

Masterproef

Externalization of the decision process from BPM to optimize business processes

Promotor :
Prof. dr. Koenraad VANHOOF

Glen Biesemans

Master Thesis nominated to obtain the degree of Master of Management , specialization Management Information Systems

2 0 1 0
2 0 1 1

FACULTY OF BUSINESS ECONOMICS

Master of Management: Management Information Systems

Masterproef

*Externalization of the decision process from BPM to
optimize business processes*

Promotor :
Prof. dr. Koenraad VANHOOF

Glen Biesemans

*Master Thesis nominated to obtain the degree of Master of Management , specialization
Management Information Systems*

Preface

The title of my thesis is 'Externalization of the decision process from BPM to optimize business processes'. At first glance this seems very complicated, but in the end of this thesis you will have a clear image about it.

The problem statement came after realizing that changes in business logic needed to be implemented on all decision parts of all business process models (BPM). When such a model is automated, there is a need to rewrite the whole code and this takes some time. Even when it is not automated, it still needs an adaption to the changes in the model itself and all the models where the decision has impact. A separation between the business logic and the processes can be a possible solution. Separation of business logic is mentioned in the book "The Decision Model: A Business Logic Framework Linking Business and Technology (IT Management)" (Larry Goldberg and Barbara Von Halle, 2009). With this book I started my research about the subject of independent management between decisions and processes. The authors explain The Decision Model, which is a sort of approach to separate decisions from all other aspects.

Further research gave me the knowledge that a separation off rules has already been deployed for many years. The use of Business Rule Management (BRM) with their more or less corresponding Business Rules Engines (BRE) and Business Rules Management Systems (BRMS) is a concept that needed some further discovery. The rather new concept of decisions is different than that from a somewhat older concept of rules. A decision can include many rules and could in that way be a higher level of the business. Questions arise to discover if the decision concept is the step in the right direction.

Soon I realized that my problem statement is for a part due to a big problem reigning businesses nowadays and that is that of the alignment between IT and business people. The collaboration and communication between IT and business people leaves much to be desired.

I will try to find ways to conquer the gap between IT and business and will try to find a way that there is no need for time-consuming tasks when changes occur. Being agile is a term often used nowadays, and I will try to discover ways for doing maintenance on decisions as agile as possible.

A word of thanks goes out to my promoter prof. dr. Koen Vanhoof for his help during the subject choice and the development of my thesis.

Summary

This summary leads the reader through the subjects which come to discussion in this thesis. The title of this thesis is: “Externalization of the decision process from BPM to optimize business processes”.

To place this title in the context of the problem statement, a first chapter speculates a possible solution with the separation of decisions.

The first chapter starts with the explanation of business rules. The next sub-chapter explains why these business rules are so important and why the last decade so much interest is arisen to handle them differently. Managing the business rules separate from processes elevates the terminology of declarative versus procedural nature of actions. The Decision Model from Barbara von Halle and Larry Goldberg (2009) gets a context. The gap between IT and business is a major problem where this model can play an interesting role, because nowadays and alignment is certainly needed.

Chapter 2 explains how The Decision Model looks like at what the theory behind it is. It starts with explaining some new terms that will come up during a development and analysis of the model. The two parts of The Decision Model are the Decision Model diagram and the Rule Family. The former represent the overall structure of decision logic, while the latter is more an in depth view of statements. After the concept explanations, the crucial link between BPM and The Decision Model gets a first theoretical review. The most important aspect of The Decision Model to come to the well defined structure, are the principles what makes it rigorous.

In chapter 3 The Decision Model will be designed for certain aspects of the EU-Rent case. The iterative designing approach for classifying rules under the logic of certain decisions that has to be made during the rental process will be extensively discussed. The case is build upon information and statements found for the fictitious car rental company EU-Rent. During the development, some concerns and challenges come to the surface and will be explored.

Chapter 4 investigates what the difference is in stating rules in catalogs or in a model. There are some benefits for a more structured approach to classify the rules according to the decisions they support. Another interesting aspect is the ease of changing and thus managing rules that are represented more structurally.

Chapter 5 discusses other representation for the business logic. There are several existing representation possibilities and a few will be explored further. Research has provided some benefits

and disadvantages of the several methods. The most common methods of presenting logic, like decision tables, decision trees, natural languages and the rather new q-charting, will be briefly explained. The intention is more to analyze them with respect to The Decision Model.

The context of The Decision Model and the closely related concepts will be described in chapter 6. It sums up concepts that are linked with the separation of rules or logic from processes. The chapter starts with discussing the architecture of a business and how The Decision Model fits in it. Terms like Service Oriented Architecture will be linked with the role The Decision Model can play with it. There are methodologies that use separation of business rules (like BRM) and there are recent ones that focus on the separation of decisions (like BDM). A short introduction to these concepts and the alignment with The Decision Model will be provided during the reading of chapter 6. A maturity model for decision managing and the problem solving of the gap with IT will come to discussion in the last part of the chapter.

Chapter 7 lists up the different standards for different ways of modeling and how important standards are for creating the universal accepted, technology-independent decision model. Therefore is the second part reserved for the Decision Model Notation (an RFP handled by OMG).

In chapter 8 will be an explanation of how the nature of decisions can be very distinct. First of all the difference with a rule should be very clear. Second will decisions depend on their impact and their volume. There are some considerations to run through to understand the nature of a decision and what kind of decision is best fit to be represented in The Decision Model. Decisions that are evolving around information quality were first not intended to be implemented in a decision model, but from real world practices, these kinds of decisions are recently used more often.

Chapter 9 is all about the constraints and challenges that come to the surface when separating decisions. The link with BPM is clearly visible and the Decision Model has a natural anchor point with processes. A short summation of several challenges, which came up with developing the Decision Model, is listed. A fact that the development of software for this rather new approach of separating is still in his early shoes, is a disadvantage. In time there will probably be more software available, with hopefully support for higher levels of maturity.

Chapter 10 discusses briefly the main benefits of separating decisions from models. Structural or economic relevant improvement due to The Decision Model and the intellectual template is represents, are briefly summed up in this chapter.

The last chapter brings, in addition to the challenges and the benefits from previous chapters, an overall conclusion about the thesis and the research that has been done.

Table of contents

Chapter 1 : The Decision Model could be a solution	11
1.1 Definition business rule:	11
1.2 Problems IT and the concept of rules:	11
1.3 Separation:	13
1.4 The Gap:	15
Chapter 2 : The Decision Model explained	17
2.1 Introduction and terminology:	17
2.2 Declarative parts in BPM:	22
2.3 Principles and normalization:	25
2.3.1 Seven principles for structural simplicity:	25
2.3.2 Three declarative principles:	27
2.3.3 Five integrity principles:	28
2.3.4 Three normal forms:	31
Chapter 3 : Designing The Decision Model	33
3.1 approaches:	33
3.2 EU-rent case:	34
3.2.1 Introduction:	34
3.2.2 Developing decision models:	36
3.3 Variable data:	47
3.4 The decision model view:	48
Chapter 4 : Modeling or listing	51
4.1 Model vs. list:	51
4.2 Changes:	52
Chapter 5 : Other representation methods	55
5.1 Decision tables:	55
5.2 Decision trees:	56
5.3 Declarative (natural) language:	57
5.4 Question charts (Q-charts):	59
Chapter 6 : Approaches based on separation	61
6.1 Architecture:	61

6.1.1 Enterprise architecture:.....	61
6.1.2 Perspectives:	63
6.1.3 SOA (service oriented architecture):	63
6.2 Separation of rules:	64
6.2.1 BRE:	65
6.2.2 BRMS:.....	65
6.2.3 BPM and BRMS:.....	65
6.3 Separation of decisions:	66
6.3.1 Somewhat a history:.....	66
6.3.2 BDM (interchangeable called EDM or DM):.....	67
6.3.3 Two way connection between BDM (EDM or DM) and SOA:	68
6.4 Maturity model:	69
6.4.1 Business decision maturity model:	69
6.5 Decisions and IT:.....	71
Chapter 7 : Finding standards and specifications	73
7.1 Specifications and standards:	73
7.2 Decision Model Notation (DMN):	75
Chapter 8 : Types of decisions	77
8.1 Terminology: rules – decisions:	77
8.2 Decision impact and context:	77
8.4 Information Quality (IQ):	80
8.5 The Decision Model and the human brain:	81
Chapter 9 : Constraints and challenges	83
9.1 Imperfections:	83
9.2 BPM and the Decision Model:	84
9.3 Software in early shoes:	86
Chapter 10 : Benefits.....	89
Chapter 11 : Conclusion	93
References	95
Appendices.....	99
Appendix A: Figures and Tables in thesis:.....	99

Appendix B: Additional appendices references:.....	100
Appendix C: EU-Rent case:	101
Appendix D: Short Glossary:.....	120
Appendix E: Architecture:.....	122
Appendix F: Question charts:	124
Appendix G: Decisions:.....	126

Chapter 1 : The Decision Model could be a solution

This first chapter will discuss the problems facing businesses caused by a bad alignment between IT and business people and a non efficient way of managing processes. The problem statement describes that business processes suffer from time-consuming IT workload when there is need for some changes. The business rules are the main aspect that causes need for change. Rules are changing very often to compete among other businesses and survive a dynamic business environment. Let's start by defining what a business rule and go further into detail about the problems facing managing efficiently. One possible solution for better alignment and eliminating such problems could be with the use of The Decision Model. The Decision Model itself will be explained in further detail in chapter 2.

1.1 Definition business rule:

The company Business Rule Solutions defines a business rule as an atomic piece of re-usable business logic, which is specified declarative.

Atomic means that the business rule can't be split into something smaller, because it then loses his semantic value.

Re-usable is used to say that the business rule isn't specific for one application, but can be used by all kinds of different applications.

The word business is used because the rules are meant for the business experts to understand and maintain.

An important characteristic of such a rule is that it is declarative. This is contrary to procedural, where a sequence has to be followed. The rules often don't need a sequence.

1.2 Problems IT and the concept of rules:

The business logic is often hidden in program code. This can cause a loss of the rules and policies.

In the past, IT people were needed to put the rules inside the applications. Automated processes therefore could have logic behind it, which is lost during the years.

When policies or regulations of the company changed, IT people were needed to rewrite those processes. This is a time-consuming effort that can be improved. The last decade enterprises needed to be more flexible according to the quick changes in the environment. The economic milieu causes

more changes than ever before. To be more flexible and to answer more quickly to changes, a business nowadays tries to be as agile as possible. Being agile means getting a quicker response on all kind of changes. Not only do companies offer more and more customized products, they also need to be prepared to answer to all sorts of external influences of the environment. To adapt a different kind of tactic, processes often need to be changed. Often this is due to different criteria for decision making that change, caused by a differentiated tactic. For example, a car company is suffering due to an increase of bad experiences with customers. This could raise a question with senior management. A different tactic could be in place, to increase the credibility of the drivers. This could be implemented with a more severe approval process for the customer. The old criteria for approval need to be renewed, or some additional criteria need to be implemented.

When these decisions are hidden in an automated process, this process needs to be rewritten. IT people are needed; and a time consuming workload is brought upon the IT department. It even could be that in the time this is rewritten, yet another rule need to be implemented due to another changed policy or business rule. This is a worst case scenario. When we could extract these decisions; we could manage them separately. This is a goal of some new developments in this area. In the past BRE at first and later BRMS (both will be described further in detail in chapter 5) came up to the surface to point out the benefits of managing rules separately from the rest.

When we speak of modeling and managing rules; a recently developed method could help eliminate most of the historical problems. Decision tables and decision trees are already well known concepts in the business, but have their flaws or shortcomings in some cases. When complexity increases, these methods bring some flaws to the surface.

The Decision Model is a rather new way of presenting decisions (and their rules). Especially when we want a clear link between processes and rules, such a method could be useful. The ideal result would be, when we change a decision, we could do this externally outside the process, and the process wouldn't suffer. Suffer in the sentence of additional changes in BP. The Decision Model gives us a clear link between different modeling techniques. Between BPM and The Decision Model, a similar terminology provides decision aware tasks. These tasks are based on a decision, which can be seen when we deepen in the according decision model. Of course are there a lot of requirements before we get a flawless distinction between the two. The principles of The Decision Model and the normalizations help to build a rigor structure. This will be explained further in this thesis in chapter 2.

1.3 Separation:

The old way of handling with rules becomes more and more unacceptable. There are enough reasons to separate business rules from business processes. An important motivation should be that business leaders are able to manage the rules separately and for the good of the business.

In separating the rules one is able to trace them to their according business motivation and measure their success against a benchmark to conclude if modifying is necessary. It can even be useful to minimize business risk and improve business performance. Another benefit is that it makes it easy to change them independent of changes in business process models. They can also be easily shared among other processes. With separation parallel and prior work is enabled. This means that the rules can be defined, analyzed and tested separately from the development of business processes. This can happen before or together with the development of business processes. The processes itself can be made decision-aware and agile.

Business automation has been a big activity in most of the businesses the last decade. The sequence of steps in a business process along with the logic of it often has been translated into program code. K. Orr described in 2007 his look about the parts of an application and how it can be separated. Functionality contained in a single framework in the old days; has been teased apart so development could be independently (database, transaction processing, presentation, workflow, security and business rules). One learned that competition is not only about how efficient the business processes are but also about how smart the decisions behind these are. For being competitive it is not just enough to optimize your processes, they also should be smart and agile.

Where we used to have applications with 'a big ball of mud', we now try to externalize as much as possible to manage different things separately.

An application is build on different components. We have a database component, that is externalized during the years and a 'relational model' was invented to manage them separately. Business logic was ironically not yet successfully externalized. This is ironically because business logic can be seen as the heart the business. Different strategies and policies are tucked in this logic. It seems logical that the next big step is to externalize this logic and manage it on a more efficient and better manner. The Decision Model is a tool to do this. One needs to distinguish between sequential tasks and declarative decisions. In the pure decisions and logic there is no sequence. This is in contrast with the purely sequential structure of processes in a BPM. Although we assume, that a BPM is purely

sequential. This is not always the case, like illustrated further in this thesis. There will be illustrated that certain tasks are not in need of a sequence. Certain tasks could be done before or after certain tasks, so there is no need of a strict sequence. We can reverse the tasks and come to the same output.

This is an interesting case for implementing a decision aware task. Now, the task contains certain decisions that could be found in The Decision Model. A first impression gives already a clear benefit. In this way, the process model is decreased substantially in complexity. When we want to go further in detail, we consult The Decision Model with its rule family tables. The concept of Rule Families will come back in the net chapter. Without decision separation, decisions will only evolve with processes.

Over the last decade, a lot of books have been published about 'business rules' and adoption of technology for these rules has increased. Business Rules Management System (BRMS) is a well known technology for handling with these rules.

As with data in the 1970's, a mere separation of business logic seems to have some flaws and unmissed opportunities. There also have been actions for common languages or grammars for business rules that became standards (e.g. SBVR). The missing link is that there is no universal model that handles independently.

What we could have learned about the data a long time ago, is that there will not be a large-scale success unless there is the adoption of such a model. The first step with the data was also separating them from the rest, but the big success came with Corr's relational model.

As this model did for data, The Decision Model would handle the separate business logic in a different representation for its maintenance and perhaps automation.

According to the book 'The Decision Model: a business logic framework linking business and technology' by Barbara von Halle and Larry Goldberg (2009) some characteristics of this model are as follows:

- It's a technology independent way of organizing the important asset, called business logic.
- It's a pure representation of business logic with a simple structure, declarative nature and optimal integrity.
- It's easy to implement in recent and future technologies.
- It's not a language or a grammar, but a model. Yet, languages and grammar can be defined over the model. This can be compared with SQL (a language that has been invented for the Relational Model).

- It's not a list or annotations attached to or buried in other kind of models, but it's a model just for the business logic.

SOA, BPM, BDM (EDM), which will be explained later on, have known significant growth in the technology area. A model of business logic is at the center of all of these emerging areas.

1.4 The Gap:

Business rules are the core of decisions and actions in the company. These decisions and actions can be automated or not by IT. The problems with alignment between IT and business people causes an inefficient way of managing. Business decisions often come from business experts and need to be communicated through to the processes.

Business decisions deserve their own model, because they are of a separate dimension (neither data nor process). The business decisions are best presented in diagrams that show relationships of logic to each other. The Decision Model fills the gap between the dimensions by introducing a stand-alone model simply for organizing business rules.

The logic in a decision model has an atomic form and is governed by 15 rigorous principles and 3 normal forms. For learning about the principles, I refer to the next chapter.

Chapter 2 : The Decision Model explained

In this chapter a further explanation of The Decision Model will occur. Business logic is a term that came back in previous chapter, but what exactly is this business logic?

The definition for the business logic according to Barbara von Halle and Larry Goldberg (2009) is that it's "a means by which a business derives a conclusion from facts". A business logic statement is therefore an expression of conditions that evaluate facts leading to a conclusion of a new fact. One can understand that this logic is behind every conclusion that is made within a company. This conclusion can then on his term help to derive at a decision. The official definition for a business decision according to Barbara von Halle and Larry Goldberg (2009) is that it's "a conclusion that a business arrives at through business logic and which the business is interested in managing".

Knowing what exactly a business decision is and what business logic is, will help understanding the concept.

The chapter will start with a brief introduction to The Decision Model. The principles and normal forms where this model is based upon will be explained to understand the rigor of the structure. Next, the methodology of designing such a model will be described. The theory comes from the book 'The Decision Model: a business logic framework linking business and technology' (Barbara von Halle, Larry Goldberg, 2009). The theory and methodology will be applied and illustrated with a case of a fictitious car rental company EU-Rent.

2.1 Introduction and terminology:

Let's start this intro with the most important definition of this chapter. The definition for the decision model according to Barbara von Halle and Larry Goldberg (2009) is that "it's an intellectual template for perceiving, organizing and managing the business logic behind a business decision".

There has been a lot of effort to make business logic tangible. This has been done by translating the business thinking into something visible and communicable. A set of business rules or business statements has been made visible with all kinds of formats. These various formats are decision tables, decision trees, free-form text, fill-in-the-black templates or sentences that follow a specific grammar.

The authors (B.von Halle and L.Goldberg,2009) state that The Decision Model is not just a list; it's an intellectual template for the full and rigorous specification of business logic. This means it's not build

for just automation or communication. It means that if the goal is automation, the model can be translated in a target technology. But if the goal is communication it can also be used or translated into any other form preferable to the people who need it. It should be seen as an independent representation with a structural design different from any other kind of model.

We can start by translating free-form textual sentences like the business statements. This means that certain conditions are needed to come to a conclusion, and these conditions and conclusion should be found in the statements.

The fundamental structural element of The Decision Model is a **Rule Family**. This is a two-dimensional table with conditions and conclusion. There can be many conditions, but only one conclusion in the same table. The condition should be tested to arrive at the conclusion. We use only one Rule Family for each conclusion column.

If different kinds of conditions are needed to arrive at the conclusion, we use a different **rule pattern**. This means that for instance if a certain amount of discount serves as conclusion, there could be many different conditions that arrive at a certain discount amount. The group of conditions that arrive for instance at a discount of 5%, could be different from the group of conditions needed to come to a discount of 10%. Each different group of conditions get's a rule pattern number. A rule pattern is a set of rule family rows with the same conditions leading to the conclusion of that rule family.

Fact type is used to state the classification of a fact (not the information itself). There are conclusion fact types and condition fact types. Each fact type has its business concept what is the subject of that fact type.

The **fact type domain** is defined as the range or set of valid values that the fact type can contain and what makes business sense for the fact type.

Operands can be all kinds of things. An operand can be a fact value what lays in the fact type domain or it can be a formula and even another fact type.

An **operator** is connecting two parts of an assertion. An assertion is just another word for saying that a condition is leading to a conclusion. The most used operands are: is, is less than, is not, etc.

Let's explain these terms with an example of the case EU-Rent. The discount depends on the status of the customer and the duration of the rental of a car. The discount serves as the conclusion where the decision is all about. To come to the conclusion, several conditions should be tested. Figure 2.1

shows a Rule Family to come to the discount conclusion. One can say that several business statements have been made visible. Statements are:

- If the renter is a VIP renter (or 'club member'), a discount of 10% is applied.
- If the number of days for a rental was less than 7 days, 0 % discount is granted.
- If the number of days for a rental was between 7 and 13 days, 5% discount is granted.
- If the number of days for a rental was higher or equal to 14 days, 10% discount is granted.

The formation of the statements consists of a condition fact type with an operator and an operand leading to a conclusion fact type with operator and operand. Some of the operands here are: '7', '7' and '13 and 14'. The operators here are: 'is', '<', 'between', 'higher or equal to'. The condition fact types are: 'number of days' and 'renter status'. The conclusion fact type is 'discount'.

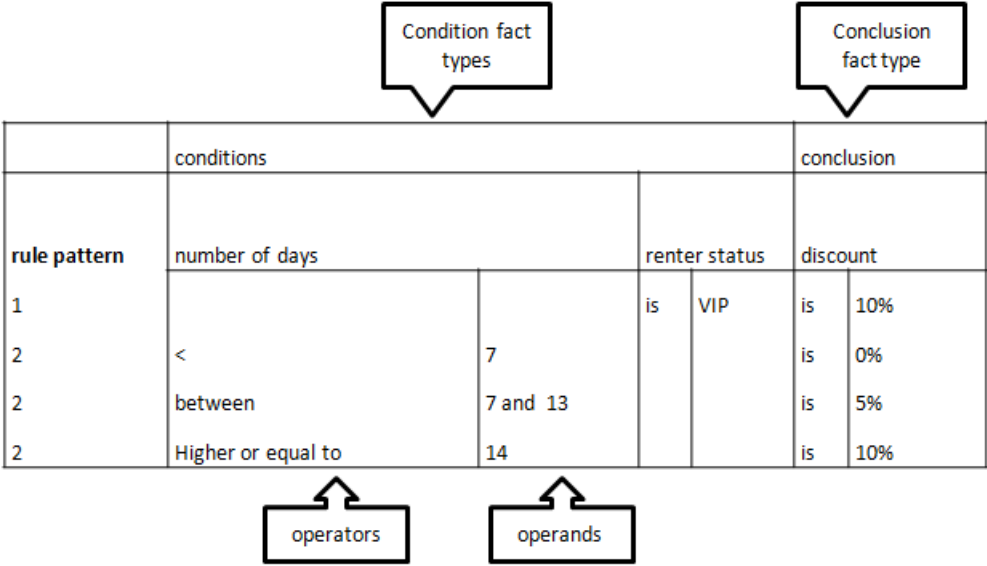


Figure 2.1: Rule Family 'Discount'

The populated Rule Family is the part of the decision model with the most details. The model itself represents the different Rule Families and their connectivity. In the decision model we see the structure and not the detailed content of rule families. A Decision Model for validating a request for booking with EU-Rent is presented in figure 2.2. The business decision is presented at the top with a special business decision shape (octagonal shape). This is the part that links will all kinds of other models like BPM (described later on). The other shapes represent Rule Families with at the top the

conclusion fact type, what represent the title of the Rule Family (above a solid line). In the figure we can see 4 Rule Families with their conclusion fact types and condition fact types (headings). Underneath the business decision shape is the Decision Rule Family. This one is directly connected with the business decision and has its name because it provides the conclusion for the entire business decision. The dotted line inside the Rule Family shapes serves to distinguish between stand alone conditions or conditions that came from a conclusion from another Rule Family. Above the dotted line are the condition column headings that serve as a conclusion column heading in another Rule Family. The conditions below the dotted line will be populated by known fact values (e.g. persistent data). The relationships between Rule Families are presented with lines with at the end a dot. A P-number can be written behind every condition to indicate what Rule Pattern numbers it uses inside the Rule Family.

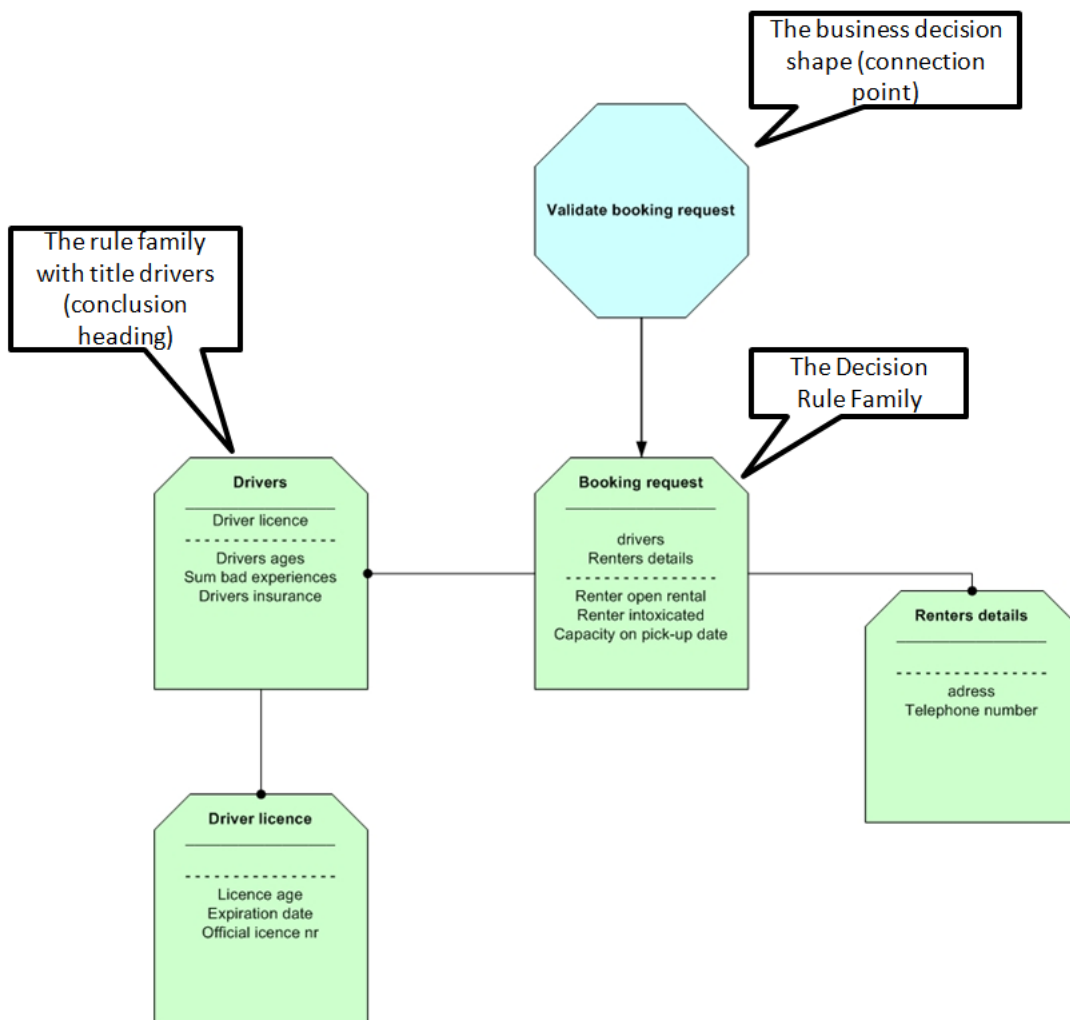


Figure 2.2: : The Decision Model 1 for validating a booking request

The connection between a Rule Family and the decision model itself is presented in figure 2.3. On this figure is only the decision Rule Family illustrated. The global structure can be seen in the previous figure. A decision model serves best to view the overall structure. When more details are needed, a Rule Family can provide an in depth view. Looking at the figure, one knows now that for instance when one want to know it the driver qualifies, one should look at the connected Rule Family of driver.

