Decision Model Change Patterns for Dynamic System Evolution

Faruk Hasić^{a,*}, Carl Corea^b, Jonas Blatt^b, Patrick Delfmann^b, Estefanía Serral^a

^a*KU Leuven, Research Centre for Information Systems Engineering (LIRIS)
Warmoesberg 26, 1000 Brussels, Belgium*

^b*University of Koblenz-Landau, Institute for IS Research
Universitätsstrasse 1, 56070 Koblenz, Germany*

Abstract

In the modern digital era, information systems must operate in increasingly interconnected and dynamic environments, which force them to be changeable yet consistent. Such modern information systems are usually decision- and knowledge-intensive. A recently introduced standard, the Decision Model and Notation (DMN), has been adopted in both industry and academia as a suitable method for modelling decisions and decision rules. Noteworthy is that, despite the dynamic nature of modern knowledge-intensive systems, DMN was only studied and implemented in a static fashion, as decision schema change patterns have not received any attention so far. This paper identifies and analyses the change patterns that can occur in a DMN decision model. A change in the decision model can require the triggering of other changes in order to safeguard consistency. As such, this paper will also investigate for each change pattern which further changes should be performed to ensure model consistency. The patterns presented in this paper will not only facilitate the understanding of decision change management and within-model consistency, but can also be capitalised on for developing and implementing flexible decision management systems. To illustrate this, we present a modelling environment prototype that provides modelling support when applying the proposed change patterns.

Keywords: DMN, Decision Model and Notation, Model Evolution, Change Patterns.

1. Introduction

Decision Model and Notation (DMN) is a recently introduced decision modelling standard that has enjoyed significant interest in literature [1–8]. DMN consists of two levels that are to be used in conjunction. First, the decision requirement level represented by the Decision Requirement Diagram (DRD) which depicts the requirements of decisions and the dependencies between elements involved in the decision model. Second, the decision logic level, which presents ways to specify the underlying decision logic. The DMN standard employs rectangles to depict decisions and subdecisions, ovals to represent data input, corner-cut rectangles for business knowledge models, and curved rectangles to represent knowledge sources. A schematic example of a DRD is provided in Figure 1. The decision logic level is usually represented in the form of decision tables.

DMN has mainly been adopted in business process management (BPM) literature, which moves towards accommodating decision management into the paradigms of Separation of Concerns (SoC) and Service-Oriented Architecture (SOA), by externalising decisions and encapsulating them into separate decision models [2, 9–13]. Hence, the decisions are implemented as externalised services.

*Corresponding author

Email address: faruk.hasic@kuleuven.be (Faruk Hasić)

This externalisation of decisions from processes, provides a plethora of advantages regarding maintainability and flexibility for the process [2, 5, 14–17]. However, all current works approach DMN from a static perspective, i.e., no attention has been given to changing or adaptable decision models. Nevertheless, the dynamic nature of modern knowledge-intensive systems demands a degree of flexibility to cope with the changing requirements [18]. In this paper we approach this research gap by identifying and defining the change patterns that can be applied to DMN decision models. Additionally, we assess the effect of each applied change pattern on decision model consistency and we investigate and determine the change propagation that is needed to be triggered to keep such consistency. A change pattern is defined as an operation that inserts, deletes, moves, or modifies model elements [19]. Note that moving or modifying model elements can be achieved by applying the insert and delete patterns, as such the insert and delete patterns form the core change patterns that will be the subject of this paper. We also refer to them as inclusion and exclusion patterns. The findings in this paper can aid in developing and implementing dynamic decision management tools and systems that meet the requirements that the digital era demands. To illustrate this, we also provide a proof-of-concept tool for modelling support.

The contribution of this paper is fourfold:

1. We identify and define change patterns for DMN decision models.
2. We analyse the influence of each decision model change pattern on the within-model consistency.
3. We investigate the propagation of decision model change patterns throughout the decision model, i.e., we discuss the chain of change patterns that can be set in motion to restore within-model consistency after applying a single change pattern on the decision model.
4. We develop a proof-of-concept modelling environment prototype which provides modelling support when applying the change patterns proposed in this paper.

This paper is structured as follows. Section 2 discusses related work and Section 3 provides preliminary formalisation needed for the construction of the change patterns. In Section 4 we introduce a running example of a DMN decision model which will be used to illustrate the change patterns. Section 5 introduces a set of DMN decision model change patterns, while Section 6 discusses the inconsistencies and change propagation induced by the change patterns. In Section 7 we present the modelling environment prototype which provides modelling support for change pattern administration in DMN decision models. Finally, Section 8 provides conclusions and directions for future research.

2. Related Work

DMN was first introduced by the Object Management Group (OMG) in 2015. Since then, several works on DMN have emerged. Most works have focused on the integration of process and decision models from a modelling point of view e.g., [2, 15, 20–26]. Others focus on the automatic discovery of decision model from enriched process event logs [9, 27]. Works on soundness of DMN models have been proposed as well [28, 29]. Furthermore, literature provides a set of tools for modelling DMN models [5, 11, 12, 30]. Before the introduction of DMN, The Decision Model (TDM) was introduced for modelling business logic and as a new requirements artifact for decision-aware business processes [31, 32]. Furthermore, works on ontologies for knowledge-intensive business processes have been suggested [33, 34]. These consider additional relevant aspects

in decision making which DMN does not cover or which cannot always be adequately represented in a decision table, such as advantages, disadvantages, risks, facts, evidence and feelings. Nonetheless, in certain knowledge-intensive settings, these aspects can cause changes that affect the consistency of a decision model.

The ability of a knowledge-based system to efficiently deal with decision rule changes is considered of paramount importance in literature [35–39]. This way, business rigidity is avoided and the ability to transform the underlying business rules to new realities is facilitated. However, the lack of research on process models at runtime is emphasised in [40], and existing DMN decision model literature addresses decision modelling from a static perspective where decision models are built and used, without any form of model evolution. Nevertheless, adaptable models are considered of paramount importance [40–42]. Change patterns are often used to define the possible evolution of models. These change patterns rely on the elementary edit operations that can be applied on the model elements, i.e., insertion and deletion, as well as substitution, which in essence is a combination of insertion and deletion [43]. Furthermore, change patterns can help facilitate the understanding of model change management as they provide a guide for implementing changes to models while maintaining model consistency.

Change propagation throughout the model is an important aspect in hierarchical structures where referential integrity needs to be upheld. Typically, such structures are connected by tables, such as the relational database structure. The changes in relational tables are propagated by rules such as on delete cascade or on update cascade [44, 45]. Decision tables that are used in decision models are of a similar hierarchical structure, and hence, similar change propagation can be expected. However, instead of cascading an update or delete action throughout the whole model, the change propagation can also be captured by applying other change patterns. Furthermore, the authors of [46, 47] show that, despite referential integrity being researched for database systems, the verification capabilities for business rules management are still lacking. As such, change patterns for decision models were not addressed yet in literature and, to the best of our knowledge, this is the first paper on change pattern propagation for DMN decision models.

3. Preliminaries

In this section, we also provide a formalisation for key DMN concepts needed for the development of the change patterns that will be discussed in the following sections. For the formalisation, we rely on [2, 48]. The backbone of a DMN decision model is formed by rectangles that depict decisions and subdecisions and ovals that represent data input. Additionally, business knowledge models and knowledge sources can be defined. The underlying decision logic is usually represented in decision table form.

Definition 1 (Decision requirement diagram (DRD)). A DRD is a tuple $(D_{dm}, ID, BK, KS, IR, KR, AR)$ consisting of a finite non-empty set of decision nodes D_{dm} , a finite non-empty set of input data nodes ID , a finite non-empty set of business knowledge model nodes BK , and a finite non-empty set of knowledge source nodes KS . These nodes are connected by requirements into a directed acyclic graph (DAG): a finite non-empty set of directed edges IR representing the information requirements such that $IR \subseteq (D_{dm} \cup ID) \times D_{dm}$, a finite non-empty set of knowledge requirements KR such that $KR \subseteq BK \times (D_{dm} \cup BK)$, and a finite non-empty set of authority requirements AR such that $AR \subseteq (D_{dm} \cup ID \cup KS) \times (D_{dm} \cup BK \cup KS)$.

A schematic DRD is represented in Figure 1. According to the DMN standard, a decision requirement graph can be an incomplete or partial representation of the decision requirements

in a decision model. The set of all DRDs in the decision model constitutes the exhaustive set of requirements. The information contained in this set can be combined into a single DRD representing the decision requirements level as a whole. The DMN standard refers to such a DRD as a decision requirement graph (DRG). We expand the notion of a DRG, in such a way that a DRG is a DRD which is self-contained, i.e. for every decision in the diagram all its requirements are also represented in the diagram.

