Emerging Standards in Decision Modeling—an Introduction to Decision Model & Notation

James Taylor, Decision Management Solutions;
Alan Fish, FICO; Jan Vanthienen, KU Leuven;
Paul Vincent, TIBCO

1. Introduction

The BPM market has expanded and matured in recent years, driven in part by the growing acceptance and broad use of process standards and common modeling notations. As companies transition to intelligent BPM, however, there is a need to focus on decision-making as well as process execution and workflow. Decision-making is important in intelligent processes, making them simpler and more agile as well as increasing the rate of straight through processing. However existing standards and notations do not readily support the modeling and specification of decision making. To address this need a new standard is being developed at the OMG, the *Decision Model and Notation* (DMN) standard.

The primary goal of DMN is to provide a common notation that is readily understandable by all business users, from the business analysts needing to create initial decision requirements and then more detailed decision models, to the technical developers responsible for automating the decisions in processes, and finally, to the business people who will manage and monitor those decisions. DMN creates a standardized bridge for the gap between the business decision design and decision implementation. As many analysts designing and building business process models are also referring to or designing decisions, DMN notation is designed to be useable alongside the standard BPMN business process notation.

In this paper four members of the submission team describe the importance and scope of decisions in intelligent BPM, introduce the basics of decision requirements modeling, discuss modeling decision logic in Decision Tables and provide an overall context for decisions in BPM more generally.

2. The Importance of Decisions in Intelligent BPM

A focus on decisions delivers on three critical elements of intelligent BPM—increased agility and capacity for business-led change; dramatic increases in Straight Through Processing / numbers of totally automated processes; and the ability to extract and operationalize value from Big Data analytics.

Increased Business Agility

Simpler and Therefore More Agile Processes

Making decisions explicit and managing them in concert with processes ensures an effective separation of concerns and a more streamlined design. Specifically, combining process management and decisioning decreases process complexity.

Most, if not all, business processes involve decisions: claims must be approved or rejected, cross-sell offers must be selected, and product discounts must be calculated. Especially when a process must handle multiple scenarios, modeling the decision-making as a process using branches and steps can become very complex. Replacing such a nest of branches and steps with a single, explicit and reusable decision point clarifies the behavior of the process, makes it easier to see when the process or decision needs to change, and allows updates to the decision-making to be independent from process change. Rather than handling all of the different subtypes of a particular transaction with branches and exception handling, a process can decide on the appropriate scenario or process state and essentially “assemble” the best process from pre-defined process tasks resulting in a much simpler, yet more flexible process design.

Once decision-making is removed from the process model it can be modeled separately as described below. With the decision-making modeled in an explicit decision requirements model it can be expressed more clearly than when process modeling is distorted to handle decision-making. Simpler processes that can be more readily changed and updated, more agile processes, are the result.

Increased Business Agility through Decoupled Lifecycles

Organizations cannot change more quickly than their business processes—when business processes become difficult to change organizations cannot then react quickly or effectively to new opportunities, new regulations or new challenges. To be responsive to change, organizations need to keep their key business parameters visible, understandable and changeable. Flexible processes cannot maintain these critical parameters when buried in software code or company manuals where the business has zero visibility into their behavior. By explicitly identifying decisions and describing the logic behind them, this business logic can be parameterized and managed separately from the process itself, dramatically increasing the business agility of an organization.

Business users like Business Process Management software because it allows them to change their workflow easily—it increases the agility of the process. Separating the decisions from the process further increases this capability as business changes often involve updates to business decisions: to pricing, eligibility or risk assessment decisions, for example. Such decisions are often the most dynamic part of a process, the part that changes most often. For instance, a company’s pricing rules are likely to change far more often that its order-to-cash process. If business users can only change the process, then they will not be able to respond to the far more numerous pricing changes without changing the process, an unnecessary step. Separately modeling decisions allows business users to control processes **and** the critical decisions within them. This increases the capacity for change built into a process and allows for a stable process even when decision-making is constantly changing and evolving.

