Separation of Decision Modeling from Business Process Modeling Using New "Decision Model and Notation" (DMN) for Automating Operational Decision-Making

Thierry Biard, Alexandre Le Mauff, Michel Bigand, Jean-Pierre Bourey

HAL Id: hal-01224703

https://hal.science/hal-01224703

Submitted on 9 Nov 2015

Separation of Decision Modeling
from Business Process Modeling
using new "Decision Model and Notation" (DMN)
for automating operational Decision-making

Thierry Biard[1], Alexandre Le Mauff[2], Michel Bigand[2], Jean-Pierre Bourey[2]

[1] Université Paris-Saclay - Laboratoire Génie Industriel - CentraleSupélec
Grande Voie des Vignes 92290 Châtenay Malabry - France
thierry.biard@centralesupelec.fr

[2] Université Lille Nord de France - Ecole Centrale de Lille
Cité Scientifique - CS 20048 - 59651 Villeneuve d'Ascq cedex - France
alexandre.lemauff@centraliens-lille.org,
michel.bigand@ec-lille.fr, jean-pierre.bourey@ec-lille.fr

Abstract. This paper presents Decision-making in Collaborative Networks and enlarges the differences between strategic-tactical and operational decisions. The common way of representing decisions into classical BPMN diagrams is shown, with its drawbacks. The new OMG's standard DMN (Decision Model and Notation) is introduced, with its context and its main elements. Then, the association between BPMN and DMN is detailed and the advantage of their separation is illustrated into examples. At last, the related experiments (already done and future works) are demonstrated: a graphical editor for modeling and an automatic code generator from a Model-To-Text transformation.

Keywords: DMN, decision model notation, BPMN, business process management, Model to Text transformation

1 Decision-making in Collaborative Networks

Decision-making is an important concern for Collaborative Networks (CN). Several papers already published address strategic-tactical decision-making, as Join / Leave / Remain a CN [1]. A few papers address operational decision-making as Demand and Capacity Sharing [2]. This paper is about a contribution to operational decision-making, directly linked to business processes.

The environment, the scope and the impact of decision-making are different depending on whether the decisions are strategic-tactical or operational. The predefined environment of operational decision usually leads to integrate them into business processes. Once it is done, due to their high frequency, there is an interesting opportunity to automate the operational decision-making. The main differences between strategic-tactical and operation decisions are listed into table 1.

Type of decision	Strategic-Tactical	Operational
Environment	Uncertain	Predefined
Scope	Global (to Local)	Local (to Global)
Term impact	Long	Short
Process oriented	Low	High
Decision-making	Human with Decision Support System	(To be) Automated
Frequency	Low	High
Period	Years-Months	Real Time
Decision example	Join / Leave / Remain	Capacity Sharing
Object example	CN Entity	Sales Order

Table 1: Strategic-Tactical versus Operational Decisions (from the authors)

2   Decision modeling inside business process diagrams

In recent years, BPMN (Business Process Model and Notation [3]; version 1.0 was published in 2003) has become the adequate notation for modeling business processes. BPMN Orchestration Diagram has even supplanted UML (Unified Modeling Language [4]) Activity Diagram for describing business processes, because BPMN provides modeling elements that are missing in UML and that are useful at business level, such as inclusive gateways and compensations. BPMN Orchestration Diagram is supposed to be simple and understandable by many people, among them the business stakeholders.

