



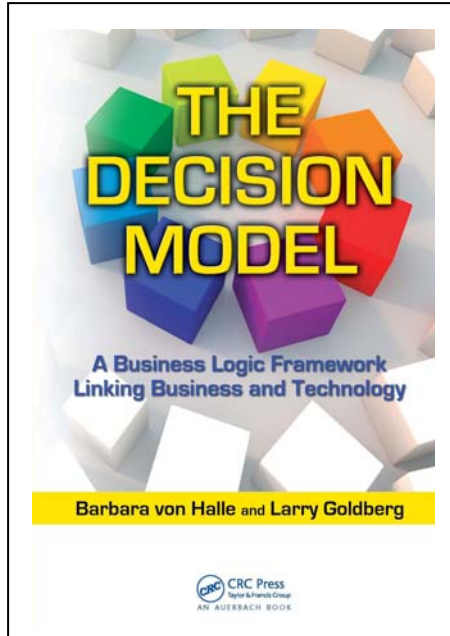
8 Calais Road
Mendham, NJ 07945
Phone: +1 973 543 1339
Fax +1 973 543 7037
www.kpiusa.com

A PRIMER ON THE DECISION MODEL

Copyright © 2009 Knowledge Partners International, LLC.
(Portions of this document are drawn from the book, *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.)

Table of Contents

- Introduction 1
- Decision Model Fundamentals 1
 - What Is Business Logic: Business Logic Fundamentals 1
 - The Structure of Condition and Conclusion Assertions 3
 - What Is a Fact and a Fact Type? 3
 - What Is an Operator?..... 4
 - The Structure of Business Logic as Conditions and Conclusions 4
 - What Is An Atomic Business Logic Statement? 5
 - Why Are Atomic Business Logic Statements Important? 6
 - Grouping business Logic Statements into Rule Families 7
 - Rule Family Structural Features 9
 - What Is a Rule Pattern? 9
 - The Agility of the Rule Family 10
 - Where Do The Fact Values Come From (Input)? 11
- The Decision Model from the Top Down..... 13
 - What Is a Business Decision? 13
 - What Is the Difference Between a Business Decision and a Conclusion? 13
 - Business Decision Scope 14
 - The Decision Model Defined: Let’s Bring Everything Together From the Top Down..... 14
 - How Does The Decision Model Connect To Rule Families?..... 17
- The Decision Model has Principles, and is Normalized 17
- Conclusion 18



Introduction

The Decision Model is a way of representing business logic that is platform and technology independent. It models logic based on the inherent structure of that logic, eliminating style and other subjective preferences, ensuring a consistent and stable representation. This is a brief introduction to the Decision Model that will enable you to read and understand Decision Models, perhaps even develop them yourself.

For details about the Decision Model, 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.

Business logic is often popularly referred to as “business rules”, but we are sure that by the time you have completed this guide you will feel comfortable with our preference for the term “business logic”.

Decision Model Fundamentals

The simplicity and elegance of the Decision Model can be best learned by building the model from the ground up. This typically is not the way you might create a Decision Model in real life, but for illustrative purposes, this approach provides an excellent perspective and understanding of the model.

What Is Business Logic: Business Logic Fundamentals

First, let us define the term “business logic”; Business logic is the means by which the business derives conclusions from facts. So, building from that definition, we may 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 simple diagrammatic representation of a business logic statement in Figure 1.



Figure 1 A Diagram of a Simple Business Logic Statement

This diagram faithfully follows our definition: we see that the condition statement contains a fact, which we evaluate for truth: it leads to the conclusion statement that we infer to be true if the condition statement is true. Thus we are lead to a conclusion of a new fact.

A simpler way of diagramming Figure 1 may be found in Figure2.



Figure 2 Simpler Diagram of a Simple Business Logic Statement

This diagram is really the same as the one in Figure 1, but it simply indicates that the business logic statement is independent of the grammar we may use to express it – so we could use the “when...then” form, we could use the “if..then” form, or we could state it in other declarative forms such as “Theshall..” and so on. We say that the condition is an assertion, and when this condition assertion is true, then the conclusion assertion may be inferred to be true. We may simplify the diagrammatic representation of the business logic statement even further, to its simplest form, as represented in Figure 3 in which the arrow represents the concept of “implies” or “infers” or “concludes.”