Figure 1: A schematic overview of DRD model elements.

Definition 2 (DRG). A decision requirement diagram *DRD* is a decision requirement graph *DRG* if and only if for every decision in the diagram all its requirements are also represented in the diagram.

The term *decision* can have a number of meanings. According to the DMN specification a decision is the logic used to determine an output from a given input. Meanwhile, in process modelling an activity can be a decision task, e.g., the business rule task in the Business Process Model and Notation (BPMN) standard [49] in which a decision is made by applying decision logic [32].

Another common meaning is that a decision is the actual result, which we call the output of a decision, or simply the decision result. We define a decision as follows:

Definition 3 (Decision). A decision $d \in D_{dm}$ is a tuple (I_d, O_d, L) , where $I \subseteq ID$ is a set of input symbols, O a set of output or result symbols and L the decision logic defining the relation between symbols in I_d and symbols in O_d .

In case of decision tables, I and O contain the variables of the input and output elements respectively, and L is the table itself, i.e. the set of decision rules present in the table.

In DRDs these decisions d_i are contained by the decision nodes $D_i \in D_{dm}$. One can state that d_i and D_i are equivalent views at different levels of granularity: d_i looks at a decision on its own, while D_i places d_i in the hierarchy of the DRD. We will use D to refer to both a decision and its representing node in a DRD. From the definition of DRGs, it is clear that every decision D

in a DMN model has a unique decision requirement graph DRG_D with D as its single top-level decision. A DRG contains all information requirements of its top level decisions. Hence, only one DRG exists with D as its single top-level decision, i.e., DRG_D . Furthermore, all the decisions in a DRG_D , except D , are consequently subdecisions of D . In other words, the top-level decision *requires* these lower level subdecisions.

4. Running Example of a DMN Decision Model

Consider a patient health monitoring system for a person diagnosed with the Chronic Obstructive Pulmonary Disease (COPD). COPD is a disease that obstructs the lungs and obstructs the airflow and breathing of the patient. Acute attacks of the disease can happen. In that case the patient can experience uncomfortable complications such as fast breathing, a fast heart rate, hyperactive use of muscles, and a cold skin [50]. In scientific literature it has been recognised as well that a patient monitoring system can help increase the life quality of the patient and decrease the risks that are inherent to the disease [50]. Multiple sensors and wearable technologies exist that can collect patient data relevant for the patient monitoring process [50]:

- Electrocardiogram (ECG) sensors monitor the heart.
- Respiratory sensors check the breathing rate.
- Skin temperature sensors monitor the skin temperature.
- Muscular Electromyography (EMG) sensors monitor the muscle activity.

All these sensors collect measurements on the patient's health. Note that a single sensor or even a few sensors combined are not enough to capture the COPD. For instance, the patient might take a walk outside in the winter and a sensor registers a low skin temperature. In that case, the patient is not necessarily suffering from COPD at that moment. However, an expert can build patient-specific decision rules to capture COPD in such a monitoring system. For instance, if the sensors register a low skin temperature, a short and fast breathing rate, together with high blood pressure, the monitoring process might decide that the patient is suffering an attack and running out of oxygen.

A suitable way of modelling such decisions in complex environments is through the Decision Model and Notation (DMN) standard, since the standard provides maintainability and understandability of decisions [1, 51, 52]. Figure 2 gives an example decision requirements graph of a DMN model for COPD severeness based on data gathered by the sensors. Figure 3 provides the top-level COPD decision table.

5. Decision Model Change Patterns

To accommodate changes, designers should be able to evolve the decision models. A number of changes can occur in the decision model. The core elements of a decision model are depicted in Figure 2, i.e. the input data and the decision nodes within a DRD, connected via information requirements arrows. The logic encapsulated in a decision node is usually modelled with decision tables, such as shown in Figure 3. To determine the change patterns, we investigate the changes that can manifest themselves on core DMN elements, provided in the metamodel of the DMN specification [1], at different levels of granularity. An overview of the full DMN metamodel is provided in Figure 17 in the Appendix of this paper (Section 9), followed by more detailed meta

Figure 2: A DRD model for COPD severeness based on data from the sensors.

COPD Severeness					
severeness					
U	Input +				Output +
	respiration	skin temperature	heart rhythm	muscle activity	severeness
	string	string	string	string	string
1	"fast"	"cold"	"fast"	"normal", "hyper"	"severe"
2	"fast"	"cold"	"normal"	"hyper"	"severe"
3	"fast"	"normal"	"fast"	"hyper"	"severe"
4	"fast"	"cold"	"normal"	"normal"	"mild"
5	"fast"	"normal"	"fast"	"normal"	"mild"
6	"fast"	"normal"	"normal"	"normal", "hyper"	"mild"
7	"normal"	"normal", "cold"	"normal", "fast"	"normal", "hyper"	"none"

Figure 3: A decision table for COPD severeness.

COPD Severeness				
severeness				
U	Input +			Output +
	respiration	skin temperature	heart rhythm	severeness
	string	string	string	string
1	"fast"	"cold"	"normal", "fast"	"severe"
2	"fast"	"normal"	"fast"	"severe"
3	"fast"	"normal"	"normal"	"mild"
4	"normal"	"normal", "cold"	"normal", "fast"	"none"

Figure 4: Decision table with changed inputs.

models for the distinctive parts of DMN. First, we assess the change patterns within a single decision rule, i.e., changing the inputs and outcomes of a single rule. Next, we look at change patterns for a decision rule in its entirety, i.e., adding or deleting decision rules from a decision table. These change patterns all pertain to a single node of the DRD, i.e., a single decision table, according to the DMN decision table metamodel (see Figure 18). Furthermore, we investigate change patterns on the topological structure of the DRD itself, i.e., the addition and deletion of core DRD elements (decision nodes and data input nodes), as specified in the metamodels in Figures 19 and 20. The change patterns are derived from the formalisation of core decision model elements in Section 3 and the elementary edit operations that can be applied on the elements, i.e., insertion and deletion, as well as substitution, which in essence is a combination of insertion and deletion [43]. We illustrate each change pattern on the running example introduced in Section 4. Table 1 provides an overview of the change patterns directly relating to core DMN elements.

5.1. Decision table change patterns

For changing decision rules, we can distinguish a plethora of change patterns. We denote a change pattern with $\Delta\Pi$, Π refers to a pattern, while Δ stands for change, e.g., ΔI_d denotes a change in the input variables of a decision node. In essence, three elements in the decision table can undergo changes, as derived from Definition 3: the inputs I_d , the outputs O_d and the logic L mapping the inputs to the outputs, i.e. the decision rules. We define these changes in what follows.

5.1.1. Changes within decision rules

Changing inputs

$\Delta\Pi$ 1 (Decision input exclusion). A change ΔI_{d-} indicates a change in the input set I_d of a decision D as follows: an input variable, or an existing value of a variable, can be deleted from a decision table.

ϵ 1 (EXAMPLE). Suppose that in the top-level decision table represented in Figure 3, the input of *muscle activity* is considered irrelevant for the decision on COPD severeness. Hence, this input variable is entirely deleted from the decision table. Figure 4 presents the decision table after change pattern $\Delta\Pi$ 1 was applied. Note that this decision table has been refactored to avoid overlapping and missing rules, since simply deleting an input variable can render the decision table to be incomplete and incorrect [53].

$\Delta\Pi$ 2 (Decision input inclusion). A change ΔI_{d+} indicates a change in the input set I_d of a decision D as follows: opposite to $\Delta\Pi$ 1, an input variable, or a new value for an existing variable, can be added to a decision table.

ϵ 2 (EXAMPLE). Adding a new input variable results in the exact opposite changes as in the previous change pattern: if the input variable of *muscle activity* were to be added again to the decision table in Figure 4, that would result in the decision table in Figure 3.

Changing outcomes

$\Delta\Pi$ 3 (Decision output inclusion). A change ΔO_{d+} indicates a change in the output set O_d of a decision D as follows: a new output value can be added to a decision table.

ϵ 3 (EXAMPLE). Suppose that in the table presented in Figure 3, the COPD severeness decision outcome of the first decision rule changes from *severe* to *lethal*. This new table is given in Figure 5. Notice that the outcome value *lethal* is a new outcome that was not represented in the decision table before.

$\Delta\Pi$ 4 (Decision output exclusion). A change ΔO_{d-} indicates a change in the output set O_d of a decision D as follows: an existing output value can be deleted from a decision table.

