blueriq
TSS
MAKE YOUR OWN RULES

Blueriq embraces the decision model

By Maarten P.D. Schadd
Product Consultant at Blueriq B.V.

Contents

Abstract.

Blueriq is a provider of a Business Process Management (BPM) software solution that enables organizations to develop and maintain critical applications. Application development in Blueriq is done through modeling (not programming) by analysts. Blueriq comprises a complete business rule management suite that allows modeling and execution of business rules. When business logic becomes complicated, overview is easily lost. The decision model is a template that structures your decisions and provides an overview of the business decisions. This article describes in detail how the Blueriq suite adapts concepts of the decision model to support analysts in developing applications for the business. We will demonstrate that the decision model improves implementation speed and reduce maintenance costs.

1 Introduction

The services that organizations offer customers are increasingly becoming more dynamic. While thirty years ago a customer had to adhere to pre-defined processes that were carried out by hand, now the customer wants the process to be customized to his or her needs and to be executed at a high pace. As these processes are knowledge-intensive it is vital to an organization to automate them in order to deliver a high level of service, in terms of speed and of consistency among customers [9]. This area of expertise is called Dynamic Case Management (DCM)[1] [8].

In a DCM modeling suite analysts can define knowledge-intensive processes without the need for programming. These processes are controlled by business rules that have to be defined and executed. Such business rules are stored in a Decision Management System (DMS).[2] The great advantage of defining a process and business rules within a DMS system is that these components are defined explicitly and are not hidden away in code. This improves transparency and development speed while reducing maintenance costs. Sometimes even the business users themselves can take control of implementing and maintaining their own business rules in the system, making business users less reliant on external parties. Blueriq[3] is such a DCM modeling suite with an integrated DMS system. Founded in 1993 it recently released version 9.2 of its software, providing its customers even more options for modeling dynamic processes.

A DMS system gives business engineers all the options they need to accurately model the business logic. Business engineers meet regularly with business experts to clarify policies and regulations and other business logic. In order to validate the model they developed, business engineers want to present it to business users. This implies that the DMS system should be able to present the business logic in a way that is understandable to business users. Business users should be able to understand (1) a single business rule as well as (2) the relation between different rules. (1) An executable business rule may be too technical for business experts as it could be a piece of code or in a special form to make it executable. However, it can be translated to a more understandable form, such as natural language, so that business experts understand it. The problem with natural language is that it is ambiguous. A standardized way of describing natural language rules to avoid ambiguity is the Semantics for Business Vocabulary and Business Rules (SBVR). It is an Object Management Group standard for the formal and detailed natural language description for complex business rules [3, 5]. SBVR enables business experts to easily understand the implemented business rules. (2) A business rule is a single statement describing a single aspect of the business. The collection of all rules create a complicated system as the outcome of one rule is used in one or more other business rules. These dependencies between business rules are often implicitly saved in the head of the business engineer. Dependency viewers often only show a long lists of elements that have some relation to this rule but are unable to visualize dependencies and give deeper dependencies.[4] In order to make this explicit and understandable to business experts, the decision model has been developed.

The Decision Model [10] 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 presentation style and other subjective preferences, ensuring a consistent and stable representation. Not only does the decision model apply restrictions on the structures of the logic, it also provides a standardized overview of a single business decision in a decision model diagram.[5] By defining the business logic in such a structured way it is possible to discuss the logic with business experts and

[1] Also called Adaptive Case Management (ACM) or Advanced Case Management
[2] This is similar to a Business Rule Management (BRM) but with a strong focus on decisions to be made.
[3] www.blueriq.com
[4] The dependencies of the dependent elements, etc.
[5] Also called decision requirements diagram.

even to let the business experts create their own logic. This reduces errors and misinterpretations that frequently occur in a natural language approach with a specifications document. The Object Management Group has received a proposal for the standardization of a decision model notation [6]. The decision model [2, 4] has been shown to not only be compatible with SBVR, but also with the Business Process Management Notation (BPMN) [11], a standardized way to define business processes.

This article describes how Blueriq embraces the decision model in its integrated DMS system. There are several reasons for choosing to investigate the decision model. It is in the process of becoming an accepted standard of the Object Management Group and it is a promising addition for Blueriq's modeling environment. Currently, analysts working with Blueriq have to do quite some work to gain an overview of a decision and this can be automated with a decision model diagram. Furthermore when this notation is used in other systems as well, the decision model provides a good basis for analysts that are not familiar with Blueriq. We show how business logic can be modeled in Blueriq and matches restrictions of the decision model. Blueriq is able to model decision logic in accordance with the principles of the decision model and is able to display a decision model diagram based on the modeled logic. This enables Blueriq to create an executable decision model. While the decision model is intended for the design and maintenance of logic, we also investigate the application to the execution of business rules.

