SAPIENS
Partnering for Success

A Primer on The Decision Model

By Larry Goldberg and Gil Segal

December 2021

Table of Contents

SAPIENS
Partnering for Success

1 Introduction

The Decision Model (TDM) is a way of representing determinative business logic that is platform and technology independent.

"Determinative" because TDM is focused on representing logic that the business can be confident that, based on facts that will be available in each circumstance, will lead to the required predetermined conclusion.

We used to think about this type of logic as "business rules" but automating processes and digitization at scale requires much more than individual business rules. Managing the interaction among the many rules necessary in such automations and aligning them to business outcomes has proven to be highly complex, and not easy to maintain over time as the business evolves.

TDM seeks to solve this problem by gathering multiple business rules into a single body of logic (hence "business logic") that provides a determination to a particular business question at a given point in the automation. It integrates these multiple rules into a structure that is "normalized", which is to say is the simplest possible to provide the determination being sought, thereby minimizing the duplication of rules and potential conflicts in logic, while making the reuse of the logic components possible across many different automations in the enterprise.

For more details about TDM, we refer you to the book from which these concepts and ideas (and many of the tables and diagrams) are drawn, "The Decision Model: A Business Logic Framework Linking Business and Technology" (von Halle & Goldberg) © 2009 Auerbach Publications/Taylor & Francis, LLC. Reprinted with permission of the Publisher.

This Primer will provide you with an understanding of the components of the model, and how they fit together; it will also describe how to approach building models that will help in a wide range of business, process, and digital automations.

2 What about AI?

AI, Machine Learning and Analytic models are all based – in one form or another – on probabilistic, rather than determinative, techniques. These augment and support TDM as inputs, and we describe in Section 6 how this is done in practice.

3 Decision Model Fundamentals

The simplicity and elegance of TDM are best learned by building the model from the ground up. This is not the optimal way to create a Decision Model in real life, but it provides an excellent perspective and understanding of the model.

3.1 What Is Business Logic: Business Logic Fundamentals

First, let us define the term "business logic": it is how the business derives conclusions from facts. Building from that definition, we say that a "business logic statement" is an expression of conditions that evaluate facts leading to a conclusion of a new fact. We can illustrate this by the diagrammatic representation of a business logic statement in Figure 1.

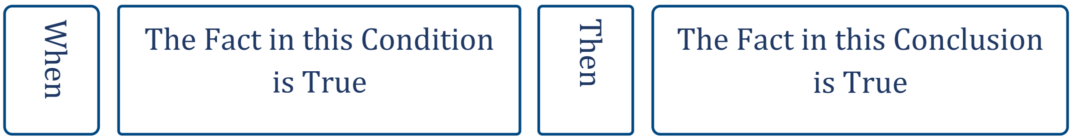


Figure 1 A Diagram of a Business Logic Statement

This diagram faithfully follows our definition. The condition statement contains a fact that is evaluated for truth and which – if true – leads to a conclusion containing a new fact.

Figure 2 is a simpler way of diagraming the statement.

Figure 2 Simpler Diagram of a Business Logic Statement

This diagram is really the same as the one in Figure 1 but is independent of the grammar used to express it. We could use the "when…then" form, the "if…then" form, or we could state it in other declarative forms such as "The…shall…" and so on. In all cases, the condition is an assertion, and when true, the conclusion assertion is inferred to be true. The simplest form of the assertion is represented in Figure 3.

Figure 3 Simplest form of Diagrammatic Representation of the Business Logic Statement

This diagram captures its declarative essence. But enough with the diagrams! We need an example to make these concepts concrete. Let's construct a real business logic statement.

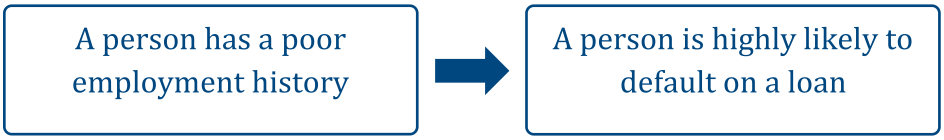


Figure 4 Example of a Business Logic Statement

In Figure 4 we have an example of a business logic statement. It concludes that a person who has a poor employment history is highly likely to default on a loan. Our condition assertion is "A person has a poor employment history", and when true, we conclude that "A person is highly likely to default on a loan" is also true. We could express this business logic statement in a natural language statement as "When (or if) a person has a poor employment history, then that person is highly likely to default on a loan." (The *logic* in the natural language statement and the diagram are the same).