ϵ 4 (EXAMPLE). Suppose that in the table presented in Figure 5, the COPD severeness decision outcome of the first decision rule changes from *lethal* to *severe*, i.e., the outcome of *lethal* is deleted from the decision table. The table in Figure 5 is consequently reverted to the situation in Figure 3.

Changing decision logic

$\Delta\Pi$ 5 (Decision rule logic change). A change ΔL indicates a change in the logic L of a decision D , i.e., a change in relating the existing input symbols I_d to the existing output symbols O_d within the decision table.

ϵ 5 (EXAMPLE). Suppose that in the table presented in Figure 3, the logic in the fourth decision rule changes. Instead of mapping the input values to an output of *mild*, the inputs are now considered to be of a *severe* nature. This change is exemplified in the decision table in Figure 6.

COPD Severeness					
severeness					
U	Input +				Output +
	respiration	skin temperature	heart rhythm	muscle activity	severeness
	string	string	string	string	string
1	"fast"	"cold"	"fast"	"normal", "hyper"	"lethal"
2	"fast"	"cold"	"normal"	"hyper"	"severe"
3	"fast"	"normal"	"fast"	"hyper"	"severe"
4	"fast"	"cold"	"normal"	"normal"	"mild"
5	"fast"	"normal"	"fast"	"normal"	"mild"
6	"fast"	"normal"	"normal"	"normal", "hyper"	"mild"
7	"normal"	"normal", "cold"	"normal", "fast"	"normal", "hyper"	"none"

Figure 5: Decision table with changed outcomes.

5.1.2. Changes on decision rules in their entirety

$\Delta\Pi$ 6 (Decision rule exclusion). *If a decision rule $i_d \in I_d \xrightarrow{L} o_d \in O_d$ is deemed irrelevant at a certain point in time, it can be deleted in its entirety from a decision table.*

ϵ 6 (EXAMPLE). Suppose that decision rule 7 is entirely deleted from the decision table in Figure 3, rendering the decision table depicted in Figure 7. Notice that the decision table does not anymore contain the output *none*, and that by deleting the decision rule, the decision table is not complete anymore, i.e., the combination of input values that was present in rule 7 in Figure 3 can perhaps still manifest itself in real life. However, the decision table is not able to return an outcome for this input combination. This could lead to a deadlock in the system. To avoid this, either the decision table should be completed and the input values at hand should be mapped to other existing decision outcomes, or the system should be redesigned to capture the possibility of no decision outcome being returned.

$\Delta\Pi$ 7 (Decision rule inclusion). *If a new decision rule $i_d \in I_d \xrightarrow{L} o_d \in O_d$ is deemed relevant at a certain point in time, it can be added in its entirety to an existing decision table.*

ϵ 7 (EXAMPLE). Suppose that the decision rule deleted in the previous change pattern is reintroduced again to the decision table. Thus, to the decision table in Figure 7, one rule is added, rendering the decision table in Figure 3.

5.2. DRD change patterns

This subsection deals with the deletion or addition of entire elements in the DRD model. Here, we focus on the core elements of a DRD model, i.e., decision nodes and input data nodes. By deleting the input data and decision nodes from the DRD, the information requirement arrows that connect them are deleted as well.

5.2.1. Decision node changes

$\Delta\Pi$ 8 (Decision node exclusion). *A decision node $D \in D_{dm}$ can be deleted from the DRD. This corresponds to deleting all decision rules ($I_d \xrightarrow{L} O_d$) from a decision node D . Hence, this change pattern is an aggregation of multiple $\Delta\Pi$ 6 changes. Note that deleting a decision node D also deletes all its incoming and outgoing information requirements arrows from the set IR .*

COPD Severeness					
severeness					
U	Input +				Output +
	respiration	skin temperature	heart rhythm	muscle activity	severeness
	string	string	string	string	string
1	"fast"	"cold"	"fast"	"normal","hyper"	"severe"
2	"fast"	"cold"	"normal"	"hyper"	"severe"
3	"fast"	"normal"	"fast"	"hyper"	"severe"
4	"fast"	"cold"	"normal"	"normal"	"severe"
5	"fast"	"normal"	"fast"	"normal"	"mild"
6	"fast"	"normal"	"normal"	"normal","hyper"	"mild"
7	"normal"	"normal","cold"	"normal","fast"	"normal","hyper"	"none"

Figure 6: Decision table with changed logic.

COPD Severeness					
severeness					
U	Input +				Output +
	respiration	skin temperature	heart rhythm	muscle activity	severeness
	string	string	string	string	string
1	"fast"	"cold"	"fast"	"normal", "hyper"	"severe"
2	"fast"	"cold"	"normal"	"hyper"	"severe"
3	"fast"	"normal"	"fast"	"hyper"	"severe"
4	"fast"	"cold"	"normal"	"normal"	"mild"
5	"fast"	"normal"	"fast"	"normal"	"mild"
6	"fast"	"normal"	"normal"	"normal", "hyper"	"mild"

Figure 7: Decision table with a deleted decision rule.

Figure 8: DRD adapted to changed inputs in the top-level decision.

ε8 (EXAMPLE). Suppose that the subdecision **Muscle activity** in the DRD in Figure 2 is excluded from the model. The updated decision model is then given in Figure 8.

ΔΠ 9 (Decision node inclusion). A new decision node D can be added to the set of decision nodes D_{dm} . This corresponds to adding a new decision table, and thus, adding multiple decision rules $(I_d \xrightarrow{L} O_d)$ encapsulated in the decision node D . Hence, this change pattern is in essence an aggregation of multiple $\Delta\Pi 7$ changes. Note that adding a decision node D also adds the necessary incoming and outgoing information requirements arrows to the set IR .

ε9 (EXAMPLE). Suppose that a decision node **Muscle activity** is added to the decision requirements diagram in Figure 8, effectively producing the DRD presented in Figure 2, as the *EMG data* input is added to the DRD as well.

5.2.2. Input data node changes

ΔΠ 10 (Input data node inclusion). A new data input node can be added to the set of data input nodes ID . By adding a data input node, its necessary input requirement arrows are also added to the set of IR and connected to the relevant decision nodes in D_{dm} . Notice that this change pattern on the DRD level corresponds to adding a new input variable to the decision table that requires the newly added data input node. Hence, this change pattern is equivalent to $\Delta\Pi 2$.

$\epsilon 10$ (EXAMPLE). Consider the decision model provided in Figure 8. Assume that an additional input data node is added that provides information on muscle activity. The new decision model is then given in Figure 9.

$\Delta \Pi 11$ (Input data node exclusion). A data input node can be deleted from the set of data input nodes ID . By deleting a data input node, all its input requirement arrows are also deleted from to the set of IR . Notice that this change pattern on the DRD level corresponds to deleting an input variable from the decision table that required the input data node. Hence, this change pattern is again equivalent to $\Delta \Pi 1$.

$\epsilon 11$ (EXAMPLE). Consider the decision model provided in Figure 9. Assume that the **Muscle activity** input data node is deleted from the requirements diagram since this information is now deemed invalid to make the top-level COPD **severeness** decision. The newly obtained DRD is then given in Figure 8.

5.3. Change patterns on non-core DMN elements

So far, we have enumerated change patterns that have a considerable impact on the core DMN elements according to the metamodel presented in the standard specification [1] (see Section 9). These are: the input data nodes, the decision nodes, and the underlying decision tables and their components. However, changes related to non-core elements of the model are possible as well. We give an overview of those changes. First, we enumerate the changes that are in essence the same as the core changes presented in this section, since the elements that undergo the changes are either composed out of the elements that have already been discussed, or they connect the previously discussed core elements.

- Business knowledge models (BKM) and associated knowledge requirements can be introduced in a DRD and, therefore, they can undergo changes as well. The encapsulated decision logic of business knowledge models is often represented in the form of decision tables as well (see the metamodel in Figure 21). Hence, the change patterns the BKMs can go through are analogous to those of the decision nodes and their related decision tables.
- Decision services are built up of core elements of the DRD, as shown in the metamodel in Figure 22. Hence, changing the decision services corresponds to changing the individual core

Figure 9: A DRD with an added input data node.

Table 1: Overview of decision model change patterns.

