

INSIGHTS FOR SUCCESS



The OMG Decision Model and Notation Spec (DMN) and The Decision Model (TDM)

By Barbara von Halle and Larry Goldberg

On September 23, 2013, the Decision Model and Notation Specification (i.e., DMN) was approved first by the Business Modeling and Integration Task Force and later accepted by the Approval Board of the Object Management Group (OMG) [1]. Final approval happens through the OMG Finalization Task Force (FTF) and should happen over the next few months. Typically such approvals occur with relatively minor, non-material, changes to the specification and software vendors tend to use the pre-FTF specification to advance products to the market.

In a nutshell, the goal of DMN is to provide a notation for decisions understandable to all audiences, including business and technical people. This is good news and is the very reason we introduced The Decision Model (TDM) to the public in 2009 [2].

Critical Points

This article is a preliminary introduction to DMN, specifically for people experienced in (or familiar with) TDM. The important points are:

- While TDM was instrumental in initiating the DMN subcommittee, the DMN specification is not based on TDM. This means that the DMN spec may not look much like the TDM notation that business analysts are already using. More about this later in this article.
- The reason for the differences between TDM and DMN are that their respective objectives are different. TDM defines a formal model for logic (in its most atomic form in one singular representation for easy validation and other functionalities). DMN defines a notation (not a formal model) that can accommodate various formats for logic expressions (from natural language, to partially rigorous, to sufficiently rigorous for generating code). DMN is also targeted at software developers creating software through which decision models are interchangeable.



The OMG Decision Model and Notation Spec (DMN) and The Decision Model (TDM)

- The first notational difference is the DMN DRD (decision requirements diagram) which is somewhat the corollary to TDM's diagram. The DMN DRD contains shapes for aspects other than the logic.
- The second notation difference is that the DMN detailed decision logic level may contain various types of expressions, one being a set of decision table formats, none of which are Rule Family format. (Rule Family format is more atomic and has only one singular representation).
- DMN includes a newly defined expression language for detailed logic. TDM instead includes standard operators, operands, and functions, but no formal language.
- TDM is based on a business-friendly glossary of fact types. DMN does not define a glossary (but includes modeled representation of input data)

The remainder of this article provides details behind the statements above. It ends with a conclusion on current thoughts about the benefits of each and a glimpse at a TDM and DMN world.

Why DMN is an Important Step for Decision Modeling

DMN is important, first of all because it validates the need for a new kind of model specifically for decision logic, separate and distinct from models we already have. Second, DMN, as an IT specification, is a confirmation that there is demand for a new kind of software product aimed at decision modeling and management. Third, the BABOK [3] update team at the IIBA [4] accepted decision modeling as a technique to be included in the BABOK, thereby removing the use of process to describe decision-making logic.

Equally important is the set of companies on the formal submitting members of the DMN. These are: Decision Management Solutions, Escape Velocity, IBM, Oracle, Knowledge Partners International, Model Systems, TIBCO, and the KU Leuven University. Some have already announced availability of, or intentions for, DMN-related software. KPI's software partners have delivered TDM-compliant software, some in use within major corporations for several years.

DMN Components at a Glance

For this article, there are five core components of DMN: a requirements level notation, decision logic level notation, expression language called FEEL (for Friendly Enough Expression Language), a supporting metamodel, and specific levels of conformance.

Of these components, this article focuses mainly on the two levels of DMN notation because these are most important for people creating, interpreting, and validating DMN decision models. This article also touches briefly on the DMN expression language since it may become important, time will tell.

Business people, business analysts, and decision modelers need not be knowledgeable in the DMN metamodel. The metamodel is for the technical audience who will create DMN-compliant software. It is not discussed further in this article.

DMN prescribes three levels of software conformance. Level 1 implies conformance to the notations. Level 2 includes Level 1 conformance plus support for S-FEEL, a subset of DMN's expression language. Level 3 includes Level 2 conformance in addition to full support for FEEL.

Before discussing the DMN notation and expression language, it is useful to understand DMN's history.



The OMG Decision Model and Notation Spec (DMN) and The Decision Model (TDM)

DMN History

Members of the OMG became aware of the idea of decision modeling during 2008 and 2009. At this time, Larry Goldberg and Barbara von Halle presented TDM to the group and James Taylor presented his approach too [5]. These presentations plus a growing awareness of decision management sparked interest in the idea that a decision model may be worthy of a formal industry-wide specification. From here, Paul Vincent (at that time, with TIBCO) and Christian De Sainte Marie (of IBM) were instrumental in forming the DMN subcommittee, which organized a successful “Decision Modeling Day” at an OMG quarterly meeting. Attended by a large group of industry practitioners and vendors, this meeting crystalized the need for a standard around this rapidly emerging field.