In fact, especially with all the features proposed by the version 2.0.2, BPMN Orchestration Diagram can become complex. The possibilities offered by its gateways for drawing multi-criteria decisions can lead to represent a business process model as a labyrinth. Indeed the result is that decisions are often drowned into these complex BPMN orchestration diagrams. One main consequence is the coupling increase between the decision elements with the process model: a slight modification on decision rules may have a strong impact on the whole process model. Here is a simple example of such Orchestration Diagram:
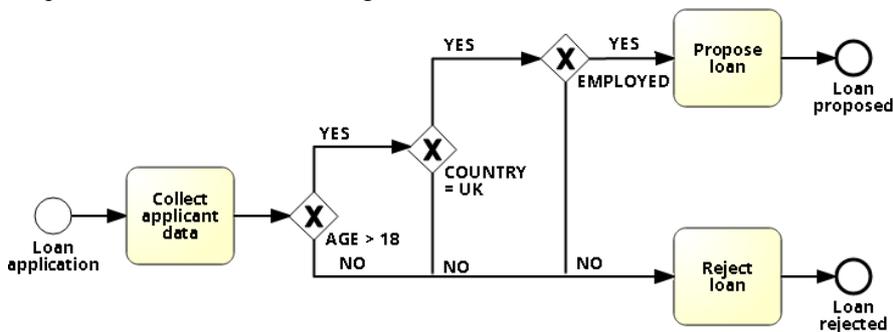
Figure 1: Classical BPMN Orchestration Diagram with cascading gateways

3 "Decision Model and Notation" (DMN)

3.1 New standard proposed by OMG

It exists now a solution for simplifying these complicated BPMN Orchestration Diagrams with multi-criteria decision. This solution is demonstrated in this paper.

This innovation was proposed last year by the OMG (Object Management Group): its "Decision Model and Notation" (DMN) specification [5] proposes a new way for modeling decisions, which can be now extracted for business process BPMN models.

This new standard DMN is not isolated. On the contrary, it can rely on the vocabulary defined in SBVR (Semantics of Business Vocabulary and Business Rules [6]) and the objectives defined in BMM (Business Motivation Model [7]), others OMG's standards. However, DMN is above all interfaced with BPMN 2.0.2. Only the strong association between DMN and BPMN is detailed further in this paper.

DMN is supported by software companies, but also by KU Leuven University. Note that, although the final version 1.0 of DMN specification was approved by OMG's board in December 2014, it is the beta version 1.0, published in February 2014, which is still available en OMG's web site.

3.2 DMN diagram elements

A DMN model, aka Decision Requirements diagram, can be drawn using four graphical elements: Input Data, Business Knowledge, Decision and Knowledge Source (this last one is optional and is used for quoting external references). The other ones are mandatory.
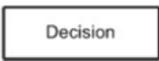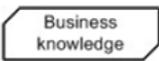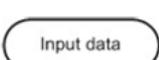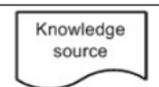
Component		Description	Notation
Elements	Decision	A decision denotes the act of determining an output from a number of inputs, using decision logic which may reference one or more business knowledge models.	Decision
	Business Knowledge Model	A business knowledge model denotes a function encapsulating business knowledge, e.g. as business rules, a decision table, or an analytic model.	Business knowledge
	Input Data	An input data element denotes information used as an input by one or more decisions. When enclosed within a knowledge model, it denotes the parameters to the knowledge model.	Input data
	Knowledge Source	A knowledge source denotes an authority for a business knowledge model or decision.	Knowledge source

Figure 2: Decision Requirements Diagram elements (from OMG specification)

DMN is supposed to be understandable by stakeholders, as business users. The fact is that its graphical representation is quite simpler than BPMN one, because DMN has neither gateways nor events, for instance:
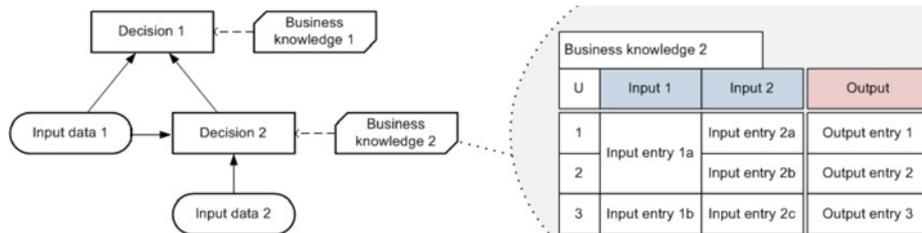

Figure 3: A simple Decision Requirements Diagram (from OMG specification)

You will notice that the Decision is the Output Data (aka the chosen option), which is determined from Input Data, potentially other sub-decisions, and pre-defined business logic described into Business Knowledge, using preferably Decision Tables, as shown on the right side of above figure 2.