3.2 The Structure of Condition and Conclusion Assertions

Each assertion is composed of:

- Facts
- Fact Types
- Operators
-

3.3 What Is a Fact and a Fact Type?

Fact Type	Business Concept	Domain (range or set of valid values)
Person Employment History	Person	Good, Average, Poor
Person Likelihood of Defaulting on a Loan	Person	High, Medium, Low
Vehicle Insured Value	Vehicle	Range 5,000-150,000
Policy Renewal Method	Policy	Manual, Automatic

Figure 5 Examples of Fact Types

We say that a fact (or, more accurately, fact value) is a piece of information (i.e., a piece of data within a context). Take the example:

"The Person has five years at his current employer."

The piece of information is the value “five” (fact value) in the context of years at current employer for the person. The fact value in this statement is “five.” All the information that gives context to the value of “five” – “Person’s number of years at his current employer” – is what we call the “Fact Type”.

The following phrases are important to remember:

- **Fact Type** is a general classification of a fact, not the piece of information itself.
- **Business Concept** is the noun, a thing that is the subject of the Fact Type.
- **Fact Type Domain** is the range or set of valid values that make business sense for a given Fact Type.

Figure 5 contains some examples of Fact Types. Fact Types play an important role in building the condition and conclusion assertions that make up business logic statements.

3.4 What Is an Operator?

An operator, (more correctly a logical operator) is a symbol – i.e., a word – that connects two assertions and applies a test of truth between the two assertions through the application of logic

- Conventional symbols, i.e., logic operators, are ones such as:
    - “Is”, “Is Not”, “Is Less Than”, and so on.
- There are many others. Many of them are Boolean (meaning that they are binary, resulting in one of two values – “true” / ”false”, “yes” / ”no”), as we would use in programming, but we can use any logical operator, and the results can vary widely, limited by the domain values of the Business Concept.

3.5 The Structure of Business Logic as Conditions and Conclusions

Now that we better understand the components of assertions, we are ready to delve deeper into the nature of a business logic statement. We begin by understanding the definition of condition and conclusion:

- A **Condition** is an assertion of a fact of a given Fact Type that is to be tested for truth in the Business Logic Statement
- A **Conclusion** is an assertion of a fact of a given Fact Type that is true if the Condition assertions of the Business Logic Statement are true

Condition assertions and conclusion assertions have identical structures:

Fact Type — Operator — Operand

Operands are frequently one or more fact values (from the pool of fact values in the Fact Type domain) but may be a formula or a Fact Type.

So, take the assertion:

Person Employment History is Poor

We model it as:

Fact Type	Operator	Operand
Person Employment History	Is	Poor

Figure 6 has some examples of assertions exemplifying their structure. Any of these examples could be either a condition or a conclusion assertion in a business logic statement.

Fact Type	Operator	Operand
Person Likelihood of Default on a Loan	Is	Average
Person Credit Score	Is Greater Than	700
Person Body Mass Index	Is Calculated As	[weight (kg)] / [height (m)]2
Policy Claim Date	Is Before	Policy Effective Date
Person Eye Color	Is In	{Blue, Green}
Customer Type	Is	Preferred
Inventory Optimum Level	Is	Two Months Sales

Figure 6 Examples Showing the Structure of Assertions

Notice how the operand can be one or more fact values, a Fact Type, or a formula. Looking at the operands in Figure 6, we see:

- **Fact values** are: Average, 700, Blue, Preferred, etc.
- **Fact Types** are: Policy Effective Date, Two Months Sales, etc.
- **Formula** is [weight (kg) / [height (m)]^2

3.6 What Is an Atomic Business Logic Statement?

In TDM, it is critical that business logic statements be atomic.

What is the meaning of the word "atomic", and why is it important?

Atomic means "indivisible". The word comes from the Greek root "atomos", first proposed by the ancient Greek philosopher Democritus, who believed that all material in the earth was composed of tiny, irreducible fractions of matter. In our context, "atomic" is used to signify a business logic statement that cannot be decomposed (into smaller business logic statements) without losing its meaning. Similarly, the concept of atomic may be applied to Fact Types; an atomic Fact Type is one that cannot be decomposed (into other Fact Types) without losing its meaning. Thus, we use atomic to denote the irreducible form of a Fact Type and of a business logic statement.