In March 2011, OMG issued an RFP soliciting proposals for a decision model and notation specification. There were two submissions, the submission teams merged, and KPI joined the subcommittee. Since then, the submission team has worked on the document culminating in the DMN specification version 1.0.

Common Philosophies of DMN and TDM

It is comforting to point out that there are five common philosophies shared by TDM and DMN.

Commonality #1: Decisions and decision logic belong to business people.

Our primary goal when introducing TDM was to return business logic back to the business people for governance. With traditional business rules techniques, the logic most often got lost within system code or was simply unmanageable. DMN echoes this sentiment in the first sentence of its Scope section, mentioning a notation for business people, business analysts, developers, and people who manage and monitor those decisions.

Commonality #2: There is a need for a new kind of model.

Our secondary goal when introducing TDM was to prove that business logic (like data) has its own existence, independent of other concerns. More importantly, it has its own natural structure which forms its own visual model, distinctly different from all other concerns for which we have models. In other words, business logic for most business decisions has never been best represented in lists or in process models, data models, or any other kind of model. We needed a new one and we needed one that elevated the importance of decision logic as a manageable business asset.

DMN confirms the need for a new model, one for decision logic. In fact, the DMN specification positions DMN as the third model in a BMI trilogy [6] of complementary model notations: BPMN 2.0, CMMN 1.0, and DMN 2013.

Commonality #3: The new model connects naturally to business process models.

With the earliest uses of TDM, decision-aware business processes emerged. A decision-aware business process is one that distinguishes between tasks that perform work and those that come to conclusions based on logic. DMN, too, relates decision models to business processes. In Figure 1 from the DMN spec, the model on the left is a process model while the model on the right is a related DMN decision model.



The OMG Decision Model and Notation Spec (DMN) and The Decision Model (TDM)

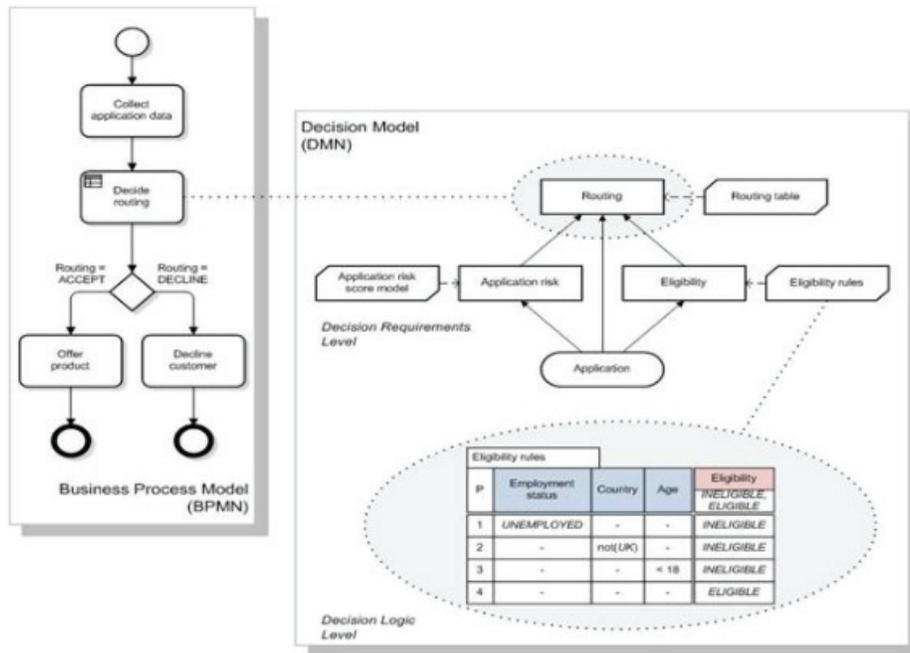


Figure 1: Aspects of Modeling

Commonality #4: Complete decision logic is automatable.

While TDM is technology-independent, it has always delivered models that are easily implemented in technology. In fact, organizations have implemented TDM decision models in many and multiple target technologies with no technical-artifacts within the models themselves, and with no coding by human beings. Likewise, the DMN specification states that, with complete (DMN) specification, decision logic becomes executable in technology.

Commonality #5: There is a need for a new kind of software.