Decision table change patterns		
Changes within decision rules		
	$\Delta\Pi 1$	Decision input exclusion.
	$\Delta\Pi 2$	Decision input inclusion.
	$\Delta\Pi 3$	Decision output inclusion.
	$\Delta\Pi 4$	Decision output exclusion.
	$\Delta\Pi 5$	Decision rule logic change.
Changes on decision rules		
	$\Delta\Pi 6$	Decision rule exclusion.
	$\Delta\Pi 7$	Decision rule inclusion.
Decision requirements diagram change patterns		
Decision node changes		
	$\Delta\Pi 8$	Decision node exclusion.
	$\Delta\Pi 9$	Decision node inclusion.
Input data node changes		
	$\Delta\Pi 10$	Input data node inclusion.
	$\Delta\Pi 11$	Input data node exclusion.

elements within the DRD that are encapsulated in the decision services. As such, these changes have been addressed above. Note that according to the latest DMN standard specification [1], there is an element which can be used in the DRD to represent a decision service. This decision service element encapsulates a subset of the core elements of a DRD, much like a subprocess task in BPMN encapsulates a subset of logically related process elements of a master process.

- The requirement connections between DRD elements can change, i.e., the requirement arrows can be added or deleted. This has possible impact on both the begin and end nodes associated to the requirement arrows, as shown in the decision metamodel in Figure 19. If the requirement arrow is deleted and the begin node has no other requirements associated to it, it needs to be deleted as well. If a new requirement arrow is added, the relevant information passed through that arrow should be incorporated in the designated end node.

Next to these non-core DMN elements that are closely related to the core DMN elements, other DMN components exist as well, as specified in the metamodels in Section 9, and, therefore, they too can be subject to change.

- The data types of the input and/or outputs can be changed (see Figure 18).
- The names of all the elements within the decision model can be changed (see Figure 17).
- Knowledge sources and associated authority requirements can be added to the DRD, as shown in the metamodels in Figures 19 and 21, and they can undergo change as well. However, the knowledge source and authority requirements in DMN are mainly used for documentation purposes and do not affect the rest of the model.
- Text annotations and associations can be added, deleted, or changed to provide additional textual information on elements within the DRD, as depicted in the metamodel in Figure 23.

These are meant for documentation and clarification purposes and do not affect the decision model in any way if they are subject to change.

- The hit policies of decision tables can be altered as well, as shown in the decision table metamodel in Figure 18. This usually leads to refactoring and remodelling the decision table [53], however, it has no further influence on the decision model elements from the input side or the output side. When the hit policy of the decision table is changed, as the decision rules will often need to undergo reordering to fit the newly assigned hit policy. However, researchers have argued to stick with unique hit decision tables and to avoid hit policies such as first hit and priority hit in order to avoid ambiguity and overlapping rules as a design principle [14, 54].
- Note that refactoring and simplifying decision tables does, however, change how they represent logic. Modelling logic with decision tables has been discussed in previous works, in particular when it comes finding overlapping and missing rules in decision tables, and on refactoring the decision tables to remedy these issues [55, 56]. Furthermore, decision table simplification was researched as well [53, 57]. Here, the goal is to represent the decision logic in a decision table in an unambiguous and understandable way by for instance contracting the inputs of decision rules which lead to the same decision outcome. As such, the notions of decision table refactoring and simplification are well researched and they have led to tools that support the automatic refactoring and simplification of decision tables.

6. Change Propagation

Applying a single change pattern to a decision model can have repercussions across multiple elements of a DRD. More precisely, an applied change pattern can render the decision model inconsistent. To restore consistency other change patterns may need to be triggered. In what follows, we investigate how each change pattern should trigger a chain of change patterns within the decision model. For each change pattern, we state which inconsistencies the change may induce into the decision model. Next, we indicate which change patterns need to be triggered and propagated to resolve the inconsistencies. We provide an overview of change pattern propagation in Table 2.

6.1. $\Delta\Pi1$ -induced change pattern propagation

6.1.1. Induced inconsistencies

Deleting an input from a decision rule ($\Delta\Pi1$) corresponds to deleting the input requirements of that decision table. The input of a decision consists of its input data elements and/or the outcomes of its subdecisions. Hence, deleting the input of the decision table without deleting its predecessors that supply the inputs, renders the decision model inconsistent.

6.1.2. Resolving inconsistencies

Given the hierarchical topology of a DRD, $\Delta\Pi1$ implies changing the elements that are required by the decision whose inputs undergo changes:

- If the deleted input is obtained from a data input node, this manifests itself by deleting input data nodes ($\Delta\Pi11$).
- If the deleted input was obtained from another decision node, i.e., a subdecision, the propagated change patterns to the subdecisions can be manifold.

- An existing decision outcome in the subdecision table can be deleted ($\Delta\Pi4$). Note that this only applies if the subdecision outcome is not used as an input in another decision table. Otherwise, other decision tables that require the subdecision outcome are rendered inconsistent.
- Another option is to exclude a whole decision rule in the subdecision whose decision outcome feeds into the decision table with the changed input ($\Delta\Pi6$).
- If an entire input variable is deleted, rather than just an existing value of an existing input variable, the entire subdecision node that feeds its outcome into the higher-level decision where the input is deleted in the first place ($\Delta\Pi8$).

Thus, applying $\Delta\Pi1$ can on its turn trigger change patterns $\Delta\Pi4$, $\Delta\Pi6$, $\Delta\Pi8$, and $\Delta\Pi11$. Take for instance the deletion of the variable *muscle activity* from the inputs in the decision table in Figure 3, rendering decision table 4. This also affects the decision hierarchy in the DRD in Figure 2. The subdecision **Muscle activity**, together with its input data, i.e., *EMG data*, are not part of the top-level decision anymore. As such, they should be discarded from the decision requirements diagram. In this case, $\Delta\Pi1$ also triggers change patterns $\Delta\Pi8$ and $\Delta\Pi11$. The updated decision requirements model is provided in Figure 8.

6.2. $\Delta\Pi2$ -induced change pattern propagation

6.2.1. Induced inconsistencies

Adding an input to a decision rule ($\Delta\Pi2$) corresponds to changing the input requirements of that decision table. The input of a decision consists of its input data elements and/or the outcomes of its subdecisions. Hence, adding an input to the decision table without changing its predecessors that supply the inputs, renders the decision model inconsistent.

6.2.2. Resolving inconsistencies

Given the hierarchical topology of a DRD, $\Delta\Pi2$ implies changing the elements that are required by the decision whose inputs undergo changes:

- If the added input is obtained from a data input node, this manifests itself by adding input data nodes ($\Delta\Pi10$).
- If the added input is obtained from another decision node, i.e., a subdecision, the propagated change patterns to the subdecisions can be manifold.
 - A new decision outcome in the subdecision table can be added ($\Delta\Pi3$).
 - Another option is to include a whole new decision rule in the subdecision whose decision outcome feeds into the decision table with the changed input ($\Delta\Pi7$).
 - If a new input variable is introduced, rather than just a new value for existing input variables, a whole new decision node that feeds its outcome into the higher-level decision where the input is added can be included ($\Delta\Pi9$).

Thus, applying $\Delta\Pi2$ can on its turn trigger change patterns $\Delta\Pi3$, $\Delta\Pi7$, $\Delta\Pi9$, and $\Delta\Pi10$. Take for instance decision table 4 and suppose that the *muscle activity* variable is added again. The DRD should reintroduce the subdecision **Muscle activity** and its required *EMG data*. Thus, this change pattern version would return the models to the initial state as shown in Figures 2 and 3, by initiating change patterns $\Delta\Pi9$ and $\Delta\Pi10$ after $\Delta\Pi2$.

6.3. $\Delta\Pi3$ -induced change pattern propagation

6.3.1. Induced inconsistencies

Given the hierarchical topology of a DRD, adding a decision outcome to a decision rule ($\Delta\Pi2$) also affects the input requirements, i.e. the inputs, of the higher level decision table that requires the added decision outcome, assuming that the added decision outcome does not belong to the top-level decision node of the DRD. Simply changing the outcome of a decision without adapting the higher-level decision tables that require that outcome, leads to an inconsistent decision model.

6.3.2. Resolving inconsistencies

Applying $\Delta\Pi3$ can trigger change pattern $\Delta\Pi2$ in all higher-level decision tables that require the decisions affected by $\Delta\Pi3$. Note that newly added outcome values can be captured in higher-level decision tables by introducing new decision rules ($\Delta\Pi7$), or even new decision nodes ($\Delta\Pi9$) if an entirely new decision outcome variable is introduced. Take for instance the decision table represented in Figure 3, the COPD severeness decision outcome of the first decision rule changes from *severe* to *lethal*. This new table is given in Figure 5. Notice that in this case the DRD in Figure 2 does not undergo any change, since the changed decision outcome belongs to the top-level decision of the DRD. Hence, the outcome does not affect any decision constructs that are higher in the decision hierarchy.