3.7 Why Are Atomic Business Logic Statements Important?

Reducing business logic to its atomic form is important because doing so delivers only one representation of the business logic, eliminating ambiguity of meaning. This ensures that the accuracy of the logic can be analyzed for precision, completeness, and consistency and is represented in its most manageable form.

So, we say an atomic business logic statement consists of ***zero to many conditions*** leading to a conclusion about a ***single Fact Type*** where:

- Each condition is an *atomic* logical expression about an *atomic* Fact Type.
- Conditions are ANDed together, never ORed.

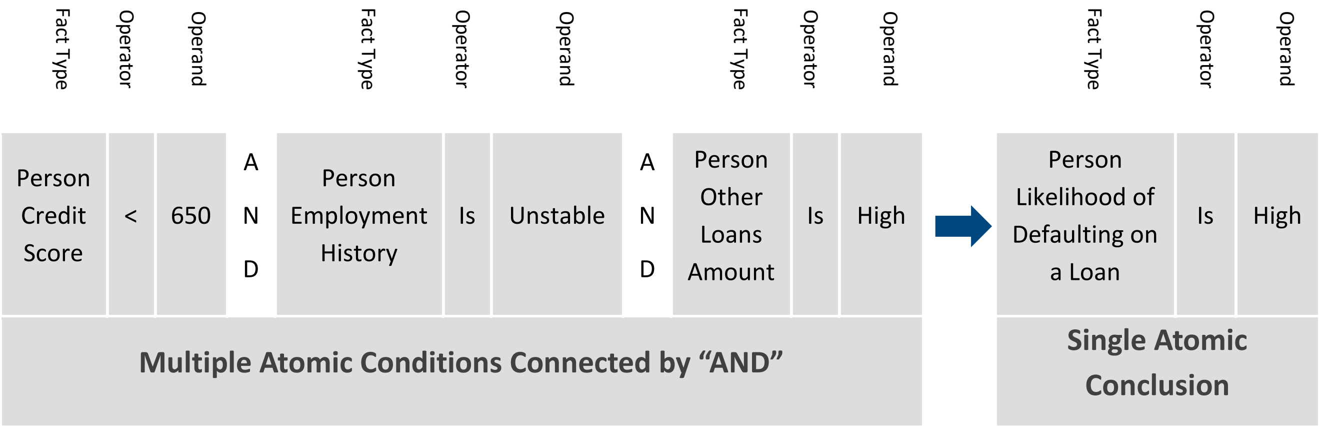


Figure 7 Diagram of an Atomic Business Logic Statement

Figure 7 is a diagram of an atomic business logic statement. It has three conditions, joined by ANDs, leading to a single conclusion. Examine it closely and you will see that each Fact Type in the conditions and conclusion is atomic and cannot be made any simpler without losing the logical intent.

Taken together with the definitions of the Fact Types, the business logic statement is clear, unambiguous and cannot be broken into smaller statements. It may be written out in longer, natural language statements using a grammar that is convenient to the user (or a grammar standard such as Semantics of Business Vocabulary and Rules (SBVR) as published by the Object Management Group (OMG)). As examples, here are a few possible natural language expressions of the business logic statement in Figure 7:

- If/when a person's credit score is less than 650 and the person's employment history is unstable, and the person's other loans amount is high then (we judge) the person's likelihood of defaulting on a loan is high
- A person with credit score of less than 650 and an employment history that is unstable and other loans amount that is high has a high likelihood of defaulting on a loan
- A person has a high likelihood of defaulting on a loan if all the following are true:
  - Person's credit score is less than 650
  - Person's employment history is unstable
  - Person's other loans amount is high

While these may be useful expressions of the business logic for everyday use, the use of natural language may introduce ambiguities, and render the statements non-atomic. (It is possible to apply a very rigorous grammar to avoid this, but that would mean all businesspeople would have to train in that grammar). Also, it becomes challenging to group the natural language forms of the logic statement in a way that they can be analyzed for logical integrity and completeness. TDM provides a simple, clear, and unambiguous method for grouping the business logic that supports this analysis.