2 Management Summary

This article describes how the decision model can work together with Blueriq, a provider of a Business Process Management (BPM) software solution. We give an introduction to the decision model and the principles that steer business engineers towards making precise and correct business rules. The Blueriq dynamic case management suite and its rule engine are introduced that is able to model executable business logic with decision tables, business rules and expressions.

We demonstrate that Blueriq is able to model business logic according to the principles of the decision model. Blueriq allows business engineers much more freedom than the decision model and does not follow the decision model notation for decision tables. An advantage of this is that the expressiveness of Blueriq is high, however, a disadvantage may be that business logic is more difficult to be maintained. An analyst is free to follow the principles of the decision model within the possibilities of Blueriq. A big advantage of creating a decision model in Blueriq is that it is directly executable. The decision model is designed in a way so that it is easy to be implemented once created. In Blueriq there is no additional effort required after creating the decision model.

Furthermore we show that a decision model diagram can be created from the business logic elements in Blueriq. This allows analysts to gain an overview of the implemented business logic. The decision model diagram is much stronger than a general dependency viewer because it is a recursive approach. Without the decision model diagram it can be a long manual task to create an overview of a decision at hand.

Next to a decision model diagram, based on the design of business logic, Blueriq is also able to create such a diagram on the actual execution of business logic. This diagram shows the user the logic components that were actually used in a decision improving the traceability of the application. Blueriq is the first application that is able to create a decision model diagram based on the actual execution of business logic.

We believe that the decision model is a valuable approach in creating business logic. This model is understandable by business users as well as the IT department and promotes a standard form of decisions across applications. With the decision model the implementation speed is increased and maintenance costs are reduced.

3 Modeling Decisions in Blueriq

Blueriq is a provider of an innovative rule-driven Business Process Management (BPM) software solution that assists corporations in the modeling, automation and integration of intelligent business processes. Using Blueriq, organizations improve business productivity while reducing business costs by implementing agile model-driven applications that quickly respond to changing business conditions. Blueriq's products empower knowledge workers and allow business users to control the process management life-cycle and outcomes facilitated by IT.

In Blueriq, business engineers model business rules in a studio environment that generates an export that is interpreted by the Runtime. The architecture of Blueriq is shown in Figure 1.

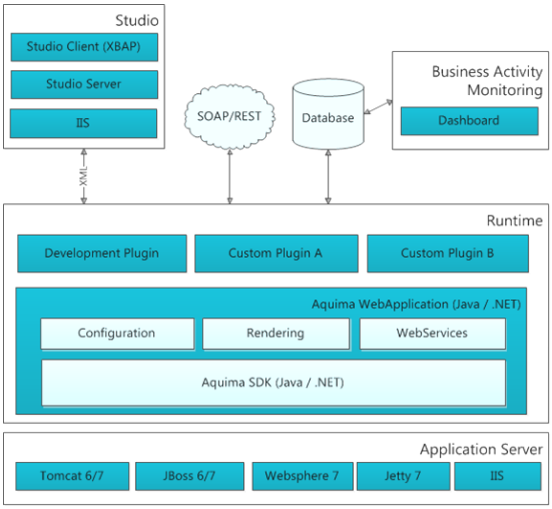

Figure 1: The Blueriq Architecture

In Studio executable business rules are modeled using logical components and an expression language. The Runtime executes the Business Rule Engine (BRE) to determine the outcome of the modeled business rules. This is done by using backwards chaining [7], i.e., when a value for an attribute is needed the BRE looks for a rule to derive the value. If a rule is found, which is again dependent on other rules, then these are also executed recursively. This article only discusses the modeling part of the business logic and we consider the modeled rules to be executable by the Runtime.

Blueriq distinguishes three logical components: (1) decision tables, (2) business rules and (3) expressions.

Decision Tables

The use of Decision tables became popular in the early 1960s as an alternative to flowcharts to describe business logic [1]. A decision table is a compact way to model complicated logic in a tabular way. A decision table consists of 4 elements; (1) conditions (e.g., Debt), (2) condition alternatives (e.g., High), (3) actions (e.g., Credit Rating), and (4) action entries (e.g., F). By adding more entries to any of these 4 elements the decision table gets larger and more complicated. Figure 2 shows an example of a decision table in Blueriq to determine the credit rating of a customer.