In 2009, we stated that TDM's greatest significance may be its potential to inspire new, related technology and business directions. The DMN spec states that the market indicates a strong demand for new software and vendors are moving to supply that demand. Therefore, these vendors need a decision model standard.

Despite the commonalities, if you are familiar with TDM, you may find the DMN to be more unfamiliar than you expected due to the differences. An overriding difference is that the DMN spec is for technical audiences. Therefore, much of the DMN content is of a technical nature and not of interest to business analysts, business people, or even decision modelers. That said, the parts of most interest in this article are the DMN notation levels, expression language, and other differences.



The OMG Decision Model and Notation Spec (DMN) and The Decision Model (TDM)

DMN Differences

Difference #1: DMN Decision Requirements Level

DMN prescribes a notation for decision logic, not a formal model. The word “model” here is meant in the strict sense of a formal system involving at least two components: a singular structural component and principles defining its structure, technology-independence, and model-based integrity.

The DMN decision requirements diagram is a notation depicting important elements of decision-making and their dependencies. The important elements are decisions, business knowledge, business knowledge source, and input data. Figure 2 illustrates a DMN decision requirements diagram.

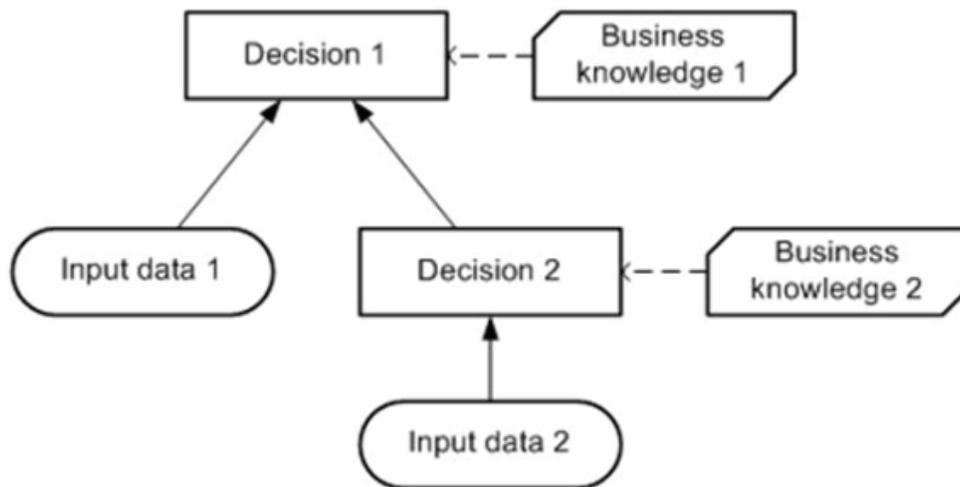


Figure 2: DMN Decision Requirements Diagram

For TDM practitioners, in Figure 2, each box labeled Decision correlates to a conclusion in a Rule Family. The arrows are dependencies among DMN decisions, similar to inferential relationships in TDM. Obviously, a DMN requirements diagram may also contain graphics for related concepts, such as business knowledge model (logic details), business knowledge source, and input data.

Figure 3 illustrates a traditional TDM decision model diagram (also known as a Decision View) on the left and how it translates on the right into a DMN decision requirements diagram. The diagram comparison is not exactly apples to apples. The DMN decision requirements diagram in Figure 3 contains only logic structures, not related concepts such as data sources – which would be a considerable level of additional detail. The TDM model reveals the detailed data inputs whether they are from another Rule Family (therefore above the dotted line) or from raw data (below the dotted line)



The OMG Decision Model and Notation Spec (DMN) and The Decision Model (TDM)