3.8 Grouping Business Logic Statements Into Rule Families

It is clear from the example in Figure 7 that there are likely to be more business logic statements beyond the statement in the diagram that conclude a person's likelihood of default. The domain of the Fact Type "Person Likelihood of Defaulting on a Loan" (Figure 5) contains values of High, Medium, and Low. For completeness of the logic, it is necessary to cover all the results that may arise from each of these values. The logic leading to some conclusions can run into many, sometimes tens, and possibly hundreds of business logic statements. It is therefore important to determine the optimal representation for this grouping of business logic statements.

Conditions						Conclusion	
Person Credit Score		Person Employment History		Person Other Loans Amount		Person Likelihood of Defaulting on a Loan	
Is Less Than	650	Is	Unstable	Is	High	Is	High
Is Greater Than	720	Is	Stable			Is	Low
Is Less Than	720	Is	Unstable	Is	Low	Is	Medium

Figure 8 Grouping Business Logic Statements in Rule Families

Now look at Figure 8: Here is grouping of three business logic statements in a single tabular form. When such a grouping complies with certain principles that bring rigor to the form it is called a Rule Family, the primary structure of TDM. One unique advantage of the Rule Family is that it groups atomic business logic statements only by their conclusion Fact Type. This is possible because we have defined an atomic business rule statement to consist of a single conclusion Fact Type.

3.9 Rule Family Structural Features

Figure 9 illustrates the structural features of the Rule Family. It is a two-dimensional tabular structure of rows and columns. The table header is divided into two sections, the first consisting of all the condition Facts Types, and the second the single conclusion Fact Type. Each row of the table is a business logic statement. Our practice is to place the conditions in the left columns, and the single conclusion on the right. From a logical point of view, this sequence is irrelevant, and the columns could occur in any order. Also irrelevant is the sequence of the rows: no logic statement has any precedence, and remains logically correct in whatever order the rows are in. The condition columns are ANDed, never ORed. (If you are presented with logic statements that have ORs in them, it simply means that they are not atomic, and need to be broken into multiple, separate atomic statements).

Person Employment History	Is	Stable

Cell

Conditions						Conclusion	
Person Credit Score		Person Employment History		Person Other Loans Amount		Person Likelihood of Defaulting on a Loan	
Is Less Than	650	Is	Unstable	Is	High	Is	High
Is Greater Than	720	Is	Stable			Is	Low
Is Less Than	720	Is	Unstable	Is	Low	Is	Medium

Header

Rows

Column

Figure 9 Features of the Rule Family

The name of a Rule Family is simply its conclusion Fact Type. So, the name of the Rule Family in Figures 8 & 9 is Person Likelihood of Defaulting on a Loan.

3.10 The Agility of the Rule Family

Apart from its high level of logical integrity, the Rule Family enables us to express business logic in a form and manner that can easily be read and understood by all the stakeholders – business, management, and IT. People today are readily familiar with decision tables and are easily able to read them. The Rule Family is not simply a decision table. It is a decision table that is given rigor because it is governed by principles of TDM.

When changes to the business logic need to be made, the TDM principles enable logic changes to be made while maintaining logical integrity. The ability to easily modify business logic is key to making an enterprise more competitive.

In Figure 8, the first row tells us that:

A person has a high likelihood of defaulting on a loan if all the following are true:

- Person's credit score is less than 650
- Person's employment history is unstable
- Person's other loans amount is high

Assume that in talking to a Subject Matter Expert (SME), we are told that there has been a revision in thinking, and when a person's employment history is unstable, and the person's credit score is less than 650, then even when a person's other loans amount is medium (and not high, as in Figure 8), we conclude the person to have a high likelihood of defaulting on a loan. How do we change the Rule Family? Simple – we change the cell in row one, under the header Person Other Loans Amount to Medium, and we are done. This is illustrated in the first row of Figure 10, which is the evolving Rule Family named Person Likelihood of Defaulting on a Loan.

On the other hand, if the SME came to us and said that they wished to add a new condition to the logic, in this case the debt-to-income ratio, how would we change Figure 8? Add a new column, with the Fact Type "Total Debt to Income Ratio". Once the column is added, the SME can go row by row and decide what, if any, operator, and fact value to add to each cell for that condition to reach the conclusion in that row.

We can also add extra rows to provide for a new business logic statement that the addition of a condition column may introduce. For example, the SME may say "regardless of any other circumstance, if the credit score is less than 650 we regard the person as highly likely to default on a loan".