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

This diagram is the simplest clearest representation of a business logic statement, and 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 An Example of a Business Logic Statement

In Figure 4 we have a good example of a simple business logic statement. This statement tells us that the business wants to conclude 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 this assertion – or fact – is true, then we know that the assertion “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, then that person is highly likely to default on a loan.” (The *logic* in the natural language statement and the diagram are exactly the same).

The Structure of Condition and Conclusion Assertions

Each assertion is composed of:

- Facts
- Fact Types
- Operators

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

Table 1 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. So the fact value in this statement is “five.” Simple enough. What about the rest of the statement? 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.

Table 1 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.

What Is an Operator?

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

- Conventional symbols, i.e. logic operators, are ones such as:
 - “Is”, “Is Not”, “Is Less Than”, and so on.
- There are many others. Most of them are boolean, as we would use in programming, but they can be any logical operator.

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 specific 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 specific 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:



- Operands are frequently a fact value (from the pool of fact values in the fact type domain), but may be a Formula or a Fact Type.

So, take the assertion:

A person employment history is poor

We can model it as:



Table 2 has some examples of assertions exemplifying their structure. It is interesting to see that 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	$[\text{weight (kg)} / [\text{height (m)}]^2]$
Policy Claim Date	Is Before	Policy Effective Date
Person Eye Color	Is	Blue
Customer Type	Is	Preferred
Inventory Optimum Level	Is	Two Months Sales

Table 2 Examples Showing the Structure of Assertions

Notice how the operand can be a fact value, a fact type, or a formula. Looking at the operands in Table 2, we see:

- **Fact values** are: average, 700, blue, preferred, etc.
- **Fact Types** are: person likelihood of defaulting on a loan, policy effective date, and two months of seasonal sales, etc.
- **Formula** is $[\text{weight (kg)} / [\text{height (m)}]^2]$

What Is An Atomic Business Logic Statement?

1

In the Decision Model, it is critical that business logic statements be atomic.

What is the meaning of the word “atomic” and why is it important?

Well, first the meaning: 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 simply 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.

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 and also eliminates ambiguity of meaning: the latter 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.

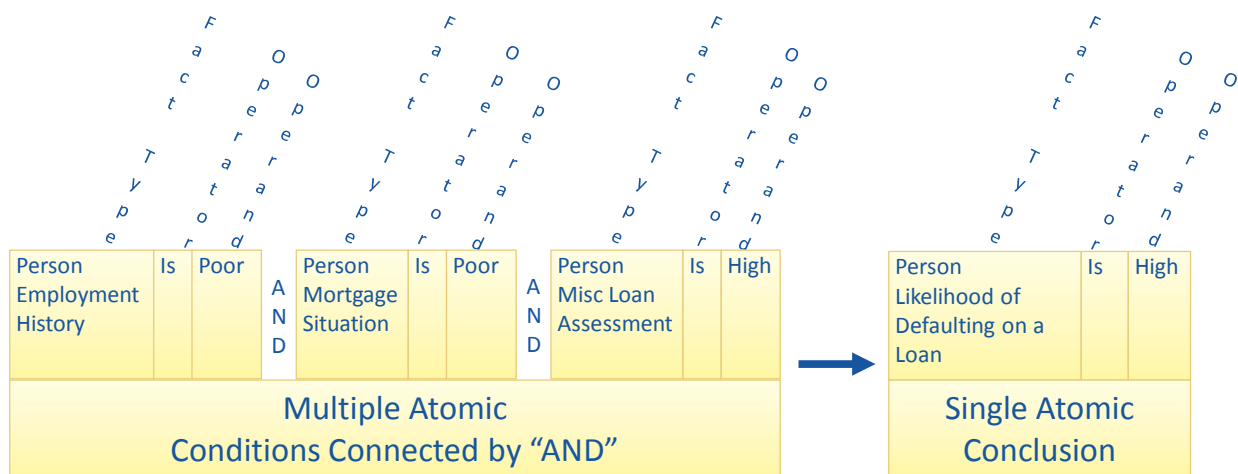


Figure 5 Diagram of an Atomic Business Logic Statement

Figure 5 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 are atomic.