In this decision table you can derive what credit rating a customer should get (action) based on the conditions of employment history and debt rating. For example, if the condition alternatives are a poor employment history and a high debt he will get the action entry rating "F". The asterisk (*) means that the value of Person.Dept is irrelevant if the employment history is excellent.[6] Please note that the decision table does not have to be complete. This table is unable to determine a rating if the employment history is good and the debt is high. In this case other

[6] This is a fictitious example, and may not be realistic.

Figure 2: A Blueriq decision table to determine a person's credit rating

decision tables or business rules are examined to find the credit rating. Please also note that the rating for the customer's employment history and dept can again be dependent on other decision tables. The backwards chaining mechanism will then first try to determine those values before executing this decision table.

Business Rules

A business rule is a rule that constrains some aspects of the business. In Blueriq these are formulated as IF...THEN...IS... statements. An example of such a business rule is: *IF Person.EmploymentHistory = Excellent THEN Person.CreditRating IS "A"*.[7] A single business rule covers only one scenario and the backwards chaining mechanism will keep looking for a logic component that is able to derive the needed value until one is found. If no value can be derived the special value *unknown* is assigned.

Please note that a decision table can be represented as a set of business rules and a business rule can be represented as a small decision table. It is up to modelers to decide which system they prefer. They will usually choose the decision table if a set of rules is similar and will choose a business rule if no other rules are similar and if it is not expected that similar rules are added in the future.

Expressions

An expression is a reusable element that produces a value and that can be used inside decision tables, business rules, or even other expressions. An example is the calculation of the age of a Person based on his birthday. A Blueriq expression called {Age} could be: YEARS BETWEEN TODAY AND Person.DateOfBirth. From now on {Age} can be used on other places to refer to a person's age.

4 The Decision Model

The decision model [10] defines a technology-independent model for organizing business logic. It purely concerns the business logic, avoiding biases from processes, data or technology. This model provides a clear view on a decision to both business experts and technology experts. This means that business users can easily manage the business decision, while developers are able to quickly implement this model.

Essentially, the decision model consists of two components. (1) 15 principles to ensure that business logic is defined rigorously and (2) a graphical notation to visualize how a decision is

[7] An ELSE can be modeled by a second business rule using the negated expression.

made in a top-down manner showing all sub-decisions belonging to the decision in question. Next to these two components a decision model notation is introduced to standardize the way of representing business logic in a tabular form and for the graphical representation.

Before discussing the principles in detail we want to familiarize you with some terms of the decision model. A **Rule Family** is the name of the collection of all rules that concern the same conclusion. This is like a decision table. A **Rule Pattern** is a pattern of conditions to derive a conclusion. For example, to derive the customer credit rating by income and by total depth is Pattern 1, while deriving the customer credit rating by the number of missed payments is Pattern 2. A **Fact Type** is a general piece of information, not the piece of information itself (e.g., the age of a person).

The 15 principles

This section provides a short introduction to each of the 15 principles that you should obey when modeling business logic. These 15 principles enforce limitations on the way you model business logic to ensure that your logic is consistent and precise. The 15 principles are presented in Table 1 and will be revisited in more detail in Section 5.

#	Name	Description
1	Tabular Principle	Rules are described in a tabular way called a Rule Family.
2	Heading Principle	The headings of a Rule Family are fact types.
3	Cell Principle	A cell contains an atomic logical expression, belonging to the heading.
4	Row Principle	The condition cells infer the corresponding conclusion cells in a row.
5	Conclusion Principle	There is only one conclusion fact type (column)
6	Conditions Principle	If all conditions are true then the conclusion is true. It is possible to draw a conclusion without conditions. The conclusion should be dependent on all conditions.
7	Connection Principle	Rule families are related if the conclusion in one is the condition in another.
8	Declarative Heading Principle	There is no order in the columns of a rule family.
9	Declarative Body Principle	There is no order in the rows of a rule family.
10	Declarative Inferential Relationship Principle	There is no predefined order in which rule families should be used.
11	Rule Pattern Transitive Conditions Principle	The condition columns should be independent from each other.[8]
12	Rule Family and Rule Pattern Consistency Pattern	The rule family should be free of inconsistencies.
13	Rule Family Transitive Conditions Principle	Remove transitive dependencies between rule families.
14	Inferential Integrity Principle	All outcomes of a rule family should be covered within every dependent rule family.
15	Business Alignment Principle	The decision model should fulfill a business need and should be connected to metrics.