This will mean a new row as illustrated in Figure 9; it will also mean that we must ask questions about the impact of the new condition on the other rows. If you review the impact of the credit score on the other rows, you will realize that we will have to ensure a condition that the credit score is equal to or greater than 650 whenever we conclude a person likelihood of defaulting on a loan is medium or better, and we must modify the rows appropriately. See if you can understand how the Rule Family helped us figure out the correct logic to implement the change in the Rule Family in Figure 9.

Conditions								Conclusion	
Total Debt to Income Ratio		Person Credit Score		Person Employment History		Person Other Loans Amount		Person Likelihood of Defaulting on a Loan	
Is Greater Than Or Equal To	80%			Is	Unstable	Is	Medium	Is	High
Is Less Than	80%	Is Greater Than Or Equal to	650	Is	Stable			Is	Low
Is Less Than	80%	Is Greater Than Or Equal to	650	Is	Unstable	Is	Low	Is	Medium
		Is Less Than	650					Is	High

Figure 10 Agility in the Rule Family: Modifying Columns and Rows

3.11 Where Do the Fact Values Come From (Input)?

In reviewing Figure 10, the question arises: where do we get the input for the information in the cells? For example, how do we know that a particular person's employment history is stable or unstable? If the information is readily available – say from a web site, user input, or from a data base (that is, the data is persistently stored somewhere), then it is clear how we get the input (fact values). But what if the information on whether a person's employment history is stable is based on additional business logic? The person's employment history logic can't be assessed in this Rule Family, since this Rule Family has only one conclusion Fact Type, which is not a conclusion Fact Type about a person's employment history.

Business logic that leads to a conclusion of a person's employment history must be structured into its own Rule Family. That Rule Family will have one, and only one conclusion Fact Type, Person Employment History. This is the same Fact Type as the condition Fact Type in the Rule Family in Figure 10. There emerges an inferential dependency between the two Rule Families, where the Rule Family for Person Likelihood of Defaulting on a Loan is dependent upon the supporting Rule Family for Person Employment History. This is illustrated in Figure 11. The conclusion Fact Type of the supporting Rule Family becomes the condition Fact Type of the related column in the dependent Rule Family. We refer to the supporting Rule Family's conclusion value as interim knowledge, since it only exists at the time the Rule Family logic is evaluated and is not maintained in a persistent state.

Conditions		Conclusion
Person Years at Current Employer	Person Number of Jobs in Past Five Years	Person Employment History

Conditions								Conclusion	
Total Debt to Income Ratio		Person Credit Score		Person Employment History		Person Other Loans Amount		Person Likelihood of Defaulting on a Loan	
Is Greater Than or Equal To	80%			Is	Unstable	Is	High	Is	High

Figure 11 Supporting and Dependent Rule Families

Looking at Figure 10 there are other condition columns in the Rule Family for Person Likelihood of Defaulting on a Loan that may depend on Rule Families other than that for Person Employment History. The Fact Types – Person Credit Score and Person Other Loans Amount – appear to have dependencies on other Rule Families.

Consider a Rule Family with many columns and many rows, where a significant proportion of the columns are dependent on supporting Rule Families. It would be difficult to conveniently display these Rule Families such that the viewer could easily trace, review, and understand the logic. To solve this important problem, TDM provides a graphical representation of the logic, providing both a summary overview illustrating the relationship between the Rule Families of each decision, as well as a method to drill into the detail of any of the Rule Families.

4 The Decision Model From the Top Down

The Rule Family structure, and how dependencies exist among amongst the Rule Families should now be clear. It is possible and advantageous to diagrammatically represent the Rule Families and their dependencies, but to do that we have to the scope to be represented. What should the scope of a single Decision Model? Do we put all the Rule Families in our project, in our department, in our business unit, in our enterprise, into a single Decision Model? Or is there some way to create a scope for the Decision Model that is meaningful and more manageable?

TDM defines the scope of a single model as that which the business wishes to manage as a discrete piece of business logic, as a "Business Decision"

4.1 What Is a Business Decision?

We define a business decision in logical and business terms:

- A business decision is a conclusion that a business arrives at through **business logic** and which the business is interested in **managing**.

A business decision is the complete group of business logic statements that lead to a conclusion which the business is interested in managing. Look at the examples of Business Decisions in Figure 12. Each of these Business Decisions is about an issue that is important in an organization's operations, and on which management wishes to focus. Hopefully there are business metrics that would be tracked to ensure that each decision – most presumably automated, but some possibly manual – yield the results that are consistent with the organization's plans and objectives.