6.4. $\Delta\Pi4$ -induced change pattern propagation

6.4.1. Induced inconsistencies

Given the hierarchical topology of a DRD, deleting a decision outcome to a decision rule ($\Delta\Pi4$) also affects the input requirements, i.e. the inputs, of the higher level decision table that requires the deleted decision outcome, assuming that the deleted decision outcome does not belong to the top-level decision node of the DRD. Here too, simply changing the outcome of a decision without adapting the higher-level decision tables that require that outcome, leads to an inconsistent decision model.

6.4.2. Resolving inconsistencies

Applying $\Delta\Pi4$ should trigger change pattern $\Delta\Pi1$ in all higher-level decision tables that require the decisions affected by $\Delta\Pi4$. Similar to the previous change pattern, deleted outcome values can be resolved by deleting decision rules that require the outcome values in higher-level decision tables ($\Delta\Pi6$), or deleting entire higher-level decision nodes if the outcome variable is deleted in its entirety ($\Delta\Pi8$). It is worthwhile to notice the exact opposite changes in the model occur as in the previous change pattern. If in the decision table of Figure 5, the *lethal* outcome were again to be replaced by the *severe* outcome, then the decision table would undergo redesign to revert back to the model presented in the running example, i.e., Figure 3.

6.5. $\Delta\Pi5$ -induced change pattern propagation

Since $\Delta\Pi5$ is only concerned with the logic within one decision node of the decision model, no additional change patterns are propagated as a result of applying $\Delta\Pi5$. This is due to the fact that the input and outcome sets of the decision table do not change, but merely the logic that maps the inputs onto the outcomes. The inputs and outcomes are linking elements for the information requirements that construct the decision hierarchy. Since this change pattern does not alter the inputs and outcomes, no change propagation to other elements in the decision model can occur.

6.6. $\Delta\Pi6$ -induced change pattern propagation

6.6.1. Induced inconsistencies

Excluding a decision rule ($\Delta\Pi6$) is in essence a combination of deleting inputs ($\Delta\Pi1$), deleting decision outcomes ($\Delta\Pi4$), and the logic that connects them ($\Delta\Pi5$). Hence, this change pattern can possibly induce the same model inconsistencies as patterns $\Delta\Pi1$ and $\Delta\Pi4$.

6.6.2. Resolving inconsistencies

Thus, applying $\Delta\Pi6$ can on its turn trigger change patterns $\Delta\Pi1$, $\Delta\Pi4$, $\Delta\Pi5$ and $\Delta\Pi6$ again. Notice that if no input variables or rule outcomes are deleted when $\Delta\Pi6$ occurs, then this pattern corresponds to $\Delta\Pi3$, i.e., a mere change in underlying logic, without any further change propagation.

When applying $\Delta\Pi6$ to decision rule 7 in the decision table in Figure 3, obtaining the decision table depicted in Figure 7, a unique decision output value, i.e., *none*, is deleted. However, this does not affect the rest of the decision model, since the affected decision table corresponds to the top-level decision. As such, these change patterns cannot propagate any patterns affecting the higher levels in the decision hierarchy. Furthermore, the lower-level elements in the decision hierarchy are not affected either, since the deleted input values are not unique, and no input variable was deleted in its entirety. Thus, in this case, $\Delta\Pi6$ only triggers a change in underlying logic ($\Delta\Pi5$) without affecting the hierarchy of the DRD. However, the decision table in Figure 7 is not longer complete and should undergo redesign according to [53] to capture all possible input combinations.

6.7. $\Delta\Pi7$ -induced change pattern propagation

6.7.1. Induced inconsistencies

Including a decision rule ($\Delta\Pi7$) is, in analogy with $\Delta\Pi6$ propagation, a combination of $\Delta\Pi2$, $\Delta\Pi3$, $\Delta\Pi5$ in the case of deleting existing values of existing input variables or outcomes. Hence, this change pattern can possibly trigger the same inconsistencies as patterns $\Delta\Pi2$ and $\Delta\Pi3$ when inputs or outcomes are added to a decision rule, assuming new values of existing variables are introduced.

6.7.2. Resolving inconsistencies

Thus, applying $\Delta\Pi7$ can on its turn trigger change patterns $\Delta\Pi2$, $\Delta\Pi3$, $\Delta\Pi5$ and $\Delta\Pi7$ again. Notice that if no unique values of input variables and rule outcomes are introduced when $\Delta\Pi7$ occurs, then this pattern corresponds to $\Delta\Pi5$, i.e., a mere change in underlying logic, without any further change propagation.

When applying $\Delta\Pi7$ by decision rule 7 in the decision table in Figure 7, obtaining the decision table depicted in Figure 3, a unique decision output, i.e., *none*, is added. As such the table is complete. However, in analogy with $\Delta\Pi6$, no other elements in the decision hierarchy undergo any changes in this example.

6.8. $\Delta\Pi8$ -induced change pattern propagation

6.8.1. Induced inconsistencies

Deleting an entire decision node ($\Delta\Pi8$) corresponds to applying $\Delta\Pi6$ on every decision rule contained within the decision table of that decision node, i.e. deleting the decision table related to that node. As such, the same chain of inconsistencies is established. All elements that are lower in the decision hierarchy and that are only connected to the deleted decision node, become obsolete as well. The elements that are higher in the decision hierarchy and that require the outcome of the deleted decision node, render the model inconsistent, since the required inputs can no longer be obtained due to the deletion of the lower-level decision node.

6.8.2. Resolving inconsistencies

All input nodes, i.e., input data nodes and subdecisions, that feed into the deleted decision node and are only connected to that decision node, need to be deleted, effectively triggering $\Delta\Pi8$ and $\Delta\Pi11$. This applies to decision model elements that are lower in the decision hierarchy than the deleted decision node. For elements that are higher in the decision hierarchy, change pattern $\Delta\Pi1$ should be triggered as well, i.e., whole input variables are deleted, since the required decision node that provides those variables is deleted as well. As such, the underlying decision logic of the decision model can be subject to changes as well $\Delta\Pi5$.

Suppose that the subdecision **Muscle activity** in the DRD in Figure 2 is excluded from the model. This also deletes the subdecision's input data, i.e., *EMG data* ($\Delta\Pi11$). The updated decision model is then given in Figure 8. Since the deleted decision node is connected to a higher-level decision node, i.e., **COPD severeness**, which requires the deleted decision node as input, inconsistencies arise in the decision model. Namely, the decision table in Figure 3 has a *muscle activity* input that can no longer be obtained due to the deletion of the **Muscle activity** decision node. As such, $\Delta\Pi8$ also triggers $\Delta\Pi1$ in this example, effectively transforming the decision table in Figure 3 into the decision table given in Figure 4 by deleting the input variable *muscle activity*. As such, the decision model is rendered consistent again.

6.9. $\Delta\Pi9$ -induced change pattern propagation

6.9.1. Induced inconsistencies

Inserting an entire decision node ($\Delta\Pi9$) corresponds to adding an entirely new decision table, i.e., applying $\Delta\Pi7$ on every decision rule contained within the decision table of that decision node. Hence, the same inconsistencies arise as in $\Delta\Pi7$.

6.9.2. Resolving inconsistencies

For all decision elements that require the inserted decision node, a new variable will need to be introduced in the inputs ($\Delta\Pi2$). For the inserted decision node itself, the relevant input requirements need to be provided, either by linking existing decision nodes and input data nodes to the inserted decision node, or by introducing new decision nodes and input data nodes ($\Delta\Pi9$ and $\Delta\Pi10$). This also affects the decision logic ($\Delta\Pi5$).

Suppose that the subdecision **Muscle activity** is added to the DRD in Figure 8, obtaining the DRD in Figure 2. This also adds the subdecision's input data, i.e., *EMG data* ($\Delta\Pi10$) to the DRD model in Figure 2. Furthermore, the decision table in Figure 4 corresponding to the **COPD severeness** decision needs to be altered as well to incorporate the addition of the new decision node. As such the *muscle activity* variable is added to reflect the information requirement between the **Muscle activity** subdecision and the **COPD severeness** decision ($\Delta\Pi2$). This way, the decision table in Figure 3 is obtained. As such, the consistency between the different decision model elements is restored.

6.10. $\Delta\Pi10$ -induced change pattern propagation

6.10.1. Induced inconsistencies

Including a new input data node ($\Delta\Pi10$) in a DRD renders the decision model inconsistent if the decision table to which the input data node is connected, is not updated as well to consider that input node.