3.3 BPMN + DMN association principle

Business Processes and Decisions can now be modelled separately, using BPMN and DMN respectively [8]. The multiple gateways into the BPMN Orchestration Diagram, often nested in cascade, are now replaced by a unique task as "Make a decision" that returns the result. It is a Business Rule Task, with a small Decision Table symbol on its top left corner:

Figure 4: Business Rule Task (BPMN)

3.4 BPMN + DMN = Separation of Concerns

The main advantage of having two different BPMN + DMN models, decoupling Decision-making from Processes, is the Separation of Concerns (SoC). SoC is an old (1974) best practice coming from computer science [9].

These loosely coupled models can evolve independently from each other and can even be supported by different stakeholders, according to their needs and skills (IT people and business analysts for instance). Each model (Business Process or Decision) can consider the other one as a black box, exchanging data against decision.

These models respect the main required property for a good Separation of Concerns: they have their own consistency; to understand one model, it is not necessary to know the other one. Moreover, Decision-making context can be explicitly detailed into the DMN diagram. Sub-decisions can be reused into several decisions too. This capitalization can lead to Knowledge Management (KM).

3.5 BPM + DMN separation applied to Collaborative Networks

Here is a couple of examples (willingly simplified), first a classical BPMN Orchestration Diagram without DMN, then a new BPMN Orchestration Diagram with DMN Decision Requirements Diagram, associated together via a Business Rule task:
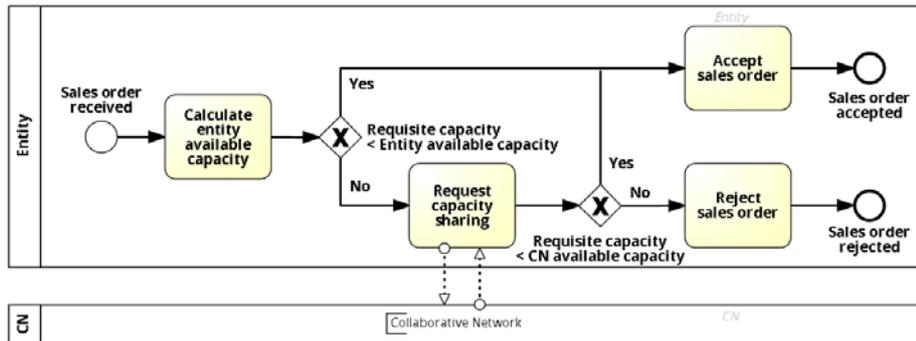

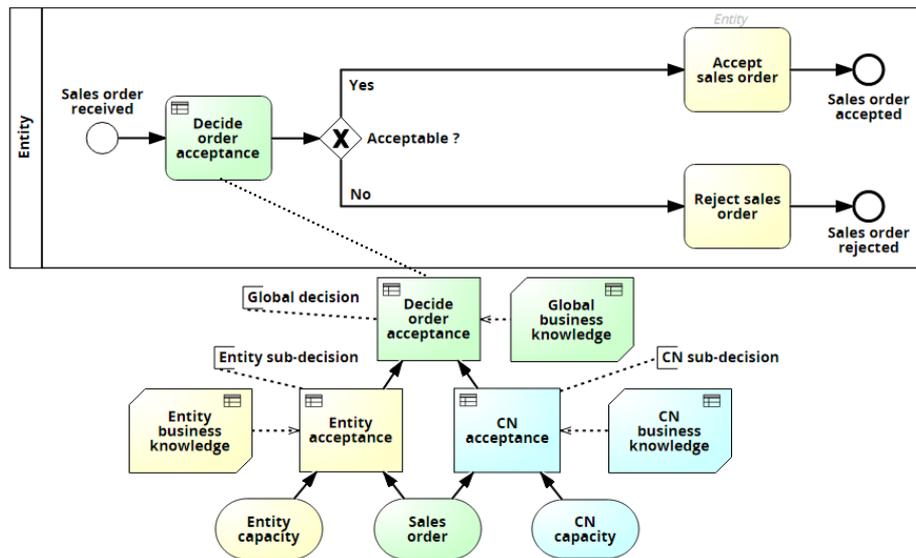**Figure 5:** Classical BPMN Orchestration Diagram without DMN

Figure 6: New BPMN Orchestration Diagram
with DMN Decision Requirements Diagram

Besides reorganizing the diagrams, this implementation of Separation of Concerns seems to be interesting for Collaborative Networks (CN): a main decision could be split into several sub-decisions made by each entity and made by the CN as a whole.