Improved Business / IT Alignment

Different groups care about a process itself and the decision-making in a process. Organizations like risk management groups or compliance groups often care about specific decisions within a process, less so about the process itself. For instance, a credit risk group will care about the way credit is assigned, how the decision to grant credit is made. It is less likely to care about the process that wraps around this decision. By separately modeling and managing this decision an organization can focus each group on the piece of the puzzle that matters to them and so improve alignment.

Furthermore, business owners must be able to effectively collaborate with their IT department to define and manage the behavior of their business processes. Separately modeled decisions and processes are simpler and easier for business owners to understand and manage. Modeling the business process **and** the decision using a graphical notation allows the widest possible business audience to effectively participate, further improving alignment.

Increased Straight Through Processing

One of the key use cases for decision modeling is to enable accurate automated decision-making. The move to real-time interactions with customers, increasing demands for mobile and self-service access as well as a desire to more explicitly manage recommendations and decisions are all driving increases in automated decision-making. Automated decision-making means that more automated processes, processes that run straight through become possible. Without such automated decision-making every process must stop each time a decision is required so that a human can make the decision.

To deliver this automation organizations must fully understanding their decisions and it is hard to do this without extricating it from the process and explicitly modeling it. A combination of explicit decisions and process management keeps transactions moving with only exceptions ending up on work-lists or in an inbox. With human experts expensive and hard to scale, capturing the know-how of experts in explicit decision logic and making it available everywhere focuses scarce expert resources on exceptions and high-value cases and customers. Staff can then focus on value-add activities that require their expertise, adding further value.

The number of exceptions can also be systematically reduced by developing new rules for the decisions as process execution is observed—observing process performance, identifying new rules to handle particular cases, and adding those rules to the automated decision results in continuous improvement and process optimization.

Effective Application of Big Data Analytics

Organizations are increasingly investing in data-driven analytics, encouraged by trends in Big Data and Big Data analytics, in an effort to improve their business results, deepen customer understanding and better manage risk. The value of these analytics lies in improving decision-making—unless a decision is improved as a result of analytics it is hard to argue that the analytics have any value.

When analytics are applied to business processes modeling without explicit decisions the result is a set of graphs or visualizations about the process—which steps execute most often, where are the delays, which steps could perhaps be omitted. While this information is useful it only scratches the surface of what is possible with analytics.

In contrast almost any explicit decision can be improved using analytics. If the decision is identified, modeled and understood then the potential for analytics to improve it is much clearer. Particularly as the details of the decision are broken down into more atomic, simpler elements as described below, the role of analytics in improving some part of the decision making can be documented. This more specific role for analytics increases the potential for using Big Data and the analytics that result to improve decision-making and thus the processes that rely on that decision-making.

Making decision-making explicit and modeling it simplifies processes, increases agility and alignment, allows for more automation and straight through processing and focuses analytics effectively. Effectively defining decisions involves business-friendly approaches to both modeling them and representing detailed decision logic.

3. Decision Requirements Modeling

Basic Modeling Concepts

DMN provides two distinct but interconnected levels of constructs for modeling decision-making: the decision requirements level, and the decision logic level.

Decision Requirements Level

The decision requirements level of DMN allows a domain of decision-making to be modeled at a high level of abstraction, using only four types of elements, corresponding to commonly-used business concepts: decision, input data, business knowledge model and knowledge source.

Input data elements correspond to the business concept of data. They are data structures whose values describe the case about which decisions are to be made. They typically group data into high-level concepts of business significance, e.g. “Application Form”, “Claims history” or “Invoices”. Input data are notated in DMN using the shape in Figure 1:

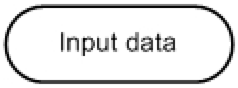


Figure 1 Input Data Notation

A *decision* element corresponds to the business concept of an operational decision. It is the act of determining an output value (a data structure) from a number of input values (also data structures), using some decision logic. The inputs to a decision may be input data elements or the outputs of other decisions. The decision logic may include the invocation of one or more business knowledge models. A decision is notated in DMN using the shape in Figure 2:

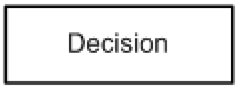


Figure 2 Decision Notation

A *business knowledge model* corresponds to business concepts such as “expertise”, “know-how” or “policy”. It is a function which encapsulates an area of business knowledge as executable decision logic, possibly expressed as business rules, an analytic model, or an algorithm. One important form of decision logic specifically supported by DMN is the decision table (see 4. Modeling Decision Logic In Decision Tables). The business knowledge model is parameterized, and is therefore a reusable component that may be called from multiple decisions, or from other business knowledge models. A business knowledge model is notated in DMN using the shape in Figure 3:

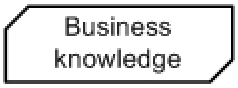


Figure 3 Business Knowledge Notation

A *knowledge source* defines an authority for decisions or business knowledge models, for example a manager responsible for a decision, a policy manual, or a piece of legislation with which a set of rules must comply. A knowledge source is notated in DMN using the shape in Figure 4:

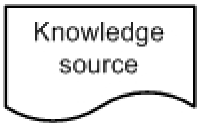


Figure 4 Knowledge Source Notation

These four elements are interdependent, and the interdependencies are characterized in DMN as *requirements*:

- A decision requires all the inputs used in its decision logic: these are called i*nformation requirements*, which are notated as solid arrows
- Decisions may require the invocation of business knowledge models (and business knowledge models may require the invocation of other business knowledge models): these are called *knowledge requirements*, which are notated as dashed arrows
- Decisions and business knowledge models may require sources of authority: these are called *authority requirements*, which are notated as dashed lines with filled circular heads.

When DMN elements are drawn connected by their requirements, the result is a *Decision Requirements Diagram* (DRD) such as Figure 5. A DRD shows the high-level structure of a domain of decision-making, revealing the relationships between a number of decisions, areas of business knowledge, areas of data and responsible authorities.

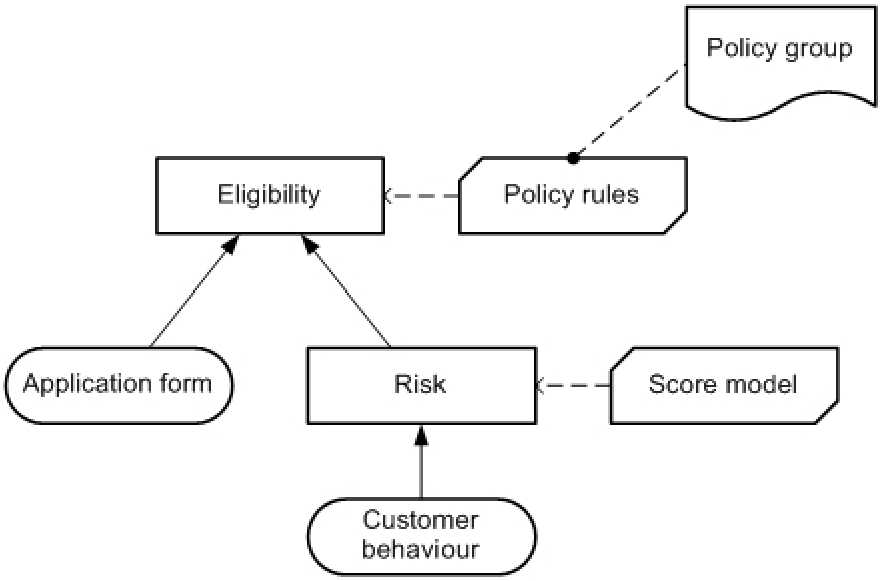


Figure 5 Decision Requirements Diagram

In this simple example, a corporate **Policy group** (a knowledge source) is responsible for defining a set of **Policy rules** (a business knowledge model), which is invoked to make an **Eligibility** decision whose output is (e.g.) ELIGIBLE or INELIGIBLE. The Eligibility decision uses input data from an **Application form**, and the results of another decision: **Risk**, whose output is a risk score. The Risk decision invokes a **Score model** to calculate the score from input data describing past **Customer behavior**.

Decision Logic Level

The decision logic level of DMN provides an expression language (called FEEL) for specifying detailed decision logic, and a corresponding notation (boxed expressions) which allows such expressions to be associated with elements in the decision requirements level.