For example, the results of business decisions (probably to be made within automated business processes) may decide the profitability or another measure of success.

4.2 What Is the Difference Between a Business Decision and a Conclusion?

A business decision, like a Rule Family, is a conclusion about a single Fact Type. It relies for its logic on a Rule Family that has a conclusion Fact Type of the business decision Fact Type. We call this Rule Family the Decision Rule Family.

Business Decision
Claim Payment Amount
Claim Payment Eligibility
Customer Likelihood of Loan Default
Insurance Policy Renewal Method
Inventory Item Minimum Stock Level
Customer Loan Prequalification Requirements
Person BMI (Body Mass Index)
Vendor Performance Index

Figure 12 Examples of Business Decisions

4.3 Business Decision Scope

A business decision is also the scope of a Decision Model, because it is a meaningful group of logic, from a business perspective, for the business to manage. A Decision Model begins with the Decision Rule Family, its supporting Rule Families and their supporting Rule Families, and the model is complete when with there are no further supporting Rule Families necessary to reach the conclusion of the decision.

4.4 The Decision Model Defined: Let's Bring Everything Together From the Top Down

Now that we have defined TDM intellectually, we can create a graphic representation of it. A complete graphical model of a Decision Model is shown in Figure 13 and Figure 14.

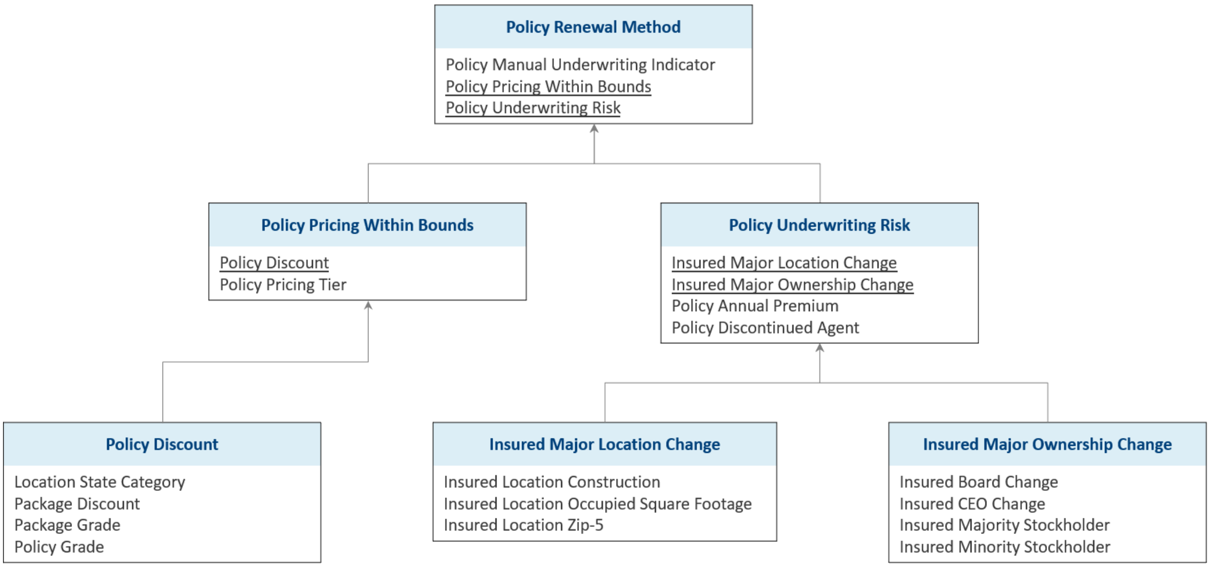


Figure 13 Example of The Decision Model Notation

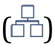
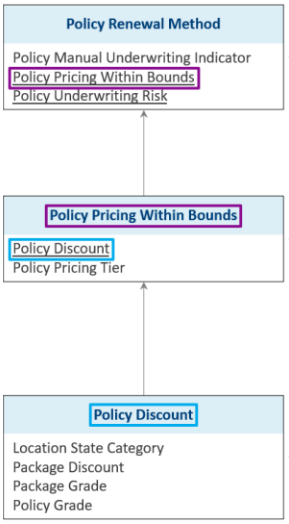
A Decision Model is represented with a simple tree icon (). This enables us to mark business decisions in other business models – such as business process models, use cases, activity diagrams, and the like. We can use this shape as an icon in these other models to indicate the existence of a business decision, whereby the logic for that business decision exists in a Decision Model external from the other models. This promotes reusability of Decision Models across many models and across different types of models. The Decision Model diagram depicts the structure of the model not the details; it does not include the business logic statements themselves. As you have already seen, those are maintained in the Rule Family tables.