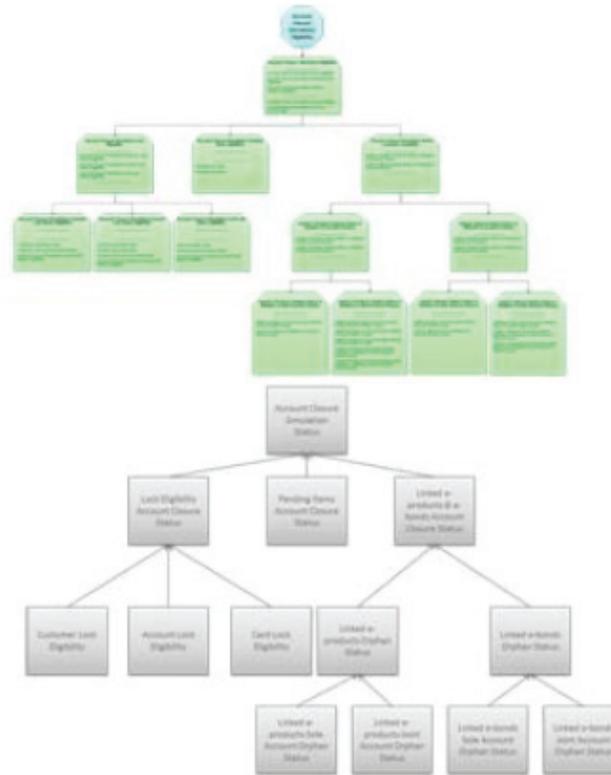


Figure 3: Comparison of TDM Decision View and DMN Requirements Diagram

Difference #2: DMN Decision Logic Level

DMN decision logic level is where to specify a complete expression of logic, potentially sufficient for automation. The right side of Figure 1 marks the boundary between the decision requirements level and decision logic level. Every Decision in a DMN decision requirements diagram may have a value expression or business knowledge model for its details. A business knowledge model may be business rules, decision tables meeting DMN decision table specifications, or analytical models.

For the TDM practitioner, the corollary to a DMN business knowledge model is the Rule Family table (also known as a Rule Family View). In TDM, the structure of the logic in the Rule Family table is evident in the TDM diagram while its content lives with the Rule Family table. Every Rule Family table conforms to 15 TDM principles: structural, declarative, and integrity. It is these principles that enable TDM model-based functionalities.

Difference #3: DMN Automation Approach

Also, as indicated earlier, DMN includes an expression language. Some of it is required for Level 2 conformance and some is required for Level 3 conformance. The parts for Level 2 conformance aim to provide standardization for representing decision model requirements for a decision model, but not necessarily sufficient for automation. The parts for Level 3 conformance aim to provide standardization for automating.



The OMG Decision Model and Notation Spec (DMN) and The Decision Model (TDM)

As stated above, TDM does not include an expression language aimed at automation. Instead, vendors of TDM- based software export decision models into various execution environments by converting them into the code base of the target execution environment [7]. This works well and there has not yet been a pressing need for a common execution language from which to translate to target technologies.

Difference #4: Glossary of Fact Types

DMN does not include a business-friendly glossary. Instead, the DMN decision requirements diagram depicts a representation of input data. Input data can therefore be in any format.

With TDM, business-oriented decision modelers use the glossary as the condition and conclusion parts of decision logic. Decision modelers or glossary administrators add fact types to the glossary using business names, business definitions, data types, and domains. These guarantee validation of the corresponding logic without having any knowledge of logical or physical data sources or models. In a technical part of the TDM business-friendly glossary, fact types are correlated to data sources. However, integrity validation and testing can happen without this correlation.

To Consider

And so we stand at an interesting fork in decision model adoption and maturity. Both TDM and DMN have commonalities. And there are interesting differences aiming for different objectives. Some differences are merely cosmetic, such as shapes of model constructs. Some are truly different in nature. The latter is where interesting conversations begin.

The Case for a Model: TDM

The Relational Model is a testimony to a formal model that was game-changing and has stood the test of time. It is defined by a singular structure (i.e., the relation) and supporting principles governing the components of that structure and its integrity [8]. The universal and simple structure opened the door to a new way of managing data (i.e., selecting rows, projecting columns, joins, unions, etc.). This new way was far more powerful than the navigational row-at-a-time data access of the day. And the relational model made clear the separation between the relational perspective and the implementation invisibly behind it [9].

Most interesting is that various diagramming techniques built upon the relational structure and integrity, gave rise to various forms of methodologies. Some described the entity-relational model as a thin layer above the relational model. Standards emerged to support relational data access.

It is in the spirit of this kind of model that TDM is similar. By definition, it has a singular structure and supporting principles, including integrity principles. These principles and simple structure open the door to new kinds of logic manipulation (e.g., model-based validation, model-based test case generation, model-based messaging, model-based views, and model-based business governance, for example). And, as one might expect, various diagramming techniques may be built upon it or may exist beside it. DMN may be one of these.



The OMG Decision Model and Notation Spec (DMN) and The Decision Model (TDM)

The Case for Standard Notation: DMN

DMN defines a standard notation that accommodates a variety of formats with discipline for some. And it extends that notation to include the context of requirements (e.g., business source, input data) rather than representing only logic constructs. It is independent of methodology. It does not include a business glossary or a formal set of comprehensive principles. Yet, it is a means by which decision models can be imported and exported from one tool to another. It is also a means by which decision models with one notation may be viewable in another. As indicated above, DMN has more degrees of freedom in its representation and supporting constructs than does TDM. This makes a great deal of sense because DMN's goals are to allow for different decision modeling methodologies – of which one may be TDM – and enable interchange of those models. DMN's various options for logic representation is an extremely valuable aspect of a specification and one that could well lead to its widespread adoption.