FEEL – the Friendly Enough Expression Language – is a simple language with inspiration drawn from Java, Javascript, Xpath, SQL, PMML, Lisp, and others. In particular, FEEL extends JSON (JavaScript Object Notation) objects: A JSON object is a number, a string, a context (JSON calls them maps) or a list of JSON objects; FEEL adds date, time, and duration objects, functions, friendlier syntax for literal values, and does not require the context keys to be quoted.

The syntax and semantics of FEEL are defined using grammar rules that show how complex expressions are composed of simpler expressions, and semantic rules that show how the meaning of a complex expression is composed from the meaning of constituent simper expressions. As a result, DMN completely defines the meaning of FEEL expressions (provided they do not invoke externally-defined functions). There are no implementation-defined semantics. FEEL expressions have no side-effects and have the same interpretation in every conformant implementation.

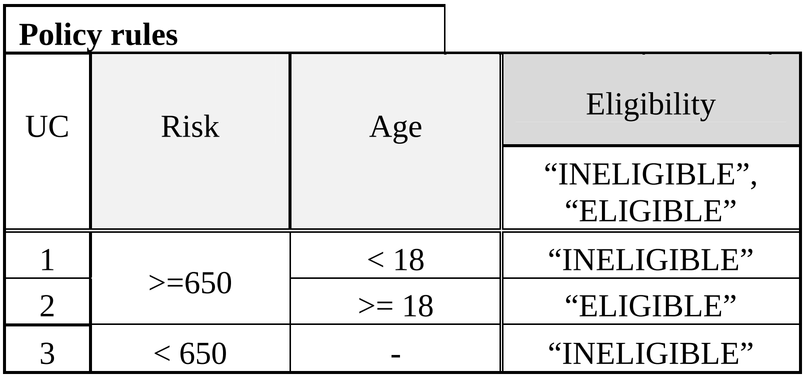
Boxed expressions allow the decision logic to be decomposed into small pieces that can be notated in a standard way and associated with elements at the decision requirements level. A DRD plus its boxed expressions form a mostly graphical language that completely specifies a decision model.

For example, the simple boxed expression in Figure 6 might be associated with the Eligibility decision in the DRD above. It first defines the applicant's age by reference to Application form input data, then calls the Policy rules, providing Age and the results of the Risk decision as parameters. The result forms the output of the Eligibility decision.

Eligibility	
Age	Application form . Applicant . Age
Policy rules (Risk, Age)	

Figure 6 Boxed Expression

One form of boxed expression which is particularly important in DMN is the decision table (described in detail in the next section). The simple example in Figure 7 might be associated with the Policy rules business knowledge model in the DRD above. It represents a set of rules for determining Eligibility from Risk and Age parameters.



Policy rules

UC	Risk	Age	Eligibility
			“INELIGIBLE”, “ELIGIBLE”
1	>=650	< 18	“INELIGIBLE”
2		>= 18	“ELIGIBLE”
3	< 650	-	“INELIGIBLE”