If we have definitions for 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 using a grammar standard such as Semantic Vocabulary for business Rules (SVBR) as published by the Object Management

Group (OMG)). As examples, here are a few possible natural language expressions of the business logic statement:

- If/when a person's employment history is poor and the person's mortgage situation is poor and the person's miscellaneous loans assessment is high then (we judge) the person's likelihood of defaulting on a loan is high.
- A person with poor employment history and poor mortgage situation and high miscellaneous loans assessment has a high likelihood of defaulting on a loan.
- A person has a high likelihood of defaulting on a loan if all of the following are true:
 - Person's employment history is poor.
 - Person's mortgage situation is poor.
 - Person's miscellaneous loans assessment 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 business people would have to train in the grammar to write it). 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. Let's look at how the Decision Model groups the business logic behind the natural language forms.

Grouping business Logic Statements into Rule Families

It is clear from the example in Figure 5 that there are likely to be more business logic statements that conclude a person's likelihood of default beyond the statement in the diagram. We need only look to the domain of the fact type Person likelihood of Defaulting on a Loan to understand that there are conditions that lead to a conclusion where the person's likelihood of defaulting has a conclusion other than high. (Table 1 illustrates a domain for the conclusion fact type, showing values high, medium and low. We should look for conditions that will result in all these values).

Figure 6 illustrates a grouping of three business logic statements with the conclusion fact type person likelihood of defaulting on a loan. This appears, on the face of it, a fairly convenient grouping. However, you can imagine that the logic leading to some conclusion fact types can run into many, sometimes tens, and possibly hundreds of business logic statements. That being the case, figure 6 may not be the most convenient representation for this grouping of business logic statements.

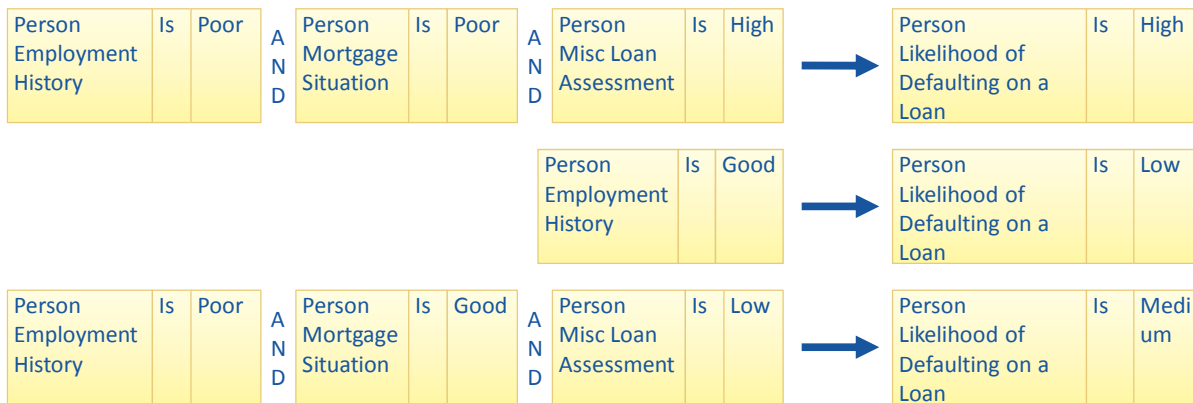


Figure 6 Grouping Business Logic Statements

Rule Pattern	Conditions						Conclusion	
	Person Employment History		Person Mortgage Situation		Person Miscellaneous Loans Assessment		Person Likelihood of Defaulting on a Loan	
1	Is	Poor	Is	Poor	Is	High	Is	High
2	Is	Good					Is	Low
1	Is	Poor	Is	Poor	Is	Low	Is	Medium

Table 3 Grouping Business Logic Statements in Rule Families

Now look at Table 3: here is that same grouping of business logic statements in a single tabular form. This is a more efficient representation of the business logic statements than that in Figure 6. In fact, when this grouping of business logic statements complies with certain principles that bring rigor to the form, this is called a Rule Family, the primary structure of the Decision Model. 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.

Rule Family Structural Features

Figure 7 illustrates the structural features of the Rule Family: it is a two-dimensional structure with rows and columns: the header of the columns is condition facts types (called condition columns) and the conclusion fact type (called the conclusion column). Each row is a business logic statement. Our practice is to place the conditions in the left columns, and the single conclusion on the right, although in fact, 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 they are in. The condition columns are considered to be 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).