6.10.2. Resolving inconsistencies

Including a new input data node ($\Delta\Pi10$) also corresponds to changing the inputs of all decision rules in all decision nodes that require the added input data node ($\Delta\Pi2$). This also changes the decision logic ($\Delta\Pi5$). Hence, all decision nodes that require the input data node need to introduce the variable concerning the newly added input data node to their inputs.

Consider the decision model provided in Figure 8 with its corresponding top-level decision table in Figure 4. Assume that an additional input data node is added that provides information on muscle activity. The new DRD model is then given in Figure 9. However the newly introduced *Muscle activity* node induces a change in the decision table in Figure 4 as well ($\Delta\Pi2$). More precisely, the inputs of that decision table need to change to incorporate the *muscle activity* variable, rendering the decision table in Figure 3. As such, the consistency between the decision model elements is restored.

6.11. $\Delta\Pi11$ -induced change pattern propagation

6.11.1. Induced inconsistencies

Excluding an input data node ($\Delta\Pi11$) from a DRD renders the decision model inconsistent if the decision table to which the input data node was connected, is not updated as well.

6.11.2. Resolving inconsistencies

Excluding an input data node ($\Delta\Pi11$) corresponds to changing the inputs of all decision rules in all decision nodes that require the deleted input data node ($\Delta\Pi1$). This also changes the decision logic ($\Delta\Pi5$). Hence, the variable derived from the deleted input data node is deleted from all the decision nodes that required the input data node. Since the input data nodes are by definition (see Definition 1) the lowest elements in the decision hierarchy, they can only propagate their changes to higher-level decision nodes. This holds both for change propagation induced by $\Delta\Pi11$, as well as $\Delta\Pi10$.

Consider the decision model provided in Figure 9 with its corresponding top-level decision table in Figure 3. Assume that the **Muscle activity** input data node is deleted. The new DRD model is then given in Figure 8. However, the deleted *Muscle activity* node should induce a change in the decision table in Figure 3 as well ($\Delta\Pi1$). More precisely, the inputs of that decision table need to change to incorporate the deletion of the *muscle activity* variable, rendering the decision table in Figure 4. As such, the consistency between the decision model elements is restored.

6.12. Overview of induced change pattern propagation

Notice that all the applied change patterns can also induce a change in the decision logic ($\Delta\Pi5$), i.e., the mapping between input and output elements. When a change pattern is applied to the decision model, it can trigger another change pattern. That triggered change pattern can in turn trigger yet another change pattern. Thus, a chain of triggered patterns may need to be propagated through the decision model until the system stabilises. Table 2 gives an overview of how this change pattern propagation chain can come into existence. Figure 10 provides a depiction of the change propagation graph. For instance, when pattern $\Delta\Pi1$ is applied, $\Delta\Pi8$ may need to be triggered. Change pattern $\Delta\Pi8$ can in turn trigger $\Delta\Pi11$ after which the propagation of changes stop and the system returns to a stable state. Note that, as indicated in Table 2, the change patterns can also induce a change in decision logic ($\Delta\Pi5$). However, we have excluded $\Delta\Pi5$ from the change propagation graph in order to render the graph more comprehensible.

Table 2: Change propagation: an overview of decision model change patterns that may need to be triggered in other elements of the decision model to restore consistency.

Applied $\Delta\Pi$	Propagated $\Delta\Pi$ possibly triggered by initial $\Delta\Pi$
$\Delta\Pi_1$	$\Delta\Pi_4, \Delta\Pi_5, \Delta\Pi_6, \Delta\Pi_8, \Delta\Pi_{11}$
$\Delta\Pi_2$	$\Delta\Pi_3, \Delta\Pi_5, \Delta\Pi_7, \Delta\Pi_9, \Delta\Pi_{10}$
$\Delta\Pi_3$	$\Delta\Pi_2, \Delta\Pi_5, \Delta\Pi_7, \Delta\Pi_9$
$\Delta\Pi_4$	$\Delta\Pi_1, \Delta\Pi_5, \Delta\Pi_6, \Delta\Pi_8$
$\Delta\Pi_5$	
$\Delta\Pi_6$	$\Delta\Pi_1, \Delta\Pi_4, \Delta\Pi_5, \Delta\Pi_6$
$\Delta\Pi_7$	$\Delta\Pi_2, \Delta\Pi_3, \Delta\Pi_5, \Delta\Pi_7$
$\Delta\Pi_8$	$\Delta\Pi_1, \Delta\Pi_5, \Delta\Pi_8, \Delta\Pi_{11}$
$\Delta\Pi_9$	$\Delta\Pi_2, \Delta\Pi_5, \Delta\Pi_9, \Delta\Pi_{10}$
$\Delta\Pi_{10}$	$\Delta\Pi_2, \Delta\Pi_5$
$\Delta\Pi_{11}$	$\Delta\Pi_1, \Delta\Pi_5$

Figure 10: Overview of change propagation: change patterns that can possibly follow each other. Note that $\Delta\Pi_5$ should be considered as a consequence of applying any change pattern, but it is not included in the graph in order to provide a more comprehensible overview.

Figure 11: The DRD view of the DMN verification tool.

7. A Modelling Environment Prototype for the Evolution of DMN Models

This section presents a modelling environment prototype which provides modelling tool support for the evolution of DMN decision models. The modelling environment is based on the open source Camunda¹ modeller which we have advanced with verification capabilities. The modeller can apply any of the change patterns discussed in this paper. After doing so, the modelling environment checks for consistency errors and displays error messages. The modelling environment can highlight the errors and suggests actions to remedy the inconsistencies. After the modeller selects an action, the modelling environment automatically performs it and checks for errors again. This way, the DMN can be evolved iteratively in a consistent way. In what follows, we explain the interface of the modelling environment and we show its verification capabilities with an example.

7.1. The modelling environment

Figures 11 and 12 show the interface of the DMN modelling environment, with numbers indicating the different parts of the tool, which are as follows:

1. Display of the DRD model.
2. Buttons to upload a DMN model into the environment, to download the model from the environment, to generate an empty DMN model, and to connect the modelling environment to an existing model repository.
3. DMN elements which can be selected by the modeller to add to the DMN model.

¹<https://camunda.com/download/modeler/>

Figure 12: The decision table view of the DMN verification tool.

- The modeller can switch between the different views of the DMN model, e.g., by selecting a decision, the tool switches from the DRD view in Figure 11 to the decision table view in Figure 12.
- Different verification mechanisms are offered. The modeller can select the verifiers which are relevant, i.e., the errors that the modelling environment should check for. Alternatively, instead of selecting a single verifier, all verifiers can be selected as to display all errors that the modelling environment discovers.
- The *Verify* button. By clicking this button the modelling environment checks for errors (specified in the previous point) in the current DMN model.
- The *Reverify* check box. By checking this box the modelling tool will perform a verification of the model every time the modeller performs one of the change patterns on the model. Unchecking this box deactivates this feature, as an experienced modeller may not need to check the consistency after every action performed. However, after applying a whole chain of changes, the modeller can again re-enable the verification feature of the tool.
- Drop-down menu to display the verification results. After the verification happens, the modeller can select which errors, i.e., which verification results to display. Alternatively, the modeller can opt to display all verification results.
- Feedback for the modeller that explains the errors. Here, the error messages as a result of the verification mechanism of the modelling environment are displayed. The error messages are explained as well, both in general terms, as well as in specific terms, i.e., indicating the names of the DMN elements that are affected.

Figure 13: Situation after applying decision node exclusion.

10. Action buttons to highlight or resolve the errors. Next to the specific error messages, automatic actions are suggested by the modelling environment that can resolve the errors. By clicking the *Show* button, the modelling environment highlights the errors in the DMN model in a red colour. By clicking any of the other suggested action buttons, the modelling environment automatically performs the selected action.
11. Display for the decision table view. Instead of the DRD model, a decision table that has been selected is displayed in the central view of the tool, as shown in Figure 12.
12. The *View DRD* button. By clicking this button the modeller can switch back from the decision table view to the DRD view in the modelling environment.

7.2. An example of change propagation as supported by the modelling environment

In this subsection we provide a short example to show how the DMN verification tool works. The DMN model presented in Figures 2 and 3 is uploaded into the modelling environment. Suppose that instead of assessing muscle activity through the *Muscle Activity* subdecision with data obtained from the EMG sensor, the physician wishes to investigate the patient and to manually enter the muscle activity classification. Hence, the *Muscle Activity* subdecision from Figure 2 is obsolete and the modeller decides to delete it (i.e., the modeller applies decision node exclusion). The modelling environment raises two errors: *Lonely Data Input*, i.e., an input data node which is not connected to any decision node, and *Missing Input*, i.e., an input column in the top-level *COPD Severeness* decision which does not have any input data node or subdecision node providing the required input. Figure 13 shows the situation after applying this change pattern and clicking the *Show* button for the unconnected input data node.