Table 1: The 15 principles of the decision model

Decision Model Diagram

A decision model diagram is a graphical representation of the underlying dependencies of the business logic. Starting from the decision to be taken, it shows what needs to be decided up to the point that only simple input is required to make a decision. An example diagram is shown in Figure 3 that uses the same decision table shown in Figure 2.

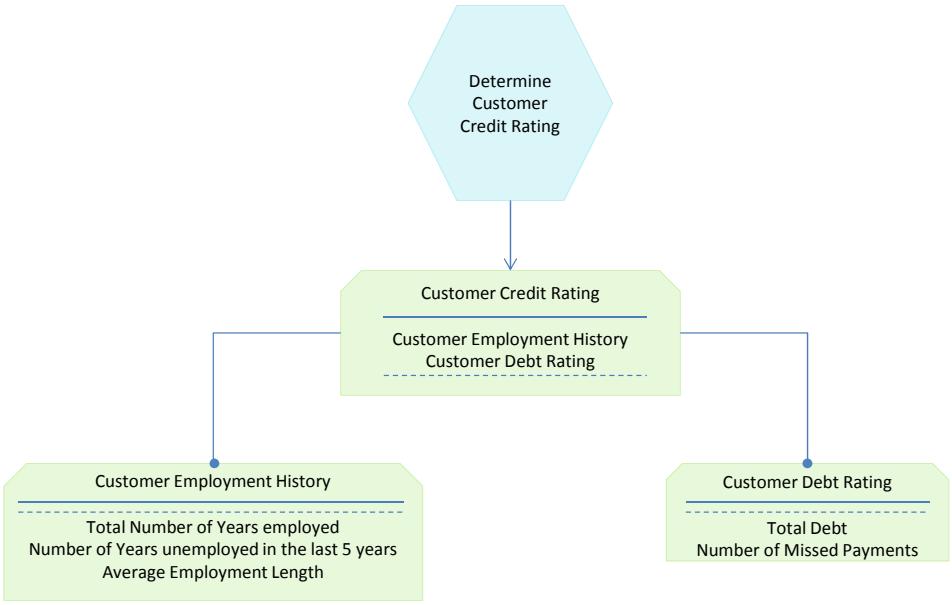

Figure 3: A Decision Model Diagram to determine a person's credit rating

On the top you see a blue decision octagon. It represents your starting point, the decision that you want to make at this point of time. From there, you can trace the decision process down to the rule families, represented as green gravestones. Denoted on the top, each gravestone has a name that is at the same time the conclusion fact type. Conditional fact types that are determined by other rule families are placed in the middle of a gravestone. Condition fact types that have no supporting rule families are placed at the bottom of a gravestone. These contain basic input data that cannot be derived.

5 The 15 principles in Blueriq

This section describes how it is possible to use the Decision Model in the Business Rule Management System that is part of the Blueriq Business Process Management (BPM) software solution. We will cover how to follow the 15 principles in detail.

[8]It should not be the case that the value of one condition can be derived based on the value of one or more other conditions.

Principle 1: Tabular Principle

This principle states that business logic should be noted in a tabular way. The Blueriq decision table, shown in Figure 2, is a tabular form of presenting business logic.

Principle 2: Heading Principle

The headings of the decision table should be fact types. In Blueriq these are not column-headings but row-headings. The decision tables are tilted by 90 degrees compared to the tables in the decision model. This is no restriction and is allowed within the decision model notation [10]. In Blueriq, expressions can be modeled as headings providing the opportunity to make more complicated statements than allowed in the decision model. In order to follow this principle business engineers have to restrain from this and only place other attributes in the headings.

Principle 3: Cell Principle

A cell contains an atomic logical expression belonging to the heading. An atomic expression is of the form: operand 1 + operator + operand 2. This principle states that the heading in combination with a cell forms such an atomic expression. In Blueriq not only the heading can be a complicated expression (see *Heading Principle*), also the cell can be a complex expression.[9] While this gives business engineers additional freedom for modeling logic, they can restrain themselves to only simple expressions (e.g. >5) to follow the rigor of this principle.

Principle 4: Row Principle

The condition cells infer the corresponding conclusion cells in a row. This principle is automatically adhered to by Blueriq without the need for analysts to take action. A conclusion cell is derived by the complete set of condition cells above it in the same column. Only if all those evaluate to true the conclusion cell is reached.