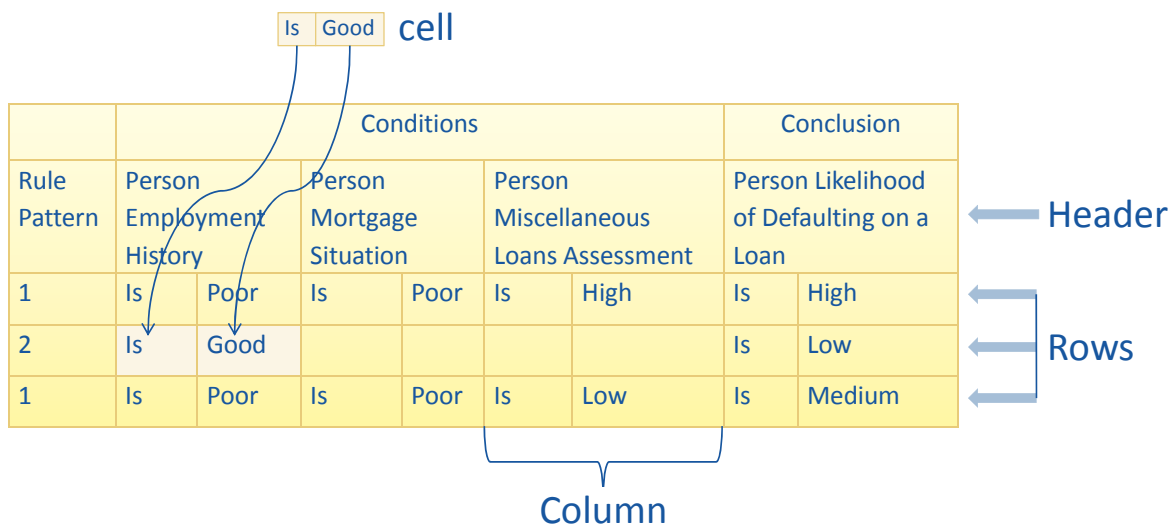


Figure 7 Features of the Rule Family

The name of a Rule Family is simply its conclusion fact type. So, the name of the Rule Family in Table 3 and Figure 7 is Person Likelihood of Defaulting on a Loan.

What Is a Rule Pattern?

One feature that should be pointed out in Table 3 (and in Figure 7) is the Rule Pattern. A Rule Pattern is a set of Rule Family rows with a common set of populated condition cells. We give each separate Rule Pattern a unique number or name. So you can see that Rule Pattern 1 in Table 3 (and Figure 7) refers to rows that have all three condition columns populated, while Rule Pattern 2 refers to rows that are populated only in the column headed by Person Employment History. The Rule Pattern is an important element in the Decision Model, and is

used in validating compliance to many of the principles that bring integrity and normalization to the logic.

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. Of course, the Rule Family is not a “mere” decision table. It is a decision table on steroids! By this, we mean it is governed by principles which give it rigor, and which you will learn about as go along.

Rule Families are also very easy to modify. The ability to easily modify business logic is a key capability to making an enterprise more competitive. In Table 3, the first row tells us (as we noted above) that:

- A person has a high likelihood of defaulting on a loan if all of the following are true:
 - Person’s employment history is poor.
 - Person’s mortgage situation is poor.
 - Person’s miscellaneous loans assessment is high.

Now 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 poor, and the person’s mortgage situation is poor, then even when a person’s miscellaneous loans assessment is medium (and not high, as in the Table 3), 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 Miscellaneous Loan Assessment to Medium, and we are done. This is illustrated in the first row of Table 4, which is the evolving Rule Family, 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 an outside credit rating, how would we change Table 3? We need simply to add a new column, with the appropriate heading, in this case the fact type “Person Outside Credit Rating”. 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 outside credit rating 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 Table 4; it will also mean that we have to ask questions about the impact of the new condition on the other rows. If you review the impact of the outside credit score on the other rows, you will realize that we will have to ensure a condition that the score is equal to or greater than 650 whenever we conclude a person likelihood of defaulting on a loan is medium or better. So we have to 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 Table 4. Also note the impact of the changes to the Rule Patterns.

Rule Pattern	Conditions								Conclusion	
	Person Employment History		Person Mortgage Situation		Person Miscellaneous Loans Assessment		Person Outside Credit Score		Person Likelihood of Defaulting on a Loan	
1	Is	Poor	Is	Poor	Is	Medium			Is	High
2	Is	Good					Is equal to or greater than	650	Is	Low
3	Is	Poor	Is	Poor	Is	Low	Is equal to or greater than	650	Is	Medium
4							Is less than	650	Is	High

Table 4 Agility in the Rule Family: Modifying Columns and Rows

Where Do The Fact Values Come From (Input)?