Figure 14: Situation after applying input data node exclusion (DRD view).

By clicking the *Remove* button next to the input data node error, the change pattern of input data node exclusion is automatically performed. The modelling environment then re-verifies the model to check for new inconsistencies (arising through the application of the change pattern), leaving only the missing input error relating to the top-level *COPD Severity* decision. Figure 14 highlights this error in the DRD view of the modelling environment, while Figure 15 highlights the error in the decision table view.

As shown in Figures 14 and 15, the modelling environment suggests a number of actions to remedy the issues surrounding the missing input in the top-level decision. Inputs can be secured in three different ways: adding a new data input (i.e., input data inclusion), adding a new input decision (i.e., decision node inclusion), or adding outputs to existing subdecision nodes (i.e. decision output inclusion). Alternatively, the situation can be remedied by deleting the input column which does not have the required inputs (i.e., decision input exclusion). After deciding that a physician will manually provide the information needed as input in the *COPD Severity* decision table, the action of adding new data input is selected and the modelling environment automatically introduces a matching input data node, yielding a consistent model as show in Figure 16.

8. Conclusion and Future Work

This paper investigates and defines possible decision model change patterns for the evolution of DMN decision models. Through a running example we have illustrated that the adaptation of DMN decision model can lead to within-model inconsistencies. In order to remedy these inconsistencies, additional changes may need to be propagated throughout the entire decision model, effectively triggering other change patterns. This way, consistency between the different decision model elements is upheld. The proposed change patterns illustrate how decision models can evolve

Figure 15: Situation after applying input data node exclusion (decision table view).

Figure 16: Stable situation after applying input data node inclusion.

and which inconsistency concerns rise up when evolving the DMN models. Furthermore, the change patterns can aid in developing and implementing flexible and dynamic decision management systems in order to move away from the static approaches present both in literature and in industry. We illustrate the feasibility of the approach by providing a proof-of-concept modelling environment prototype which provides modelling support and automation for change pattern administration on DMN decision models.

In future work, we will investigate how changing decision models impact other systems and models that rely on the logic encapsulated in the decision models. More specifically, we will examine how the proposed change patterns influence process and decision model consistency in an integrated process (BPMN) and decision (DMN) model environment. Changing the underlying decisions of a process can lead to a construction of a new process variant. Additionally, in the case of knowledge-intensive processes, instead of BPMN, change patterns can be considered for Case Management Model and Notation (CMMN) models, as they allow to model activities that can be performed in an unpredictable order by knowledge workers. Furthermore, it should be investigated whether DMN can be extended with a decision ontology to better cope with knowledge-intensive aspects that cannot adequately be represented in a decision table [58]. Moreover, flexible decision models are of particular interest to Internet-of-Things (IoT) process settings [59, 60], as IoT process are inherently subject to a dynamic and changeable environment.

9. Appendix

This appendix gives an overview of the DMN metamodel, as specified in the latest DMN standard specification [1].

Acknowledgements

This work was in part supported by the Deutsche Forschungsgemeinschaft (grant DE 1983/9-1).

References

- [1] OMG, Decision Model and Notation (DMN) 1.2 (2019).
- [2] F. Hasić, J. De Smedt, J. Vanthienen, Augmenting processes with decision intelligence: Principles for integrated modelling, *Decision Support Systems* 107 (2018) 1 – 12.
- [3] D. Bork, D. Karagiannis, B. Pittl, Systematic analysis and evaluation of visual conceptual modeling language notations, in: *2018 12th International Conference on Research Challenges in Information Science (RCIS)*, IEEE, 2018, pp. 1–11.
- [4] F. Hasić, J. Vanthienen, Complexity metrics for dmnn decision models, *Computer Standards & Interfaces* 65 (2019) 15 – 37. doi:<https://doi.org/10.1016/j.csi.2019.01.006>. URL <http://www.sciencedirect.com/science/article/pii/S0920548918303647>
- [5] K. Hinkelmann, A. Pierfranceschi, E. Laurenzi, The knowledge work designer-modelling process logic and business logic, in: *Modellierung (Workshops)*, 2016, pp. 135–140.
- [6] M. Deryck, F. Hasić, J. Vanthienen, J. Vennekens, A case-based inquiry into the decision model and notation (dmn) and the knowledge base (kb) paradigm, in: C. Benz Müller, F. Ricca, X. Parent, D. Roman (Eds.), *Rules and Reasoning*, Springer International Publishing, Cham, 2018, pp. 248–263.

- [7] D. Bork, R. Buchmann, D. Karagiannis, M. Lee, E.-T. Miron, An open platform for modeling method conceptualization: The omilab digital ecosystem, *Communications of the Association for Information Systems* 44 (2019) 673–697. doi:10.17705/1CAIS.04432.
URL <http://eprints.cs.univie.ac.at/5462/>
- [8] F. Hasić, J. Vanthienen, From decision knowledge to e-government expert systems: the case of income taxation for foreign artists in belgium, *Knowledge and Information Systems* doi:10.1007/s10115-019-01416-4.
URL <https://doi.org/10.1007/s10115-019-01416-4>
- [9] J. Campos, P. Richetti, F. A. Baião, F. M. Santoro, Discovering business rules in knowledge-intensive processes through decision mining: an experimental study, in: *International Conference on Business Process Management*, Springer, 2017, pp. 556–567.
- [10] F. M. Santoro, F. A. Baião, Knowledge-intensive process: A research framework, in: *International Conference on Business Process Management*, Springer, 2017, pp. 460–468.
- [11] C. Corea, P. Delfmann, A tool to monitor consistent decision-making in business process execution, *Proceedings of the Dissertation Award, Demonstration, and Industrial Track at BPM* (2018) 9–14.
- [12] K. Hinkelmann, Business process flexibility and decision-aware modeling-the knowledge work designer, in: *Domain-Specific Conceptual Modeling*, Springer, 2016, pp. 397–414.
- [13] S. Nagel, C. Corea, P. Delfmann, Effects of quantitative measures on understanding inconsistencies in business rules, in: *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.
- [14] K. Figl, J. Mendling, G. Tokdemir, J. Vanthienen, What we know and what we do not know about dmn, *Enterprise Modelling and Information Systems Architectures (EMISAJ)* 13 (2018) 2–1.
- [15] F. Hasić, J. De Smedt, J. Vanthienen, An illustration of five principles for integrated process and decision modelling (5pdm), Available at SSRN 3082752.
- [16] J. Hu, G. Aghakhani, F. Hasić, E. Serral, An evaluation framework for design-time context-adaptation of process modelling languages, in: *IFIP Working Conference on The Practice of Enterprise Modeling*, Springer, 2017, pp. 112–125.
- [17] F. Hasić, J. De Smedt, S. Vanden Broucke, E. Serral Asensio, Decision as a service (daas): A service-oriented architecture approach for decisions in processes, *IEEE Transactions on Services Computing* (2020) 1–14doi:10.1109/TSC.2020.2965516.
- [18] F. Hasić, E. Serral Asensio, Change patterns for decision model and notation (dmn) model evolution, in: *Proceedings of the 18th Belgium-Netherlands Software Evolution Workshop (BENEVOL)*, CEUR-WS.org, 2019.
- [19] A. Hallerbach, T. Bauer, M. Reichert, Configuration and management of process variants, in: *Handbook on Business Process Management* 1, Springer, 2010, pp. 237–255.
- [20] F. Hasić, J. De Smedt, J. Vanthienen, Developing a modelling and mining framework for integrated processes and decisions, in: C. Debruyne, H. Panetto, G. Weichhart, P. Bollen,