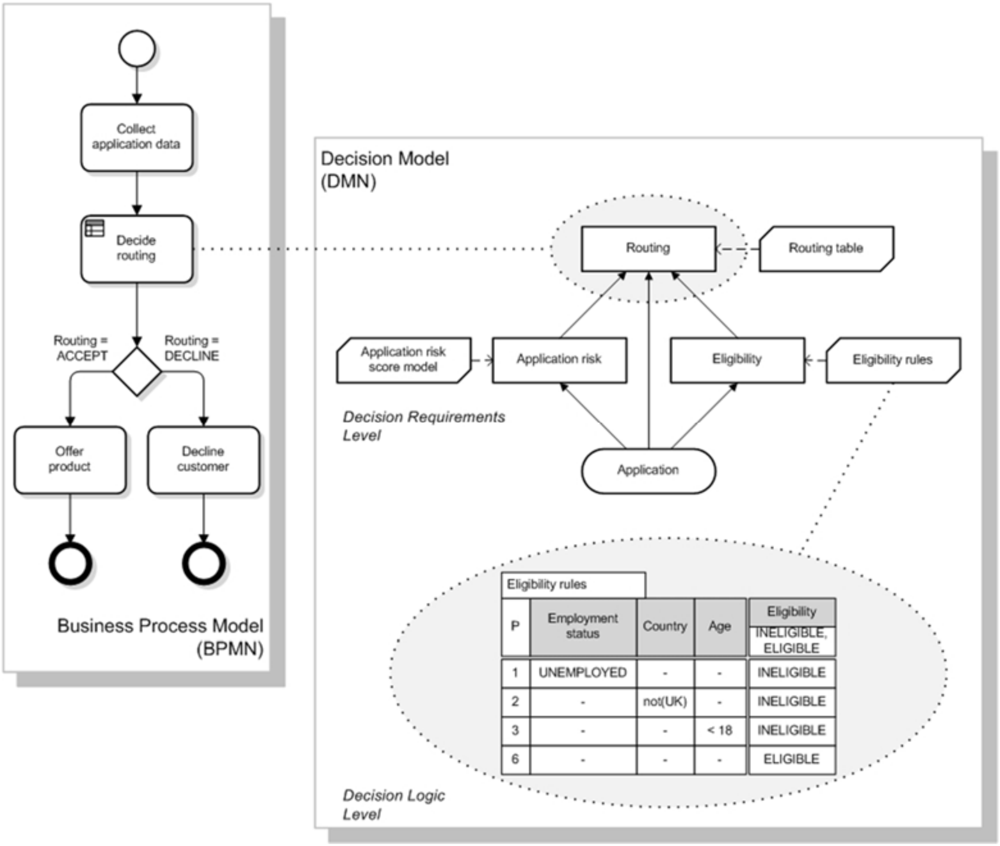
Figure 7 Decision Table

Decision Models

The two levels of DMN—decision requirements and decision logic—together provide a complete *decision model*. At the decision requirements level, the notation of the DRD is simple enough to make the structure of the model immediately apparent, yet the decision logic level provides a specification of the decision-making which is precise enough to allow automatic validation and/or execution.

Figure 8 (taken from the introduction to the DMN specification) summarizes the relationship between the levels of a decision model in DMN, and one possible relationship of the decision model with a business process model in BPMN. Decision models are complementary to business process models, and may be used to specify in detail the decision-making carried out in process tasks. This is discussed in more detail in *5 .The Context For Decisions In BPM*. Here it can be seen that the Decision Requirements Diagram is able to form a bridge between a business process model and decision logic expressed (for example) as a decision table.

One of the most common ways to represent decision logic in decision modeling is using a decision table.



Eligibility rules

P	Employment status	Country	Age	Eligibility INELIGIBLE, ELIGIBLE
1	UNEMPLOYED	-	-	INELIGIBLE
2	-	not(UK)	-	INELIGIBLE
3	-	-	< 18	INELIGIBLE
6	-	-	-	ELIGIBLE

Figure 8 The Relationships Between Decision and Process Models

4. Modeling Decision Logic In Decision Tables

Determining the appropriate outcome value for a number of input values is a central purpose of decision logic. The decision logic for operational decisions is specified in policy documents, scenarios, decision manuals, instructions, reports, all describing what the decision outcome should be for certain value combinations of input conditions. This logic can be discovered and described using a complex analytical model, a simple mathematical function, a list of decision rules, and a variety of other formats. Decision tables traditionally allow the visualization of these input-outcome combinations in a concise tabular format, such that it is easy for business users to specify and maintain business logic in a complete and consistent way, ready for automation or human decision making. DMN provides a standard notation and semantics for decision tables.

Decision logic is built from simple sentences, e.g.:

The decision outcome for A is a when B (and C, or D, ...) have a certain value.

Other values for B or C might lead to a different outcome for A. These are simple statements, or decision rules, all concluding something about A, depending on some conditions. A group of such rules about A is listed in a tabular format: the decision table.

Structuring Decisions and Subdecisions In Tables and Subtables

In real life, the simple sentences are not simple. They are usually disguised and spread out over a more complex document. Structuring the decision therefore requires the grouping of sentences (or rules) according to the decision (or subdecision) they are referring to. Every group of rules can be combined into a table concluding something about the decision (or subdecision) and using the results from other decision tables.