4   DMN related experiments

4.1 Our reached goals: graphical editor and automatic code generation

The OMG specification of DMN provides both a metamodel of DMN and an expression language called FEEL (Friendly Enough Expression Language) for defining and assembling decision tables, calculations, etc. From this specification, we had a couple of reached short-term goals:

1- To create our own graphical editor for drawing Decision Models (the first two letters of DMN), compliant with this metamodel. When we started one year ago, only one tool was available [10]. They are ten now (e.g. [11]), which certainly demonstrates the interest of software editors for this new notation, but some of these tools have only drawing capabilities and they are not free of charge.

2- To generate automatically, from these decision models, the decision code expressed with FEEL by using a model driven engineering approach and more precisely Model-to-Text transformation technique. In DMN, FEEL is the language for Notation (the third letter of DMN). FEEL stands for Friendly Enough Expression Language. Our approach is to consider FEEL as a DSL (Domain-Specific Language), where the domain studied is that of Decision.

4.2 Our targeted goal: bring DMN to life for automated Decision-making

Our targeted long-term goal is to provide a complete computer-aided method going from decision-making models to automatically generated FEEL code, interpreted or executed by a BRMS (Business Rules Management System), as open source Drools [12]. This BRMS will be interfaced with a business process engine, on which the corresponding BPMN business process model will be executed.

This approach is fully aligned with the Model-Driven Architecture (MDA [13]), which clearly specifies a system independently from the platform that supports it, with primary goals of portability, interoperability and reusability. With its Decision Requirements Diagram for modeling and its FEEL language for notation, DMN covers the two MDA upper levels CIM and PIM as shown in table 2. DMN can also represent Decision Tables graphically (at CIM level) and logically (at PIM level).

MDA levels	Decision-making
CIM (Computation Independent Model)	DMN (Decision Requirements Diagram)
PIM (Platform Independent Model)	DMN (FEEL considered as a DSL)
PSM (Platform Specific Model)	e.g. DRL (Drools Rule Language)

Table 2: Model-Driven Architecture (MDA) applied to Decision-making

4.3 DMN implementation based on Eclipse tools

We chose best-of-breed modules for Eclipse Modeling Tools dedicated to model driven engineering:

- Sirius [14] to create our graphical DMN modeling tool (without programming but with advanced setting),

- Acceleo [15] to generate FEEL code by using a Model-to-Text transformation (with programming and advanced setting too).

Below is a couple of screen captures about what we have done with theses Eclipse modules. DMN graphic elements special shapes are for the moment replaced by colours: it is a temporary solution without Java programming.
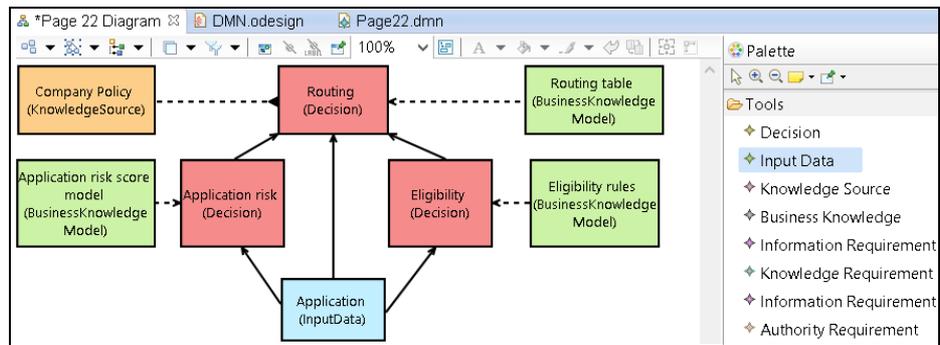