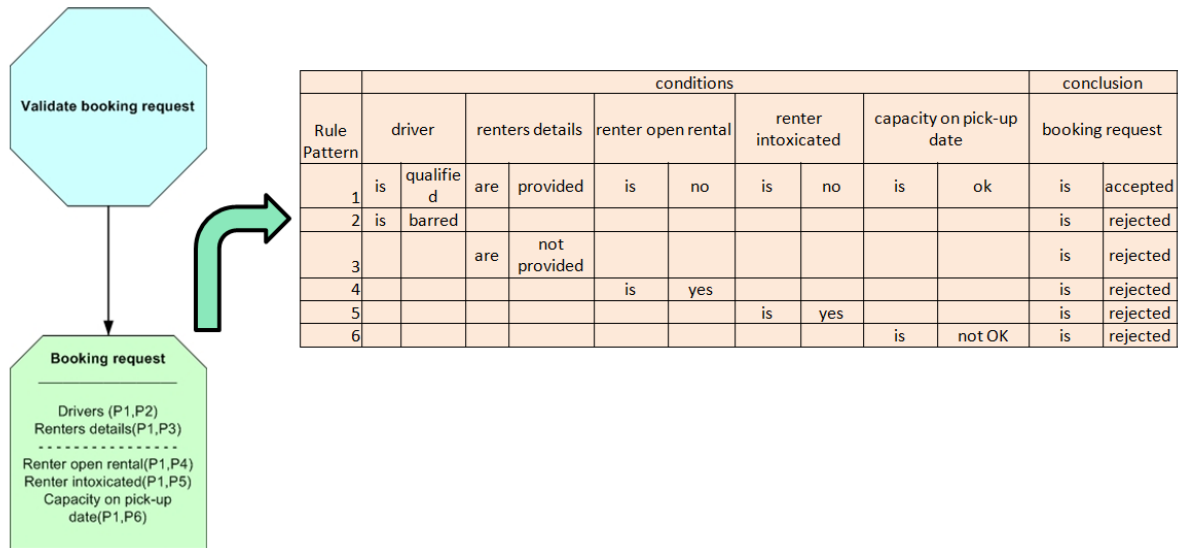


Figure 2.3: The link between the decision model diagram and its rule families

The figure assumes that the date about the capacity on pick-up date, if the renter is intoxicated and if the renter has and open rental is given data. This can also not be given data, but dynamic data (explained later on). Maybe some conditions will be needed to be tested to come to the values of those conditions. The conclusion about these will be in an additional rule family and these conditions need to be written above the dotted line, when this is the case.

2.2 Declarative parts in BPM:

There has already been a short introduction to the procedural or declarative concept. Now, this will be explained with an example of the EU-Rent case. The declarative part has no sequence like we discussed earlier. Figure 2.4 illustrates declarative decisions in a procedural model. When we analyze this figure, we see that the sequence doesn't matter. Let's assume we first check renters details (like address, telephone number and name) before checking if the driver license fulfills the rules (e.g. expiration date is after the end of the drop-off date). This has little meaning, because we easily can shift this procedure around, by first checking the driver license and thereafter the renters' details. It's even worse because all the specific actions and tasks inside this BPM don't ask for a certain sequence. This illustrates clearly that the tasks are of a declarative nature.

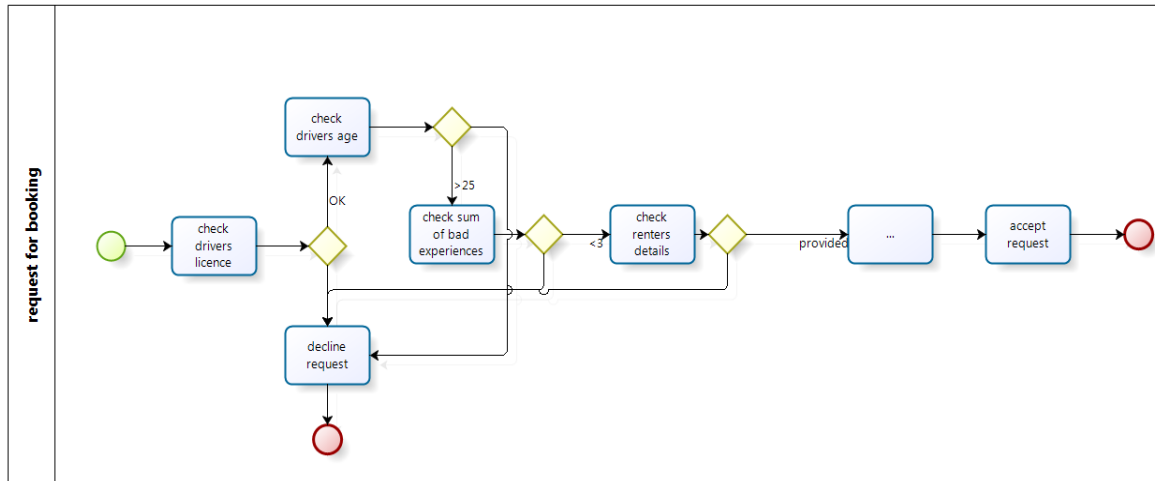


Figure 2.4: Declarative decisions in a procedural model

The declarative parts represented in a procedural model, is not the most efficient way of modeling. The BPM becomes unnecessary complex. This process presents only a part of the request for booking decision from figure 2.2, because it becomes quickly very complex and comprehensive. Referring to the problem statement of this thesis, this is a source for the workload on IT. Let's assume that the process from figure 2.4 is automated in some kind of way. When a decision changes (e.g. drivers age will change, or will be eliminated in the process for accepting a request for booking), the process needs to be rewritten. The danger is also that there could be missing business logic in the BPM. In this example in figure 2.4 is a task called check driver license. The real logic behind it, like checking the driver license age, the expiration data and the official license number are buried in this task. One can say that this is not the ideal representation of tasks from this nature.

A lot of business process models include tasks of declarative nature. One can simplify this process by externalizing the declarative part of a process. In this extreme example the process only represents one decision aware task in figure 2.5, because everything is based on business logic to come to a conclusion. The decision aware task is represented with the same shape as the decision in figure 2.2. When one wants to see the logic, one should look at the according decision model with its possibility to look in depth to the populated rules families.

Managing the business logic becomes easier now, because we can find them inside the according Rule Family. Here we can make the necessary changes when for instance a policy of the business changes. How changes are made; will be discussed further in this thesis. When business logic requires a certain sequence, we put different decision aware tasks behind each other with the different models behind each step.

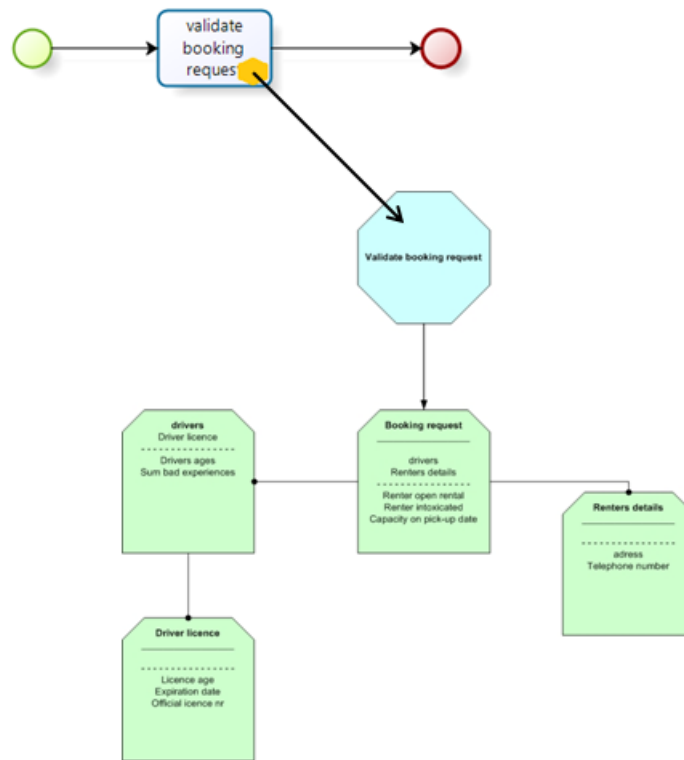


Figure 2.5: Separation of business logic out of the BPM with a decision aware task

The Decision Model also provides connection points with other models than a BPM. It links between a Decision Motivation Model, use-case, etc. Additional information will be provided later on when discussing the different standard models for different business aspects. A detailed description falls outside the content of this thesis.

2.3 Principles and normalization:

The principles together with the three normal forms serve as guidance to eliminate things that do not look quite right. They assist in identifying what is wrong about the content of a decision model and help correcting those things. They are an essential assistance when it comes to developing a decision model with a correct structure, a purely declarative character and completeness in content with full integrity. They ensure that the model is stable, flexible and stays technology independent. The theory for the principles comes from Barbara von Halle and Larry Goldberg (2009).

2.3.1 Seven principles for structural simplicity:

The decision model needs a structure. Putting business logic in a list could make managing the list difficult, especially when the list is very large. When a list contains for instance of more than thousand rules and changes happen often, it would be better to group the logic. The grouping could happen in various ways. If there is no strict way of developing groups (or a structure), different users would use different ways of grouping. Process modelers are likely to put together the logic associated to the same process tasks. Technical developers are likely to group them based on the target technology. Data modelers would then group them again in a different way according to for instance their data tables. This being said, it seems clear that the goal should be a common and uniform decision model. The structural representation must be independent of the process tasks, the data or other models or the technology.

Principles 1 through 7 support the common goal of coming to structural simplicity. This means it minimizes complexity and maximizes comprehension of the structure of the decision model. There is a short explanation of each principle in what follows. The principles are named according to the terminology of Barbara von Halle and Larry Goldberg.

Principle1: The tabular principle

The basic element of a decision model is the rule family. The rule family has two dimensions. One is the heading and the other is the body (content). The rule family represents a two-dimensional table. This principle leads to structures that are intuitive and familiar to most people. The two-dimensional table is something that is well known for most people.

Principle2: The heading principle

Each label in the heading in a rule family contains a fact type. There is a distinction between a fact value and a fact type. A fact type is for example 'sum of bad experiences', while '5' could be the fact value of this type. Each fact type has his domain (possible fact values), which should be defined together with the definition of the fact type. This ensures a high-quality, understandable decision model.

Principle 3: The cell principle

The content of a cell in the body of a rule family is a logical expression adequate to the heading of that rule family. These logical expressions contain operators and operands. The expressions are in the following form: "heading (fact type) + body (operator + operand)". For example customer's 'sum of bad experiences (+) is greater than (+) 3'. This principle provides an easy understanding.

Principle 4: The row principle

In a row of a rule family we read one (or several) conditions that lead to one conclusion. Here can a further distinction in the fact types be made according to their role (fact type role). There are condition fact types. These play the role of being tested for a conclusion. The conclusion fact types are the one where the conditions have been for concluded. The conception that each statement can be written in a conditional form, has some remarks. There could be an unconditional computation for example. The principle ensures a structural presentation that can't be interpreted differently than it has to be.

Principle 5: The conclusion principle

A rule family contains only one conclusion fact type. By ensuring that there is only one conclusion column, there could only be one representation of the decision model. This representation will be the simplest and respects the atomic characteristic we desire. This way it will be easy to make any changes, because there is only one place to do so.

Principle 6: The conditions principles

The populated conditions in a rule family have an 'AND' relation, and no 'OR' relation. The latter relation should be decomposed into a different rule pattern, so that there are only 'AND'- relations.

This principle has some additional sub-principles about the rule patterns. This will not be described in this thesis.

Principle 7: The connection principle

There could be inferential relationships among rule families. This is the case when a conclusion fact type of one rule family serves as a condition fact type of another rule family. To grant a customer a booking in EU-Rent, a driver needs to be valid. This validation process can be seen in the Rule Family where the conditions for validation will be tested. One of the conditions is a valid driver license. For the conditions to come to a valid driver license, one should look at the Rule family for driver license, and so on. These are the relations that give structure to the model. There are also some additional principles contained in this principle, which will not be discussed further. This principle gives an integrated decision model structure.

2.3.2 Three declarative principles:

The decision model must be declarative to structure logic fully independent of technology and other considerations. Declarative solutions separate themselves from procedural ones. According to Date, it's all about what needs to be done, not the sequence of how it should be done. The declarative nature of the business logic is crucial to understand the development of the decision model. A business process should contain the step-by-step sequence of tasks, while the decision model should offer a structure of the declarative business logic. By separating those two concepts, we make the decision model independent.

Principle 8: The declarative heading principle

This principle is all about eliminating sequence in the columns of a rule family. There should be no order of the fact types in the heading of a rule family. If there is an implied sequence in the rule family, the declarative nature of is lost. The order of conditions should not have an effect on the conclusion. Hidden condition sequence should not be implemented in the decision model, like other unnecessary sequences of business logic should not be in a business process model. The separation provides a fully independent model.

Principle 9: The declarative body principle

This principle can be compared with previous principle, but now we assure that there is no implied sequence in the rows of a rule family. There should be no order in the entries of the body of a rule family. This also means that duplicate rows are not allowed, because they are redundant and do not add new business logic.

Principle 10: The declarative inferential relationship principle

This principle has its focus on the path among related rule families. Here should also not be any sequence through the inferential relationships of rule families. There should be no implication that one rule family should be evaluated before another rule family. Sometimes it seems that one should be evaluated before it can be used as a condition in another rule family. This can be refuted when we know there are two chaining methods. Forward-chaining is when we start with obtaining the values for the conclusion of one rule family to use it as a condition in another. There is also another chaining technique, which we call backward-chaining. Here we start with filling in an amount for the second rule family and go back to the first where we select the corresponding conclusion value for which we arrived at our initial amount. This is more complex to understand and derives values thru an iterative approach. Of course, performance can vary among these 2 methods, but this is not important for our model, which is technology independent.

2.3.3 Five integrity principles:

Decision model integrity is there to make sure that its content fully makes sense. It should make sense structurally, logically and according to the business. The model should avoid logical and business anomalies. The structure is made in a way that content is represented in one and only one right place in the model. The model is ought to be free from logical contradictions and should not miss any logic. The content is also correct to the leading business directions, assuring integrity businesswise.

Principle 11: Rule Pattern transitive conditions principle

Conditions in a rule family should not depend on each other for coming to a conclusion. This principle is related to the third normal form, which will be briefly discussed later on. This principle ensures that conditions are truly independent of each other in reaching to a conclusion. If dependencies

exist, there are too many representations of the same thing. The principle is able to remove such redundancies so maintenance is simplified and logical errors are avoided. To illustrate this, let's make a possible partial rule family for EU-rent's accepting of a booking request in table 2.1. This Rule Family is not according to principle 11 and the third normal form, because driver license influences the qualification of the driver. Driver license should be incorporated in the Rule Family driver. The discussion for this can be an issue of terminology what makes a driver qualified, but to illustrate this principle, this may be neglected.

conditions						conclusion	
driver license		driver		renters details		...	Booking request
is	valid	is	qualified	are	provided	is	accepted

Table 2.1: Rule family booking request (fault against principle 11)

Principle 12: Rule Family and rule pattern consistency principle

This principle is all about the elimination of any inconsistencies within a rule pattern and among overlapping rule patterns. This principle can be split down in several sub-principles. One describes that the execution of a rule pattern should not result in more than one conclusion for any set of valid input values. Another sub-principle states that the conditions of a rule pattern need only to considerate the input values for the conditions that the rule pattern is expected to process. This is also true for the conclusions; they just need to cover only the subset of the domain that is within scope. There are also sub-principles that describe overlapping in the condition operand or across rule patterns. Yet another sub-principle states that there should be at least one conclusion value for any set of valid input values for the condition fact types. They can also derive at more than one conclusion value; further rules could cover the selection in this case. Principle 12 removes in this way the inconsistencies in the content of the decision model. It helps uncovering situations where there could be inconsistencies in business logic; this happens before the implementation in software and testing.

Principle 13: Rule Family transitive conditions principle

This principle removes redundancies among rule families, much like the way principle 11 does among rule pattern conditions. It states that there are no inferential dependencies among these related rule

families. We use a term called transitive dependency to describe the problem. It's best understood by using an example. In figure 2.6 a decision model for the validation of a booking request for EU-Rent is drawn without the confirmation of principle 13. Practical this means that we start to look for circular paths. By eliminating the redundancies, maintenance is simplified and risk of logical errors is reduced.

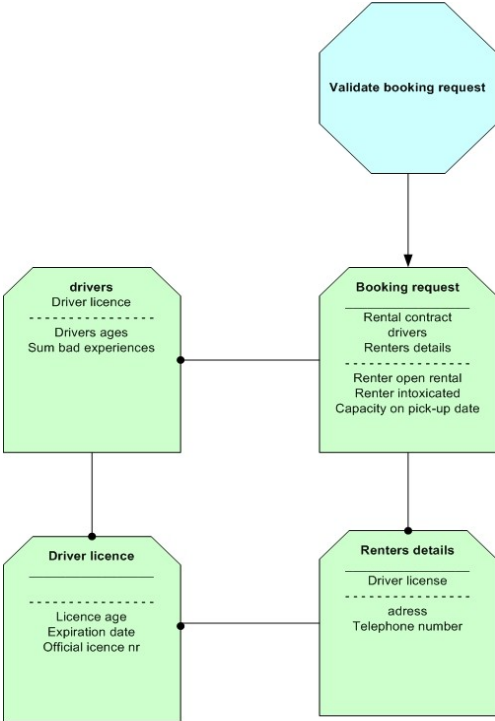


Figure 2.6: Circular path in a decision model (fault against principle 13)

Principle 14: Inferential integrity principle

This principle states that there should be no conclusions in a supporting rule family that are not covered by the associated dependent rule family. To understand this better, a definition is given for a supporting- and dependent rule family. A supporting rule family is one whose conclusion fact type is condition fact type in a different rule family. A dependent rule family is one with a condition fact type that is a conclusion fact type in a different rule family. This principle ensures completeness in a logical perspective. In that way that it makes all inferential relationships in a decision model complete.

Principle 15: Business alignment principle

This principle is all about the alignment of the decision model with business directions. Using a decision model should help you manage your business. This should be done with comparing business metrics and business results. Metrics are needed to evaluate The Decision Model, and see if it's performing as expected. Measuring effectiveness of business decisions is crucial. This is maybe the most important principle of them all, because this helps achieving the desired goals.

2.3.4 Three normal forms:

A part from concept of the principles, there is also the concept of normalization. Normalization is in line with the described principles. There are three normal forms which should be applied to the decision model. A decision model wherein those 3 normal forms are adapted delivers logic with maximum integrity and manageability.

First normal form

This is needed to interpret and represent the model in one and only one way. This means more concrete that a rule family row makes only one conclusion; and that there is no nesting of any conclusions inside these rows. We use AND relations among the rows of conditions and not OR relations. The latter would be in contrast with the first normal form. This normal form is aligned with principle 6 called 'the condition principle'. As stated above, this principle ensures the AND relations among conditions. The first normal form is about reducing business logic to its atomic form.

Second normal form

This is needed to eliminate any redundancies in the business logic. This normal form ensures that no conclusion is partially dependent on the populated conditions. A conclusion should be fully dependent on the conditions. Let's illustrate this with the help of the EU-rent case. In table 2.2 the partial rule family for car's service for EU-Rent is drawn. Reading the first condition key, reads :When a car has a mileage of over 5000 miles (8000 km), and the car group is A, then the car is scheduled for service. When one looks closer to the table, one sees that the Car group doesn't matter in coming to a conclusion about scheduling service or not. The conclusion is hereby partially dependent on the condition key. In this example the dependency is only with the car mileage since last service and the car's time since last maintenance (or service). The table is therefore not according to the second

normal form. There is a presence of a redundancy. This normal form is aligned with the principles about the Rule Patterns. More concrete, it is linked with one of the sub-principles of principle 6, named 'Rule pattern partial condition key dependency principle'.

		conditions					conclusion	
RP		mileage since last service	car's time since last maintenance		car group		Car's service	
1	>	5000			is	A	is	sheduled
1	<	5000			is	B	is	not sheduled
2			is	> 3 months	is	A	is	sheduled
2			is	< 3 months	is	A	is	not sheduled

Table 2.2: Partial Car's service schedule Rule Family EU-Rent (fault against normal form 2)

Third normal form

This is needed to eliminate functional dependencies among conditions. There is no condition that is partly or fully dependent on another condition in the same rule family. There should be no hidden logic within business logic statements. This is closely linked with principle 11.

Chapter 3 : Designing The Decision Model

In this chapter the development of a decision model becomes more clearly. There are two approaches to start developing The Decision Model. After a short explanation, the EU-Rent case will be used to design a decision model for certain tasks in EU-Rents' business process models.

3.1 approaches:

There are different approaches to start building up a decision model. There has been some discussion on that top-down approach should be better, but other differ from opinion. This discussion is based on the fact that data structures are often developed according to the top-down approach. With the top down proponents an interesting question has been asked: "Why not deliver iterative decision models whereat can be started even way before we know all the business rules?" With the bottom-up we produce first lists and with the top down we start modeling very quickly (Barbara von Halle and Larry Goldberg 2010). The benefit of these models can be that it is able to create value for the business people early on.

Top down

Here one should start with a simple list of ideas or condition fact types that serve for the target business decision. The list will evolve into related decision model structures. The next step is to place the conditions and conclusions in the correct places based on the normal forms described earlier. It's an iterative way of modeling, so no harm is done when the models are incomplete. It's easier to get a quick understanding of the complexity and of business decision values and their according data sources.

Bottom-up

According to Barbara von Halle and Larry Goldberg (2009) the bottom-up approach starts with gathering business logic statements from different sources. These sources can be people, documents and even program code. These collected statements will be transformed from textual form to a tabular form. This comes down to applying principle 1 of the decision model principles.

The next step is identifying fact types and logical expressions which have their relevance for the decision model's scope. A logical expression contains a fact type followed with an operator and an

operand. This step comes down to applying principles 2 and 3.

The third step would be making a distinction between condition fact types and conclusion fact types. This is done by checking which fact types inferentially leads to another fact type. The third step comes down to applying principle 4.

The fourth step should be creating the two-dimensional structures for the conclusion fact types. This is done by applying principle 5.

The next step involves making rule patterns whenever necessary. This comes down to applying principle 6.

The last step comes down to connecting the rule families which have an interferential relationship. Hereby principle 7 should be applied.

The bottom-up approach has mainly been applied to develop The Decision Model for the fictitious car rental company EU-Rent. The reason for this approach was based on the fact that the available info for this case was given in statements and data terminology on a more IT-level. Off course the link with BPM has been applied to align with the problem statement.

3.2 EU-rent case:

3.2.1 Introduction:

This case implemented in this thesis as example and link for the problem statement because the case is relatively easy to understand. One more or less knows a little about what a car rental business does. This case has been widely used by different companies for demonstration purposes.

The link that was in every reference list when something about EU-Rent came in, named www.eurobizrules.org/eurentcs/eurentoview.htm has been broke down. The needed information for this case of EU-Rent came from different sources on the internet. It came mainly from an SBVR formulation of business rules (OMG, 2008). The sources lacked completeness and consistency. This made it rather difficult to vision the whole structure of the enterprise and its processes.

Let's start with some short introduction of how the company is built. EU-rent is a fictitious car rental company. They have different branches in different countries. The main purpose of the company is off course to rent cars to customers. These customers could be individuals or organizations. Bookings for a car can be in advanced or by walk-ins on the day of the rental. The rental booking specifies the start and end-dates of the rental, the car group required and the branch where the

rental starts. The reservation may specify also if the rental is a one-way rental. This means that the car won't return to the pick-up branch, but will be dropped off at another branch. The cars are grouped in several car groups, where the price rate is the same.

There are a lot of rules coupled with this company, which we will explore. An important aspect is for instance that the records of the customers are kept and that 'bad experiences' will be recorded.