In reviewing Table 4, 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 good, medium or poor? 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 good is based on additional business logic? The person’s employment history logic can’t be 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.

So, it follows that, if there is business logic that leads to a conclusion of a person’s employment history, then that business logic can be structured into its own Rule Family. And, regardless of the business logic, we already know that such a Rule Family will have one, and only one conclusion fact type, person employment history, the same fact type as the condition fact type in the Rule Family in Table 4. So 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 8. 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 executed.

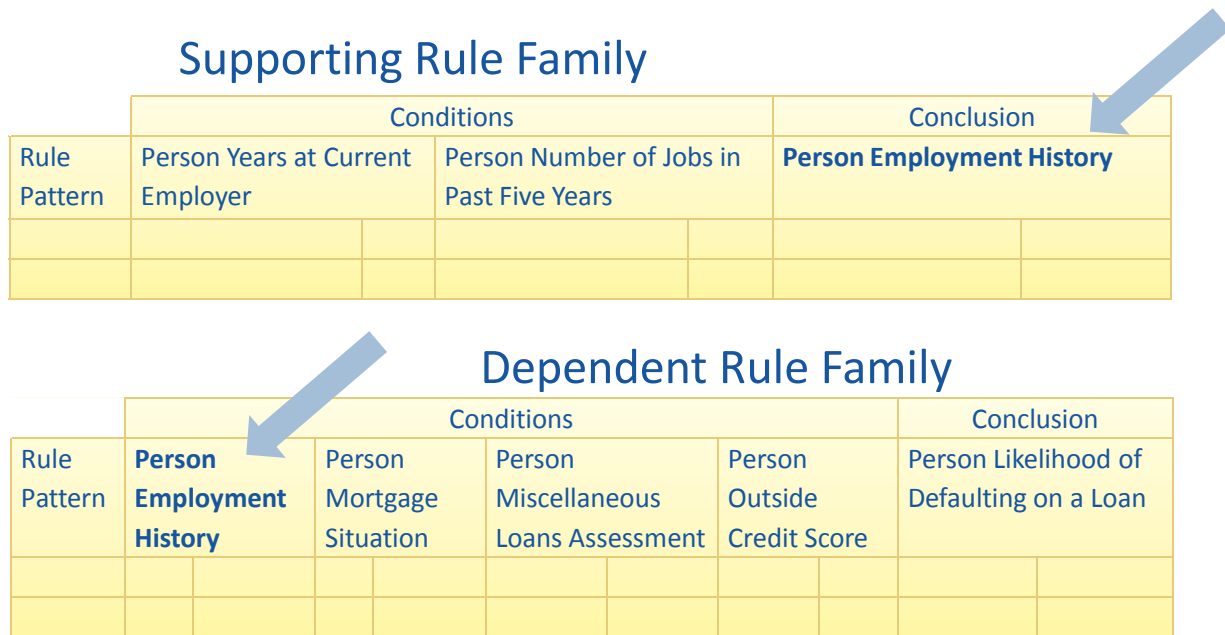


Figure 8 Supporting and Dependent Rule Families

It is also clear when looking at Figure 8 that 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. Both fact types – person mortgage situation and person likelihood of defaulting on a loan – may have dependencies on Rule Families. It would soon become difficult to illustrate Rule Families and their dependencies, particularly for Rule Families with a large number of rows, and a large number of dependencies. Ideally, we would

like to represent all the logic in a single, easy to understand graphic. This leads us to the Decision Model, which is designed to provide us with that view.

It is now convenient to consider how to represent the whole Decision Model.

The Decision Model from the Top Down

At this point you should have an understanding of how the Rule Family is structured, and how we create dependencies among the Rule Families. It is possible, and advantageous, to diagrammatically represent the Rule Families and their dependencies, but to do that we have to come to a determination about scope: how do we decide which Rule Families belong in 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?

The Decision Model achieves this by relying on the structure of the logic itself, while at the same time giving weight to business expediency. To understand this, we have to understand the term “business decision”.

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**.

We see that the definition of business decision only differentiates it from a business logic statement (which after all, arrives at a conclusion through business logic) by the qualification that it is at the granularity at which the business is interested in managing. Look at the examples of Business Decisions in Table 5. 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.

