 1 shows an exemplary decision table. The columns of the table represent the input (income and assets) and output (credit worthiness). The rows constitute individual rules. The rules in Fig. 1 can be understood such that: *if* the income is ≤20 and the assets are >50, *then* the customer is not creditworthy (etc.).

U	Income (n)	Assets (n)	Creditworthy?
1	≤ 20	>50	false
2	[20..50]	>50	false
3	[30..40]	>100	true

Fig. 1: Exemplary decision table with modelling errors (units in thousands)

While DMN provides a standard on how to *represent* decisions, the actual content of the business rules is still the responsibility of the modeler. In turn,

[1] https://www.omg.org/spec/DMN/About-DMN/

the modelled decision logic can potentially contain errors such as inconsistencies or redundancies. An example of this is shown in Figure 1. First, the income conditions for rules 1 and 2 are overlapping (e.g., for an income of exactly 20). Furthermore, rule 2 completely subsumes rule 3. However, the respective outputs are inconsistent, meaning that this modelling error must be attended to by experts before rules can be merged.

In practice, such modelling errors can occur frequently, as decision tables are mostly maintained by multiple modelers. For example, Batoulis and Weske [1] reported on a case study with a large insurance company, where those authors found that 27% of analyzed rules contained overlaps. This calls for (automated) means to support companies in the verification of decision logic [1].

In this context, Smit et al. [10] have recently proposed the business rule management (BRM) capability framework. This framework identifies specific decision logic level verification capabilities, derived from qualitative research with industrial partners. Thus, those authors present a comprehensive set of verification capabilities actually needed in practice. Therefore, our tool implements these verification capabilities, which are: *identical rule verification, equivalent rule verification, subsumed rule verification, interdeterminism verification, partial reduction verification, overlapping condition verification* and *missing rule verification* (We will discuss these capabilities in in Section 2).

		Detection capabilities						
Literature		Identical Rules	Equivalent Rules	Subsumed Rules	Inter-determinism	Overlapping Conditions	Partial Reduction	Missing Rules
Calvanese et al. (2016)	[5]	X	o	X		X		X
Laurson et al. (2016)	[9]	X		X		X	o	X
Batoulis et al. (2017)	[2]	X		X		X		X
Calvanese et al. (2017)	[7]	X	o	X		X		X
Batoulis et al. (2018)	[3]	X		X		X		o
Batoulis et al. (2018)	[4]	X		X		X		
Calvanese et al. (2018)	[6]	X	o	X		X	o	X
Corea et al. (2018)	[8]	X			X	X		
This work		X	X	X	X	X	X	X

Table 1: Overview of capabilities from [10] covered by existing approaches.
(X = Full support, o = partial support/not aligned with [10])

Table 1 shows an overview of DMN decision logic verification approaches that have been introduced to the BPM community in recent years. To the best of our knowledge, our tool is the first to offer all decision logic level verification capabilities by Smit et al. [1][2]. Most prominently, we extend the works of Laurson and Maggi [9] by covering the capabilities missing in their approach. Also, those authors do not distinguish between identical rules, subsumed rules and overlapping rules, but denote all these error types as overlaps. Here our approach

[2] Please note that we do not implement "unnecessary facts verification", as this is geared towards analyzing case-dependent facts and is beyond the scope of this report.

distinguishes errors aligned along the definition in Smit et al. [10], to provide a more fine-granular understanding for companies as a basis for resolution. Also, to the best of our knowledge, whereas existing tools only allow to analyze individual DMN decision tables, our tool is the first to allow checking *multiple* tables at once. In case that different modelers have created tables, inconsistencies or overlaps between them can be analyzed.

2 Tool Description

Our tool integrates camunda-dmn[3], which is a Java library for DMN by Camunda. Our project can be viewed at `https://gitlab.uni-koblenz.de/fg-bks/br-verification-tool`. Also, an online-demo[4] and screencast[5] are available.