The EU-organization has the following structure:

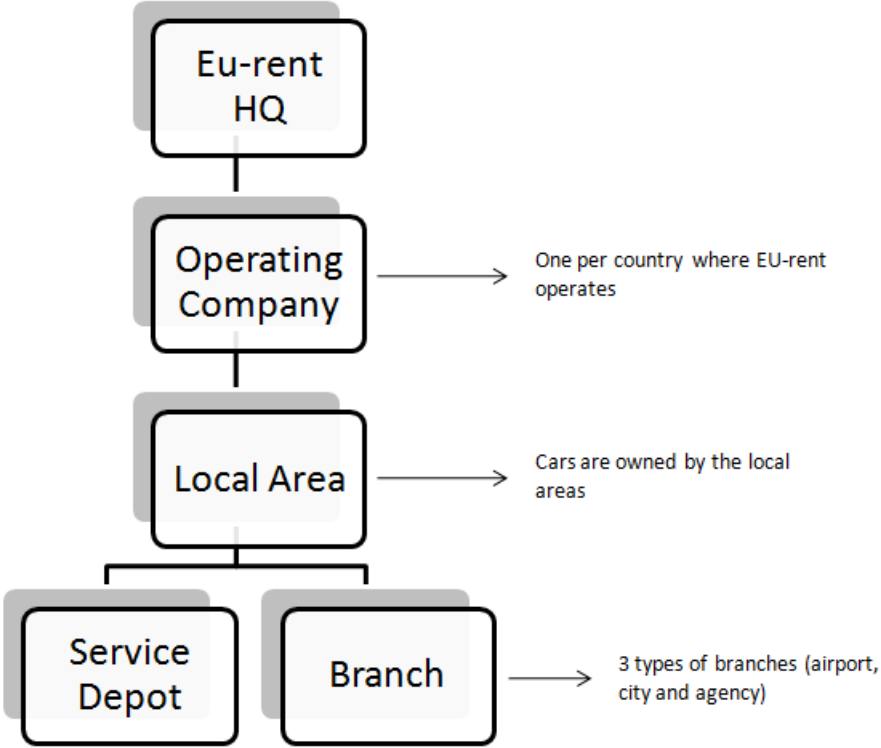


Figure 3.1: The organizational structure of EU-Rent

At the top is the world head quarter of EU-rent. In each country there is an operating company which sets rental tariffs, select car models and adapts global policy to local regulation. Within each country are several local areas and each local area contains a number of branches. A branch is the place where a pick-up or a return happens and there are three sorts of branches. The cars are stored at the branches but owned by the local areas. Car ownership can change when the rental demands a movement to another local area inside a country (when this is another country; ownership will not be transferred). This happens for instance when there is a one way rental or when a car is being transferred by an EU-rent employee.

The categories of branches all have their characteristics. An airport branch is open 24 hours a day and 7 days a week, have a storage capacity of hundreds of cars. A city branch is open during business hours and has storage space for tens of cars. An agency branch is operated on demand (service desks in hotels, travel agents, etc.) and has only space for a few cars.

The internal actors for each branch are a manager and booking clerks who arrange rentals and reservations for future rentals.

The local managers get significant flexibility and authority in the way that they can influence the number of cars owned by the branch, change tariffs to beat competitors' prices, add promotions, etc. The dynamic nature of the rules makes it interesting to separate decisions from processes. It is complicated to rewrite the processes each time a manager makes a change. The ideal way would be an independent managing of rules and processes. This way the organization can become agile and increases its competitiveness.

3.2.2 Developing decision models:

When we look at the request for booking (which has been used previously to illustrate the structure of The Decision Model), it informs EU-Rent of the car group required, the scheduled pick-up date/time and scheduled return date/time, and the pick-up branch and return branch, and provides details of the renter. It's difficult to envision this request for booking process, because the context is unknown. A booking request is evaluated when one speaks about an advance rental booking. Nothing is known from how this advance booking happens. It can be via telephone, via internet, personally, etc. Off course the conditions change in all these different situations. For example, when a customer makes an advance rental booking via the telephone, it is hard to be 100% sure about the given information (like is the drivers age indeed above the 21 years). When the advance rental is made through an internet application, it's often only the rental agreement represented in a checkbox that has to be ticked. This agreement can contain for instance that the driver has to be at least 21 years old or the driver license needs to be 3 years old at least. When a customer request a pick up, the driver will have to present information about its driver license again and assures EU-Rent that the person showing it, is the person owning the driver's license. This makes that the conditions to come to a conclusion about a valid driver's license has to be tested differently twice. There is definitely room for improvement. A lot of the info about EU-Rent comes from SBVR statements, which are not compliant with business processes. It's also a challenge to distinguish between definitions and

structural business rules. Structural rules supplement definitions. They are closely linked like described in appendix C. A whole list of rules for EU-Rent can also be found here.

To implement The Decision Model, I used a lot of these rules and tried to classify them according to decisions that come up during a process. A possible customer servicing process from EU-Rent is illustrated in figure 3.2 on a tactical level and in figure 3.3 on a lower level. This is done for an advance booking, because it's more interesting to evaluate than a walk-in rental which is more simplistic. The decision tasks are visualized with an orange octagon figure. The most common decision words used for decision tasks are: determine, validate, calculate, assess, select and choose. Additional information is added in text format to the figure. Some data object states are implemented to illustrate what their lifecycle is and how some of these objects change during a customer servicing process.

The process starts when the customer contacts EU-rent to make a booking for a car rental. The customer fills in (online) or transmits (telephone) his/her name and necessities to analyze the customer's acceptance. If the customer is a new one, he/she gets registered. After the obtained information a decision needs to be taken to accept or decline that customer (based on several things like bad experiences, age, etc.) After a request for booking is accepted, the process proceeds with obtaining the information for requirements (car group, pick-up date/branch, etc.). There are some rules for an acceptance of these requirements. When the requirements are accepted, a booking will be registered and the price will be calculated.

On the date of the pick-up, the customer will have to bring the needed documents (drivers' license, credit card, passport, etc). The request for pick-up will be analyzed and accepted or rejected. The rental contract that has been finalized by EU-Rent needs to be signed. After this is done a sub-process called hand-over car is started. This process includes a provisionally charge to the renter's credit card and the rental object becomes 'open'. The car is now in possession of the renter and the start date/time of the rental is registered.

After the rental duration, the customer drops off the rental car and the rental object transforms into 'returned'. When the grace period (contracted end date/time + 1 hour or closing time of branch) is over and the car is still not received, some additional actions need to be taken. For example after 48 hours the brand manager will be informed and has to inform the police with the claim of uninsured and stolen car. When the car is received, the time/date is registered and the drop off location is analyzed. A wrong drop-off branch includes a location penalty for the customer. A late return

includes a late return penalty. Ownership often changes under certain conditions; these will be analyzed to come to a possible change of ownership to the branch where the drop-off occurred. The car will be checked for any damages, where a possible damage can be registered. This will be recognized in the payment sub-process, where the bill needs to be paid by the customer.

A lot of the tasks illustrated in figure 3.3 are responsibilities of the sales clerk. The car park assistant is responsible for the check up in the figure. For simplicity the two actors are combined in the BPM on figure 3.3 with respect to the servicing of the customer. In reality a lot of the aspects from the BPM are more complex than illustrated. Insurance is an aspect that is not included into the SBVR EU-Rent case (OMG, 2008) or other sources. Damage to cars or unforeseen events during the rental period may or may not be included in the insurance selected by the customer. Although this aspect is not described, it is interesting to provide The Decision Model for decisions that include the possible conditions for certain insurance packages.

Another interesting implementation of The Decision Model is with the pricing calculation. This process task has the potential to become automated. Tax rates are implemented according to the necessary tax rate of the country where the branch is situated. The additional penalties are added and damage claims are analyzed according to the insurance of the renter, which may or may not be added to the bill. A volume discount can be implemented or all kinds of promotions that branch managers add to be competitive in their environment. Because the case study is simplistic described over all available sources of information, a lot of decisions are unknown due to a lack of knowledge. This falls for this reason outside the scope of the case study. Additional pricing information can be found in appendix C.

Let's analyze the two requests in the BPM process. These can be put in a decision model. As illustrated here, the customer will need an accepted request for booking. There is the complication that all drivers (also additional drivers) has to be checked by bad experiences for EU-rent and the number of additional drivers is at most three, so the number of drivers are at most four. There is a need to revise the request as presented before when explaining the model. When assuming that several requirements can't be presented by telephone or online, a need to revise the model is necessary.

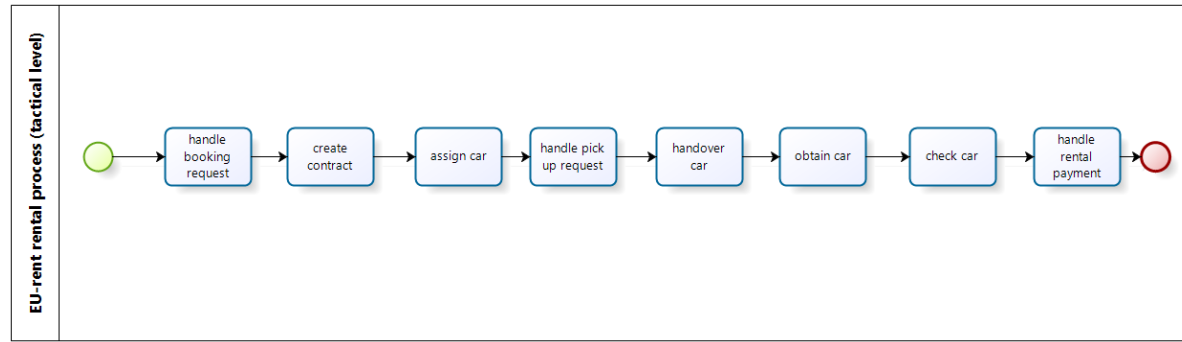


Figure 3.2: Rental process of EU-Rent on a tactical level

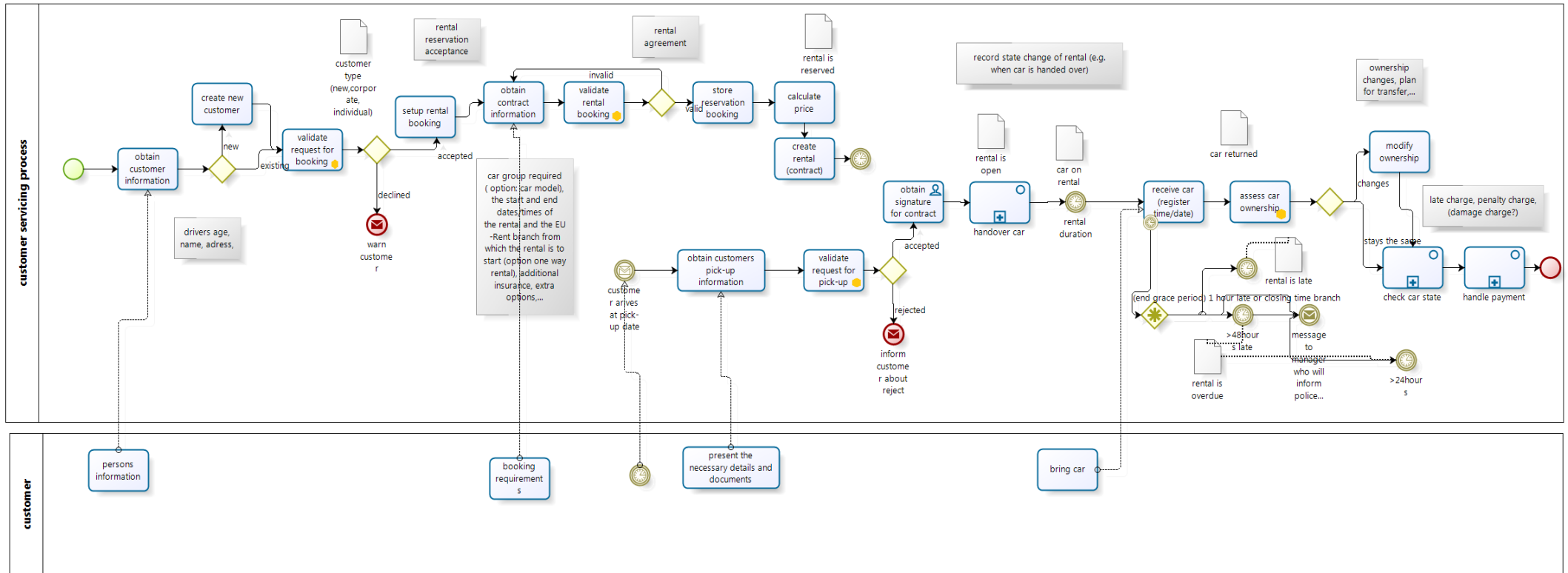


Figure 3.3: Possible customer servicing process of EU-Rent

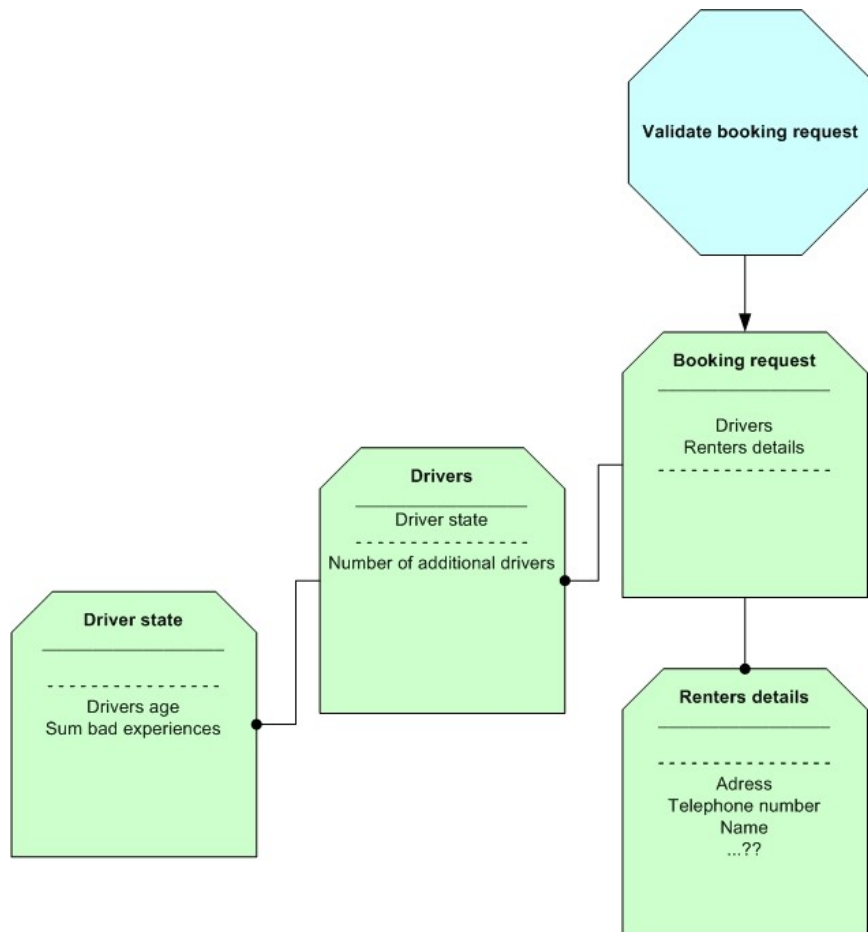


Figure 3.4. Attempt for Decision Model Diagram 'validate booking request'

The revised Decision Model for a booking request is illustrated in figure 3.4. The decision to accept the booking requests happens when the drivers are 'authorized' and the renter's details are provided. A renter is authorized when the driver state is 'qualified' for each driver and the number of additional drivers is at most three. Drivers come to a 'qualified' state when driver's age is above 21 and sum of bad experiences is less than three. Although the driver's age is not a fact because the lack of presenting an ID, this has been obtained by most of the online car rental companies. The reason is probably to have an early selection of probable customers. Often a company may accept a customer younger, but an additional fee for young drivers is added.

The reason for a first step of analyzing the customer is because modification will not make any sense. When a renter is blacklisted or has an age that is younger than 21 (according to SBVR OMG 2008), no modification can help for getting an accepted rental contract from EU-Rent. When one puts the request for validation and the validation of the rental booking together, the process can be stuck.

With this solution, an accepted customer can modify his/her contract, so that the customer gets a validation in the second validation task. The challenge here is the additional drivers, who also need to be qualified. If the renter isn't qualified, the rental will never happen. But when an additional driver is not qualified, a modification can happen to eliminate a termination of the process. This is not a perfect solution, so let us put the first 2 validations together and let's assume that getting stuck in the process is farfetched.

The rules used for a possible next solution are listed below. Additional rules from EU-Rent can be found in appendix C. Rules for the next Decision Model are:

- It's necessary that each rental authorizes at most 3 additional drivers.
- It is necessary that the booking date/time of a points rental is at least 5 days before the scheduled start date/time of the rental.
- It is necessary that the renter of each points rental is a club member.
- It is necessary that the rental duration of each rental that is the responsibility of a corporate renter is not greater than the maximum rental duration of the corporate rental agreement that is available to the corporate renter.
- It is obligatory that the start date of each reserved rental is in the future.
- It is obligatory that the rental duration of each rental is at most 90 rental days.
- If rental1 is not rental2 and the renter of rental1 is the renter of rental2 then it is obligatory that the rental period of rental1 does not overlap the rental period of rental2.
- Each car movement has exactly one movement-id.
- Each car movement has exactly one receiving branch.
- Each car movement has exactly one sending branch.
- Each car movement specifies exactly one car group.
- Each rental has exactly one return branch at a given date/time.
- Each advance rental specifies exactly one car group at a given date/time.
- Each rental booking has exactly one booking date/time.
- It's necessary that the booking date/time of the rental booking that establishes a cash rental is before the scheduled pick-up date/time of the rental.
- It's necessary that the booking date/time of the rental booking that establishes a points rental is at least 5 days before the scheduled pick-up date/time of the rental.

- The renter of each points rental is a club member.
- It is obligatory that each driver of a rental is qualified.
- ...

The Decision Model, presenting the logic of the rules stated here, is visible in figure 3.5. Here are the validations of the driver and the renter's details extended with rules about the validation of a rental (contract). The conclusion about a valid rental period is given by testing the Rule Family 'rental period'. To be valid, a rental overlap is nonexistent; the book date/ time has some separate rules that are influenced by for example the payment method. The appropriate maximum rental duration is dependable from the type of customer (corporate vs. individual). The according Rule Families are provided below The Decision Model in table 3.1 to 3.7. The Rule Families are often partly populated to keep the volume restricted; but they are enough populated to understand the logic which is necessary to come to the validation of the booking.

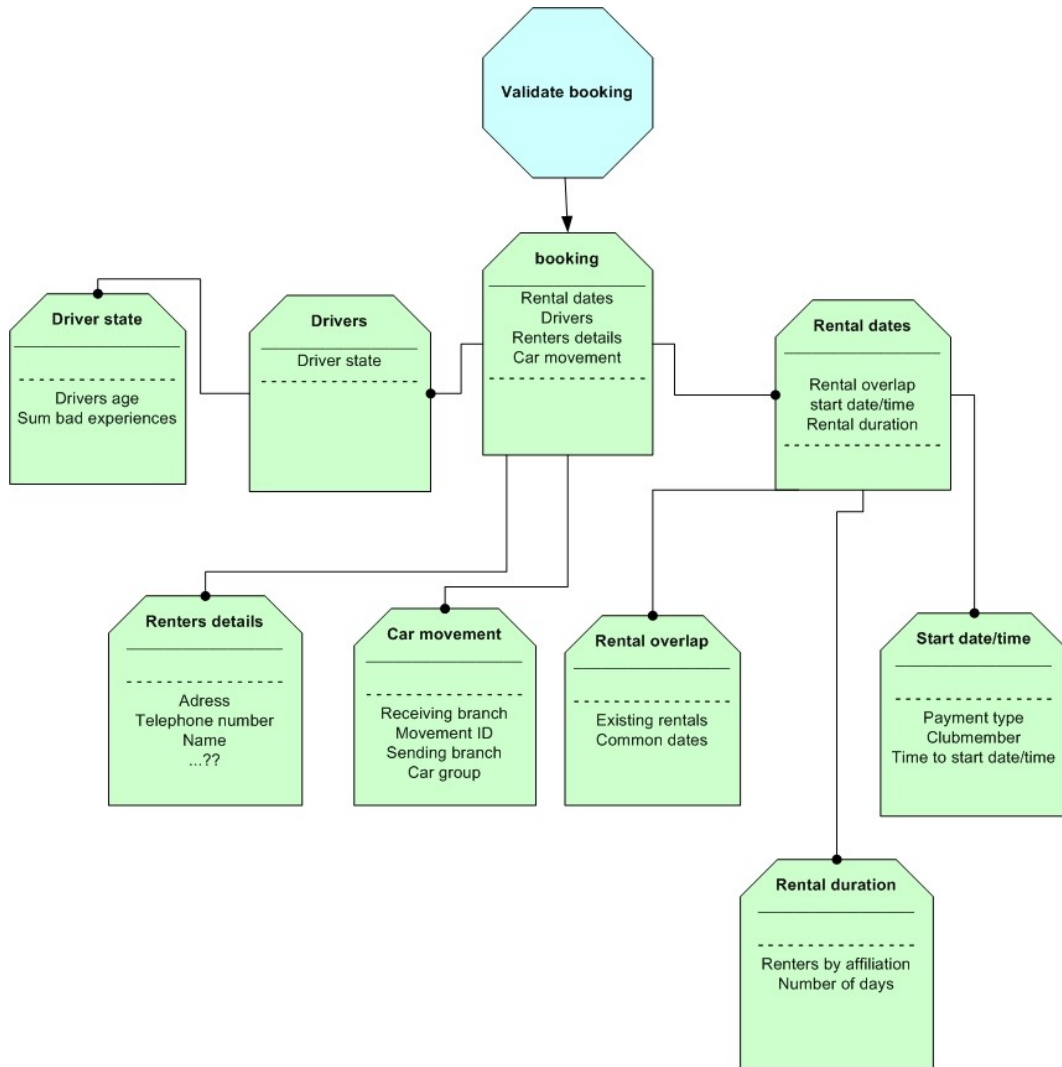


Figure 3.5: The Decision Model 'validate booking'

conditions								conclusion	
car movement		renters details		drivers		rental dates		booking	
is	correct	are	provided	are	authorized	are	acceptable	is	valid

Table 3.1: Rule Family 'booking'

conditions						conclusion	
rental duration		start date/time		rental overlap		rental dates	
is	accepted	is	valid	is	none	are	acceptable

Table 3.2: Rule Family 'rental dates'

conditions						conclusion	
time to start date/time		clubmember		payment type		start date/time	
>	5 days	is	yes	is	points rental	is	valid
≥	1 day			is	cash rental	is	valid

Table 3.3: Rule Family 'start date/time'

	Conditions				conclusion	
RP	common dates		existing rentals		rental overlap	
1	is	yes	is	yes	is	yes
2			is	no	is	none
1	is	no	is	yes	is	none

Table 3.4: Rule Family 'rental overlap'

Conditions				conclusion	
number of days		renters by affiliation		rental durations	
≤	90 days	is	individual customer	is	accepted
≤	corporate contracted	is	corporate renter	is	accepted

Table 3.5: Rule Family 'rental duration'

	conditions				conclusion	
RP	number additional drivers		driver state		drivers	
1	≤	3	is	qualified	are	authorised
2			is	barred	are	unauthorised
3	>	3			are	unauthorised

Table 3.6: Rule Family 'drivers'

	conditions				conclusion	
RP	sum of bad experiences		driver's age		driver state	
1	<	3	≥	21	is	qualified
2	≥	3			is	barred
3			<	21	is	barred

Table 3.7: Rule Family 'driver state'

In other sources of documentation, a statement called: "Reservations may be accepted only up to the capacity of the pick-up branch on the pick-up day." comes often to play. If this should be added, an additional rule family should be provided. The rule families for car movement and renter's details are not given. The reason for not giving the renters details table, is because this speaks for itself. The reason for not given the car movement rule family is because personally this can be hardcoded and doesn't mean anything for the business. That a rental only has one receiving branch, one return branch, only one requested car group, etc. seems to me a definite fact that would not change in the future because there is little business value. This can easily be coded into java or something similar as a data validation implementation and eliminated from The Decision Model itself.

Other requirements to rent a car are tested when there is personal contact. This is done with the pick-up of the rental car. A possible decision model for validating a request for pick-up is presented in figure 3.6. The statements used to format this model are presented in appendix c. One of them is: "It is obligatory that a request for pick-up is accepted only if an assigned rental matches the request for pick-up and the renter that is responsible for the assigned rental has a valid credit card and the renter provides current contact details and each driver of the rental has a valid driver license". Some additional sources of information are used because of a lack of the information in the SBVR (OMG,

2008) case study. The Rule Family 'Ability to control car' is added because often there was vaguely stated that too long or too short persons, drunken persons, etc. (that is the cause of inability to drive the requested car) will not be validated for pick-up. The reason for the Rule Family 'Rental match' is to be sure that the person physical present to pick up the car is the one registered in the contract and the details are the same as the one in the database. There can be some discussion to eliminate that rule family because it lacks a certain business value worth managing. These considerations illustrate that developing a decision model is an iterative operation. Every new decision model should be better than the previous one.