Structuring decisions into a network of decision tables is common practice in decision tables. In Figure 9, in order to know the outcome for A, we need information about B. This information might be readily available or in its turn, B might be concluded from some other elements (E and F). In the latter case, B is a subdecision. It is again represented using a group of rules, now concluding something about B. The structure indicates which subdecisions are required for a decision, and which subsubdecisions are required for every subdecision, and so on.

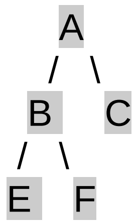


Figure 9 A Decision Table Network

The fact that B is a subdecision is clear from the observation that sometimes B is used as a *condition* in a statement concluding about A, and sometimes it is used as a *result* in a statement concluding about B.

This concept of decision table networks is further elaborated in DMN by extending the table network into a decision requirements model (see 3. Decision Requirements Modeling above).

Representing The Rules For A Decision

Decision tables are a powerful and proven technique for representing and validating a set of related rules in the form of a table. They have proven a useful aid in modeling complex business decisions and are easy to check for consistency, completeness and correctness.

Because of these interesting properties, decision tables are an important part of DMN. Throughout the years, however, the concept has often been reinvented, renamed or misinterpreted. One of the aims of DMN is to clearly define different types of decision tables, such that there is a common level of understanding and possible exchange when working with decision tables.

A decision table is a tabular representation used to describe and analyze situations where the value of a number of evaluation terms (conditions) determines the value of a set of outcomes. The tabular representation of the decision situation is characterized by the separation between conditions and outcomes, on one hand, and between names and entries (for conditions or outcomes), on the other. In this representation all rules are shown as rows (or columns depending on orientation) in a table. Each column (or row) refers to one condition or outcome.

The fact that in a decision table, conditions and outcomes are recorded in the same order for every rule has a number of advantages over just listing a group of rules:

- Well-designed decision tables avoid common anomalies in rule systems, such as redundant rules, conflicting rules, subsumed rules, unnecessary conditions, circular rules, and missing rules or combinations.
- The acquisition process is well served through the overview and communication abilities of well-structured decision tables.
- When modeling and representing the complex decision logic of real business situations, we want to ensure the quality of the set of rules from the start. Also, because maintaining the rules. by end user domain experts is not a trivial task and often introduces unnoticed anomalies, it is important that this quality is maintained so that the rules remain correct, complete, consistent and simple. Decision tables, with their fixed condition ordering, help with both these qualities.

Decision Table Types

Even though decision tables have been established for some time, and best practices have been defined decades ago, there is still a lot of variation in business practice. DMN will not prescribe a single best format (because it is methodology independent), but will clarify and standardize a number of decision table formats such that exchange is possible and the meaning is clear.

A distinction will be made between tables that return the outcome of one rule and tables that return multiple outcomes. Tables returning one outcome select the appropriate outcome value from the group of rules in the table. Tables returning multiple outcomes select all matching rules in the table and then perform an additional operation, such as adding the outcome values (as in a scorecard).

Tables Returning the Outcome of One Rule

Tables returning one rule outcome select the appropriate rule with its outcome value from the group of rules in the decision table. This may look straightforward (and it usually is), but there can be different rules matching a given set of input values and then a choice has to be made. A distinction is made between tables where (a) there can be only one match, or (b) there can be multiple matches, but with the same outcome, or (c) multiple rules with different outcome can match and a selection has to be made.

DMN therefore distinguishes different table types, identified by the first letter:

- unique hit tables: every input case is included in one rule only. There is no overlap between rules.
- any hit tables: every input case may be included in more than one rule, but the outcomes are equal. Rules are allowed to overlap.
- priority hit tables: multiple rules can match, with different outcome values. This policy returns the matching rule with the highest output value priority (e.g. highest discount).

- first hit tables: multiple (overlapping) rules can match, with different outcome values. The first hit by rule order is returned (and evaluation can halt). This is a common usage, because it resolves inconsistencies by forcing the first hit. It is important to distinguish this type of table from others because the meaning depends on the sequence of the rules. Because of this sequence, the table is hard to validate manually and therefore has to be used with care.