Principle 6: Conclusion Principle

This principle states that there can only be one conclusion fact type. Halle and Goldberg [10] argue that the reason for this is that when there are multiple conclusion fact types one is often dependent on the other instead of that both fact types are dependent on all condition fact types. If this is not the case a second rule family can be made with the same condition fact types. Blueriq does not enforce this restriction, and you can add multiple conclusion fact types.[10] This can be beneficial for maintenance reasons, as there are no duplicate decision tables where only the conclusion differs. However, if a business engineer wants to follow this principle, this is perfectly possible.

Furthermore, this principle indicates that in order to make the conclusion true all conditions must be true. This is also how decision tables in Blueriq work; all cells above the conclusion have to become true in order to reach a conclusion.

It is also possible to draw a conclusion in Blueriq without a decision. For this, business engineers can use the asterisk (*) character that translates to 'any value' or 'don't care'.

[9] A complex expression can range from a simple sum up to counting instances or evaluating regular expressions.
[10] There are some restrictions on multiple conclusions in order to avoid side effects.

The last point of this principle is that the conclusion should be dependent on all conditions. In other words, there should be no conditions that do not matter for a conclusion. This is mainly a modeling question and could be supported by a verification algorithm, which is still subject to further research.

Principle 7: Connection Principle

If the heading in one column is the conclusion in another column, then the rule families are related. This principle is automatically adhered to in Blueriq if the analyst follows the *Heading Principle*.

Principle 8: Declarative Heading Principle

There should be no order in the condition fact types of the rule family. This principle follows from the declarative nature of the decision model and avoids a hidden sequence to the columns in a rule family. Therefore, you can switch columns without problems. In Blueriq there exists a sequence in the rows[11]. They are evaluated from top to bottom. Because of this structure a lot of repeating statements can be avoided. Take for example Figure 2. First the employment history is evaluated and second the debt. We see that the "Poor" entry of the employment history spans 2 columns indicating that for both cases of debt the employment history is "Poor". This split resembles in some way the working of a decision tree [7]. It is still possible to turn around the rows in a Blueriq decision table, however the effect will be that 2 cells with the entry "Poor" are present, one under Debt="High", and one under Debt="Low". The more entries you have in a decision table, the more work it is to maintain it.

Principle 9: Declarative Body Principle

There should be no order in the body of a rule family. You can start with every row in the rule family to find a matching entry. This principle ensures that there are no procedural constraints. In Blueriq, where the rows of a rule family are the columns in the decision table, it also does not matter in which order you model the columns. In Figure 2 you could just as well have started with the entries of "Good" and "Excellent" for the employment history.

There is one exception to this principle. If the value ranges of two alternatives overlap then the first alternative that matches is chosen. For example, if the customer rating is 100 and two decision table cells accept ratings 0-200 and 50-500 then the most left one that matches is chosen. Modeling logic like this is not advisable though, and it is much better to use distinct value ranges. By using distinct value ranges the *Declarative Body Principle* is obeyed.

Principle 10: Declarative Inferential Relation Principle

There is no sequence in which subdecisions are evaluated. Blueriq uses a backwards chaining mechanism. This means that values of attributes are derived when needed. This can happen in any order, as the rule engine will consult its rule library and can encounter matching rules in any order. Therefore, business engineers cannot assume a fixed sequence in which sub-decisions are evaluated.

[11] The Blueriq decision tables are tilted 90 degrees compared to a rule family.

Principle 11: Rule Pattern Transitive Principle

The rule patterns should be independent from each other. This means that a condition fact type should not be deducible from other condition fact types within the same rule family. In other words, no sub-decision is hidden in the condition columns. In order to check if there may be a dependency between the condition fact types you can inspect the correlation between them [10]. However, business experts still have to decide whether there really is a dependency. When a dependency is found the table should be split up. This principle still applies in the same way when modeling decisions in Blueriq.

Principle 12: Rule Family and Rule Pattern Consistency Principle

This principle encapsulates 7 of smaller points.

(1) When executing a rule pattern there should be either one result or no result, but not multiple results. This ensures predictable behavior. If business engineers follow the *Declarative Body Principle* in Blueriq then either one answer or no answer is given in a predictable way.

(2) There is no need to cover the complete domain of a condition fact type. If values are not relevant to the rule family you can leave them out. In Blueriq you also have the freedom to not cover the complete domain of a variable. Furthermore, the special keyword "[]" is available to denote 'all other values'. This will be true if the value at hand is not covered by alternative entries in the same decision table.