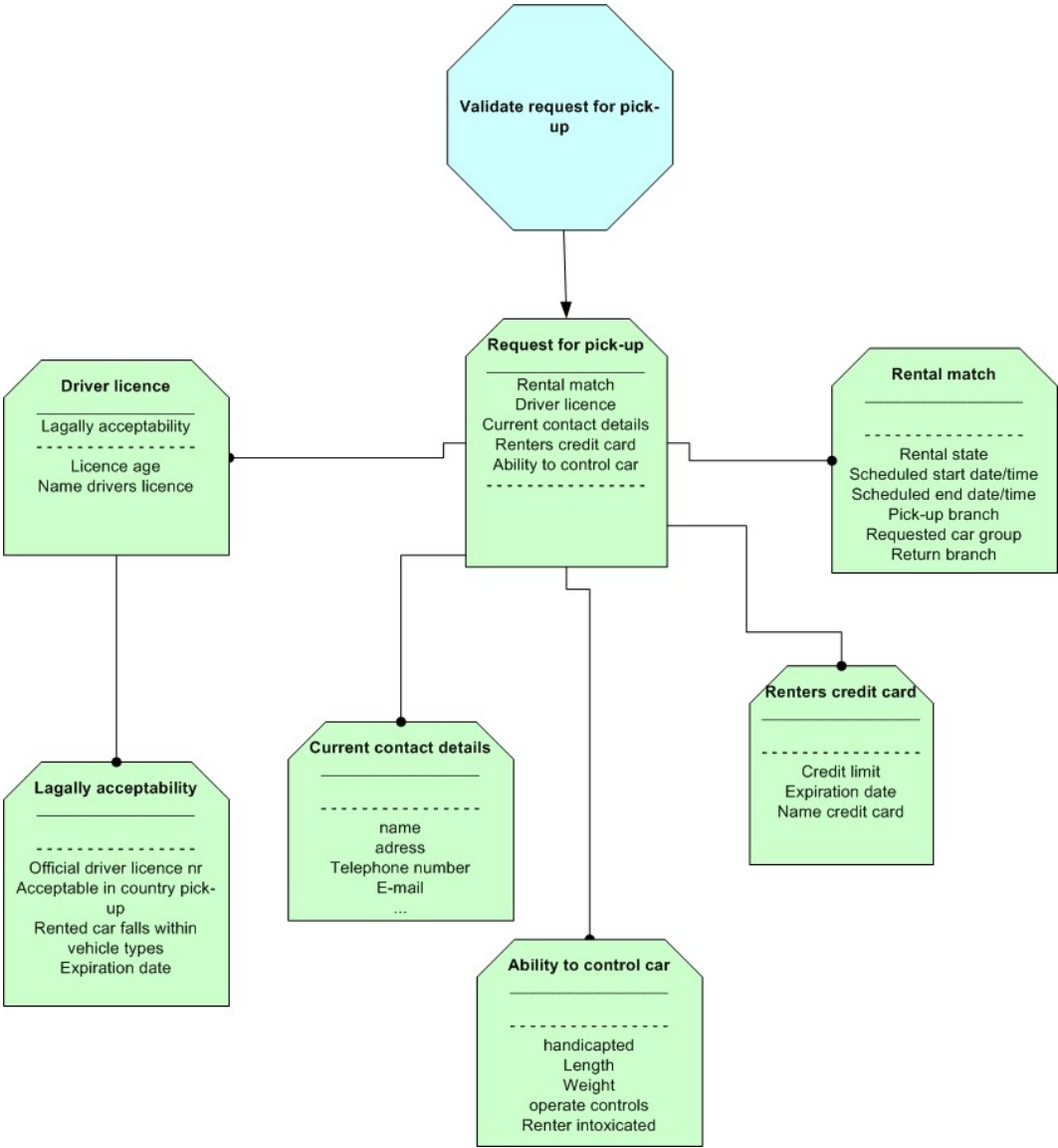


Figure 3.6: The Decision Model diagram 'validate request for pick-up'

The iterative approach and challenges of developing a decision model has been described for the EU-Rent case. Additional information about the EU-Rent case is provided in appendix C.

Not all business rules can be implemented in a decision model. The model represents the business logic, and out of the definition one can know that these decisions are coming from conditions that lead to conclusions. A rule of thumb is that a decision leads to at least two outcomes. There are several constraints or event based actions that can't be put into such a decision. There can be little to no structure when we would wish to model them. An example of a business rule that can't be implemented is: "At the end of each day cars are assigned to reservations for the following day." The implementation is done in a BPM with the use of an event. It's the time event that triggers the actions to assign cars. Other frequently used concepts like pre- and post conditions, for a certain task, are most likely in the form of rules. Preconditions are often easy to consider in a decision model, but post conditions can be more complicated.

There are some gaps in the logic of this case. In the real world a business expert can be consulted to obtain of the necessary information. The Decision Model is a great tool to find those gaps and recognize the need for further information at some aspects.

It's difficult to know which data is stored and which data only exists at the time of executing the Rule Family logic. In the real world an IT-expert can be consulted to understand what persistent data is and what not. When the data for filling in the Rule Family exists in a database, no additional inferential Rule Family is needed. I refer to appendix C to illustrate this for EU-Rent car availability. It's only for the interim knowledge further relations exists in The Decision Model. In the next sub-chapter is explained what the importance is of this interim knowledge that's not getting stored in a database, but is crucial to test for conclusions.

3.3 Variable data:

For coming to a conclusion of something, another condition can have to be met. This other condition can serve as a conclusion in another rule family, where there are again different conditions to set a value for the conclusion. This relation is called an inferential relationship. We know that not all data is being stored in databases. Some values of a conclusion fact types can be derived from a combination of conditions which are stored at a database, but the conclusion fact type itself is not stored and maybe is useful in another rule family. Let's call the many conclusions that eventually

serve to come to a final conclusion ‘mini-conclusions’. These mini-conclusions are judgments from which the data need not to be stored in a database. The data that comes out of the mini-conclusions is called dynamic data, because it is dynamically created in-flight throughout the logic. This dynamic data can be seen as an asset with substantial importance for the business. The decision model ensures the logical thinking behind every mini-conclusion. We can conclude that a decision model can serve as the glue which keeps the dependencies among rule families together. Barbara von Halle and Larry Goldberg introduce the concept of an ‘interim decision’ which is similar with the mini-conclusions.

3.4 The decision model view:

After the publication of the book about The Decision Model, the decision model has known some evolvment. Barbara von Halle and Larry Goldberg (2010) implemented the term decision model view to reach to a higher level of maturity with BDM. The challenge was to model very complex rules that for a company working worldwide has differences according to for instance locations or different lines of business. Let’s use the example of EU-rent. EU-rent has given some freedom to the branches, which can use their own promotions and discounts. The company has branches all over Europe, where there could be some difference about policies on insurance, rules about damages cars, rules which involves country regulations, and so on. This can cause that there will be many additional rule families and conditions or conclusions will have to be added. Assuming the rules are quite different among the different locations, one should be questioning the overall simple structure of the decision model. Although, this is not the problem of the decision model itself, it creates the need for improvement. Complexity can be severe within the ‘base’ view. The base view is the overall enterprise view which the decision model originally only handles with. Because multi-branch, multi-line companies can become complex within a decision model’s base view, there is the possibility to add view for specific areas. The notion of the view can be seen as looking at the decision model through a filter. The view can be added to the rule family shapes of the decision model. An ‘*’ symbol will be added to the view and the text shall be made ‘bolt’ to make clear that in that rule family an extension or change is present in comparison to the ‘base’ view. Let’s assume that the request for pick up validation is somewhat different in America than in Europe. The required license age is higher if it’s an American driver license. Different credit cards are acceptable when the country (of pick-up) is America. These are just examples represented in figure 3.7 to make The Decision Model View clear.

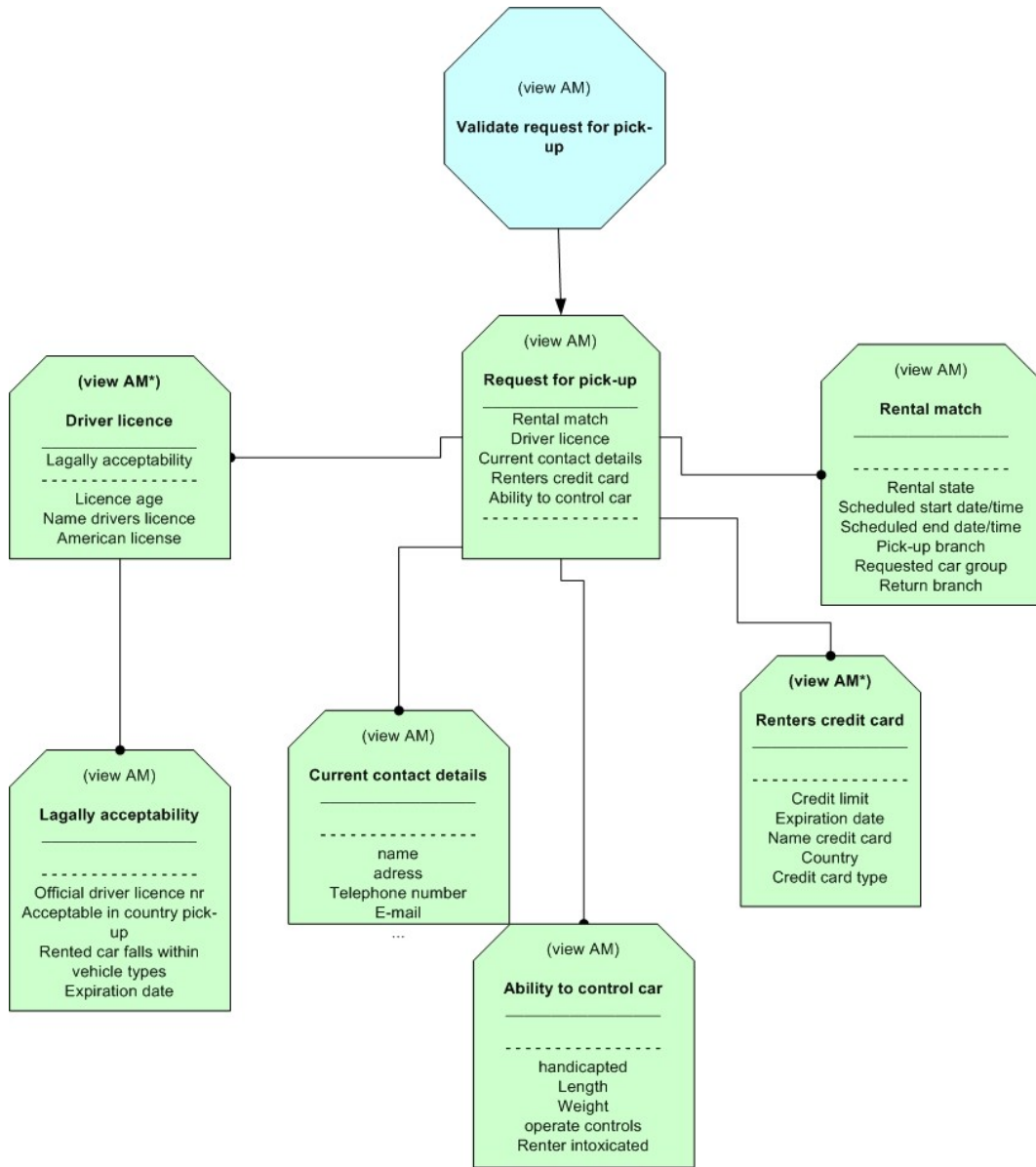


Figure 3.7: The Decision Model 'validate request for pick-up' with view: 'America'

Chapter 4 : Modeling or listing

Chapter 4 explains what the difference is with modeling or listing business rules. The ease of changing a model will be explained in the second sub-chapter.

4.1 Model vs. list:

There is no doubt that a business process model is an important technique nowadays for redesigning and automating business systems. Although business rules guide these processes, it still is difficult to represent these rules. Rules are often stated in a catalog form; but are catalogs the optimal way of presenting them?

There is a crucial difference when it comes to managing and automating a list versus managing and automating a model. The old business rules approaches were dealing with individual business rules, one at the time. Software for business management of technology-independent models of business logic is different than the software for business management of lists of business rules. Not only will the software be different for managing, it is also significantly different for automating. The decision model has a function of being an ideal vehicle for sharing logic among business and technical audiences. The model gives a great graphical representation of the logic structure in a way that is understandable for business experts. The rule families of the decision model serve to see a more detailed and atomic format of the logic. These rule families are an important aspect when we speak of automation goals. The company Open Rules has demonstrated that their business rule engine can execute Rule Families. This happens entirely without program code. Within the business rules practices prior to the decision model, there has been promoted to make a distinction between business management of logic and technical management of logic (BRMS).

MDA (model driven architecture) is a software design approach with main focus on separating design from architecture. MDA intends that a platform independent model (PIM) will survive changes in technologies and software architectures. A PIM represents a conceptual design realizing the functional requirements for an application. The Decision Model is a technology- and platform independent model.

Over the years a lot of aspects of the business have got their own model. The reasons for this are simple. The complexity decreases when we put something in a well structured model. Another reason is that changes can be done independently from other aspects. Different aspects need

different representation because their context varies. For example a data model shows the relation of each entity to another. Icons are used to represent the entities and lines are used for showing the relationships among them. The way of modeling process flow on the other hand, is best done with a flow of process tasks. Here the icon represent a process task and the lines the process sequence. In this way of thinking business logic should also have its own kind of model. The decision model is this model for business logic what never before has come to light.

4.2 Changes:

Changing business logic in the decision model is straightforward. This is done due to the fact that each conclusion has only one place in the model. The home of that conclusion can easily be traced. In previous sub-chapter were the difficulties of managing a list described in comparison with a model. In this sub-chapter the changes due to changing rules are applied to illustrate the straightforwardness of changes in a Decision Model. There are a couple of different changes that could occur. One could add a condition because a new rule wants to add this for testing to reach a conclusion. Another change could expect adding a conclusion or adding an interrelation between rule families. The simplest change is changing a populated Rule Families value. Let's assume that the driver's license has to be over 5 years instead of over 3 years to qualify as an accepted license for EU-Rent. This means one should only change that value in the rule family.

Adding a new condition

Let's explain how it works by taking an example of EU-Rent. Imagine the pick-up of the car from the EU-rent pick-up branch. The customer wants to pick-up his/her booked car to start the rental. There are some rules for accepting a pick-up from a customer. One of the rules is to provide a driver license which is valid for EU-Rent. The possible conditions to come to the validation conclusion are presented in table 4.1. Assume that EU-Rent headquarter has decided to make the validation more strict. This could be due to some experiences where customers represented another driver license than their own (reasons: loss, not yet graduated for driving, etc.). This could increase the risk for bad experiences with customers and could put EU-Rent in bad light with insurance companies. To assure that something like this does not happen anymore, EU-rent can add another condition to check if the customer presenting the driver license has the same name as the name on the driver license. To add this new condition no other rule family is needed. One should just add a column to the conditions.

The new Rule Family is presented in table 4.2. To find the fact type ‘name drivers license’ back, it is off course also added to the decision model in the rule family ‘drivers license’. When we look back at previous chapter about the link between BPM and The Decision Model, we can conclude that if a condition needs to be changed when decisions are merged into a BPM is a much more complicated process than changing within a Rule Family.

conditions							conclusion	
legally acceptable		licence age		expiration date			drivers licence	
is	yes	>	5 years	>	sheduled end-date		is	valid

Table 4.1: Initial rule family ‘drivers license’

conditions							conclusion		
legally acceptable		licence age		expiration date	name drivers license		drivers licence		
is	yes	>	5 years	>	sheduled end-date	is same as	name driver	is	valid

Table 4.2: Added condition to rule family ‘drivers license’

Adding a new conclusion

A new conclusion is a broader concept and can have different variations. There can be some reasons to add a new conclusion into an existing decision model. This can be due to an additional rule that has to be implemented in the company. If a conclusion is added in the model, this means that there should be an additional Rule Family. There is a need to add a Rule Family to The Decision Model because conclusions are results from tested conditions. The conditions to come to that conclusion should be gathered and structured in a Rule Family and the according decision model diagram. In an existing decision model, the adding of a conclusion fact type causes an extra interferential relationship. This is somewhat more complicated than just adding a column when a new condition is implemented. But The Decision Model ensures a structure where everything has its place and searching for the right place is more natural.

It could be that when a new conclusion is the conclusion from a decision rule family, a new model

should be made. This can happen when a process needs an additional decision; the decision can then be modeled in a new decision model if it's worth managing.

Chapter 5 : Other representation methods

Some other ways for visualization of business rules and decisions will be explored in this chapter. The differences with a decision model and the benefits and disadvantages will be explained. This briefly exploring in other methods can help deciding what representation is best used in a certain context.

5.1 Decision tables:

Decision tables can be a great visual aid when there are not too many rules grouped. A disadvantage according to Smals Research is that they can't give a general view from the rules and their connectivity. There is no option to link different decision tables that have inferential relations. The forms of those tables can vary and there are no strict policies in developing a decision table. When there are some complex rules, a decision table is not ideal. Because there are multiple conditions linked with multiple actions, it can be very complicated to model complex rules within such tables without losing the general view.

Rule families and decision tables seems to be the same at first glance. They both have a two-dimensional structure and work more or less with conditions and conclusions (actions in case of the decision tables). In fact there are some big differences between the two. First of all, rule families are built upon certain principles described previously, decision tables are not. Second, there is no view of relations among decision tables. Rule families can be related to each other, and this can be seen easily with the decision model. One can say that a decision table has its benefits in representing logic, but to reach a higher level of maturity a Rule Family is the next step forward. The table 5.1 can be a decision table of logic represented In the Rule Family in chapter 2 (figure 2.1).

CONDITIONS							
VIP		YES	YES	YES	NO	NO	NO
number of days < 7		YES	NO	NO	YES	NO	NO
number of days ≥7 and ≤13		NO	YES	NO	NO	YES	NO
number of day ≥14		NO	NO	YES	NO	NO	YES
ACTIONS:							
discount	10%	X	X	X			X
	0%				X		
	5%					X	

Table 5.1: A decision table for the volume discount at EU-Rent

5.2 Decision trees:

Nayab (2011) states that decision trees are ideal as modeling tool for very simple rules. When the rules are simple, a decision tree provides an easy to view illustration of rules. The construction can be impossible to make, when the rules are more complex. Even if the complex construction is made, it loses its purpose as an easy to view presentation of decisions. Preparing large decision trees with many branches is a complex and time-consuming affair. Maintenance is also very difficult, certainly when there are nodes that should be eliminated or added. A small change in input data can cause very large changes in the tree. With certain changes, a redraw can be the only possible solution. Another point of critique is that the decision tree suggests a sequence, so this is a procedural representation of logic. It's known by now that declarative description is the best for dealing with rules and decisions. Figure 5.1 represents a decision tree again for the volume discount in the EU-Rent case (a different terminology is used for the number of days). This is not according to the official presentation, but is rather intended to illustrate the concept of a decision tree. Normally here is a symbol for the conditions (called antecedents) and one for the conclusions (called consequents), connected with lines. This figure is a decision tree on a very low level. We could also represent a decision tree where we go from a known discount to the next step to take at the end a decision on the final price for instance. Once the trees have been drawn out, it can be put into Rule Families, because these are technology independent and declarative.

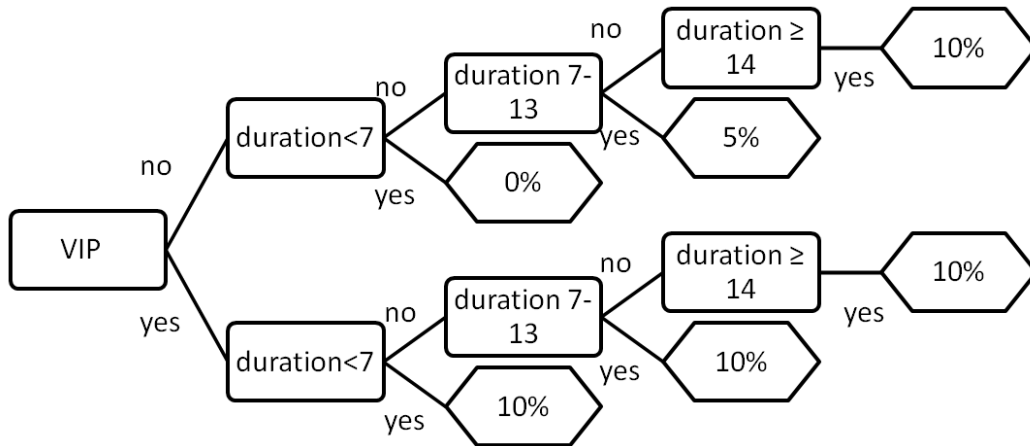


Figure 5.1: A decision tree for the volume discount at EU-Rent

5.3 Declarative (natural) language:

This is not really a model, but often used to describe rules. It uses a business vocabulary whereby business people easily can understand the rules. It's easy to make errors against consistency with this method. The business people should all use the same rigorous grammar when making statements, but this means that people have to be trained in the grammar to write the language. It is also challenging to group the logical statements made by the natural language.

Semantics of Business Vocabulary and Business Rules (SBVR) is such a natural language approach that is a standard made by OMG. The meaning of natural languages is that the vocabulary is aligned with the business and its enterprise glossary. SBVR has not much to do with the needs of IT operations. This is a language more for business people, hereby not really crossing the gap between IT and business people. An example of this language is: "It is obligatory that the drop-off date of a rental precedes the expiration date on the driver's license of the customer responsible for the rental." Like stated above SBVR (OMG) elevates business rules to be understood mainly by business audiences, not only technical audiences. It has a clear business perspective. SBVR creates a context for the rules and the business vocabulary, by introducing a grammar for business rule statements. From a business perspective, there are structural business rules and operative business rules. The syntax is complex, but is very rich and can cover a wide range of statements and logic. As an example, some operative business rule statements with their classification:

- Obligation statement:

"If the drop-off location of a rental is not the EU-Rent site of the return branch of the rental, then it is

obligatory that the rental incurs a location penalty charge.

“A rental must incur a location penalty charge if the drop-off location of the rental is not the EU-Rent site of the return branch of the rental.”

-Prohibitive Statements:

“It is prohibited that a rental is open if a driver of the rental is a barred driver.”

“A rental must not be open if a driver of the rental is a barred driver.”

-Restricted Permissive Statements

“It is permitted that a rental is open only if an estimated rental charge is provisionally charged to the credit card of the renter of the rental.”

“A rental may be open only if an estimated rental charge is provisionally charged to the credit card of the renter of the rental.”

SBVR and The Decision Model play different roles and serve different purposes. SBVR represents fact types and expresses individual rules about the fact types. It uses a more natural language instead of a program language. SBVR translates statements with rules and policies into such a language. The decision model is a graphical model for business decisions. The Decision Model groups the business logic in a way according to decisions. This creates a structure that is done with applying the principles and normalizations. Inside a rule family are not complete language sentences, but a strict structure where rows can assemble statements. There is no conflict between SBVR and The Decision Model; The Decision Model just creates a visual presentation and a very interesting higher business context. SBVR statements can be used to populate Rule families and create a visual model (and vice versa). Here we can make the distinction between top-down and bottom-up decision models. The development can start from the SBVR statements and then organize them into normalized Rule Family structures. A first step of this approach has been done by implementing when of the statements inside a Rule Family in table 5.2. Another way is to start with finding conditions and conclusion, organize them into rule families (according to the normalization and principles) and then translate each instance into SBVR statements.

	conditions				conclusion	
rule pattern	rental's driver	rental's estimated charge			rental	
1	is barred				is	not open
2		is provisionally charged to	renter's credit card		is	open

Table 5.2: Example of putting a statement in a Rule Family

5.4 Question charts (Q-charts):

This rather new method by Ronald G. Ross (Business Rules Solutions, 2010) provides a means to represent the structure of decisions. In an elongated hexagon, which stands for a decision, is a question present that represents that decision. This method has an intermediate part named Q-COE that represents the step in between before reaching the actual complete decision logic. The 'Q' stands for 'question', the 'C' stands for 'considerations', the 'O' stands for 'outcomes' and the 'E' stands for 'exceptions'. A chart for the volume discount for EU-rent has been illustrated in figure 5.2. This is a more or less similar self-made presentation of the Q-COE. The official version can be seen in the appendix F. The different charts can have connections to each other through dependencies between the decisions. There has been made a distinction between two kinds of decision dependencies. One is a consideration dependency and the other is called a relevance dependency. A consideration dependency is existent between two decisions where the first one needs the outcome of the second one. The relevance dependency is another way of saying that one decision is meaningless unless the other has an outcome that makes the first one relevant. The term 'independent subdecisions' is defined in a way that is very similar with the concept of interferential relationships among rule families. Each of the subdecisions has their own unique set of considerations. The logic for an overall decision can be presented as a Question Chart (Q-chart). A figure of this can be found in appendix F.

When developed the charts, the full logic will be made visual in decision tables. The advantages and disadvantages of decision tables are already explained. The Q-charts themselves are not based on strict principles, so errors could occur easily. It is an easy and great tool for sketching the scope and decisions that are necessary for a project. A more strict structured approach should be following next to this.

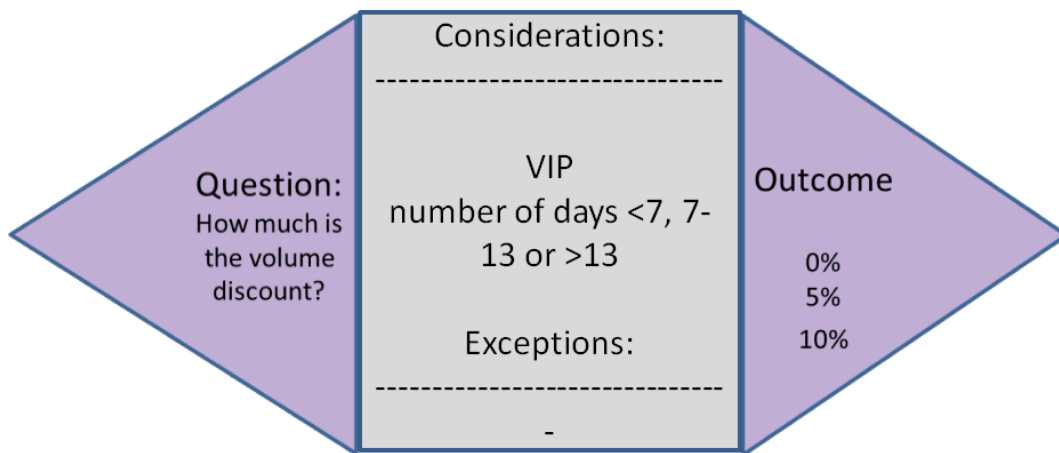


Figure 5.2: A Q-CEO for the volume discount at EU-Rent

To end this chapter, let's notify that there are more representations than the ones handled in this chapter. These have the most relevance for this thesis, but be noted that these are not a summation of all the representations of logic. Alcedo Coenen (2011) wrote on his blog a theory of looking at rules from different perspectives, at which we will come back later on. He states that The Decision Model is a revolution in particular for the system designer (or analyst). In a discussion with Larry Goldberg he explains that his attentions were not to limit the value of The Decision Model only to that perspective. For some perspectives of a business, The Decision Model will not be the most obvious representation, but The Decision Model reaches to a lot of aspects in a business and is therefore a tool that could limit some gaps between different perspectives. Modeling of such a model can be learned rather easy and the factor of understanding by different audiences makes it interesting.