Tables Returning the Outcome of Multiple Rules

Tables returning the outcome of multiple rules collect the outcome value of all matching rules in a list and then return the list or the result of an operation on the list (e.g. the sum of scores). These tables are called multiple hit tables.

DMN distinguishes several forms of multiple hit tables, based on the order of or the operation on the resulting list:

- no order: returns all hits in a unique list in arbitrary order.
- output order: returns all hits in decreasing order of output value priority.
- rule order: returns all hits in rule order. Note: the meaning will depend on the sequence of the rules.
- sum: a common table form, returning the summed output value of all matching rules (a scorecard).

The Advantages Of Well-Defined Decision Tables

Although DMN will allow multiple forms of decision tables, different table types have different properties in terms of validation abilities, ease of construction, etc. This is a modeling issue and not in the scope of DMN. Because various modeling methods exist it is important to recognize and standardize these different forms, and so allow interchange.

Completeness

Not every list of rules in a tabular format represents a complete picture of the decision logic. Completeness means that every possible combination of input values will result in an outcome value (no missing rules). Different table types offer different opportunities to ensure or check completeness.

Exclusivity

Not every tabular representation offers equal opportunities for validation. Validation however is important for the quality of the decision logic. As indicated in the CODASYL report on decision tables (Codasyl 1982), a complete table with mutually exclusive columns offers a number of advantages. This means that each possible combination of conditions can be found in exactly one (one and only one) row (or column). A rule however may include multiple single cases and is then called a contracted rule. Rules are mutually exclusive: no condition combination occurs in more than one rule. If the rules are not exclusive at least one combination of conditions is present in more than one rule. This may look like a harmless redundancy but it opens the door to (future) inconsistencies. Only an exclusive table allows easy checking for consistency and completeness (as in Figure 7).

5 .The Context For Decisions In BPM

Decisions Within BPM

BPM has become the main focus for a large majority of business IT, centering as it does on the process tasks carried out by the business in its day-to-day operations. Organizations and departments in those organizations typically describe themselves by the work they do—the business processes they carry out to achieve the goals of their remit. The discipline of Business Process Management helps to achieve efficient and controlled business processes through its coverage of processing modeling, testing, governance, and management and control or process execution, either by the organizations' staff or through automation and Straight Through Processing. The focus of DMN is on modeling the business decisions that are made alongside, associated with, driving or being driven by the business processes themselves. So it is no surprise that there is a related discipline of "Decision Management" that covers the same aspects of decision modeling, testing, governance, and execution management and control.

History of Decisions and BPM

It is important to understand that, from a BPM perspective, there has always been a close relationship between "process" and "decision." Workflow and early BPM tools in the 1990s usually provided a means of scripting decision logic (sometimes referred to as "business logic" or "business rules"). By the 2000s there was an increasing trend to link to BRMS (Business Rule Management Systems) to provide this logic, relying on these vendors to provide the modeling, vocabulary handling and runtime capabilities. With the increase in vendor consolidation there has been continued integration of "decision technologies" —especially decision table representations and engines—into BPM tools, albeit without the modeling standardization seen by BPMN.

Anyone familiar with BRMS tools would note that these often provide an orchestration mechanism for defining what is effectively, but never called, a "decision process" —the terms used are "decision flow" or "ruleflow". Nonetheless such processes are used to model some fixed, ordered sequence of decisions—ordered to ensure that subdecisions are available for subsequent decisions—until some primary decision is made whose results are used directly in some business process or processes. Most BRMS tools are of course primarily concerned with rule execution, together with providing content management for the rules and decisions that are to be executed—very little attention is generally made to the aspect of "modeling" these subdecisions or their dependencies per se outside of providing decision representations like decision tables and trees. Ruleflows are a half-way house between a DMN Decision Requirements Diagram (see above) and a BPMN Process Model.

It is worth noting that the importance of decision logic was noted even during the formative days of BPMI during the creation of BPMN—a Business Rules Task Force was even planned, and BRMS vendors participated in BPMI knowing that the success and adoption of BPM would also drive the success and adoption of decision technologies. This has led to the BPMN 2.0 definition of "business rule task" to specify a work task to make decisions.