(3) There should be no overlapping condition key coverages. While this is not enforced in Blueriq it is strongly advisable not to model this.

(4) There should be no overlapping condition keys between different rule patterns in the same rule family. In Blueriq there is no notion of rule patterns, so by following (3) this will also be followed.

(5) A rule family should result in a conclusion for any valid input values. This is the responsibility of modelers to make sure this is covered. This is the same in Blueriq.

(6) A rule family can have more than one outcome. For example, if a customer qualifies for multiple discounts then each discount is concluded from different entries in the rule family.[12] This is the first principle which works somewhat different in Blueriq. A decision table in Blueriq only has one result. The result itself, however, can be multivalued, indicating that multiple discounts are applicable. The drawback of this approach is that the decision table has to be structured in a way to accommodate this situation, and will get more complicated compared to the same table within the decision model. Another alternative would be to split the decision up into different sub-decisions that are handled separately and from which the results can be combined.

(7) The values of the conclusion do not need to cover the complete domain of the conclusion fact type. This works the same in Blueriq and business engineers can choose which values are set.

Principle 13: Rule Family Transitive Conditions Principle

This principle is similar to the *Rule Pattern Transitive Principle*, only that this now concerns rule families instead of conditions. You can recognize transitive dependencies with triangular structures in the decision diagram and these dependencies should be

[12] These have to be concluded from different rule patterns though, as otherwise point 1 of this principle is violated.

removed in collaboration with business experts. As this is a modeling question the same approach can be used in Blueriq.

Principle 14: Inferential Integrity Principle

Two rule families are dependent on each other if the conclusion in one is a condition in the other. This principle states that if a value is concluded in one rule family, the other rule family should use this value in its conditions. If this is not followed, then the situation can occur that a rule family comes to a conclusion that is not tested in the other rule family. It is up to analysts to follow this principle in the decision model, and in Blueriq. An automatic verification could be useful to follow this principle.

Principle 15: Business Alignment Principle

A decision model should be aligned with business metrics. This principle highlights the importance that a decision model is made to solve a business problem. Therefore, business metrics should be used to measure its effectiveness. This principle holds for decisions in Blueriq in the same way.

Business Rules and Expressions

As discussed in Section 3 analysts have more than only a decision table available when modeling business logic in Blueriq: (1) Business Rules and (2) Expressions are also available. In fact, both these constructs can be seen as special cases of a decision table.

(1) A business rule has an expression in its IF clause that can be translated to one or multiple condition columns and condition cells. The THEN clause is translated to the conclusion column and conclusion cell. (2) An expression is a special case of a business rule, in which the IF clause is always true. This closely aligns with the *Conditions Principle*, and can also be translated to a decision table.

Both the business rule and the expression can be regarded as a single entry in a corresponding decision table. As a business rule is not always true, multiple business rules are often needed to model a good decision. These business rules can be combined with any number of decision tables to come to a conclusion. The business rule engine therefore has to keep looking for a logic component to evaluate the current (sub-)decision. This is somewhat different to the decision model notation, which places all these rules in one rule family, using different rule patterns.

6 Proof of Concept

Blueriq created two proof of concepts that demonstrate how the decision model can work together with the Blueriq studio environment for modeling business logic and with the runtime execution of business logic.

Decision Model Diagram for decision design

This proof of concept shows that it is possible to create a decision model diagram in accordance with the decision model notation based on modeled business logic. The analyst models the business logic using a combination of business rules, decision tables and expressions. When the

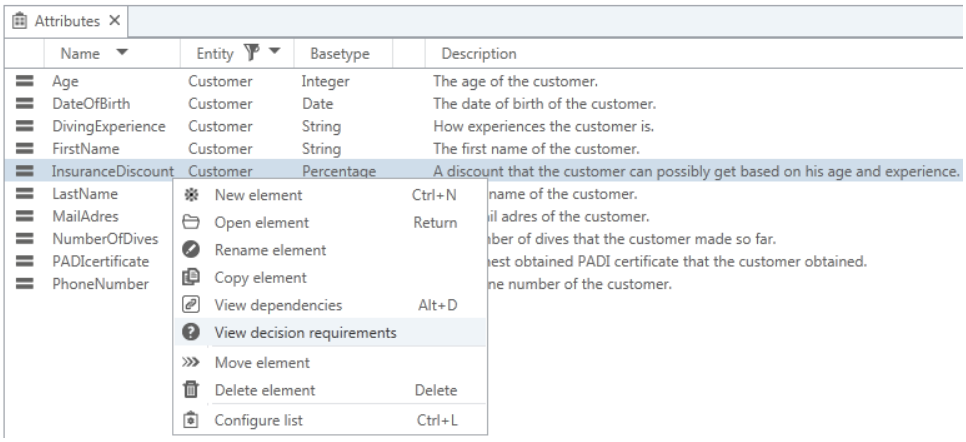

Figure 4: Opening a decision model diagram in Blueriq

analyst is interested in obtaining overview of how the value of an attribute is derived he or she can open a decision model diagram, as shown in Figure 4.