Chapter 6 : Approaches based on separation

6.1 Architecture:

A lot of terms came up in previous chapters that relate to the separation of rules approaches. To enjoy the outmost benefits of such an approach, it should be implemented across an enterprise. This sub-chapter starts with the enterprise architecture and where The Decision Model fits in it. One can read how the aspect of a decision outgrows that from a rule when one speaks of the architecture of a business. After that service oriented architecture (SOA) will be briefly discussed to see what the role of The Decision Model is according to SOA.

6.1.1 Enterprise architecture:

In the book “The Decision Model a logic framework linking business and technology” (Barbara von Halle, Larry Goldberg, 2009) the Decision Model get’s a place within the overall architecture of a company. When hearing about Enterprise Architecture (EA), the Zachman Framework comes to mind. This framework is the standard for EA. Enterprises have to handle with a lot of challenges nowadays. A product’s complexity and rate of change increases and there is a reduction in time to market. It’s not only the industrial products that are feeling those challenges; it’s the entire enterprise that deals with increasing complexity and changes. When reviewing thousands of years in the history, we can say that humans only came up with ‘architecture’ as a device to deal with complexity and change. Architecture is here the broad term containing a set of descriptive representations. John Zachman developed a framework that shows how the enterprise architecture looks like. The architecture falls into a two-dimensional classification system. There is the interrogative dimension (with question words like: what, how, where, who, when and why) and there is an audience dimension (with terms like: scope, owner, designer, builder, sub-contractor and operator).The latter dimension appears as rows, while the first dimension appears as columns. The whole framework can be found back in appendix E. We can see that rules have their place in this framework. Column 6, Row 3 is called ‘the business rule model’. The decision model is in EA a solution that has not yet been available for the architect. It provides the opportunity for advances in information technology. There should be a place where the decision models can be deployed inside such an EA. The techniques of EA lead to a decomposition of the enterprise into basic components which can relate to each other. When we look at the Zachman Framework, we know that column 6 is the column where the decision model plays its most important role. There are several connection to

other columns, like the WHAT (data) and HOW (function) columns. There should be some necessary modifications to the cells of the framework, so the decision model could reach its full potential. The first three rows of column 6 should be modified, because the fourth row is platform specific. To understand this better I refer to appendix E.

Starting with row 1; the role of the decision model is to contribute a mechanism for scoping which decisions support the goals and strategies of an enterprise. The audience perspective for this row is the planner, so the decision model can help in this perspective with scoping what decisions should be made.

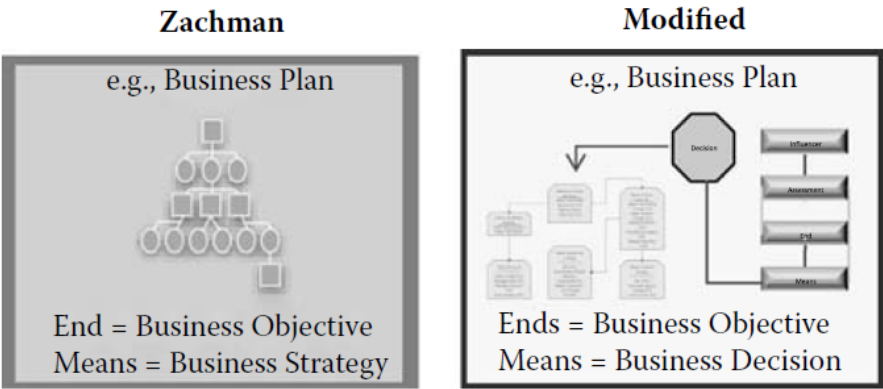


Figure 6.1: Modification of row 2 column 6 of the Zachman Framework (Adapted from John Zachman’s “The Framework for Enterprise Architecture (The Zachman Framework)” included in the book “The Decision Model a business logic framework linking business and technology” (Barbara von Halle and Larry Goldberg, 2009)

The audience for row 2 is the owner, which needs a business perspective relating to the use of products (or services) of the operating business. The role which the decision model could play is that it serves this audience as it describes architecture of the use of business logic related to the products and/or services. In this cell should be a clear connection between a Business Motivation Model (BMM) and the decision model itself. The decision model is more likely to have more connections to row 3 and 4 than the BMM has. Row 2 is the most appropriately position of the decision model. The owner is the one that needs the business perspective that the decision model assures.

Row 3 has as audience the designer, which needs a system model. This system model is all about the mediation of what is desired at row2 for products/services and what is technically feasible. The decision model can help specifying this feasibility. The decision model plays also a role in rows 4

through 6, but modifications are not necessary.

By making these modifications as seen for row 2 in figure 6.1, the decision model gets its place in the Zachman Framework. This will enable that the business logic of an enterprise is connected to relevant views and domains of the enterprise. Objectives of the business connect to the according business logic through business decisions. This is true throughout the whole developing from the decision model to the implementation and deployment of those models.

6.1.2 Perspectives:

An interesting theory has been provided on Alcedo Coenen's blog (2011), where the enterprise and the right approaches for handling with rules can be seen from different perspectives. When one looks from the perspective of the manager, the term Business Rule Approach comes to mind. This approach accentuates the organizational perspective. A business rule is defined as a specification of policy and practice within the organization. The rules are kept often in a repository in a natural language what is easy to understand for the managers. The standard from OMG used as a natural language is called Semantics of Business Vocabulary and Business Rules (SBVR).

When looking at the perspective of a system designer, business rules are specification of business logic. The specification of business rules together with those of business objects, business processes and business services determine how the to-built system has to behave. Unified Modeling Language (UML) can be used to specify the logic with Object Constraint Language (OCL), which will not be discussed further. The only thing to know is that it does not represent the logic in an easy to read way. It looks more like coding than a good and easy to read presentation. In the Zachman Framework is a dimension defined that makes the reference to the business rules concept. This column is called 'motivation'. The problem is that business rules have a detailed level that doesn't fit within the more global specifications of business architecture. This makes it inconvenient to specify business rules on the architecture level of business objects and business processes. A solution to this came with the term 'decision' which couples different rules together. The gap between the architecture level and a more detailed level is filled up with the Decision Model and its rule families.

6.1.3 SOA (service oriented architecture):

Obtained from the theory of Barbara von Halle and Larry Goldberg (2009), SOA and The Decision Model are compatible and are helping each other. Major application vendors nowadays are transforming their products into collections of smaller services, tied together by processes, and

executed on a SOA platform. An application can be seen as a process-driven compilation of several services. Services accommodate functions, decisions and information that support processes. They are modular units of business functionality. The three basic roles they play in supporting business processes are: tasks, entities and decisions.

Task services are used for a procedural action; decision services are services with the business logic inside of it and entity services are used for data access. The decision model is useful in the decision services. It's common sense that those services can be developed with the decision model as a clear and complete representation of the logic. Each service often has some business logic inside of it. The decision service should always be separated even if the decision service is never used in a stand-alone mode. Each rule family has the potential of being a stand-alone decision service. The Decision Model can also be a starting point for the creation of inventory of services for business logic. The Decision Model should be seen as playing a role in decision services inside the SOA (service oriented architecture). These decision services can be used across multiple processes; this makes them easier to manage.

Software as a Service (SaaS) is increasing in popularity. The users of SaaS don't need to assemble the necessary hardware or software, but just use the software on demand. The software is often delivered through a cloud (internet via a web browser). There is even the term 'RaaS' that means rules as a service. SOA plays a major role in this kind of software service. SOA enables to make better, more modular and flexible solution for the enterprise, with the necessary support for business processes (Barbara von Halle, Larry Goldberg, 2009).

6.2 Separation of rules:

The concept of separating rules from processes has become a well known technique during the last decade. A business rule engine (BRE) is one of the oldest terms in this area. This sub-chapter will briefly explain what BRE, BRMS and the link between BRMS and BPM present. One can make a distinction between platform independent and platform specific. BRMS and BRE are platform specific and represent technology dependent software packages what can be obtained from different vendors. BPM and The Decision Model are platform independent models (PIM) what mean that for automation, there are some lower levels required to implement these models. The link with previous sub-chapter can be seen in the appendix E.

6.2.1 BRE:

Business rule engines have already a long history. It is in a technical way a kind of interpreter. The rules are read, interpreted and executed like data during the execution of the application. An advantage is that the rules can be changed even during the execution of the program (run time adaptability). This was already a big step in making the enterprise and systems more agile. Business Rule Management (BRM) was the next step and an important tool for separation and agility.

6.2.2 BRMS:

A business rule technology is great for separating the rules from processes. It enables executing business logic at run time. A task in a BPM can call up a BRMS to execute the rules that are necessary. Discovering, documenting and managing the business rules are not the main tasks of a BRMS, so BRMS will not be the best solution for these tasks. Within a BRMS, business logic could also be getting easily lost. It is not a technology-independent, rigorous and normalized method. So it is rather hard to understand by everyone (IT and business). It's more useful for IT as an IT tool. Although a lot has been improved within the recent BRMS systems, working with rules will not get one to a higher level of maturity. Different additional functions and options have been implemented by vendors of recent business rule management systems. This is a step in the right direction, but Barbara von Halle and Larry Goldberg (2009) are saying that unless the grouping according to decisions is well applied, there will be a loss of the huge potential of separating rules and processes. The decision model creates business logic which is compatible with BRMS. It also can help with creating decision services which are compatible with BPMS technology. Visual Rules is a BRMS system which also can model decisions and rules, but on a less structural way. They also work with decision tables instead of rule families.

6.2.3 BPM and BRMS:

On the blog-page of Tom Debevoise (2009) a perception of the collaboration between BPM and BRMS is illustrated. In BPMN 2.0 there is a special notation available what he calls a 'decisioning activity' (previously also called in a more general way: a decision task) what will be followed with a gateway. A 'decisioning activity' is an activity in a BPM which relies on a BRMS. The decision behavior is a name for the output of such an activity. Combining business process modeling with a business rules approach is the right way to go to a more agile solution. When we look at it technically, we see that the process logic is placed within the BPM suite and the decision logic lies within the BRMS.

Process logic is procedural and forms the sequences; it controls activities and launches or cancels processes. Control is attained with the help of timers and exception handlers. The BRMS has the task to organize and execute the rationale behind a process decision. What one surely should understand is that The Decision Model is not a replacement for the BRM systems. The model is not a system and so it can work collaborative to improve processes today.

6.3 Separation of decisions:

In this sub-chapter we arrive at the separation of decisions. This level ensures the separation of rules, but they are grouped according to the decision that they support. The management term used for managing these decisions separately is called Business Decision Management (BDM, EDM or DM). This in contrast with previous chapter where rules were separated and managing these was called Business Rules Management (BRM).

6.3.1 Somewhat a history:

Let's look at what has been happening around decisions and rules for the last couple of years. In 2007 Addison-Wesley published a book by James Taylor and Neil Raden called Smart {Enough} Systems. In this book some clear statements come to the horizon. Ones like operation decision are important, or operational decisions can and should be automated; and that taking control of operational decisions is a crucial source of competitive advantage. Herein came also a clear call for what is called Business Decision Management. The business rules era has somewhat ended since. The business rule manifesto was created around the end of the nineties. Here was the focus on separating and managing business rules. Since the book, a higher level component was introduced, called the business decision which included these business rules. The business decision has a link with objectives, motivations and operations of the business. It is manageable and has an impact on operations what could be tracked. It became too difficult to manage the often thousands of business rules. 2008 was a year where the early work in BDM began to mature. In 2009 came the book out, mostly referred to during this thesis, about The Decision Model; written by Barbara von Halle and Larry Goldberg.

6.3.2 BDM (interchangeable called EDM or DM):

According to James Taylor (2009) decision management is more than just the business rules. Business Decision Management is the practice of managing smart decisions that can be changed and managed with the goal of being more agile.

The business motivations should be aligned with the decisions. This means that the objectives of a business plan should be shown in the smart decisions the company makes. These business objectives should be measured over time to evaluate certain decisions. Business metrics are used to find out that the business is right on track for its objectives. The use of analytics is also implemented in a good BDM. Predictions can be made with more complex analytic tools. The analytic tools are used to track the performance of business decisions. It's sometimes difficult to automate the decisions behind such an analysis. It's a challenge to set up logic behind for instance a segmentation decision, but sometimes it's possible and pays off. The role of a decision model in BDM is to assure a stable, normalized and complete representation of the business logic.

An example of decision modeling tools for business analysts are MS/Visio and rule Guide; for business people is rule guide popular and for enterprise architects is Enterprise Architecture popular.

Business decision management (BDM) is also called Enterprise decision management (EDM) and decision management (DM). Between the terms is not really a difference, it's more a matter of emphasis. When we use the term EDM, the emphasis is on the importance of enterprise ownership of decisions. With BDM the emphasis is somewhat more on the parallel nature and link of the concept to BPM. Decision management is just a short term for describing either of those terms. Enterprise decision management is an approach to automate and better manage the omnipresent operational decisions. Operational decisions are used frequently. They deliver a high volume. This are rules that processes use a lot of times to come to the next action. Focusing on these operational decisions, it creates decision services. These decision services are using business rules to automate the decisions.

It's most efficient to use a SOA; by which the decision service can be seen as the brain of the application component. The next element of EDM is the use of business rules (and a BRMS) to manage the logic inside the decision services. The business rules in these decision services come from policies, regulation, expertise or analysis of information. These business rules can be presented in all kind of ways (decision trees, decision tables...). Another part of EDM is the analytics aspect, which adds insight to the decision services. The adding of analytic insight to the services happens

with the use of what J.Taylor and N.Raden (2009) call predictive analytics. To control and analyze the impact of decisions, there is a need to track the results. Adaptive control is used to optimize the decisions and reach the preferred goals. The big difference according to Barbara von Halle and Larry Goldberg (2009) between the Decision Model and EDM is that EDM is focused on the automation and improvement of decisions. The decision model doesn't make a distinction between automated and manual decisions. Another difference is that an EDM decision may contain many decision models because they combine procedural and declarative steps. The Decision Model only uses declarative steps. This means that if a bigger decision service needs some kind of procedure (procedural steps); the EDM will contain several Decision Models of the separate declarative parts. These differences ensure that not every decision from the decision model is relevant to EDM. As a consequence, EDM is not used to automate the Decision Model. The Decision Model can be used in several steps of the EDM. It can help finding the right decision where EDM can be applied, help with analyzing, help understand and rationalize the decisions, and more. The conclusion should be that the decision model doesn't substitute for EDM and vice versa. They are compatible and both add value to each other.

6.3.3 Two way connection between BDM (EDM or DM) and SOA:

Clay Greene, Principal Consultant of Technology Blue Inc., wrote an article (2010) about how the links is between SOA and BDM. He states that SOA needs decision management and decision management needs SOA. When speaking of static services, one can implement the services and reuse them relatively trouble-free because they don't change often. When speaking of services which implement business logic that is more dynamic in nature, it must be able to change more rapidly. The need for decision management with this kind of dynamic services can be by seen by looking for instance at a policy that changes often. If the policies remain in the code of the service, maintenance will be more difficult and on a technical level. This will lose some agility and flexibility even if the SOA approach is very robust. These policies should be maintained separately from the process to obtain optimal benefits. This is where decision management comes to the horizon. Decision management will enable to move the business policies out of the program code into a central repository. There should not be searched for policies in the code, which reduces the time factor for maintenance. The services which implement the policies are the decision services, which use those policies to deliver functionality. The changes can be made inside the repository by business users.

Decision management enables smarter, more consistent decisions across the whole enterprise. The

implementation across the whole enterprise has some challenges (decision management is not a technology) but is needed to reach the maximum value of decision management. Enterprise-wide adoption is crucial and means that making changes with enterprise scope should be enabled. This is where SOA helps a hand as an architectural foundation for expanding beyond boundaries of business units. Decision management technologies should be service-oriented, therefore maximizing deployment and reuse.

6.4 Maturity model:

Like with a lot of approaches, a maturity model should also be in order to measure the maturity in handling and managing decisions. The approach around decisions is a rather new concept, so there should be a new maturity model in place. This model is called the Business Decision Maturity Model (BDMM) from Knowledge Partners International, LLC (KPI).

6.4.1 Business decision maturity model:

The quality of a business is not only defined by the quality of its business processes and software infrastructure, but also by the quality of business decisions that really drives those two (Barbara von Halle and Larry Goldberg, 2009). The BDMM is a tool for connecting business objectives with business decision management practices; by seeing how close the practices come to the objectives. When improvement for these practices has been made, a bigger level of maturity has been reached.

There already has been developed a maturity model addressing the management of business rules. It is called the Rule Maturity Model (RMM) (Von Halle, 2006). As with most maturity models it contains levels. Each level presenting a state of maturity for managing business rules (BRMS). Although it has its success, Barbara von Halle and Larry Goldberg (2009) now state that it was difficult to reach the higher levels of maturity in a decent time horizon. Maybe the concept of business rules is at a too low level to justify high-level business management processes. It would have more relevance to use decisions instead of rules. Decisions talk more to the executive management and decisions stand closer to the objectives of an enterprise. Now that there is a way of modeling decisions, a new maturity model should be implemented. The impact of the decision model on business and information technology is significant. Therefore a maturity model focused on business decisions rather than business rules is justified.

The BDMM (KPI, 2009) doesn't measure the effectiveness of modeling the Decision Model. It measures the Business Decision Management process; trying to improve the quality and agility of business decisions. The model can be seen in figure 6.2. The 'business value' part of this model is kind of unique among maturity models. It indicates the business consequence of each level. Organizations use this business value to set a target BDMM level for their organization. The other parts of this model define characteristic and measurement criteria for reaching to that level. The 'business architecture' vector describes the maturity of BDM in the context of business architecture. There is also a 'business governance' (also called 'business stewardship') vector that describes the maturity of the business governance of BDM. By scoring and judging each of the business architecture and business governance characteristics, a level of the BDMM can be assessed.

Level 0 Unmanaged	Level 1 Visible	Level 2 Agile	Level 3 Aligned	Level 4 Predictive	Level 5 Autonomic
MINIMUM Business Value MAXIMUM					
Risk of loss of business control is high. Risk of business change is high. Ability to predict business impact of change is low. Cost of change is high.	Risk of loss of business control & business change is lower. Cost of change is lowered. Ability to predict business impact of change is still low. Analysis of business decisions is possible, but is manual.	Risk of loss of business control greatly reduced at the project level; business change becomes possible through automated analysis. Ability to predict business impact of change is still low.	Risk of loss of business control greatly reduced across projects. Ability to predict business impact of change is improved. Consistency between business units improved. Cost of change and testing reduced further.	Firm control of business policy established. Ability to predict short-term futures, ability to assess the impact of change on the future is possible.	Optimize business policy to changing conditions in real time and against predicted changes in business models and metrics. Management focus on evolving business objectives and policy with a firm business control; the birth of the Agile Enterprise
IMMATURE Business Architecture MATURE					
No business architecture; to speak of.	Informal Business Decision Management architecture.	Project level process and business decision standards established within broader architectural standards.	Cross project level process and business decision standards defined within broader architectural standards.	Detailed standards for process and Business Decision Architecture established and managed.	Continuous improvement of process and Business Decision architecture within the broader architectural process.
NOT PRESENT Business Stewardship ENTERPRISE					
No stewardship.	Business Analysts lead business decision discovery for local logic development.	Integration of Business Decisions with use cases and process flows with business metrics.	Stewardship of business process and Business Decisions across project boundaries.	Stewardship of business process and Business Decisions at enterprise levels.	Full integration of process and Business Decision Management into business planning.
DECISIONS SHARED ACROSS: PROJECT LEVEL ONLY PROJECTS ENTERPRISE					

Figure 6.2: The business decision maturity model from KPI source: “The Decision Model a business logic framework linking business and technology”(Barbara von Halle and Larry Goldberg, 2009).

Organizations use this maturity model to see where they are currently and set a realistic target for where they want to be in maturity of BDM. They can use the model to see which incremental steps should be taken to reach their target of maturity.

6.5 Decisions and IT:

The decision model can be an aid to further development of applications. In this sub-chapter one learns about the link with IT and how the approach of separating decision could help with better alignment between business and IT.

Brian Stucky (2009) describes that there can be some problems when we want to reach the web 2.0 approach in businesses today. Web 2.0 is a well known term for describing the internet (World Wide Web) nowadays. The internet has known an evolution from web 1.0 to web 2.0 and in the future even web 3.0. It's this kind of evolution where businesses today want to evolutes together with. Working with a Business Rules Approach offers a top down approach to describing a business, its processes and the rules within. A characteristic of this approach is that the requirements should be fully described to start developing. That is why a waterfall approach is still used often. Technicians have to wait until everything is described and defined. After getting some Process models and spreadsheets of business rules (not organized in standard order), IT have to start their development process. On several points they are forced to make their own interpretations or to equip the business people again for rewrite the requirements more clearly. It is difficult to describe the requirements and business concepts in a format that IT people easily understand. Modern BRMS and BPMS platforms are great in representing, executing and maintaining rules. But their strength lies in handing over a solution that only needs to be maintained; building it right in the first place is something different. A lot of software projects fail because of poor management of requirements. It's not only the business-centric approach that suffers from a tunnel vision; technologic-centric approaches have the same problem. Apart from the business gap and the technology gab, there is also the organization gab. Many companies still have an old hierarchical organizational structure that creates separation. Many are putting their money in a lot of technology approaches, but will never succeed unless they change their structure. Maybe over time, the dilemma of silos structure will be eliminated, because maybe future enterprises will be flatter. There should be a way to bridge the gap between business and technology.

A business decision management approach is a step in the right direction. With the right framework

in place, some better understandings will rise to the surface. Business processes with their decisions that support those processes are easier to understand and evaluate. Also the business rules that comprise decisions and the requirement for management of these rules are better understandable. The decision model helps in getting to an agile enterprise. It helps the clarity and completeness of the business model. Business specifications and technical specifications can merge to a uniform, traceable, consistent and understandable form. The Decision Model enables better communication and collaboration, what is a critical point in becoming agile (Barbara von Halle and Larry Goldberg, 2009).

David L. Haslett and Tracy Williams (2009) describe that adoption of business decision management (and the decision model) requires several things. It requires a vision for creating it and finding a way to measure the added value. Sometimes it will require a big cultural change. A cultural change is needed when business rules are getting instituted. The support of senior management is crucial to the amount of success.

Well formulated business rules are a gift for IT that will improve quality and reduce costs. The decision model and its techniques helps with getting the business rules clear on a map. Rules are communicated to IS to implement testing cases. Rules that are not fully clear are often interpreted in different ways by different IT people. The decision model removes all doubt about business expectations thanks to a clear communication.

Chapter 7 : Finding standards and specifications

Standards are an important aspect; certainly for getting recognition. Smals (ICT for Society research) mentioned that there has been a stimulating discussion with the standards BRE and BRMS uses. Two different standards from two different worlds were fighting for the format of business rule presentation. We have PRR from OMG and RIF from W3C which were leading to this conflict. The comparison can be made with the conflict among Blu-Ray and HD DVD. A direct consequence of this conflict is that different vendors of BRE and BRMS are not using the present standards. They often use a format of representing rules specific for their solution. This leads to a bad interoperation of different solutions. The first part of this chapter includes a summary of different standards, presented for different perspectives of a business. The OMG group is handling a request for proposal of a new standard called the Decision Model Notation (DMN) what could help pushing The Decision Model. This will be briefly discussed in the second part.

7.1 Specifications and standards:

Barbara von Halle and Larry Goldberg describe in their book “The Decision Model a business logic framework linking business and IT” (2009) that different aspects of the business have been gotten their own standards and specifications. Let’s take a short look of how and where standards are used in a business and how The Decision Model relates to these.

Business planning

The Business Motivation Model (BMM), originally proposed by the BRG (Business Rule Group), has the purpose of providing the tools to support creating a business plan. The Decision Model is a complementary model to the BMM. It assists the objectives and goals that a business wants to reach. The link with The Decision Model and BMM came up in discussing the enterprise architecture in chapter 6.

Business processes

Business process modeling (BPM) is a well known modeling method. The connection between a BPM and the Decision Model is an anchor point called a business decision. It’s not a link to individual business rules, because individual links are hard to manage and aren’t necessary.

Unified Modeling Language (UML) (developed by Grady Booch, James Rumbaugh and Ivar Jacobson) is also used for business processes, but more used for software systems. It's an Object Management Group (OMG) specification for a graphical notation for defining, visualizing and documenting software systems. This model is more situated among row 3 and 4 in the Zachman Framework (lower than The Decision Model). It has a broader range of models that reach a higher level in this framework, like a use case diagram. The link with The Decision Model and a use case diagram is that it can connect to a decision step in this use case diagram. This way we can separate the logic and manage it better.

Information (fact type) modeling

Information modeling is the structural representation of vocabulary. Vocabulary is defined as “a structured set of terms and other symbols together with their meanings and relationships among them, for use by a business community” (Object Management Group, 2006). The terms in this definition are fact types, which we can see in the decision model. We make a distinction between fact models, data models and business object models. These models are mostly used for database design. If decision models will be automated (for instance in a BRMS); an information model will be necessary for technical oriented purposes. The glossary of the fact types of a decision model should create some kind of interface with other models.