Figure 13 illustrates the TDM diagram for the business decision Determine Policy Renewal Method. The Decision Rule Family is called Policy Renewal Method. Recall that the name of the Rule Family is the name of the conclusion Fact Type. Dependent condition Fact Types are shown underlined; condition Fact Types that have no dependency have no underlining. The Decision Rule Family Policy Renewal Method has two dependent Fact Types, Policy Pricing Within Bounds, and Policy Underwriting Risk. We would expect to see supporting Rule Families by those names, and so the inferential dependency connectors lead to the supporting Rule Families.

Figure 14 shows the Rule Family structures represented by the branch of the Decision Model having to do with the Policy Pricing Within Bounds Fact Type.



Policy Manual Underwriting Indicator	Policy Pricing Within Bounds	Policy Underwriting Risk	Policy Renewal Method	Informational
Is ⁝ Off	Is ⁝ Yes	Is ⁝ Acceptable	Is ⁝ Automatic Renewal Process	-
Is ⁝ On	-	-	Is ⁝ Manual Renewal Process	"Policy Manual Underwriting Indicator"
-	Is ⁝ No	-	Is ⁝ Manual Renewal Process	"Policy Pricing out of bounds"
-	-	Is ⁝ Unacceptable	Is ⁝ Manual Renewal Process	"Policy Underwriting Risk is unacceptable"

Policy Discount	Policy Pricing Tier	Policy Pricing Within Bounds
-	Is Less Than or Equal To ⁝ 1	Is ⁝ No
Is Less Than or Equal To ⁝ 10%	Is Greater Than ⁝ 1	Is ⁝ Yes
Is Greater Than ⁝ 10%	Is Less Than or Equal To ⁝ 1.5	Is ⁝ No
Is Less Than or Equal To ⁝ 20%	Is Greater Than ⁝ 1.5	Is ⁝ Yes
Is Greater Than ⁝ 20%	Is Less Than or Equal To ⁝ 2	Is ⁝ No
Is Less Than or Equal To ⁝ 25%	Is Greater Than ⁝ 2	Is ⁝ Yes
Is Greater Than ⁝ 25%	Is Less Than or Equal To ⁝ 2.6	Is ⁝ No
-	Is Greater Than ⁝ 2.6	Is ⁝ Yes

Location State Category	Package Discount	Package Grade	Policy Grade	Policy Discount
-	-	-	Is ⁝ D	Is ⁝ 0%
-	-	Is ⁝ B	Is In ⁝ {B,C}	Is ⁝ 5%
-	-	Is ⁝ A	Is In ⁝ {B,C}	Is ⁝ 10%
-	-	Is ⁝ B	Is ⁝ A	Is ⁝ 15%
Is In ⁝ {B,C,D,E,F}	-	Is ⁝ A	Is ⁝ A	Is ⁝ 20%
-	Is Greater Than or Equal To ⁝ 5%	Is ⁝ A	Is ⁝ A	Is ⁝ 20%
Is ⁝ A	Is Less Than ⁝ 5%	Is ⁝ A	Is ⁝ A	Is ⁝ 25%

Figure 14 Key to The Decision Model Notation

5 TDM Has Principles, and Is Normalized

There are 15 principles that are applied to the Rule Families to ensure rigor in the model. These are broken into three groups:

- Structural Principles – Leading to structural simplicity. These principles define the two-dimensional structure of the Rule Family.
- Declarative Principles – Ensuring the declarative structure. These principles define the declarative technology-independent nature of the structure, ensuring that it is free of any sequential or implementation constraints.
- Integrity Principles – Encouraging optimal logical integrity. It is essential that the business logic in The Decision Model has business and logical integrity. This means that the business purpose has been understood and aligned, and that there are no logic errors, conflicts, or duplication.

One of the most important concepts introduced by the 15 principles is that of normalization, which means decomposing structures into more desirable structures that have greater levels of integrity – in the context of The Decision Model, this means logical integrity. The Decision Model introduces three levels of normalization:

- **First normal form** – enables the business logic to be represented and interpreted in one and only one way.
- **Second normal form** – eliminates redundancies in the business logic statements.
- **Third normal form** – ensures there is no hidden logic within business logic statements.