Figure 7: Graphical DMN modeling tool with Sirius module

This is an example of FEEL code generation from the above DMN model:
on the left, there is M2T (Model-To-Text Transformation Language) code **[16]**;
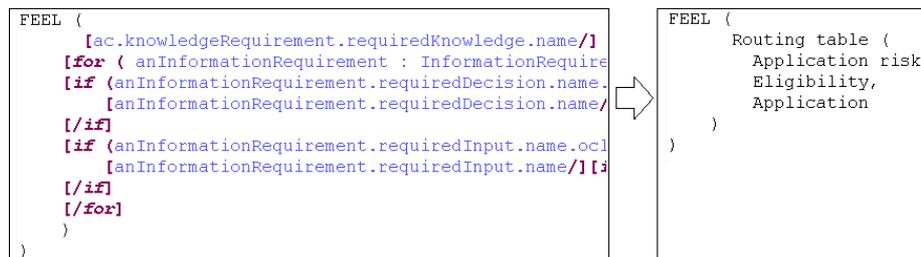on the right, the FEEL code automatically generated from the above DMN model.

Figure 8: Model-To-Text (FEEL) transformation with Acceleo module

5   Conclusion and future works

The separation of Decision Modeling from Business Process Modeling is a good principle, whatever the models, notations and languages are. It will enhance agility when changes are required, in reducing their impact, among them the risk of failure, and in increasing the resilience of the Information System (IS).

Our contribution is to present DMN as a relevant standard for Decision-making. However, DMN cannot deal with uncertainty and is limited to pre-defined decisions made from known criteria. The emphasis is now on operational Decision-making.

DMN makes it possible to represent Decision Models with sub-decisions, business knowledge, input data and knowledge sources distributed over various CN entities. Therefore, DMN can model decentralized and collaborative operational Decisions, based on standard and international specification. The decision models, shared on-line by all entities, can provide the transparency required by Collaborative Networks.

DMN is already usable for modeling these decisions, in separating them from business processes and applying the principle of separation of concerns. In a few months, due to Model-To-Text transformation, its FEEL code will be runnable and it will be possible to automate operational Decision-making, in a fast, reliable and repeatable way from business processes already modelled.

References

1. Chituc, C.-M., Nof, S.Y.: The Join/Leave/Remain (JLR) decision in collaborative networked organizations. Comput. Ind. Eng. (2007).
2. Yoon, S.W., Nof, S.Y.: Demand and capacity sharing decisions and protocols in a collaborative network of enterprises. Decis. Support Syst. 49, (2010).
3. OMG: Business Process Model and Notation (BPMN), http://www.omg.org/spec/BPMN/.
4. OMG: Unified Modeling Language (UML), http://www.omg.org/spec/UML/.
5. OMG: Decision Model and Notation (DMN), http://www.omg.org/spec/DMN/.
6. OMG: Semantics of Business Vocabulary and Rules (SBVR), http://www.omg.org/spec/SBVR/.
7. OMG: Business Motivation Model (BMM), http://www.omg.org/spec/BMM/.
8. Debevoise, T., Taylor, J.: The MicroGuide to Process and Decision Modeling in BPMN/DMN: [Building More Effective Processes by Integrating Process Modeling with Decision Modeling]. (2014).
9. Dijkstra, E.W.: On the role of scientific thought, http://www.cs.utexas.edu/users/EWD/transcriptions/EWD04xx/EWD447.html.
10. Decision Management Solutions: DecisionsFirst Modeler, http://decisionsfirst.com/.
11. Signavio: Decision Manager, http://www.signavio.com/products/decision-manager/.
12. Red Hat: Drools, Business Rules Management System, http://drools.org/.
13. OMG: MDA (Model Driven Architecture) Specifications, http://www.omg.org/mda/specs.htm.
14. Eclipse: Sirius, http://www.eclipse.org/sirius/.
15. Eclipse: Acceleo, http://www.eclipse.org/acceleo/.
16. OMG: Model To Text Transformation Language (M2T), http://www.omg.org/spec/MOFM2T/.