Our browser-based tool allows to upload and analyze DMN decision tables. The analysis is based on the framework by Smit et al. [10]. As mentioned, this framework was derived based on interviews with industrial partners, i.e. the capabilities reflect analysis tasks needed in practice. Our tool therefore implements all of the verification capabilities proposed in [10] as follows:

- **Identical rule verification.** Detecting rules, which have an identical input, i.e. are redundant.
- **Equivalent rule verification.** Detecting rules, which are not identical, but still semantically equivalent. Here, our tool can detect equivalent rules, based on synonym relations.
- **Subsumed rule verification.** Detecting individual rules, which are subsumed by other rules, i.e. they are not necessary.
- **Interdeterminism verification.** Detecting rules, which will *always* be activated together, but have differing or contradicting conclusions. For example, rules must not yield that a customer is both credit worthy, and not credit worthy, as this is logically inconsistent.
- **Partial reduction verification.** Checking whether ranges can be combined to simplify decision tables.
- **Overlapping condition verification.** Detecting whether there are any overlaps in rule conditions.
- **Missing rule verification.** Detecting whether there are any missing business rules, e.g., gaps in condition ranges.

Our tool also allows an analysis of multiple decision tables at once, e.g. the tool can be used to find identical rules which are distributed across multiple tables. A more detailed explanation and examples for the individual verification capabilities can be found in the supplementary documentation[6].

Figure 2 shows an actual usage example. For this example, we uploaded the decision table shown in Figure 1. The tool provides an overview of all errors

[3] `https://github.com/camunda/camunda-engine-dmn`
[4] `http://inconsistency.fg-bks.uni-koblenz.de:8090/`
[5] `https://youtu.be/yTXTKi3s6LM`
[6] `https://gitlab.uni-koblenz.de/fg-bks/br-verification-tool/wikis/home`

found in the tables (1). There are 2 main tabs, namely for errors found *within* single tables, or errors including *multiple* tables (2). Users can browse the errors and click on the "show" button (3) to highlight the concerned rules. Users can also edit and re-verify the tables directly in the browser. We would like to remind the reader of the available online-demo and screencast referenced above.

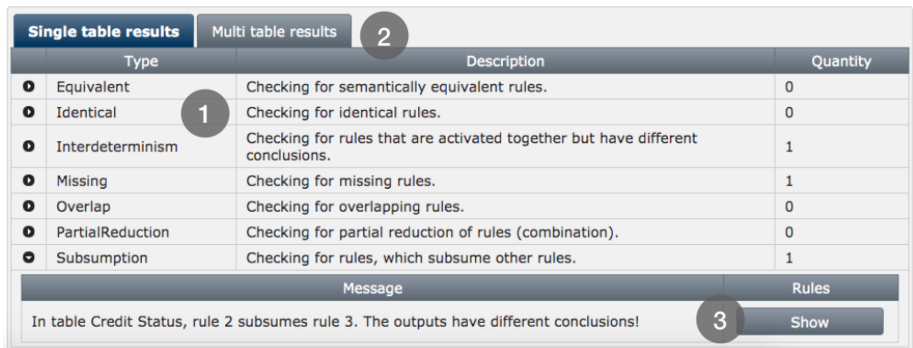

Fig. 2: Usage example: Results of analyzing the table in Figure 1.

3 Maturity and Outlook

For evaluation, we performed run-time experiments. To this aim, we analyzed a total of 300 synthetic decision tables. As parameters for generating these tables, we chose the number of table columns from {1,2,...,10}, and the number of table rows from {50,100,...,500} (i.e., 10x10 possible combinations). For each of the 100 possible combinations of rows and columns, we generated 3 decision tables with different random rules (i.e., 300 decision tables). The respective rules of these tables were randomly generated by using random integer conditions, with one of the operators from $\{=, [a..b], \leq, \geq, <, >\}$. These random conditions allow to create synthetic decision tables with actual errors, which are meant to be analyzed by our tool, such as redundancies or condition overlaps. We consequently applied our verification tool and computed the average run-time for each parameter configuration, which is shown in Figure 3. Our experiments were run on a Windows 10 PC with i7 processor, 16GB DDR4 RAM and 512 GB SSD memory. As can be seen in Figure 3, the run-time for analyzing 500 rules with 10 columns averages to roughly 5s. Thus, for our analyzed data-sets, our tool allowed for a feasible analysis.