A Decision Model compliant with these normal forms delivers its logic with maximum integrity and manageability.

6 Integrating TDM with Stochastic Models

As explained in Section 2, TDM is a determinative model that returns a certain result for a given set of inputs. Stochastic models – whether AI or statistically derived models, providing probabilistic results from data – are extremely important tools in Decision Management practice. These models detect patterns in data from which can be derive important and actionable insights. TDM provides the means to implement and operationalize the results from these models, while ensuring that the business can place controls and limits ("guard rails") around the outcomes from these models.

An illustrative example is provided in Figure 15, where a decision uses, as input facts, the outcome from four separate stochastic models to make an insurance underwriting decision. TDM enforces the underwriting rules that the insurance company has filed with the relevant regulatory agencies, but within the business logic, stochastic models are used to determine the risk levels of various aspects of the insurance policy – the driver, vehicle and policy holder credit and insurance risks. These are represented in the respective Rule Families as scores, and are used as inputs, along with other factors (e.g., Driver License status, vehicle registration compliance) to reach a determination.

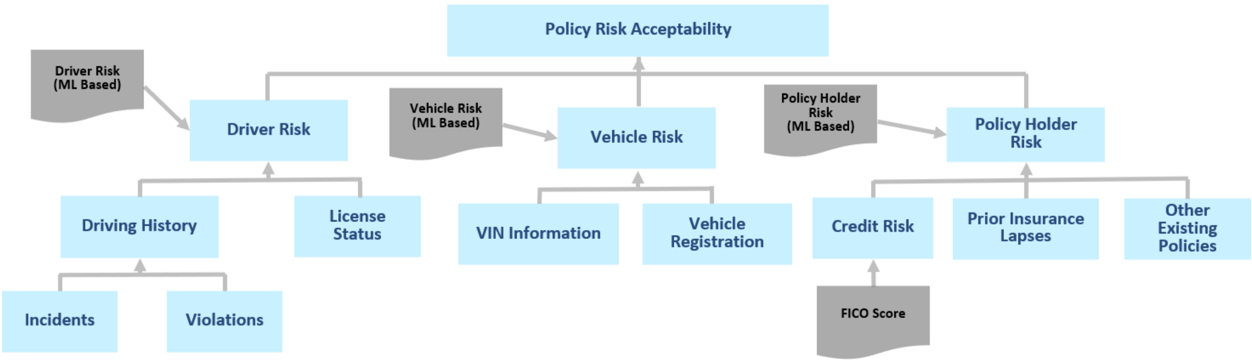


Figure 15 Using Inputs from Stochastic Models in an insurance underwriting decision model.

7 Conclusion

The introduction of TDM, a technology-independent model of business logic, has opened new opportunities for significant improvement in management and information technology (IT) practices.

The ability to relate management objectives directly to business decisions and to manage those business decisions against performance over time has created a new discipline of Decision Management (DM). This new discipline promises dramatic opportunities for increased business agility, particularly in effecting automation and digitalization.

For IT, TDM provides the missing link – a model of business logic. This is the one remaining aspect of business application systems for which IT does not yet have a universally accepted model. Consequently, until now, business and systems analysts bury the business logic in various models such as process models, use cases, activity models; or they list what they consider to be the business rules in catalogs of rule statements using various proprietary (or just random) methods of expression and grouping. The result is that no two models agree, leaving the designers, and ultimately the developers of systems to make their own determination of the correct logic. This results in wasted costs and higher error rates, failures in projects, and application systems that are sub-optimum and lack agility.

TDM has improved a wide range of IT and business practices, including business planning, decision support, Business Process Management (BPM), Business Architecture (BA), Enterprise Architecture (EA), Business Requirements, Business Analysis, Service-Oriented Architecture (SOA), System Testing, and development methodologies both classical and Agile. The Decision Model, and its impact on these practices is the subject of a book published 2009 called "The Decision Model: A Business Logic Framework Linking Business and Technology" (von Halle and Goldberg 2009). Certain diagrams in this document and example Rule Families are drawn from the book and are protected by copyright, © 2009 Auerbach Publications/Taylor & Francis, LLC. Reprinted with permission of the Publisher.

Contact us

For more information, please visit us or contact us at:

www.sapiens.com or info.sapiens@sapiens.com