A decision model diagram created on the modeled business logic is shown in Figure 5. In this example the discount rate of a diving insurance is decided based on the customers age and diving experience. His diving experience is dependent on his obtained certificate and the number of dives he did so far. His age is dependent on his date of birth.

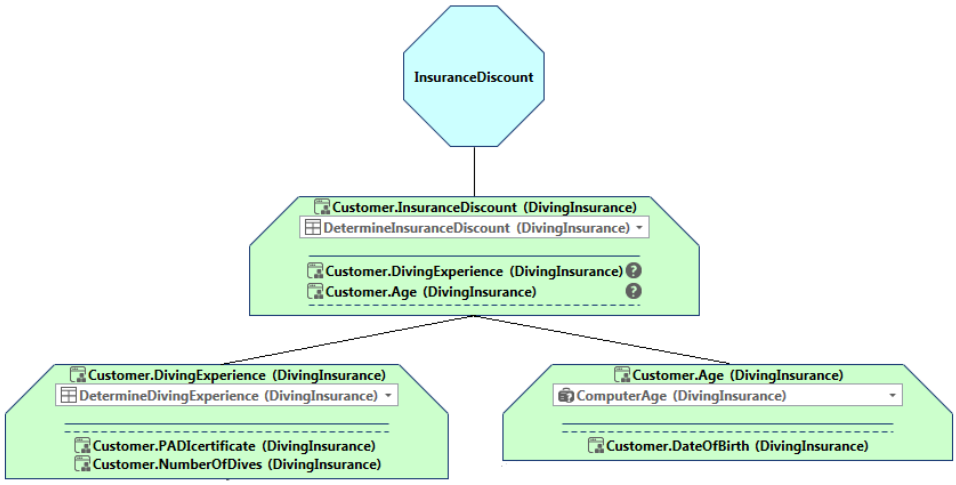

Figure 5: A Blueriq decision model diagram

Decision Model Diagram for decision execution

The decision model diagram shown in Figure 5 is generated during design time. It displays all possible paths that can be taken when making a decision. When a decision is executed not all situations require to visit every rule family that is shown in this diagram. An example of a decision

Customer.DivingExperience	"High"		"Medium"		"Low"
Customer.Age	>25	<=25	>25	<=25	*
Customer.InsuranceDiscount	20	10	10	0	0

Figure 6: The decision table to determine the discount for a scuba diving insurance

table for this situation is illustrated in Figure 6, which shows a decision table to determine if a person is eligible for a discount on a scuba diving insurance.

In this figure we see that when the scuba diving experience of the customer is low his age does not matter and 0% discount is awarded. In this case it is not needed to visit the rule family to determine the customers age based on his date of birth. For this reason Blueriq is also able to generate a decision model diagram during runtime, which accurately displays which rule families were actively used during the current decision. We call this a decision model execution diagram, and an example of such a diagram is shown in Figure 7.

In this example a risk of a customer is calculated who applies for insurances. The risk is dependent on many factors such as age and past behavior. For this particular application, the applicant is too young for insurance. The minimum age is 18 while the applicant is 16. This means that other sub-decisions do not have to be taken. Figure 7 therefore only shows the decision on the required age of the applicant. The bottom left part of this figure calculates the age of the applicant, while the bottom right part decides on the minimum age requirement. Those two are combined and determine that the applicant is to young resulting in a high risk.