Achieving Synergy

So, the question arises: are there benefits to providing both DMN and TDM in the same software product? After all, the components of each have their own merits and uses.

It is only natural for TDM-compliant tools to also be DMN-compliant so that customers have flexibility. This includes the ability to export TDM models to DMN format and vice versa. Where there is a mismatch in either direction (for example, there is no DMN glossary to export), a conversion will be partially complete. It is also important that TDM-compliant software continue to support the full and evolving functionality of TDM for the value it brings beyond DMN.

As for automation? Currently there are approximately 30 defined standard functions to use in TDM Rule Family cells. Existing TDM clients extend these for their own for domain-or-industry-specific functions. FEEL would permit an unlimited extension of those functions in a standardized expression language that will simplify interchange. And, FEEL enables the encapsulation of non-TDM compliant structures into FEEL expressions that could be incorporated into TDM. In a future world, if FEEL became the widely adopted standard, decision models or notations would be exported as FEEL expressions, eliminating the need for the multiple conversion methods of today. This requires that vendors of target (rule) execution systems adopt FEEL [10]. So we will wait and see.



The OMG Decision Model and Notation Spec (DMN) and The Decision Model (TDM)

A Moment in Time

Someday we may all remember this moment in time. The DMN specification is significant in that it confirms what some of us have known for a while - that decision logic is a business asset worth managing. We, at KPI, will continue to support DMN for the value it brings. At the same time, we will continue to evolve and endorse aspects of decision modeling that are purely for the business audience.

Whether you are a veteran decision modeler or a beginner, this is an exciting time to be a decision modeler. Stay tuned for more news on DMN and TDM as it happens.

Authors: Barbara von Halle and Larry Goldberg of Knowledge Partners International, LLC (KPI)

References & Footnotes:

- [1]. The Object Management Group (OMG®) is an international, open membership, not-for-profit computer industry standards consortium. Founded in 1989, OMG standards are driven by vendors, end-users, academic institutions and government agencies. OMG Task Forces develop enterprise integration standards for a wide range of technologies and an even wider range of industries. For more information, go to (<http://www.omg.org/>)
- [2]. von Halle and Goldberg, 2009, Taylor & Francis LLC, The Decision Model: A Business Logic Framework Linking business and Technology
- [3]. BABOK stands for “Business Analyst Book of Knowledge”
- [4]. IIBA stands for “International Institute of Business Analysts”
- [5]. See Taylor, James, 2011, IBM Press, Decision Management Systems: A Practical Guide to Using Business Rules and Predictive Analytics
- [6]. BMI stands for “Business Modeling and Integration Domain Task Force.”
- [7]. To our knowledge, TDM is the only common model of business logic that today is widely automated into many (and often several different kinds) of automation technology.
- [8]. There are two integrity principles: entity integrity and referential integrity. There is a third for user-defined integrity. It is common opinion that the relational model only provided the minimum for integrity principles.
- [9]. It is interesting that, missing from the Relational Model was the connection to business data names and definitions (e.g., glossary).
- [10]. If the past is a predictor, vendors are not often open to such adoption, preferring their proprietary turf, as witnessed in the lethargic adoption of standards such as Rule Interchange Format (RIF).

Originally published: <http://www.modernanalyst.com/Resources/Articles/tabid/115/articleType/ArticleView/articleId/2757/The-OMG-Decision-Model-and-Notation-Spec-DMN-and-The-Decision-Model-TDM.aspx>



The OMG Decision Model and Notation Spec (DMN) and The Decision Model (TDM)



Larry Goldberg has over thirty years of experience in building technology based companies on three continents, and in which the focus was rules-based technologies and applications. Commercial applications in which he played a primary architectural role include such diverse domains as healthcare, supply chain, and property & casualty insurance.



Barbara von Halle is co-inventor of the Decision Model and co-author of The Decision Model: A Business Logic Framework Linking Business and Technology published by Auerbach Publications/Taylor and Francis LLC 2009.

Contact us

For more information please visit us:

www.sapiensdecision.com

info@sapiensdecision.com

+ 1-800-858-9473 (US)

+ 44 1895 464 000 (UK)

Published in: www.modernanalyst.com, October 27, 2013

© 2014 Copyright Sapiens International. All Rights Reserved. Dec. 2014