Decisions in Process Design

Processes are designed with respect to process goals; the effect of DMN on process designers is to encourage them to identify the business decisions that occur in their processes up front, and how their processes relate to these decisions.

For example, the process for Order Processing may include the decision DecideToAcceptOrder which may be the first task. There may be associated decisions at other stages in the process called DecideToRevokeOrder, based on additional information derived during processing as the process proceeds.

Some process designs will need to be reassessed as decision designs. For example, a process called AllocateCallCenterGroupToCall is really a decision based on the attributes of the caller and attributes of the available Call Center groups. Once this is realized some decision analysis methodology can be applied.

Decisions in Process Models

DMN is mostly related to the "modeling" aspect of BPM—indeed the term DMN covers "Decision Model and Notation" in a directly equivalent way to BPMN covering "Business Process Model and Notation". The main observation for process modelers is that decision modeling is *not* a subset of process modeling but an entirely separate discipline: they are usually related of course, as processes refer to decisions and decisions can require processes. However, both are "first class" models from the business modeling perspective.

To demonstrate this, consider that some decision results can be re-used across processes: one might assess the CreditWorthinessDecision of a loan applicant and then for any co-applicant—two instances of the same decision. This decision may of course need to be made at some point in a LoanApplicationProcess, and may need to be remade at different parts of the customer case such as when there is a subsequent re-application or application for a change in the loan. Note the inherent applicability of decisions in Case Management too!

It is worth noting that Process Gateways may or may not signify a business decision: some such gateways are purely related to the management of the process, while others will rely on some business decision—a Business Rule Task containing decision logic that can then be used in the Process Gateway.

Business Process Models Versus Decision Models

It is certainly not the intent of DMN to replace in any way the need for creating process models: on the contrary it is the experiences of many process modelers in accidentally embedding decision designs in process models that has led to the realization for the need for DMN. It is also the case that a common notation for describing decisions—just as BPMN provides for process orchestrations—would be useful for all the business analysts moving between tools in their job.

Decisions Role in Case Management and “Dynamic Intelligent Processes”

It may seem to some that the compartmentalization of decisions in DMN and processes in BPMN runs contrary to the concepts of Case Management and other types of BPM; “Dynamic Intelligent” processes and so forth. However this is not the case; the recognition of specialization in models and execution is key to understanding how to achieve and deliver on these.

In Case Management, there is a new OMG standard called CMMN Case Management Model and Notation, which is expected by its authors to utilize DMN to represent case decisions in a future version.

For "Dynamic Intelligent Processes" there are several considerations. Firstly the "dynamic" aspect implies ability to change and adapt—effectively to make decisions about which processes and process tasks apply on a continuous or near-continuous fashion, responding rapidly to changing events as required. The "intelligent" aspect implies good performance in its decision making that in turn implies advanced decision technologies. Some would argue that the best platform today for implementing dynamic intelligent processes would be the rules engines of the decision technology vendors, using declarative rules to organize decisions and process tasks… but a discussion of that is beyond the remit of this paper!

6. Conclusion And Next Steps

The Decision Model and Notation standard is expected to be submitted to the Object Management Group for approval in 2013 with publication expected in 2014. In the meantime the core elements of the standard—a focus on decisions as peers to process, the value of modeling decisions, the power of decision tables to describe decisions and the use of a simple language to specify decision logic—can be put to work today. Identifying, modeling and describing decisions alongside your business processes helps you manage complexity, drive alignment and improve your business. Adopting the key elements of the standard as part of your approach to intelligent business processes offers tangible, immediate benefits that will only increase in value as the standard is published and adopted.

References

(Taylor 2011) James Taylor. Decision Management Systems: A Practical Guide to Using Business Rules and Predictive Analytics. New York: IBM Press, 2011.

(Fish, 2012) Alan Fish. Knowledge Automation: how to implement decision management in business processes. New York. Wiley, 2012

(Codasyl 1982) Codasyl. A Modern Appraisal of Decision Tables. Report of The Decision Table Task Group, ACM, New York, 1982.