Information systems development approaches

Barbara von Halle and Larry Goldberg (2009) ensure that with the decision model there is the possibility to develop business logic parallel to the development of process models, use cases, business scenarios and other models. Decision models can simplify business processes, deliver them quicker and can define decision services for a SOA catalog. Business analyst can get more involvement because of The Decision Model. It can shift responsibility from technical to business resources, accelerate testing practices and reduce software errors. The testing can happen in the decision model first prior to the testing of the code. This can save a lot of time. With the separation the increasing accuracy and traceability also play a beneficial role. One can test the Rule Families before the actual implementation. This serves for early error detection in test cases, which then again can implicate lower volume testing.

When we speak of agile methodologies, the decision model can play an important role. With agile development there is a delivery of small increments in a short time horizon. The application is

divided in several smaller increments (for instance different functions). The decision model can help because first of all it creates the separation of business logic as a subset of a software component. Second, the Decision Model itself can be handled in several pieces.

Link with system transformation methods

The decision model helps the mining of business logic inside systems. A system can be needed to be transformed by a reengineering process, or maybe just to upgrade to a better technology platform. The decision model can give a view from a low level code to a high level architecture; that's why it helps when system transformation is needed.

7.2 Decision Model Notation (DMN):

Let's start finally with explaining what the object management group (OMG) is. OMG is a standard group; what means that they don't invent models, notations or languages, but they try to find commonality among existing ones and they try making them common with some simple extensions. The object management group is also called a consortium what has his focus on developing standards.

Barbara von Halle and Larry Goldberg (2010) explain that the Decision Model Notation (DMN) has come to life after some members of OMG recognized the value after a presentation has been given by Larry Goldberg about the decision model. DMN wants to provide a standard framework for decision models, a standard notation for these models, a common notation for decisions in BPMN and interchangeability for decision models and tables among the modeling tools. There is a difference about what DMN means by decision model and what is called The Decision Model. The one DMN uses can refer to various ways of representing logic.

During 2010, the DMN committee was busy with investigating several things like how to represent the logic, looking for existing standards and the use of decision tables with combination of BPMS and BRMS and even SBVR. There was been a search from OMG to the relation of SBVR, BMM, BPMN and other OMG models with The Decision Model. A request of proposal is at the moment of this writing evaluated by OMG to create a standard for a Decision Model Notation. The ultimate intent for the DMN standard is that it will include all kinds of other representations for decision logic (e.g. decision tables, decision trees, etc.) together with that of The Decision Model. Hopefully the standard will

provide a framework where The Decision Model fits in. In addition it should provide guidance to how these representations are connected and will evolve with the other OMG standards (e.g. BPMN).

The last word for this is for a DMN RFP author. Paul Vincent stated in his blog, “Decision modeling could become the dominant analysis aspect in future years: consider a business process like “loan approval” and you don’t need a degree in computer science to work out that the business process is there just to support the business decision on whether to approve a loan! So in summary: DMN is looking like it could be very useful.”

Chapter 8 : Types of decisions

There are different types of decisions, just as there are different kinds of rules, although the difference is distanced by different aspects. A decision can be for instance context driven. The context could include data and time, so where one is in a business process. The Decision Model is made for manageable decisions. What the different kind of aspects are and what makes a decision worth of managing, will be covered in this chapter. The chapter starts with explaining the difference between rules and decisions. The impact of decisions plays an important role when it comes to managing. Further several considerations will be covered to know the optimal characteristics of a manageable decision.

8.1 Terminology: rules – decisions:

There has been some discussion about the use of the term 'rules'. James Taylor (2010) explains that nowadays there are a lot of rules inside a business. He speaks of business rules in user interfaces, in business processes, in data quality, in event correlation and business rules in operational decision-making. The usage of the term rules brings sometimes complexity and misunderstanding. With the term 'decision' a lot of the complexity is vanished. It is a term used for decision making. Rules about data quality should be used when data is created; rules about processes routing and escalation should be used during a process design and rules about event correlation should be used with your Complex Event Processing (CEP) engine. CEP is about processing events in all kinds of layers of the business, but will not be described further into detail. The rules about business decision-making should not be mixed with rules in previous sentences. The different "roles" that "decisions" can provide are: Classification, Evaluation, Selection, Approval, Assessment, Assignment, Allocation, Diagnosis and Prediction. (Ron Ross). The rules about business decision-making only change when your business is also changing (e.g. different policy, new regulation about eligibility, etc.). These are the rules that should be inside the decision-making components or decision services.

8.2 Decision impact and context:

Business decisions all have a different value (von Halle, 2009). They vary in complexity, volume-based economic impact and operative context as explained by the authors of the book "The Decision Model a business logic framework linking business and IT" (Barbara von Halle and Larry Goldberg, 2009).

The number of rules families will drastically increase with the complexity of a decision. The economic impact is different for a strategic decision than for a simple process decision. The frequency on the other hand, will probably be much lower for a strategic decision. The context varies because the environment may range from simple to complex. There are decision that are made during a crisis, in contrast to decision that are made daily in a business process. The impact of a decision will be a function of all these characteristics described above.

There are decisions that have to be made when a certain event happens (like a crisis). These are sometimes founded on a decision maker's intuition or expertise. They are not fully fact based and therefore difficult to be modeled by a decision model.

David J. Snowden and Mary E. Boone (2007) define four operative contexts for business decision based on what the input is. There is the simple context, the complicated context, the complex context and the chaotic context. These contexts are described differently based on what we know about the input. It ranges with the simple context where we have a clear understanding of the facts, to the chaotic context where the input consists of what Barbara von Halle (2009) calls the 'unknowables'. Further details about these contexts are not included in this thesis. The table about the operative contexts can be checked in appendix E. What is useful to remember is that the complexity of developing a decision model varies along with these contexts. It is common sense that it will be much easier to develop the model in a simple context, than in a chaotic context. The decision model can help by getting from a very complex context to a simple context. It creates more visibility to the construct of decisions. This is often done in an iterative way.

The number of times a decision is made can be stated as the volume of the decision. The discount for a customer EU-Rent may be a decision that is made many times a day, while the decision to add a new branch in a certain location is made infrequently. Here the distinction between operational decision and strategic-level decision is clear. The economic value of an operational decision by itself can be very low, but because of a high frequency this may not be ignored. A lot of small decisions add up to a volume that should not be neglected. The decision model provides the support in managing such operational business decisions.

The complexity of business logic is the last characteristic. The quantity of rules families, interrelationships, conditions and conclusions increases together with increasing complexity. For the simplest complexity the decision model is a good way of documenting, sharing, standardizing and

changing the logic. When there is more complexity to the logic, the decision model may be of perfect assistance for simplifying it. 8.3 Considerations:

The Decision Model is most useful for business rules or statements that can be written as conditions that lead to conclusions that have a metric to evaluate the business impact (von Halle, 2009).

Not all business rules can be implemented in the decision model. Even if we can implement them, it sometimes is not that useful to do.

Barbara von Halle and Larry Goldberg stated that : “Business logic that serves best for a decision model is of a purely business nature, involves evaluation of facts leading to a conclusion, represents one business decision, is not easily represented by another means, and is subject to change.”

They explained these considerations in their statement in further detail:

The logic is off a purely business nature

Data validation rules are not the target rules for implementing in a decision model. They are just rules that validate data and so the quality of data. They don't have a business meaning behind it that leads to conclusions. Data transformation rules are also not in the category of best practice for developing a decision model. They have logic behind it that changes formats of data to another format, but they don't result in a business oriented conclusion. The validation or transformation of data most likely happens before the task that relies on that data, and where the decision model does come handy. User or system interface rules are another example of rules that are not that suitable for the model.

The logic can be put in conditions that lead to conclusions

An important consideration is that the logic can be stated in a kind of “if-then” form. Although data validation rules can be put in conditions that lead to conclusions, there could be better ways of representing them. They can conclude in either ‘valid field’ or ‘invalid field’, but the rules are often minimal related to each other.

The logic is not known otherwise

If the logic is buried in system code, manuals or inside peoples' heads; it's often useful to gather all of this logic. Once the logic is obtained, better management and smarter decisions can be enabled.

The logic needs to change often

If the business logic hardly ever changes, the traditional way of documenting may suffice. Data validation rules are usually very static. For example, a validation rule could be that in the renter's address for EU-Rent a postcode should involve 4 digits. This can be a static given, that hardly ever changes.

8.4 Information Quality (IQ):

In the considerations above was explained that the decision for implementing in a decision model should be of pure business context. Barbara von Halle et al. explain in an article for BPMInstitute (2010) that the Decision Model was originally not intended to deal with data quality, data transformations, workflow routing or driving interface sequences. There was clearly stated that a decision has to be in a purely business context. This means that the previous aspects are not intended to be used by the decision model. In reality, there has been use of the Decision Model for such kind of aspect that has been summed up. Information quality (IQ) is an aspect where the Decision Model has been used for. IQ is becoming more and more important within organizations and it has some challenges. The Decision Model turns out to be a unique solution for part of the IQ challenge. The table with the entire content of how the Decision Model interacts with different kinds of data quality expectation is given in appendix G

According to Barbara von Halle (2010) the Decision Model ensures that the IQ logic is visible and tangible in a better way than with textual statements. The variations in the IQ logic based on business needs will be easier to handle when represented as a view. Possible errors will be better traceable with the use of the decision model. The general benefits of using a technology-independent form are also accurate with IQ.

An extensive study about this is not represented in this thesis. Although a short example within the case of EU-rent is given. If rules about data validation in the development of the rental contract are not tested to come to a valid rental, the process shouldn't go further. A Rule Engine can create a message, indicating what went wrong and what should needed to be done to go forward (D. Pedersen, 2010). This process is much more complicated when there is no decision model in place; because a row from the rule family can suggest such a question according to D. Pedersen.

8.5 The Decision Model and the human brain:

To end this chapter, let's look at how the Decision Model and the human brain cooperate in the sense of understanding logic. The representation of logic done by the use of The Decision is a simple rational presentation. Barbara von Halle explains in an article for *Modern Analyst* (2011) that this means that our rational brain is closed linked with the decisions within such a model.

She explains that if we assume a top-down approach for creating a decision model, it is known that the iterative creating of skeletal decision models before knowing the actual logic is a useful way in brainstorming of the brain. By focusing on conclusions first and making the necessary considerations proves to be useful in stimulating business innovation. The models present only one part of our human decision-making brain. There are some ways one can benefit from the emotional brain (i.e. human wisdom and how we learn). For wisdom one could include experienced business experts in decision model validation. For learning one could incorporate Business Intelligence (BI) and analytical tools, so that results from execution of decision out of the past can be improved in the future. With a clear focus on business objectives and a measurement of outcomes, one should make smarter decisions in each version of a decision model. To give a real world example of when one uses its rational and emotional brain is probably poker a good example. A part of the game is based on probability and statistics which is triggering the rational, logical, computational brain. Another part is the bluffing and estimation of other players' bluffing which is triggering the emotional brain. A famous poker player (J.Lehrer, 2010) stated: "The only way to win is to make better decisions than everyone else at the table." This is also valid for businesses today with decision management as an emerging survival and competitive competency. There is also another quote from J.Lehrer (2010) that predicts a promising future for decision management. He quoted: "The most astonishing thing about the human brain: it can always improve itself. Tomorrow we can make better decisions." This can also be interpreted into a business perspective by saying that we always can improve the decision models so that the business becomes increasingly smarter, one decision model at a time.

Chapter 9 : Constraints and challenges

This chapter will enlighten the challenges The Decision Model has to cross. Most of these challenges come further from studies which are described in previous chapters. To ensure that rigor form of structure that The Decision Model enables, it should be formatted according to several principles and normal forms (described in chapter 2). The sorts of decisions for which The Decision Model can make an asset worth managing are discussed in chapter 8. The Decision Model as a model in comparison with different other models, can be evaluated according to the audience it is needed for. The Decision Model has a broad audience and is easy to understand by as well IT as Business people, but other representations of logic can still have their benefits (like described in chapter 5). Other perspectives of possible challenges and constraints will be briefly noted in this chapter together with how well the implementation with BPM works.

9.1 Imperfections:

Let's look at obtaining the information for formulating the decisions and what challenges it brings to the table. There is the need of knowledge to come to a successful model. One must have an idea of how thing being modeled actually work. It is not self-evident that this is always the case. Gathering the information to establish a correct model can be a very complex process with many actors playing their role. Business rules can have been stored in documents, rule guides, software, people's minds, etc. It is very rare that a person knows all the facets of the corporation, especially in medium to large corporations. Often the gathering of the information about decisions is complex and happens across different departments. Not only can it be complex to gather all the necessary logic, it can be that there are inconsistencies among different people. One has to be sure that everybody thinks the same thing. It seems obvious, but every decision maker has to agree with the conditions leading to certain conclusions.

It's important to know the value of the decision model, before implementing. Change in a business culture is often a factor that involves resistance from the people. Employees are afraid of change in their company, especially when they think their jobs are influenced by it. By knowing what the value is, this resistance can be eliminated or reduced to a minimum.

Businesses are dynamic systems in a way that change happen often. Even with the best models of decision logic, a major change can cause everything to collapse. It is very difficult to predict the

future and a major crisis is always possible. Although it seems that modeling is in a negative perspective in a crisis a waste of time and effort; it helps with fast adaption to the environment and gives a certain control over the logic of a business (von Halle, 2009). Let's end this challenge with saying that logic plays an important role in times of chaos.

An important aspect of decision making is knowing what data is available and for which dynamic data can be important. Barbara von Halle (2010) admits that it would make little sense to make a decision model for important business decisions based on data that already exists in data models and databases. This could be a costly mistake.

The quality of a business decision is only as good as the quality of the business logic inside the Decision Model itself and the quality of the dynamic data created by the Decision Model. These dynamic data values need proper names, definitions and domains. If they are not managed properly, the decision model falls apart and this could have as consequence that conclusions may be risky and business may suffer.

Some people were questioning if the decision model would have its benefits when the rules are enormous. For many this was ought to be a big challenge but Larry Goldberg (2010) explains that this is exactly what the decision model can handle. When there are thousands of rules in a business, it seems very complicated to manage. The decision model ensures that those thousands of rules can be brought back to maybe a few hundred decisions. The rules maybe not known, but these decisions are mostly known to the business. Managing a few hundreds of decisions is easier than managing an amorphous mass of several thousand business rules. Each decision has its own logic and this way of grouping the rules seems a much more efficient way of managing.

9.2 BPM and the Decision Model:

When we speak about a good collaboration between BPM and the rules or logic behind processes like what is constructed using The Decision model, one often comes to some challenges. The people who define the process steps are often different from the people who are defining the decision logic. This synchronizing of people is not simple. To help collaboration in these cases, value should be clear for both parties. By studying the separation of rules and processes, it seems that there was a conflict several years ago between the supporters of process model vendors and those who were focusing on rule separation. The supporters of BPM were convinced that the modern BPMS were as agile as one

could be and that consulting a BRMS within is unnecessary. They even had their doubt about the fact that rules more often change than processes. Looking at how The Decision Model fits in the BPM artifact, one can state that neither BPM as The Decision Model handle all of the business logic in most cases. They often define distinct and complementary aspects of business logic.

BPMN is the standard language for Business Process Modeling. It describes an end-to-end process as sequential tasks that need to be taken. It shapes the relation in time to other tasks and events in the process. We can say that it has routing logic in the step to step sequence of a process. BPM helps to get a better overview of processes that often are cross-functional. This way optimization can be enabled, by a better management of cross-functional processes. Although, BPM doesn't cover all aspects of what happens in a process, it is very useful. The other details are part of the implementation design. The Decision Model can be linked to a BPM by the use of a 'decision task' which has been known from previous chapters. In the new BPMN version will be a new task called a 'business rule task', because the term decision creates a conflict. A decision gateway is a diamond shape that defines a branch or split in the flow of activities. This is why the term 'decision' reaches some conflict, although a gateway does not really 'decide' anything. It's really the step before such a diamond that includes a decision to be made. According to the outcome, the flow chooses a direction from the diamond. In that way is the gateway pure routing logic. The Decision Model plays its role in managing decision logic external to process logic. This makes it centrally for all processes and greatly improves governance and agility.

The separation of decisions and processes could have an additional challenge. Within The Decision Model can be no actions. There can be no separation between actions and decisions, for example when something needs to be registered during the running of the logic in a decision model. The challenges obtained from chapter 3 where the EU-Rent case has been explored can be added to this subchapter. It is rather difficult to start with SBVR statements and try to link them to the processes. This was a major reason for some challenges obtained in that chapter. There has been talked about the difficulty of collecting all the rules for decisions, but the information for the decision making is also not always immediately available. The Decision Model assumes that all values are available to fill in the rule families to come to conclusions. Another point learned from that was that not every rule can be implemented in the model. A redesign of the process can help to implement more decisions in decision aware tasks that need to link to The Decision Model. Because the model is an intellectual

template and is not some kind of system, it's the method of separating decisions that can be surely revolutionary.

9.3 Software in early shoes:

The software designed specifically for The Decision Model stands early in his shoes (Barbara von Halle and Larry Goldberg, 2011). Looking at the Business Decision Maturity Model (BDMM), the authors stated that little software tools help to get higher than level 2 of the BDMM, most of them only to level 1. The software vendors that are following are partners of Knowledge Partners International (KPI). These are the vendors that can bring managing of decisions together the Decision Model to a higher maturity level.

In 2010, many organizations have created decision models in office automation software (e.g. MS/Visio). Several software vendors are providing tools for business management of The Decision Model. The two most known vendor products are NewWisdom Software's Rule Guide and eDev Technology's inteGREAT. They both have some different aspects to offer and provide different capabilities for business management of decision models.

RuleGuide provides a tool for capturing business rules and the management of it. Lee Lambert, founder of Lambert Consultants and former of New Wisdom Software adapted features in RuleGuide to support The Decision Model. The results of this software were significant compared to the default ones of for instance MS/Visio. The support for The Decision Model is visible in for instance when an analyst enters decisions and rule families, RuleGuide automatically links dependent and support rule families within the decision structure. It shares the decision model rule family structural principles. It uses also a visual decision rule family tree which has a lot of similarities with the graphical decision models.

InterGREAT is a requirements tool and provides a glossary with links to object and data models, Rule family population, Graphical decision model structures and all kind of connection from decision models to process models or business motivations.

Apart from the software for business management of decision models, there has been some effort in software for technical management of decision models. The execution environment for decision management solutions is supported by a wide range of technology. The traditional Business Rules Management Systems (BRMS) are an example of this (e.g. IBM/ILOG, Drools, Open Rules, etc).

Knowledge Partners Inc (KPI), the company that owns The Decision Model has actively worked with OpenRules and ILOG to make processing capability for The Decision Model.

There is still no commercially available software designed specific for the Decision Model and Decision Management throughout their full life cycle. The challenge is to develop software what can reach to the higher BDMM levels. Let's end with a positive note that it is still early and that even without software support, the Decision Model proves his value. There is little doubt that further more sophisticated software with decision model adoption will arise. The software support is catching up and it will only be a matter of time. Latest news from KPI brought the confirmation of software that can elevate decision managing up to level 4 in maturity. This is a big help, but even less sophisticated software, that helps business audiences, accelerates projects about business rules and process management by reducing time and money.

Chapter 10 : Benefits

This short chapter will sum up several different benefits of separating the decisions from the processes. The economically advantages are substantial. Practice has been proven that The Decision Model reduces costs and time with capturing, managing and testing the business rules. One can state that a lower cost will be created with the adoption of The Decision Model. Apart from the benefits rule separation already has with the traditional means, the Decision Model brings the level even higher.

In chapter 8 was described how decisions could have different contexts. When the context is complicated, the formation of the model is not that easy. The big benefit is that one can decrease the complexity during the iterative development of a Decision Model. This is often thanks to logic that becomes clearer or easier detection for the need of additional information in certain areas. Once obtained a model, one is prepared for different kind of dynamics in the business environment. Threatening times and times of crisis are easier to struggle through if we have a clear model of the logic behind crucial processes.

The Decision Model's structure is easy to understand. It promotes a higher level of business understanding of the rules and simplifies business processes (see chapter2). The business logic and business rules get visualization. This enables finding errors and missing rules, overlapping rules, conditions and condition values. When leaders in a business can see their business logic in a clear structure like the decision model, better business decision-making is ensured.

Decision models give the decision making back to the business people, because IT people are not needed for subjective implementation of rules. In the old days interpretation of rules was often applied by IT people, because there was a lack of communication. Now, the Business people can with only a limited support from IT quickly setup rule families and test for expected outcomes. Barbara von Halle and Larry Goldberg (2009) say that the efficiency of a business rule approach is improved. The decision model is one tool in a whole toolset, but it's the tool that brings clarity and predictability of outcomes from a business rule initiative. The decision model is a great help in discovering business logic that is incomplete, inconsistent, duplicated and sometimes even just wrong.

Looking at managing decisions (and rules); a big step forward has been taken. The rigor structure of The Decision Model ensures benefits in managing. There is the option for a structured view on a higher level without the populated rule families. When there is a need for more details, it's easy to just dig deeper into the structure to the rule families. Rule families are very easy to modify. This ability to easily modify business logic is an important capability for making an enterprise more competitive. When automation has been applied, Decision model automation reduces the time from policy that change to the automation.

The Decision Model plays a significant role in improving system testing. The model helps testing the system for the conformity to the functional requirements (Barbara von Halle, Larry Goldberg, 2009). They explain that the decision model can shorten the development and testing of systems. Rules families can get input values, which than can be tested for conformance to the functionalities. Logical test cases can be written from a decision model. There is the opportunity to write a test case for each rule family role in simple manner. The testing results are improved, because there is less time and costs needed to test the outcomes. The time aspect is that this can happen before an actual implementation of the model. In this way, faults can be detected early on.

Barbara von Halle and Larry Goldberg (2009) also explain that the development time has been shortened. Different types of rules can be implemented differently. Because this Model is independent of technology there is a separation of business rules from implementation. If implementation aspects changes don't mean that the underlying rule should be changed. When rules often change, it's better to put them into a BRMS. Business rules that never or seldom change can be written in a programming language like JAVA. By this we can state that the benefits reach to an IS function.

The Decision Model has the capability to connect to several other models, what makes cooperation possible. The connection points with BPM are mainly discussed in this thesis, although there are connection points with several others like use-cases, business motivation models, etc. The connection of business rules groups to their natural anchor points in processes enables a separation between declarative and sequential processes. The scope of changes and their impact can be easily understood. There are different kinds of changes, sometimes crossing more models. Barbara von Halle states that there is a common understanding across the enterprise thanks to The Decision Model. The Decision Model creates an atomic form of the business rules. Natural languages can be easily created from this model. There is the opportunity to express rules in a way that everybody can

understand. By making a glossary, an enterprise wide terminology can be created. The tangible separation of process and decisions takes process improvement and reengineering to a new level. (D.Pedersen, 2010)

Another given what brings the Decision Model in the spotlight, is the importance within a service oriented architecture (described in chapter 6). The Decision Model supports and facilitates SOA. The grouping of logic by decisions supports decision services. These services are an important part of that approach.

The Decision Model supports an agile way of development. If we look at how agile software approaches handle, this model has its value. In contrast with a waterfall approach, agile development does not require a detailed requirement list of functions. The Decision Model can help within the scope of the functionality which is based often on only one model. The iterative approach with the modeling is aligned with the iterative agile development approaches. Application maintenance and IT integration costs can also be reduces substantially with a separation of decision from processes.

Chapter 11 : Conclusion

The introduction of Decision Model has opened up doors and created opportunities for significant improvement in management and information technology practices. It provides the ability to relate management objectives to business decisions (B. von Halle and L. Goldberg 2009). These objectives should be measured on performance over time to fully utilize the benefits. This has created a new discipline of Business Decision management (BDM). Increased business agility can be easily the benefit obtained from this.

For IT, the decision model fills the gap that there was. There was no separate model of business logic, although other aspects of a business application system have had their separate models throughout the years. Prior to The Decision Model business and system analysts have been burying the business logic in various models (e.g. process models, use case models, etc.) or have been listing business rules in catalogs of rule statements without optimal grouping methods. The result is that there could be many versions according to a lack of a general method. This leaves the developers to make their own determination of the correct logic. This results in unnecessary costs, higher error chances, failure in projects and applications systems that can't reach their optimum and lack agility. Separating decisions out of the processes creates the grouping of the rules on the basis of decisions. This way of grouping enables connection points with processes and with The Decision Model the structure of the supporting rules for a decision can be visualized in a way that is understandable for as well IT as business people. This brings an alignment between IT and business.

The separation of declarative and procedural nature is crucial for agile and reusable processes and decisions. This is a two-way reusability. The process stays often the same, but the decisions are inconsistent. For example different countries have different regulations, but the processes can be the same. By externalizing the decisions, a lot of processes are reusable. The other way is also possible. Decisions can be reusable across all kinds of different processes.

Out of the previous chapters and the listed benefits of separating decisions from processes, the model of such a separation helps in different perspectives of a business. The Decision Model has the capability to improve a lot of IT and business practices. Out of previous chapters one can say that it can improve business planning, business process management, decision support, business architecture, enterprise architecture, business analysis, business requirements, system testing,

service-oriented-architecture and development methodologies. Although it assists in these different aspects of business, it still has some challenges to conquer (described in chapter 9). The future may enlighten that it is maybe not the optimal solution for creating agility, but it definitely a big step in the right direction.