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

In Table 5 the only thing that distinguishes the business decision from its related fact type is the “decision word” in front of the fact type (we underlined the decision word just in case you didn’t notice!). So a business decision, like a Rule Family, is a conclusion about a single fact

type. In fact, 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	Fact Type
<u>Estimate</u> the claim payment amount	Claim Payment Amount
<u>Determine</u> Claim Payment Eligibility	Claim Payment Eligibility
<u>Determine</u> Customer Likelihood of Loan Default	Customer Likelihood of Loan Default
<u>Determine</u> insurance policy renewal method	Insurance Policy Renewal Method
<u>Assess</u> the Inventory Item minimum stock level	Inventory Item Minimum Stock Level
<u>Determine</u> loan prequalification requirements for a customer	Loan Prequalification Requirements
<u>Calculate</u> Person BMI	Person BMI (Body Mass Index)
<u>Calculate</u> the Vendor Performance Index	Vendor Performance Index

Table 5 Examples of Business Decisions

Adapted from: The Decision Model: A Business Logic Framework Linking Business and Technology, von Halle & Goldberg, © 2009 Auerbach Publications/Taylor & Francis LLC. Reprinted with the permission of the Publisher.

Business Decision Scope

A business decision is also the scope of the Decision Model, because it is a meaningful group of logic, from a business perspective, for the business to manage. So the Decision Model begins with the business decision, followed by 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.

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

Now that we have defined the Decision Model intellectually, we can create a graphic representation of it. A complete graphical model of a Decision Model is shown in Figure 9. Figure 10 and 11 provide an explanation and key to the notation of the model.

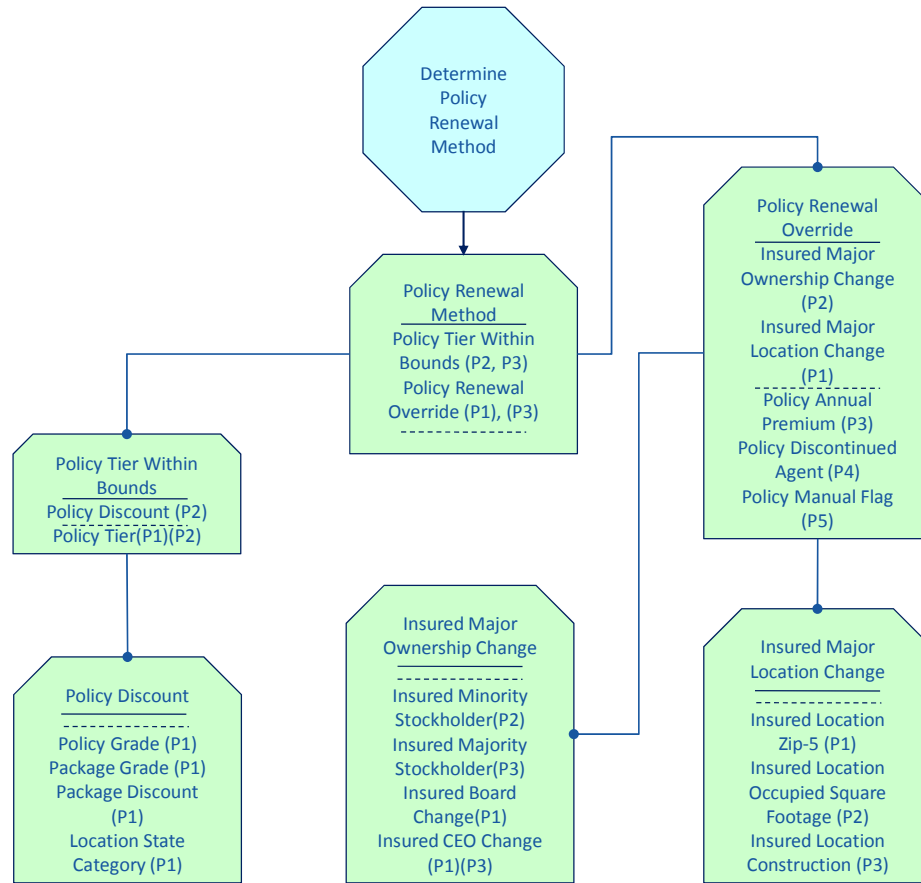


Figure 9 Example of Decision Model Notation

Source: The Decision Model: A Business Logic Framework Linking Business and Technology, von Halle & Goldberg, © 2009 Auerbach Publications/Taylor & Francis LLC. Reprinted with the permission of the Publisher.