To conclude, the tool presented in this work allows to analyze multiple DMN decision tables. The tool currently supports the unique hit policy, which will be extended in future work. Regarding the verification capabilities, our tool implements the capabilities proposed in the practice-based framework by [10], and thus extends existing works such as [9]. Hence, our tool supports companies in decision logic verification and facilitates sustainable business rules management. In future work, we aim to apply our tool to industrial data-sets.

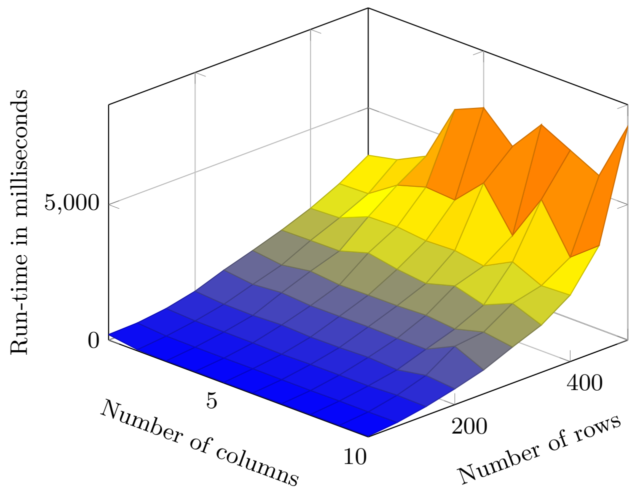

Fig. 3: Run-time for the analysis of synthetic decision tables with up to 10 columns and 500 rows.

References

1. Batoulis, K., Nesterenko, A., Repitsch, G., Weske, M.: Decision management in the insurance industry: Standards and tools. In: Proceedings of the BPM 2017 Industry Track (2017)
2. Batoulis, K., Weske, M.: A tool for checking soundness of decision-aware business processes. In: Proceedings of the BPM Demo Track (2017)
3. Batoulis, K., Weske, M.: Disambiguation of DMN decision tables. In: Business Information Systems - 21st International Conference (2018)
4. Batoulis, K., Weske, M.: A tool for the uniqueification of DMN decision tables. In: Proceedings of the BPM Demo Track (2018)
5. Calvanese, D., Dumas, M., Laurson, Ü., Maggi, F.M., Montali, M., Teinemaa, I.: Semantics and analysis of dmn decision tables. In: International Conference on Business Process Management. Springer (2016)
6. Calvanese, D., Dumas, M., Laurson, Ü., Maggi, F.M., Montali, M., Teinemaa, I.: Semantics, analysis and simplification of dmn decision tables. Information Systems **78** (2018)
7. Calvanese, D., Dumas, M., Maggi, F.M., Montali, M.: Semantic dmn: Formalizing decision models with domain knowledge. In: International Joint Conference on Rules and Reasoning. Springer (2017)
8. Corea, C., Delfmann, P.: A tool to monitor consistent decision-making in business process execution. In: Proceedings of the BPM Demo Track (2018)
9. Laurson, Ü., Maggi, F.M.: A tool for the analysis of dmn decision tables. In: Proceedings of the BPM Demo Track (2016)
10. Smit, K., Zoet, M., Berkhout, M.: Verification capabilities for business rules management in the dutch governmental context. In: Research and Innovation in Information Systems (ICRIIS), 2017 International Conference on. IEEE (2017)