References

- Goldberg, L., & von Halle, B. (2009). *The Decision Model: A Business Logic Framework Linking Business and Technology*. Auerbach Publications, Taylor & Francis, LLC
- Goldberg, L., & von Halle, B. (2010, March 1). The Decision Model: Step by Step Decision Modeling for Data Professionals (Turning Poor Calls into Good Business). *The data administration newsletter*. Retrieved from <http://www.tdan.com>
- Goldberg, L., & von Halle, B. (2009, December 1). The Decision Model: "Bottom up". *The data administration newsletter*. Retrieved from <http://www.tdan.com>
- Goldberg, L., & von Halle, B. (2010, June 1). The Decision Model: Dynamic Data, a New Data Asset? Why manage it, Where and How to Find It. *The data administration newsletter*. Retrieved from <http://www.tdan.com>
- Nayab, N.(2011, February 9). Advantages of Decision Tree Analysis. *Bright hub*. Retrieved from <http://www.brighthub.com/office/project-management/articles>
- Nayab, N.(2011, February 9). Disadvantages to using Decision Trees. *Bright hub*. Retrieved from <http://www.brighthub.com/office/project-management/articles>
- Goldberg, L., & von Halle, B. (2011, April 12). Software Tools for The Decision Model –part1. *The BPM institute*. Retrieved from <http://www.bpminstitute.org>
- Goldberg, L., & von Halle, B. (2011, April 21). Software Tools for The Decision Model –part2. *The BPM institute*. Retrieved from <http://www.bpminstitute.org>
- Business Rules Management System (n.d.) from Smals onderzoek website, <http://inventarisoss.smals.be/nl/351-dsy/318-dsy/329-dsy.html>
- Coenen, A.(2011, January 18). Bedrijfsregels en wendbaarheid. [Web log post]. Retrieved from <http://alcedocoenen.wordpress.com/2011/01/18>
- Ross, R. G. (2010). Introducing Question Charts (Q-Charts™) for Analyzing Operational Business Decisions: A New Technique for Getting at Business Rules. *Business Rules Journal*, 11 (12). Retrieved from <http://www.BRCommunity.com>

- Ross, R. G. (2010) Decision Analysis Using Decision Tables and Business Rules: Introducing Question Charting (Q-Charting™) Version 1.2. Business Rules Solutions. Retrieved from <http://www.brsolutions.com>
- Debevoise, T.(2009, March 20). BPMN versus business rules. [Web log post]. Retrieved from <http://www.tomdebevoise.com/blog/?currentPage=4>
- Taylor, J. (2009). Decision Management contrasted. *Business Rules Journal*, 10 (9). Retrieved from <http://www.BRCommunity.com>
- Green, C. (2010, February 2). Transforming the Enterprise: SOA +DM. *The BPM institute*. Retrieved from <http://www.bpminstitute.org>
- Snowden, D. J., & Boone, M. E. (2007). A Leader's Framework for Decision Making. *Harvard Business Review*. Retrieved from <http://www.mpiweb.org>
- Debevoise, T. (2007, February 5). Converging BPM and Business Rules Maturity Models. *The BPM institute*. Retrieved from <http://www.bpminstitute.org>
- Stucky, B. (2009, February 27). Collaboration and Business Decision Management. *The BPM institute*. Retrieved from <http://www.bpminstitute.org>
- Taylor, J. (2010, October 7). What, exactly, do you mean by business rules? [Web log post]. Retrieved from http://www.ebizq.net/blogs/decision_management/2010/10/
- Pedersen, D. (2011, March 28). Better, Faster, Cheaper part 2: The Decision Model Meets Data Quality Head on. *Modern Analyst*. Retrieved from <http://www.modernanalyst.com>
- von Halle, B., Pedersen, D., Goldberg, L., Grohs, M. (2010, September 22) The Decision Model Column: MIT Industry Symposium on Information Quality, The Decision Model, and an IQ Framework. *The BPM institute*. Retrieved from <http://www.bpminstitute.org>
- Pedersen, D. (2009, January 22). Surviving and growing during an economic downturn. *The BPM institute*. Retrieved from <http://www.bpminstitute.org>
- Schacher, M. (2008, June 23). Mini EU-Rent Business Model. *Know Gravity*. Retrieved from <http://www.knowgravity.com/>

Schacher, M. (2006, May 21). Moving from Zachman Row 2 to Zachman Row 3 Business Rules from an SBVR and an xUML Perspective. *Know Gravity Inc.* Retrieved from www.omg.org/news/whitepapers

Object Management Group (2008, January 2). Semantics of Business Vocabulary and Business Rules (SBVR), v1.0, 267-342. Retrieved from http://knmgt2.univera.de/documents/SBVR_08-01-02.pdf

Appendices

Appendix A: Figures and Tables in thesis:

Figure 2.1: Rule Family ‘Discount’	19
Figure 2.2: : The Decision Model 1 for validating a booking request.....	21
Figure 2.3: The link between the decision model diagram and its rule families	22
Figure 2.4: Declarative decisions in a procedural model	23
Figure 2.5: Separation of business logic out of the BPM with a decision aware task	24
Figure 2.6: Circular path in a decision model (fault against principle 13)	30
Figure 3.1: The organizational structure of EU-Rent	35
Figure 3.2: Rental process of EU-Rent on a tactical level	39
Figure 3.3: Possible customer servicing process of EU-Rent	39
Figure 3.4: Attempt for Decision Model Diagram ‘validate booking request’	40
Figure 3.5: The Decision Model ‘validate booking’	43
Figure 3.6: The Decision Model diagram ‘validate request for pick-up’	46
Figure 3.7: The Decision Model ‘validate request for pick-up’ with view: ‘America’	49
Figure 5.1: A decision tree for the volume discount at EU-Rent	57
Figure 5.2: A Q-CEO for the volume discount at EU-Rent.....	60
Figure 6.1: Modification of row 2 column 6 of the Zachman Framework (Adapted from John Zachman’s “The Framework for Enterprise Architecture (The Zachman Framework)” included in the book “The Decision Model a business logic framework linking business and technology” (Barbara von Halle and Larry Goldberg, 2009).....	62
Figure 6.2: The business decision maturity model from KPI source: “The Decision Model a business logic framework linking business and technology”(Barbara von Halle and Larry Goldberg, 2009).	70
Table 2.1: Rule family booking request (fault against principle 11).....	29
Table 2.2: Partial Car’s service schedule Rule Family EU-Rent (fault against normal form 2).....	32
Table 3.1: Rule Family ‘booking’	43
Table 3.2: Rule Family ‘rental dates’	44
Table 3.3: Rule Family ‘start date/time’	44
Table 3.4: Rule Family ‘rental overlap’	44
Table 3.5: Rule Family ‘rental duration’	44
Table 3.6: Rule Family ‘drivers’	45
Table 3.7: Rule Family ‘driver state’	45
Table 4.1: Initial rule family ‘drivers license’	53
Table 4.2: Added condition to rule family ‘drivers license’	53
Table 5.1: A decision table for the volume discount at EU-Rent	56
Table 5.2: Example of putting a statement in a Rule Family.....	59

Appendix B: Additional appendices references:

During the appendices different sources will be revealed. There are some references in addition to the ones previously stated..

Koehler, J. (2010).The Role of BPMN in a Modeling Methodology for Dynamic Process Solutions. *IBM Research*. Retrieved from <http://user.enterpriselab.ch/>

Frankel, D. S., Harmon, P., Mukerji, J., Odell, J., Owen, M., Rivitt, P., Rosen, M., Soley, M. R. (2003)
The Zachman Framework and the OMG's Model Driven Architecture. *Business process trends*.
Retrieved from <http://www.bptrends.com>

Appendix C: EU-Rent case:

Concepts:

EU-Rent is a fictitious car rental company. The company has branches all over Europe and in some sources also in the USA and Canada.

To make the case more clear the roles will be explained. These are obtained from “The Role of BPMN in a Modeling Methodology for Dynamic Process Solutions” (Jana Koehler, 2010 IBM Research).

- The branch manager, who oversees a particular branch office of the rental car company, is responsible to buy, sell, or transfer cars, keep the contact to the police in case of missed cars, and bar customers who misbehaved when renting a car.
- The car park assistant oversees all technical operations on the car fleet, including activities to service cars and receive returned cars.
- The sales clerk interacts with the customer. For example, she will hand over the keys and the contract when the customer arrives to pick up the car, she might grant a contract extension upon a call of a customer, which may well interrupt the registration of a new customer who waits at the counter.

Most of the rules and information used to indicate examples during this thesis come from the EU-rent case provided by Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 (OMG, 2008).

The rental company has other processes than the customer service implemented in the thesis in chapter 3. There are activities that have nothing to do with service, like for instance assigning cars, planning acquisitions or selling of cars, planning car transfers, planning maintenance services, etc.

In the thesis came the fact type ‘rental duration’ often to the surface. A rental duration is measured in Rental Time Units (RTU). It’s measured as a whole number. An example should make this concept clear. Let’s assume that there was a rental with an end date/time (actual return date/time) that was 11 days and 7 hours after its start date/time (actual pick-up date/time). The calculation would be 1 x 1-week RTU plus 5 x 1-day RTU. The calculation contains 6 rental time units. The base unit price is then the sum of the prices for the 6 RTU’s. It should be clear that although not a complete day has

been passed at the end, the final part-day is charged as a rental day. Hourly tariffs are only used when there is a need to calculate a late charge.

A late return charge is calculated at an hourly rate when the rental is less than 5 hours late. Between 5-24 hours late, the rate is calculated via a daily rate. A location penalty contains 3 parts. There is a fixed amount that will be charged to the customer. There can be a cost when the car is not dropped off at an EU-Rent branch (but for example at an airport). This will include an extra cost for retrieving the vehicle. There is also the cost of moving (at another branch). This will be calculated with the measured distance to the branch of the contracted rental and the drop-off branch. This amount will be according to the car group price of the rented car. The lowest rental price is based upon the base rental price which is calculated according to the RTU's discussed above.

The price can be calculated according to previous penalties and additional charges. The rental charge contains the lowest rental price + additional charges. The additional charges are for optional extras, for additional drivers and for the penalties discussed above. The price is based on duration and on the car model (from car group) that is assigned (or requested, because free upgrades can be possible when capacity is low). Discounts and promotions vary from location to location, because managers are relatively free with these to be able to compete with local competitors. In this thesis is the example of a volume discount used to illustrate several things.

The terminology used in SBVR (OMG, 2008) is sometimes confusing, because it seems like redundancies at first. There are different terms for more or less the same thing. At first side this seems complicated. For example a rental has initially a sending and receiving branch. At a later stadium these terms are replaced with a pick-up branch for the former and a return branch for the latter. Later a return branch can be compared with a drop-off branch. The reason for this is the different states an object can have. When the rental states changes from reserved to assigned, the terminology changes. The reason for the term drop-off branch is that it is the actual drop off place (when it is a branch of EU-Rent).

There is some difference in how a sentence is written. There are obligations, permissions, prohibitions, etc. Obligations may be considered as triggers of business activities in some cases, whereas prohibitions, as well as permissions, may be considered as (necessary) preconditions for business activities. What follows are the lists of structural and procedural business rules about EU-Rent given by Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 (OMG,2008).

List of Structural rules:

- It is necessary that each rental has exactly one requested car group.
- It is necessary that each rental includes exactly one rental period.
- It is necessary that each rental has exactly one return branch.
- It is necessary that the scheduled pick-up date/time of each advance rental is after the booking date/time of the rental booking that establishes the advance rental.
- It is necessary that the rental charge of each rental is calculated in the business currency of the rental.
- It is necessary that each cash rental honors the lowest rental price of the cash rental.

3 parts:

- It is necessary that a cash rental price for a cash rental that is calculated because of EU-Rent price changes and that is less than the lowest rental price honored by the rental replaces the lowest rental price honored by the rental.
 - It is necessary that a cash rental price for a cash rental that is calculated because of changes to the requested car group or rental duration of a rental replaces the lowest rental price honored by the rental.
 - It is necessary that the lowest rental price honored by a rental is not replaced after the actual return date/ time of the rental.
-
- It is necessary that if a rental is open and the rental is not an international inward rental then the rented car of the rental is owned by the local area of the pick-up branch of the rental.
 - It is necessary that the booking date/time of a points rental is at least 5 days before the scheduled start date/time of the rental.
 - It is necessary that the renter of each points rental is a club member.
 - It is necessary that each rental has exactly one requested car group.
 - It is impossible that the pick-up branch of a one-way rental is the return branch of that rental.
 - It is possible that a rental is an open rental only if the rental car of the rental has been picked up.

- It is necessary that the cash rental price of each rental that is the responsibility of a corporate renter is based on the cash rental rates of the corporate rental agreement that is available to the corporate renter.
- It is necessary that the rental duration of each rental that is the responsibility of a corporate renter is not greater than the maximum rental duration of the corporate rental agreement that is available to the corporate renter.

List of operative rules:

- It is permitted that a rental is open only if an estimated rental charge is provisionally charged to a credit card of the renter that is responsible for the rental.
- If the renter of a rental requests a price conversion then it is obligatory that the rental charge of the rental is converted to the currency of the price conversion.
- It is permitted that a rental is open only if each driver of the rental is not a barred driver. A rental must not be open if a driver of the rental is a barred driver.
- It is obligatory that each driver of a rental is qualified.
- It is obligatory that at the actual return date/time of each in-country rental and each international inward rental the local area of the return branch of the rental owns the rented car of the rental.
- It is obligatory that the country of the return branch of each international inward rental is the country of registration of the rented car of the rental.
- If the actual return date/time of a rental is after the end date/time of the grace period of the rental then it is obligatory that the rental incurs a late return charge.
- If the drop-off location of a rental is not the EU-Rent site that is base for the return branch of the rental then it is obligatory that the rental incurs a location penalty charge.
- If a rental is assigned then it is obligatory that the rented car of the rental is stored at the pick-up branch of the rental.
- At the actual start date/time of each rental it is obligatory that the fuel level of the rented car of the rental is full.
- It is obligatory that the start date of each reserved rental is in the future.
- It is obligatory that the rental duration of each rental is at most 90 rental days.

- If rental1 is not rental2 and the renter of rental1 is the renter of rental2 then it is obligatory that the rental period of rental1 does not overlap the rental period of rental2.
- It is obligatory that each rental car in need of service has a scheduled service.
- It is obligatory that the service reading of a rental car is at most 5500 miles
- If the rented car of an open rental is in need of service or is in need of repair then it is obligatory that the rental incurs a car exchange during rental.
- At the transfer drop-off date/time of a car transfer it is obligatory that the transferred car of the car transfer is owned by the local area that includes the transfer drop-off branch of the car transfer.
- It is obligatory that the country of the transfer drop-off branch of an international return is the country of registration of the transferred car of the international return.
- At the drop-off date/time of an international return it is obligatory that the transferred car of the international return is owned by the local area that includes the transfer drop-off branch of the international return.
- It is obligatory that a request for pick-up is accepted only if an assigned rental matches the request for pick-up and the renter that is responsible for the assigned rental has a valid credit card and the renter provides current contact details and each driver of the rental has a valid driver license.
- At the actual start date/time of a rental it is obligatory that the estimated rental charge is provisionally charged to some credit card of the renter that is responsible for the rental and that the renter has possession of the rented car of the rental.
- It is obligatory that the insurer of each operating company is notified of each overdue rental that has a pick-up branch that is in the operating company.
- If the drop-off location of a rental is not the EU-Rent site of a branch then it is obligatory that the rented car of the rental is recovered from the drop-off location to some branch.

List of Necessities:

These are found in the SBVR vocabulary and are closely linked with structural rules when one rewrites these more or less as 'it is necessary that'.

- Each car movement has exactly one movement-id.
- Each car movement has exactly one receiving branch.

- Each car movement has exactly one sending branch.
- Each car movement specifies exactly one car group.
- In-country car movement is included in Movements by Geography.
- International car movement is included in Movements by Geography.
- Local car movement is included in Movements by Geography.
- Movements by Direction contains the categories 'one-way movement' and 'round-trip movement'.
- Movements by Geography contains the categories 'in-country car movement' and 'international car movement' and 'local car movement.'
- One-way car movement is included in Movements by Direction.
- Round-trip car movement is included in Movements by Direction.
- At most one rental car is assigned to each car movement.
- The rental car that is assigned to a car movement is of some car model that is included in the car group that is specified in the car movement.
- The concept 'agency' is included in Branches by Type.
- The concept 'airport branch' is included in Branches by Type.
- The concept 'branch' is included in Organization Units by Function.
- The country of a branch is the operating country of the operating company that includes the local area that includes the branch.
- Each branch is included in exactly one local area.
- Branches by Type contain the categories 'airport branch' and 'city branch' and 'agency.'
- The concept 'city branch' is included in Branches by Type.
- Service depot is included in Organization Units by Function.
- Each EU-Rent site is located in exactly one operating country.
- Each local area is included in exactly one EU-Rent operating company.
- Each operating country has exactly one currency.
- Organization Units by Function contains the categories 'branch' and 'local area' and 'service depot'.
- Each rental organization unit is based at exactly one EU-Rent site.
- Service depot is included in Organization Units by Function.
- Each service depot is included in exactly one local area.

- Each car model that is included in a car group is charged at the rental rates of the car group.
- Each car model is included in exactly one car group.
- Each car model is supplied by exactly one car manufacturer.
- Each car model has at least one fuel type.
- Convertible is included in Models by Body Style.
- Coupe is included in Models by Body Style.
- Models by Body Style contains the categories 'convertible' and 'coupe' and 'sedan'.
- The request for pick-up matches exactly one rental.
- The renter presents a valid credit card.
- The renter provides current contact details.
- Each driver of the rental has a valid driver license.
- The pick-up branch of a rental is not changed.
- A rental has an actual pick-up date/time if and only if the actual pick-up date/time is the start date/time of the rental period that is included in the rental.
- A rental has an actual return date/time if and only if the actual return date/time is the end date/time of the rental period that is included in the rental.
- A rental has a pick-up branch if and only if the pick-up branch is the sending branch of the car movement that is included in the rental.
- A rental has a rental duration if and only if the rental duration is the duration of the period that is the rental period that is included in the rental.
- A rental has a requested car group if and only if the requested car group is the car group that is specified in the car movement that is included in the rental.
- The requested car group of an advance rental is not changed after the actual pick-up date/time of the advance rental.
- A rental has a return branch if and only if the return branch is the receiving branch of the car movement that is included in the rental.
- A rental has a scheduled pick-up date/time if and only if the scheduled pick-up date/ time is the scheduled start date/time of the rental period that is included in the rental.
- A rental has a scheduled return date/time if and only if the scheduled return date/time is the scheduled end date/time of the rental period that is included in the rental.

- A rented car is assigned to a rental if and only if the rented car is the rental car that is assigned to the car movement that is included in the rental.
- At a given date/time each rental has exactly one requested car group.
- Each rental has exactly one return branch at a given date/time.
- Each advance rental specifies exactly one car group at a given date/time.
- The car group specified for an advance rental is not changed after the actual pick-up date/time of the advance rental.
- The concept 'advance rental' is included in Rentals by Booking Mode.
- Each rental requests at most one car model.
- The car model specified for an advance rental is not changed after the actual pick-up date/time of the advance rental.
- The concept 'in-country one-way rental' is included in In-Country Rentals by Direction.
- In-Country Rentals by Direction contains the categories 'round-trip rental' and 'local one-way rental' and 'in-country one-way rental.'
- The concept international rental is included in Rentals by Movement Type.
- International Rentals by Direction contains the categories 'international inward rental' and 'international outward rental.'
- The concept local one-way rental is included in In-Country Rentals by Direction.
- Each advance rental is established by exactly one rental booking.
- Each rental booking has exactly one booking date/time.
- The booking date/time of the rental booking that establishes a cash rental is before the scheduled pick-up date/time of the rental.
- The booking date/time of the rental booking that establishes a points rental is at least 5 days before the scheduled pick-up date/time of the rental.
- Rentals by Booking Mode contains the categories 'advance rental' and 'walk-in rental.'
- Rentals by Movement Type contains the categories 'in-country rental' and 'international rental.'
- The concept round-trip rental is included in In-Country Rentals by Direction.
- The concept walk-in rental is included in Rentals by Booking Mode.
- Additional Charges by Basis contains the categories 'extras charge' and 'driver charge' and 'penalty charge.'

- If the rental duration is not for an exact number of days, the final part-day is charged as a Rental Day.
- The concept cash rental is included in Rentals by Payment Type.
- The concept cash rental price is included in Rental Prices by Basis.
- Each cash rental price is based on exactly one cash rental rate.
- Each cash rental honors exactly one lowest rental price.
- The lowest rental price of a rental is honored after the booking date/time of the booking that establishes the rental.
- The lowest rental price of a rental is honored before the actual return date/time of the rental.
- The concept drivers charge is included in Additional Charges by Basis.
- A cash rental price of a rental that is calculated because of EU-Rent price changes and that is less than the lowest rental price of the rental replaces the lowest rental price of the rental.
- A cash rental price of a rental that is calculated because of changes to the car group or rental duration of the rental replaces the lowest rental price of the rental.
- The lowest rental price of a rental is not replaced after the actual return date/time of the rental.
- The concept penalty charge is included in Additional Charges by Basis.
- Each points rental has a points rental price.
- The renter of each points rental is a club member.
- The concept points rental price is included in Rental Prices by Basis
- The concept points rental is included in Rentals by Payment Type.
- A rental has a business currency if and only if the business currency is the currency of the operating country of the operating company that includes the local area that includes the pick-up branch of the rental.
- Each rental charge of each rental is calculated in the business currency of the rental.
- Rentals by Payment Type contains the categories 'cash rental price' and 'points rental price.'
- Rentals by Payment Type contains the categories 'cash rental' and 'points rental.'
- A rented car has a drop-off branch if and only if the drop-off branch is the branch that is based at the EU-Rent site that is the drop-off location of the rental.
- Each rental car has exactly one vehicle identification number.

- Each rental car is of exactly one car model.
- A rental car is of a car group if and only if the rental car is of some car model that is included in the car group.
- Each rental car is owned by exactly one local area.
- Each rental car is stored at at most one branch.
- Each rental authorizes at most 3 additional drivers.
- Each corporate renter is a person who is accredited by a corporate customer and who is responsible for at least one rental.
- The concept corporate renter is included in Renters by Affiliation.
- Each corporate renter is authorized for at least one corporate rental agreement.
- Each individual customer is a given person who is not a corporate renter and who is responsible for at least one rental that is a Reserved Rental or an Assigned Rental or an Open Rental or a Returned Rental that has an end date that is less than 5 years earlier than the current day date.
- The concept individual customer is included in Renters by Affiliation.
- A rental has a driver if and only if the driver is the renter that is responsible for the rental or an additional driver that is authorized in the rental.
- Each rental has exactly one renter.
- The renter of a rental is not changed.
- Renters by Affiliation contains the categories 'individual customer' and 'corporate renter'.

According to SBVR (OMG, 2008) a request for pick-up matches a rental when:

- The rental is assigned (has a rental car assigned to it).
- The pick-up branch of the rental is the branch at which the request is made.
- The renter of the rental is the person making the request for pick-up.
- The scheduled start date of the rental is the day of the request for pick-up.
- ...

According to SBVR (OMG, 2008) a valid credit card has following characteristics:

- The card is of a type that EU-Rent accepts.
- "Expiry date" on the valid credit card is after the scheduled end date/time of the rental.

- “Cardholder” is the person presenting the card.
- ...

A Valid driver license has some of the following characteristic according to the EU-Rent case in SBVR (OMG, 2008):

- “Expiry date” on the valid driver license is after the scheduled end date/time of the rental.
- Necessity: “Driver” is the person presenting the license.
- Necessity: The rented car falls within “vehicle types.”
- Necessity: The license is legally acceptable in the country of the pick-up branch.
- ...

Further studies of EU-Rent:

Another interesting explanation of the rules from EU-Rent comes from “Moving from Zachman Row 2 to Zachman Row 3 Business Rules from an SBVR and an xUML Perspective” (Markus Schacher, 2006). He explains how rules are interpreted on 2 different levels. There is the business people and IT. When moving from row 2 to row 3, business rules are formulated differently.

Structural business rules (business perspective):

Markus Schacher (2006) states that structural business rules are true by definition. Some examples are:

Rule R1:

It is necessary that each rental *uses* at most one car.

Rule R2:

It is necessary that each rental *is signed by* exactly one renter.

Rule R3:

It is necessary that each rental *books* exactly one car model

Rule R9: (where R8 is based upon, see further)

It is necessary that the open amount of each renter *is the sum of* the total price of each rental that *is signed by* the renter.

In the next figure is clearly visible that a rental uses at most one car (structural rule). This is visible with the connection lines between the different objects. Rules R1,R2 and R3 are in this way stated.

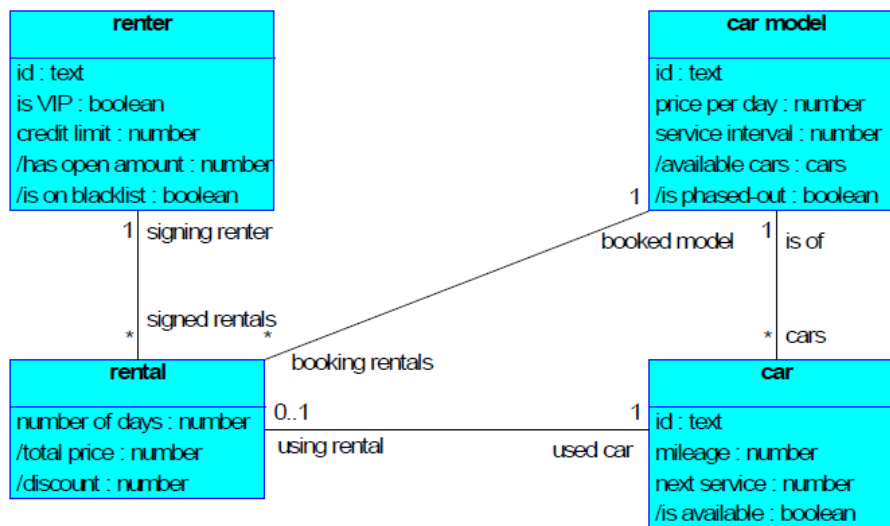


Figure 1: Objects renter, rental, car model and car and their interrelations. Source: “Moving from Zachman Row 2 to Zachman Row 3 Business Rules from an SBVR and an xUML Perspective” (Markus Schacher, 2006)

An example of a derivation rule, which is initially structural business rule implemented by IT as a derivation rule, is:

Rule R4:

It is necessary that the total price of each rental is the number of days of that rental x the price per day of the car model that is booked by that rental x (100 - the discount of that rental) / 100

Operative business rules (business perspective):

Markus Schacher (2006) states that in general, operative business rules affect the behavior of humans and/or IT systems. They have an effect on business processes carried out by those humans and/or IT systems. One can make a distinction by looking at the effect operative business rules may have on business processes:

- They may **initiate** business processes (**obligations**)
- They may **allow** business processes (**permissions**)
- They may **disallow** business processes (**prohibitions**)

Markus Schacher (2006) explains that Process rules are initially operative business rules (business view) implemented by IT as process rules (IT view). Some examples, with the process context, are:

Rule R5: → rental contract

It is prohibited that a renter has signed a rental if that renter is blacklisted.