In creating the Decision Model graphic representation, the shape we use to signify the business decision is an octagon. 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 decisions exists in a Decision Model external from the other models. This promotes reusability of Decision Models across many models and of different types. 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 9 illustrates the Decision Model diagram for the business decision Determine Policy Renewal Method. The business decision is denoted by the decision shape, the octagon. The business decision is connected to the Decision Rule Family called Policy Renewal Method. Recall that the name of the Rule Family is the name of the conclusion fact type, and is displayed above

the solid line. Dependent condition fact types are listed below the solid line, and above the dotted line; condition fact types that have no dependency are listed below the dotted line. So the Decision Rule Family Policy Renewal Method has two dependent fact types, Policy Tier Within Bounds, and Policy Renewal Override. We would expect to see supporting Rule Families by those names, and so the inferential dependency connectors lead to the supporting Rule Families.

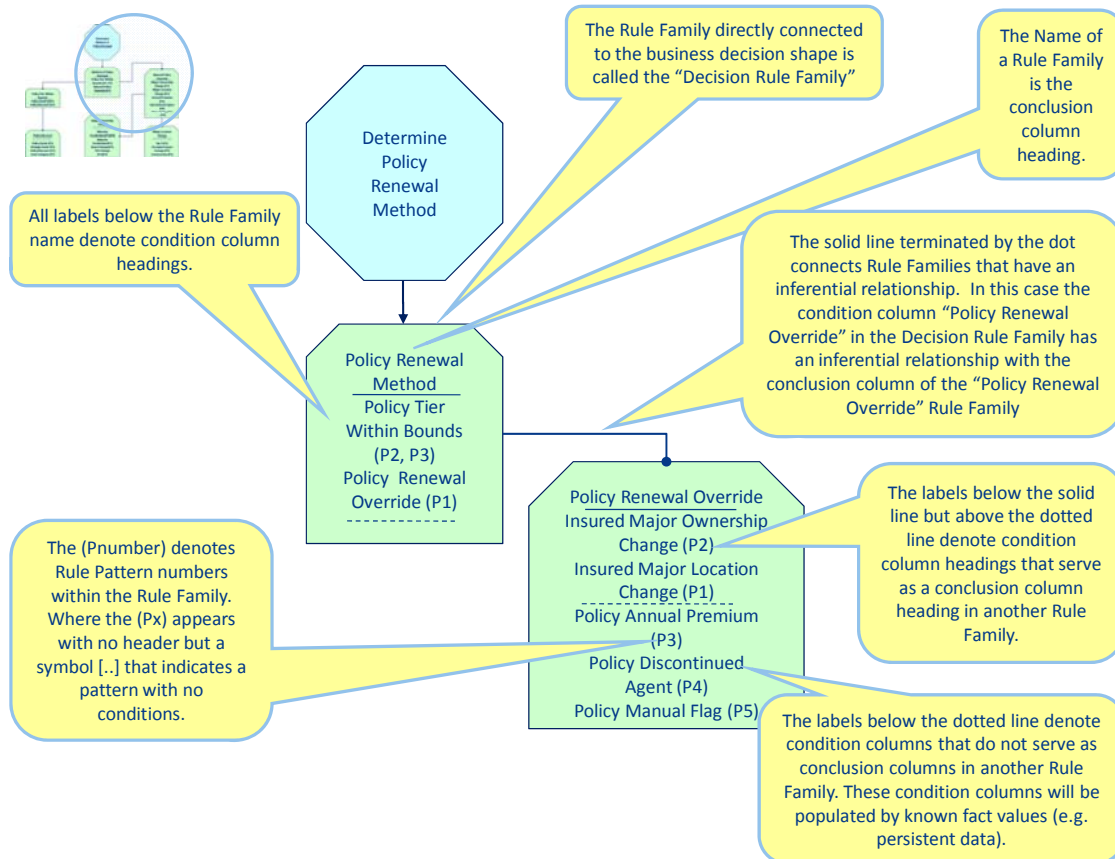


Figure 10 Key to Decision Model Notation – Part 1

Source: The Decision Model: A Business Logic Framework Linking Business and Technology, von Halle & Goldberg, © 2009 Auerbach Publications/Taylor & Francis LLC. Reprinted with the permission of the Publisher.