- I. Ciuciu, M.-E. Vidal, R. Meersman (Eds.), *On the Move to Meaningful Internet Systems. OTM 2017 Workshops*, Springer International Publishing, Cham, 2018, pp. 259–269.
- [21] F. Hasić, J. De Smedt, J. Vanthienen, Redesigning processes for decision-awareness: Strategies for integrated modelling, in: *International Conference on the Quality of Information and Communications Technology*, 2018.
 - [22] F. Hasić, L. Vanwijck, J. Vanthienen, Integrating processes, cases, and decisions for knowledge-intensive process modelling, in: *International Workshop on Practicing Open Enterprise Modeling*, CEUR, 2017.
 - [23] T. Biard, J.-P. Bourey, M. Bigand, J.-C. Bocquet, Modélisation des prises de décisions dans les processus métier grâce à dmn (decision model and notation), in: *12ème Congrès International de Génie Industriel 2017*, 2017.
 - [24] T. Biard, J.-P. Bourey, M. Bigand, Dmn (decision model and notation): De la modélisation à l’automatisation des décisions, in: *INFORSID 2017*, 2017.
 - [25] T. Biard, A. Le Mauff, M. Bigand, J.-P. Bourey, Separation of decision modeling from business process modeling using new decision model and notation(dmn) for automating operational decision-making, in: *Working Conference on Virtual Enterprises*, Springer, 2015, pp. 489–496.
 - [26] K. Batoulis, A. Baumgräß, N. Herzberg, M. Weske, Enabling dynamic decision making in business processes with dmn, in: *International Conference on Business Process Management*, Springer, 2015, pp. 418–431.
 - [27] J. De Smedt, F. Hasić, S. vanden Broucke, J. Vanthienen, Holistic discovery of decision models from process execution data, *Knowledge-Based Systems* 183 (2019) 104866. doi:<https://doi.org/10.1016/j.knosys.2019.104866>.
URL <http://www.sciencedirect.com/science/article/pii/S0950705119303405>
 - [28] K. Batoulis, M. Weske, Soundness of decision-aware business processes, Working Paper. URL goo.gl/nhtCZ7.
URL goo.gl/nhtCZ7
 - [29] K. Batoulis, S. Haarmann, M. Weske, Various notions of soundness for decision-aware business processes, in: *International Conference on Conceptual Modeling*, Springer, 2017, pp. 403–418.
 - [30] K. Hinkelmann, K. Kritikos, S. Kurjakovic, B. Lammel, R. Woitsch, A modelling environment for business process as a service, in: *International Conference on Advanced Information Systems Engineering*, Springer, 2016, pp. 181–192.
 - [31] B. von Halle, L. Goldberg, *The decision model: A business logic framework linking business and technology*, Taylor and Francis, LLC.
 - [32] B. von Halle, L. Goldberg, *The decision model# 2: Improving process models and the requirements process*, Knowledge Partners International, LLC.
 - [33] J. B. dos Santos França, J. M. Netto, J. do ES Carvalho, F. M. Santoro, F. A. Baião, M. Pimentel, Kipo: the knowledge-intensive process ontology, *Software & Systems Modeling* 14 (3) (2015) 1127–1157.

- [34] F. Santoro, T. Slaats, T. T. Hildebrandt, F. Baiao, Dcr-kipn a hybrid modeling approach for knowledge-intensive processes, in: *International Conference on Conceptual Modeling*, Springer, 2019, pp. 153–161.
- [35] Z. Bellahsene, Schema evolution in data warehouses, *Knowledge and Information Systems* 4 (3) (2002) 283–304.
- [36] N. F. Noy, M. Klein, Ontology evolution: Not the same as schema evolution, *Knowledge and information systems* 6 (4) (2004) 428–440.
- [37] W. M. W. Kadir, P. Loucopoulos, Linking and propagating business rule changes to is design, in: *Information Systems Development*, Springer, 2005, pp. 253–264.
- [38] J. Boyer, H. Mili, Agile business rule development, in: *Agile Business Rule Development*, Springer, 2011, pp. 49–71.
- [39] F. Corradini, A. Polzonetti, O. Riganelli, Business rules in e-government applications, *arXiv preprint arXiv:1802.08484*.
- [40] M. Szvetits, U. Zdun, Systematic literature review of the objectives, techniques, kinds, and architectures of models at runtime, *Software & Systems Modeling* 15 (1) (2016) 31–69.
- [41] G. Blair, N. Bencomo, R. B. France, Models@ run. time, *Computer* 42 (10) (2009) 22–27.
- [42] S. Loukil, S. Kallel, M. Jmaiel, An approach based on runtime models for developing dynamically adaptive systems, *Future Generation Computer Systems* 68 (2017) 365–375.
- [43] A. Wombacher, M. Rozie, Evaluation of workflow similarity measures in service discovery., *Service Oriented Electronic Commerce* 80 (2006) 51–71.
- [44] C. Calero, M. Piattini, M. Genero, Empirical validation of referential integrity metrics, *Information and Software Technology* 43 (15) (2001) 949 – 957. doi:[https://doi.org/10.1016/S0950-5849\(01\)00202-6](https://doi.org/10.1016/S0950-5849(01)00202-6).
URL <http://www.sciencedirect.com/science/article/pii/S0950584901002026>
- [45] C. Ordonez, J. Garca-Garca, Referential integrity quality metrics, *Decision Support Systems* 44 (2) (2008) 495 – 508. doi:<https://doi.org/10.1016/j.dss.2007.06.004>.
URL <http://www.sciencedirect.com/science/article/pii/S0167923607000887>
- [46] K. Smit, M. Zoet, M. Berkhout, Verification capabilities for business rules management in the dutch governmental context, in: *2017 International Conference on Research and Innovation in Information Systems (ICRIIS)*, IEEE, 2017, pp. 1–6.
- [47] K. Batoulis, A. Nesterenko, G. Repitsch, M. Weske, Decision management in the insurance industry: Standards and tools., in: *BPM (Industry Track)*, 2017, pp. 52–63.
- [48] E. Bazhenova, F. Zerbato, B. Oliboni, M. Weske, From bpmn process models to dmn decision models, *Information Systems* 83 (2019) 69–88.
- [49] OMG, Business process model and notation (BPMN) 2.0 (2011).
- [50] M. Hassanalieragh, A. Page, T. Soyata, G. Sharma, M. Aktas, G. Mateos, B. Kantarci, S. Andreescu, Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges, in: *2015 IEEE international conference on services computing (SCC)*, IEEE, 2015, pp. 285–292.

- [51] F. E. A. Horita, D. Link, J. P. de Albuquerque, B. Hellingrath, odmn: an integrated model to connect decision-making needs to emerging data sources in disaster management, in: System Sciences (HICSS), 2016 49th Hawaii International Conference on, IEEE, 2016, pp. 2882–2891.
- [52] F. E. Horita, J. P. de Albuquerque, V. Marchezini, E. M. Mendonça, Bridging the gap between decision-making and emerging big data sources: an application of a model-based framework to disaster management in brazil, *Decision Support Systems* 97 (2017) 12–22.
- [53] D. Calvanese, M. Dumas, Ü. Laurson, F. M. Maggi, M. Montali, I. Teinemaa, Semantics, analysis and simplification of dmn decision tables, *Information Systems* 78 (2018) 112–125.
- [54] G. Wets, J. Vanthienen, H. Timmermans, Modelling decision tables from data, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 1998, pp. 412–413.
- [55] Ü. Laurson, F. M. Maggi, A tool for the analysis of dmn decision tables., in: BPM (Demos), 2016, pp. 56–60.
- [56] K. Batoulis, M. Weske, A tool for checking soundness of decision-aware business processes., in: BPM (Demos), 2017, pp. 1–5.
- [57] C. Corea, J. Blatt, P. Delfmann, A tool for decision logic verification in dmn decision tables, in: Proceedings of the Dissertation Award, Doctoral Consortium, and Demonstration Track at BPM 2019 co-located with 17th International Conference on Business Process Management, BPM 2019, Vienna, Austria, September 1-6, 2019, 2019.
- [58] D. Calvanese, M. Dumas, F. M. Maggi, M. Montali, Semantic dmn: Formalizing decision models with domain knowledge, in: International Joint Conference on Rules and Reasoning, Springer, 2017, pp. 70–86.
- [59] F. Hasić, E. Serral Asensio, Executing IoT processes in BPMN 2.0: Current support and remaining challenges, *IEEE RCIS 2019 proceedings*.
- [60] C. Janiesch, A. Koschmider, M. Mecella, B. Weber, A. Burattin, C. D. Ciccio, A. Gal, U. Kannegiesser, F. Mannhardt, J. Mendling, A. Oberweis, M. Reichert, S. Rinderle-Ma, W. Song, J. Su, V. Torres, M. Weidlich, M. Weske, L. Zhang, The internet-of-things meets business process management: Mutual benefits and challenges, *CoRR abs/1709.03628*. [arXiv:1709.03628](https://arxiv.org/abs/1709.03628). URL <http://arxiv.org/abs/1709.03628>

Figure 17: Overview of the DMN meta model.

Figure 18: Decision table meta model.

Figure 19: Decision meta model.

Figure 20: Input data meta model.

Figure 21: Business knowledge model meta model.

Figure 22: Decision service meta model.

Figure 23: Definitions meta model.