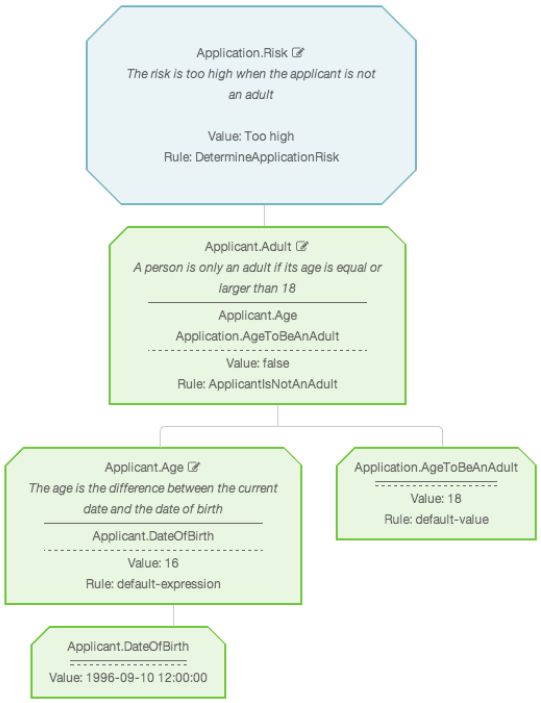

Figure 7: A Blueriq decision model execution diagram

7 Epilogue

In this article it was explained that Blueriq is compatible with the decision model notation. Analyst can model business logic adhering to the 15 principles. This will bring structure to the modeled logic. Furthermore, we have created two proof of concepts that demonstrate how powerful a decision model diagram can be. The first proof of concept enables analysts to show a diagram during design time, while the second proof of concept enables analysts to create a diagram during execution time. This second variant is new and shows the dependencies of logic for the actual case at hand. This means that unused logic is omitted and a clear answer can be given on why a certain decision was made. We believe that the decision model improves the implementation speed of projects and reduces maintenance costs.

Further research includes two topics. (1) While generating a decision model diagram from logic is bottom-up, you may want to work top-down. You may want to start with drawing a decision model diagram and let it create blank logic components for you. (2) Multiple principles of the decision model rely on the judgment of a business expert such as checking if there are dependencies between fact types. In some situations these dependencies can be recognized automatically. It would be interesting to develop such algorithms that support analysts during the modeling of business logic.

Contact Us

If you have any questions about this article or if you would like to start a discussion, do not hesitate to contact us.

Email the author	:	m.schadd@blueriq.com
Email Blueriq	:	info@blueriq.com
Call Blueriq	:	+31 (0)73 6450467
Website Blueriq	:	http://www.blueriq.com

Blueriq BV
Veemarktkade 8
5222 AE s-Hertogenbosch
The Netherlands

About Blueriq

Blueriq is a global provider of an innovative rules-driven Business Process Management (BPM) software solution that assists corporations in the modeling, automation and integration of intelligent business processes. Using Blueriq, organizations improve business productivity while reducing business costs by implementing agile applications that quickly respond to changing business conditions. Blueriqs products empower knowledge workers and allow business users to control the process management lifecycle and outcomes facilitated by IT. Blueriq, based in s-Hertogenbosch, the Netherlands, is part of Total Specific Solutions (TSS). For more information, please visit www.blueriq.com.

References

[1] D. L. Fisher. Data, Documentation and Decision Tables. *Communications of the ACM*, 9(1):26–31, 1966.

[2] P. Harmon. Processes and a Decision Model Notation. *Business Process Trends*, 10(22), 2012.

[3] M. H. Linehan. SBVR Use Cases. In N. Bassiliades, G. Governatori, and A. Pascke, editors, *Rule Representation, Interchange and Reasoning on the Web*, number 5321 in Lecture Notes in Computer Science (LNCS), pages 182–196, Berlin Heidelberg, Germany, 2008. Springer.

[4] M. H. Linehan and C. de Sainte Marie. The Relationship of Decision Model and Notation (DMN) to SBVR and BPMN. *Business Rules Journal*, 12(6), 2011. URL: http://www.BRCommunity.com/a2011/b597.html.

[5] Object Management Group. Semantics of business vocabulary and business rules specification. formal/08-01-02, 2008.

[6] Object Management Group. Decision Model and Notation - Request For Proposal. bmi/11-03-04, 2011.

[7] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, Upper Saddle River, NJ, 2010.

[8] K. D. Swenson. *Mastering the Unpredictable: How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done*. Meghan-Kiffer Press, Tampa, FL, 2010.

[9] J. Taylor. Decision-Centric, Intelligent Processes. White Paper. Decision Management Solutions, 2013. URL: http://www.blueriq.com/resources.

[10] B. von Halle and L. Goldberg. *The Decision Model*. Auerbach Publications, Boca Raton, FL, 2010.

[11] S. A. White and D. Miers. *BPMN Modeling and Reference Guide*. Future Strategies Inc, Lighthouse Point, FL, 2008.