Figure 10 is a detailed description of the Decision Model, zooming in on the area of the model depicted by the icon in the top right corner of the diagram. This diagram provides detailed explanations of the Rule Family shapes, including the connections among them and the notation for patterns.

How Does The Decision Model Connect To Rule Families?

Figure 11 highlights a different Decision Model area of detail, illustrating the relationship between the Decision Model diagram, and the populated Rule Families and detailed business logic statements that underlie the diagram.

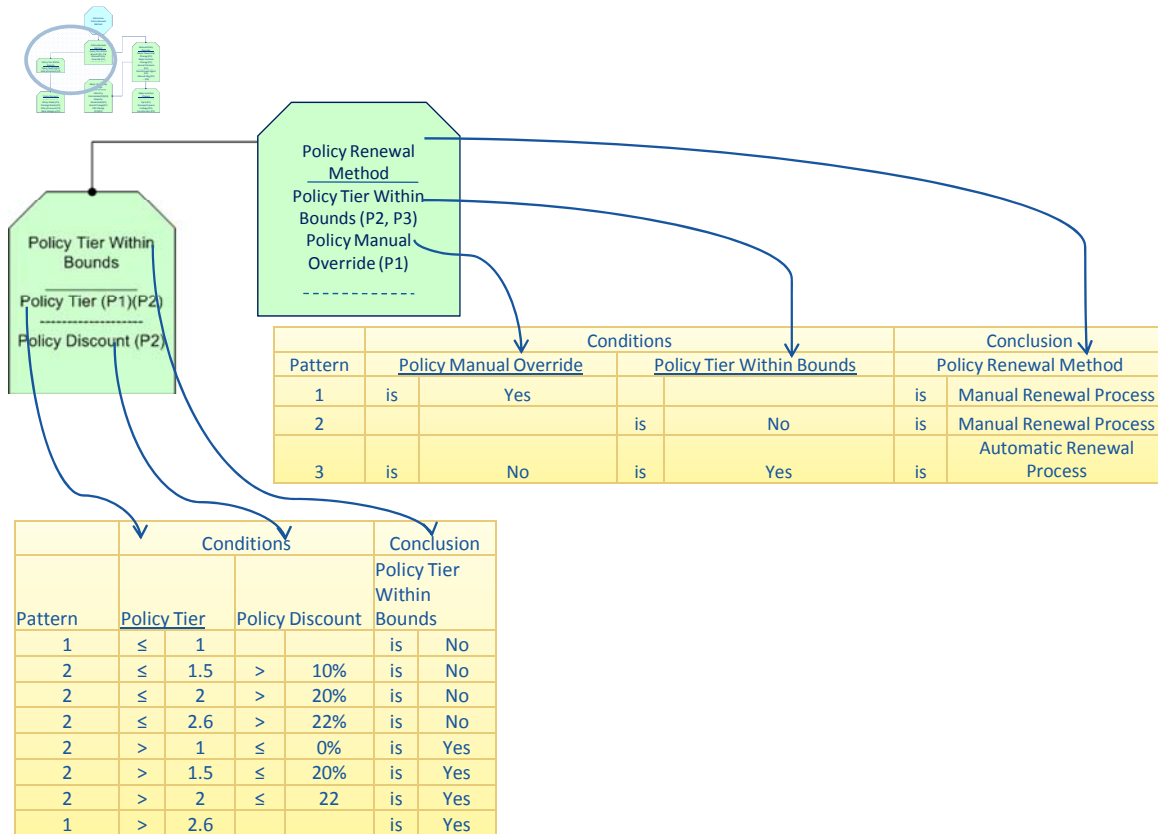


Figure 11 Key to Decision Model Notation - Part 2

Adapted from: The Decision Model: A Business Logic Framework Linking Business and Technology, von Halle & Goldberg, © 2009 Auerbach Publications/Taylor & Francis LLC. Reprinted with the permission of the Publisher.

The Decision Model 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 that logic with maximum integrity and manageability.

Conclusion

The introduction of the Decision Model, a technology independent model of business logic, has opened up 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) (sometimes called Business Decision Management (BDM) or Enterprise Decision Management (EDM)). This new discipline promises dramatic opportunities for increased business agility.

For IT, the Decision Model 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.

The Decision Model promises to improve 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 these practices is the subject of a book to be published in late October of 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 copyright © 2009 Auerbach Publications/Taylor & Francis, LLC. Reprinted with permission of the Publisher. The book may be found at www.TheDecisionModel.com.