Rule R6: → rental contract

It is prohibited that a rental books a car model that is phased-out.

Rule R7: → assign car

It is obligatory that a rental uses a car that is available and that has mileage that is minimal.

One can rewrite those and couple them with the business activities. The previous rules become:

Rule R5a:

It is obligatory to reject rental for a renter if that renter is blacklisted.

Rule R6a:

It is prohibited to create rental contract for a car model that is phased-out.

Rule R7a:

It is permitted to assign car for a car only if that car is available and has mileage that is minimal.

Markus Schacher (2006) explains also that constraints come further from operative business rules (business view). An example of such a rule is:

Rule R8:

It is prohibited that a renter has an open amount that is larger than the credit limit of that renter

Markus Schacher (2006) explains that this rule (R8) is non-trivial because this may happen in several situations. These situations could be:

- If a renter wants to sign a new rental, but already has some active rentals close to his credit limit.
- If a renter wants to extend an existing rental, but his open amount is already is close to his credit limit.
- If the credit limit of a renter is decreased below the open amount of the renter.
- If the price per day of a car model rented by the renter is increased and the renter's open amount is already close to his credit limit.¹
- If the discounting rules change.

There are also activities resulting from rule obligations. One of them is:

“At the end of each day cars are assigned to reservations for the following day.”

These kinds of decisions are come to the surface in a study by Jana Koehler (2010)

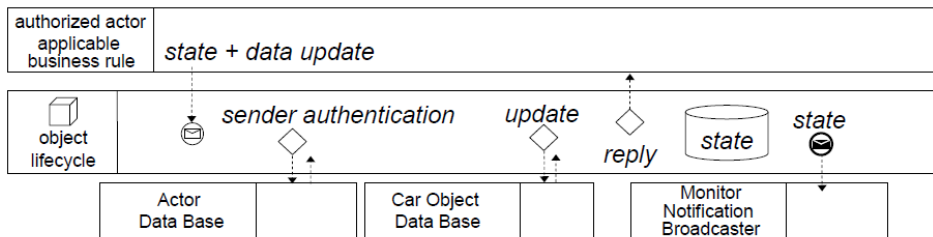
Link with BPM:

Jana Koehler (2010) explains a methodology for dynamic BPM of collaboration models in BPMN.

There is certain granularity obtained to react to certain business rules. Her methodology handles pre conditions as each rule at a time and uses it in BPMN models according to the nature of it. She also describes that data objects can be given a “state” that shows how a document may be changed or updated within the Process.

The object life cycle of a car starts when the car is bought. The car can only be expected to become available if the previous state is acquired. When the car is acquired an inspection by the responsible actor determines if the car can become available. The message comes in and the state transition can put the car state from acquired to available. After data validation this becomes persistent in the car database.

Object Life Cycle Collaboration



- 1 model for each state transition of each object

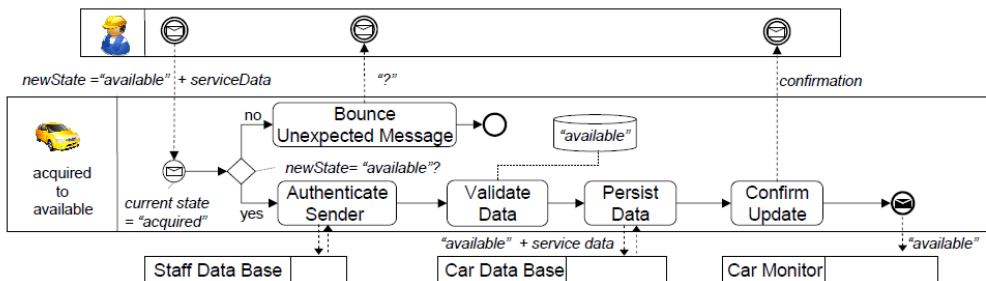
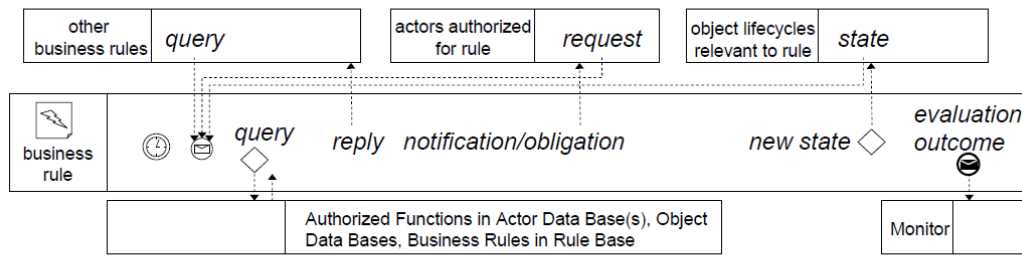


Figure 2: model for transition of state. Source: "BPMN for Dynamic Process Solutions" (Jana Koehler, 2008)

A business rule can be triggered by a time-event, a query from another business rule, a request from an actor or a state from an object life cycle. The business rule and the collaboration model are shown in figure 2. Here it is assumed that the end of the day is at 6pm. This shows the communication for the obligation rule to assign cars.

Business Rule Collaboration



- 1 model for each rule

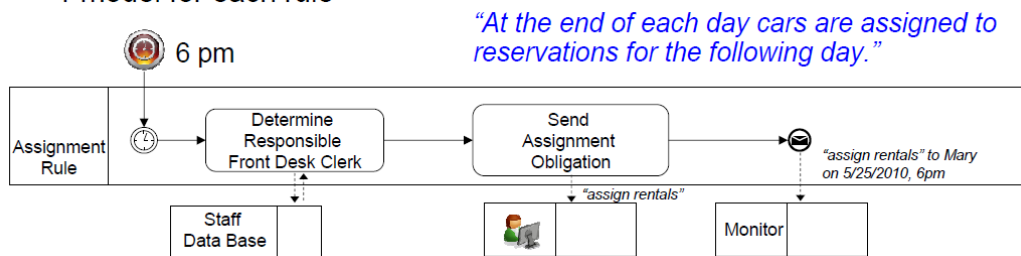


Figure 3: Collaboration model for rental assignment. Source: *Role of BPMN in a Modeling Methodology for Dynamic Process Solutions* (Jana Kochler, 2010)

My own BPM of this process is shown in the next figure. At the end of the day are cars assigned for the next day's rentals. A car pool is created in a looping process. Then the cars are checked for their availability. When an available car matches the requested car and the available car has the least mileage of all the available ones, the car is assigned to the rental. If not all the cars have been assigned, an upgrade is possible. Cars requested out of car group A can get a free upgrade to car group B, and so on. If no cars are available, the manager has several options to complete the assigning of cars. He can ask for instance to nearby branches to transfer a car.

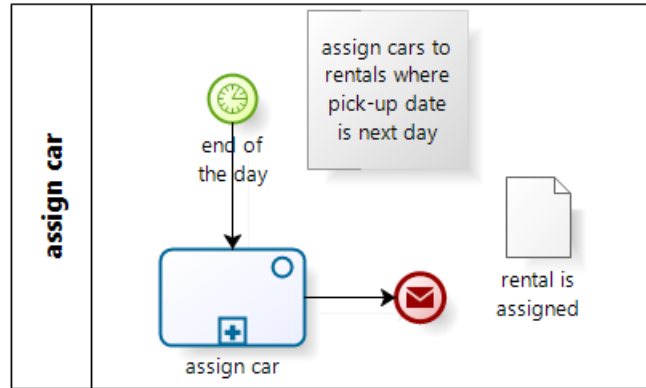


Figure 4: Assigning cars on a high level as a subprocess.

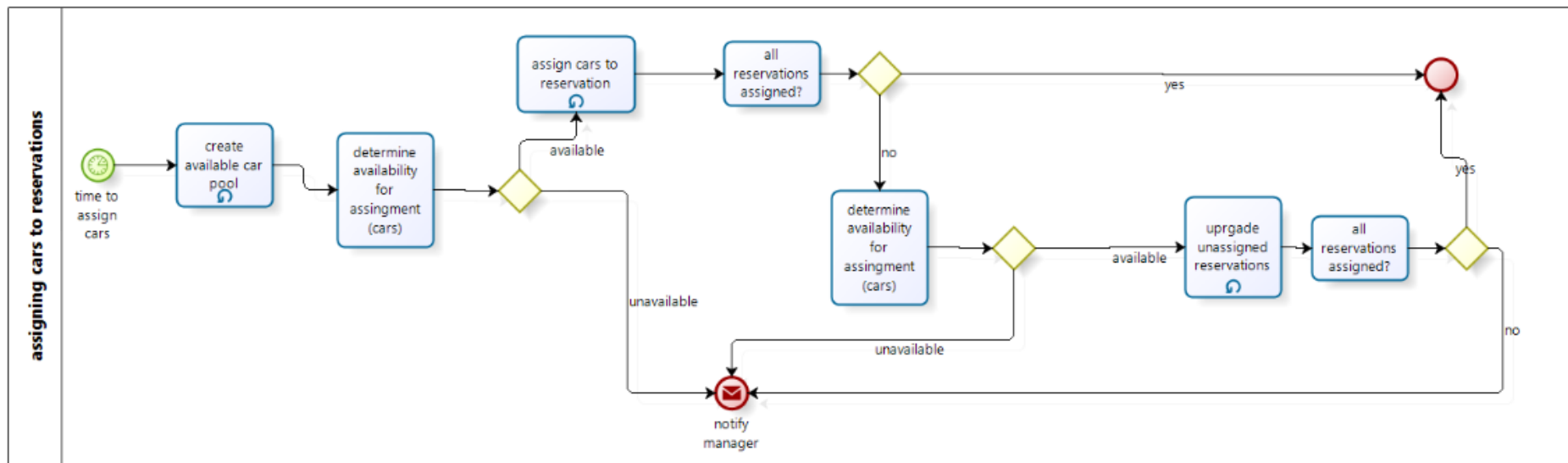


Figure 5: EU-Rent business process 'assigning cars to reservations'

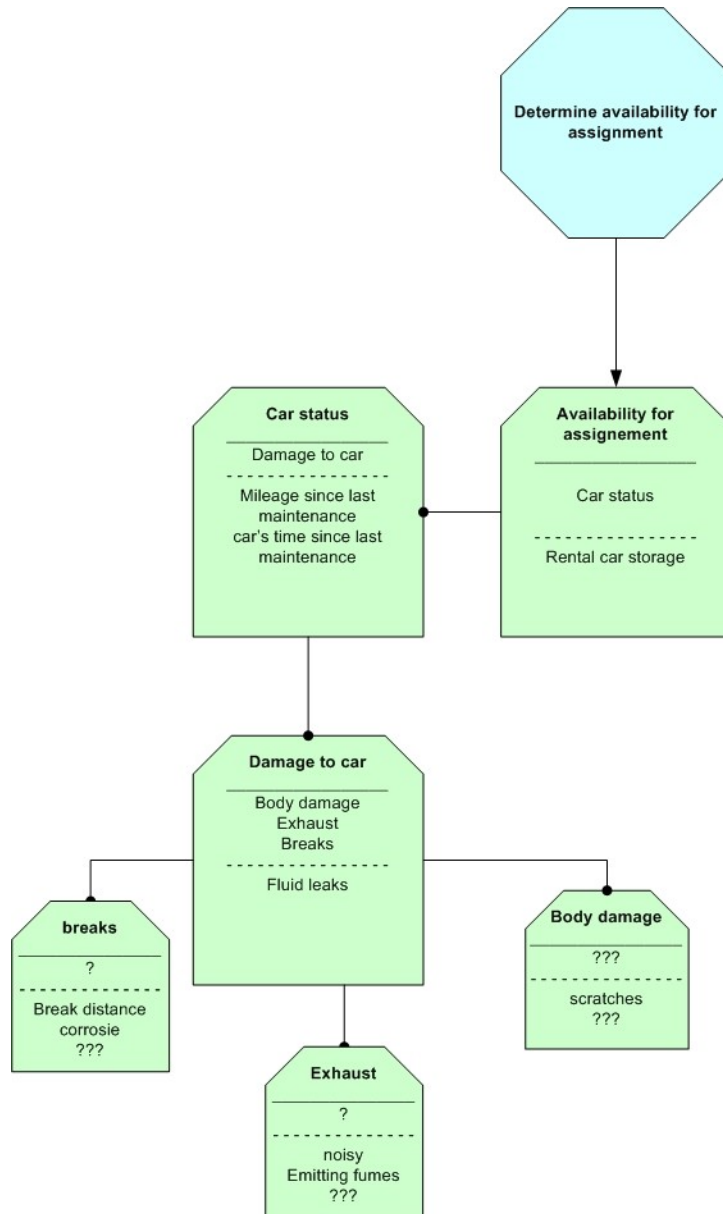


Figure 6: Decision Model 'Availability for assignment (Car is available / unavailable)

This decision model can be wrong, because on the moment of assigning the cars, the car status is known and stored data. There is a double check of the cars. Once they are returned and once when they are handed over. A triple check-up would be inefficient. The difficulties with unknown aspects are made clearly in chapter 3.

The following two figures illustrate other processes that are driven by an event. This time is it the customer who initiates a request. Figure 7 describes what should happen when a customer wants an extension from his/her rental period. The extension will be approved if the conditions that are tested

come to a conclusion of validation. Figure 8 describes it broader in the sense that certain contractual data may be changed before the pick-up.

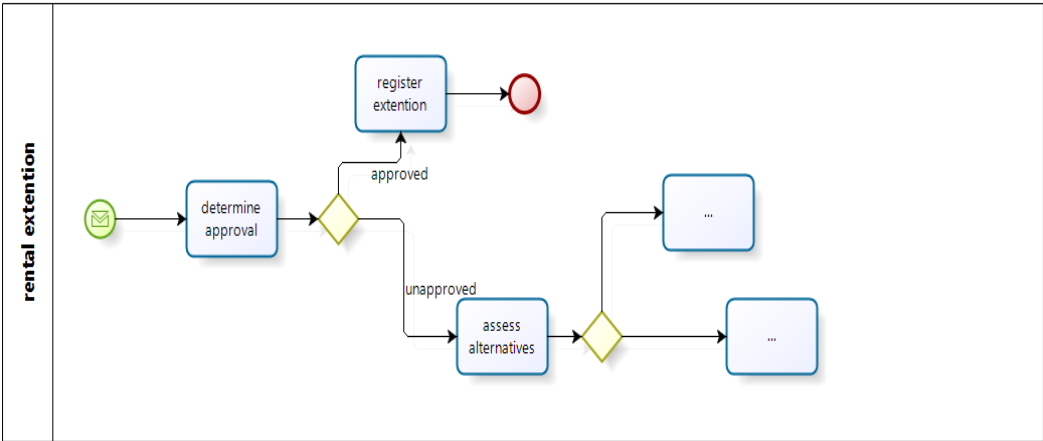


Figure 7: Rental extension process

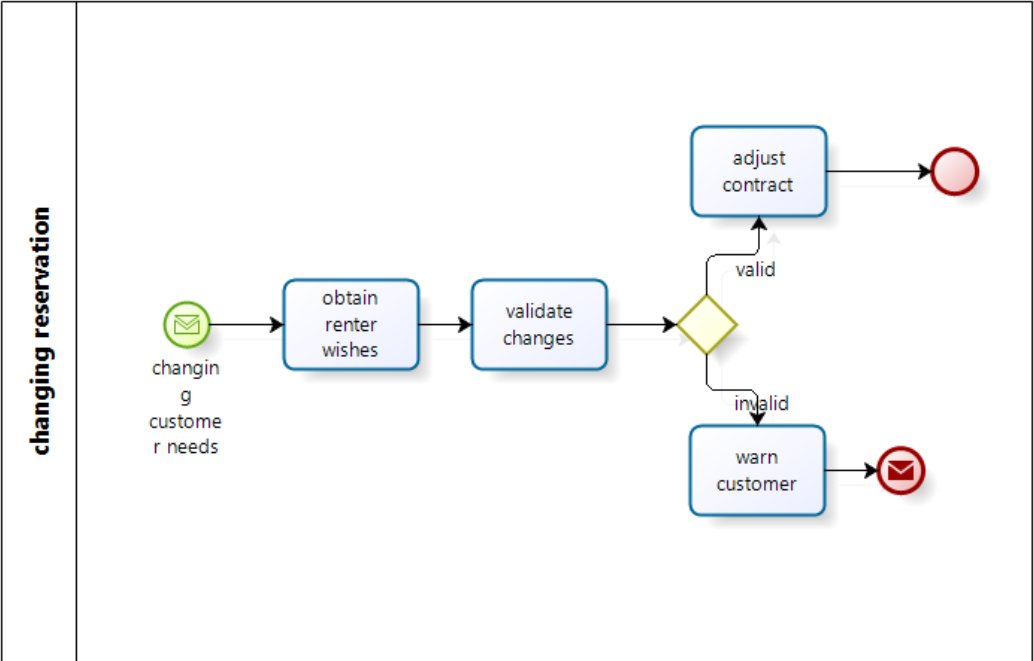


Figure 8: Changing reservation process

Appendix D: Short Glossary:

The list below is a limited list of fact types used by SBVR (OMG, 2008). The selection is based on the context of this thesis, to eliminate confusion.

Advanced reservation = A rental that is contracted with EU-Rent at least one day before the day of the scheduled pickup date/time of the car.

Walk-in rental = A rental that is contracted with EU-Rent on the day that the car is picked up.

Branch = A rental organization unit that has rental responsibility.

Car = A vehicle owned by EU-Rent and rented to its renters

Rented car = Rental car that is assigned to a rental

Car model = A named type of car built to the same specification, e.g., body style, engine size, fuel type.

Mileage = The current odometer reading of a car.

Rental = A contract with a renter specifying use of a car of a car model for a number of days (rental period) to drive from branch to branch (a car movement).

Optional extra = An item that may be added to a rental at extra charge if the renter so chooses.

Renter = A person contractually responsible for a rental.

Number of days = The contractual duration of a rental.

Price per day = The list price for renting a specific car model for one day.

Open amount = The sum of all outstanding amounts of a renter.

Credit limit = The maximum allowed open amount of a renter.

Discount = A deduction from the rental price in percent.

Actual return = A time point (date) at which a car of a rental is actually brought back.

Expected return = A time point (date) at which a car of a rental is expected to be brought back.

Penalty = An amount that has to be paid in addition to the total price of a rental, if a car is not returned on time.

Rental request = An information received by EU-Rent that a customer wants to rent a car.

Rental contract = This is the signed document that defines the conditions for a rental.

Renter = A person or organization that is contractually responsible for a rental.

Current contact details = Contact details that have been confirmed as up-to-date by the renter.

Car transfer = A logistical movement of a car by EU-rent staff.

Car movement = A planned movement of a rental car of a specified car group from a sending branch

to a receiving branch.

Rental organization unit = organization unit that operates part of EU-Rent's car rental business.

Service depot = A rental organization unit that has servicing responsibility.

Branch = A rental organization unit that has rental responsibility.

Agency = A branch that does not have an EU-rent location and has minimal car storage and has on-demand operation.

Airport branch = A branch that has an EU-rent location and has large car storage and has 24-7 operation.

City branch = A branch that has an EU-rent location and has moderate car storage and has long business hours.

Return branch = branch stipulated in the rental contract for return of the rented car

In country one-way rental = A one way rental that includes an in-country car movement.

International rental = A one-way rental that includes an international car movement. (inward-outward)

Local one-way rental = A one-way rental that includes a local car movement.

Base rental price = The price charged for the use of the rented car of a rental before any additional charges are added.

Cash rental = A rental that is charged in money.

Points rental = A rental that is charged in loyalty club points.

Rental charge = The total amount estimated or charged for a rental.

Bad experience = An undesirable occurrence during a rental that is the fault of one of the drivers.

Club member = A renter who has joined EU-rent's loyalty club. (also used as VIP in this thesis)

Loyalty club = EU-rent's incentive scheme for its frequent renters.

Appendix E: Architecture:

The figure below represents the entire architecture of an enterprise according to Zachman. This is called the Zachman Framework. This refers back to chapter 6 (6.1).

	DATA	WHAT	FUNCTION	HOW	NETWORK	WHERE	PEOPLE	WHO	TIME	WHEN	MOTIVATION	WHY	SCOPE (CONTEXTUAL)
SCOPE (CONTEXTUAL)	List of Things Important to the Business	List of Processes the Business Performs	List of Locations in which the Business Operates	List of Organization Important to the Business	List of Events/Cycle Significant to the Business	List of Business Goals/Strategies	Planner	BUSINESS MODEL (CONCEPTUAL)	Owner	SYSTEM MODEL (LOGICAL)	Designer	TECHNOLOGY MODEL (PHYSICAL)	Builder
Planner	ENTITY - Class of Business Thing	Process - Class of Business Process	Node - Major Business Location	People - Major Organization Unit	Time - Major Business Event/Cycle	Ends/Mean - Major Business Goal/Strategy	BUSINESS MODEL (CONCEPTUAL)	Owner	SYSTEM MODEL (LOGICAL)	Designer	TECHNOLOGY MODEL (PHYSICAL)	Builder	DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)
BUSINESS MODEL (CONCEPTUAL)	e.g. Semantic Model	e.g. Business Process Model	e.g. Business Location Link - Business Linkage	e.g. Work/Flow Model	e.g. Master Schedule	e.g. Business Plan	Owner	SYSTEM MODEL (LOGICAL)	Designer	TECHNOLOGY MODEL (PHYSICAL)	Builder	DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)	FUNCTIONING ENTERPRISE
Owner	Ent = Business Entity Rel = Business Relationship	Proc = Business Process I/O = Business Resource	Node = Business Location Link = Business Linkage	People = Organization Unit Work = Work Product	Time = Business Event Cycle = Business Cycle	End = Business Objective Mean = Business Strategy	Owner	SYSTEM MODEL (LOGICAL)	Designer	TECHNOLOGY MODEL (PHYSICAL)	Builder	DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)	FUNCTIONING ENTERPRISE
SYSTEM MODEL (LOGICAL)	e.g. Logical Data Model	e.g. Application Architecture	e.g. Distributed System Architecture	e.g. Human Interface Architecture	e.g. Processing Structure	e.g. Business Rule Model	Owner	SYSTEM MODEL (LOGICAL)	Designer	TECHNOLOGY MODEL (PHYSICAL)	Builder	DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)	FUNCTIONING ENTERPRISE
Designer	Ent = Data Entity Rel = Data Relationship	Proc = Application Function I/O = User Views	Node = IS Function (Processor, Storage, etc) Link = Line Characteristics	People = Role Work = Deliverable	Time = System Event Cycle = Processing Cycle	End = Structural Assertion Mean = Action	Owner	SYSTEM MODEL (LOGICAL)	Designer	TECHNOLOGY MODEL (PHYSICAL)	Builder	DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)	FUNCTIONING ENTERPRISE
TECHNOLOGY MODEL (PHYSICAL)	e.g. Physical Data Model	e.g. System Design	e.g. Technology Architecture	e.g. Presentation Architecture	e.g. Control Structure	e.g. Rule Design	Owner	SYSTEM MODEL (LOGICAL)	Designer	TECHNOLOGY MODEL (PHYSICAL)	Builder	DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)	FUNCTIONING ENTERPRISE
Builder	Ent = Segment/Table/etc Rel = Point-to-Point/etc	Proc = Computer Function I/O = Data Element/Set	Node = Hardware/Software Link = Line Specifications	People = User Work = Screen/Format	Time = Execute Cycle = Component Cycle	End = Condition Mean = Action	Owner	SYSTEM MODEL (LOGICAL)	Designer	TECHNOLOGY MODEL (PHYSICAL)	Builder	DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)	FUNCTIONING ENTERPRISE
DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)	e.g. Data Definition	e.g. Program	e.g. Network Architecture	e.g. Security Architecture	e.g. Timing Definition	e.g. Role Specification	Owner	SYSTEM MODEL (LOGICAL)	Designer	TECHNOLOGY MODEL (PHYSICAL)	Builder	DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)	FUNCTIONING ENTERPRISE
Sub-Constructor	Ent = Field Rel = Address	Proc = Language Statement I/O = Control Block	Node = Address Link = Protocol	People = Identity Work = Job	Time = Interrupt Cycle = Machine Cycle	End = Sub-condition Mean = Step	Owner	SYSTEM MODEL (LOGICAL)	Designer	TECHNOLOGY MODEL (PHYSICAL)	Builder	DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)	FUNCTIONING ENTERPRISE
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	Owner	SYSTEM MODEL (LOGICAL)	Designer	TECHNOLOGY MODEL (PHYSICAL)	Builder	DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)	FUNCTIONING ENTERPRISE

Figure 9: The Zachman Framework. Source: The Decision Model a business logic framework linking business and technology (Barbara von Halle & Larry Goldberg, 2009)

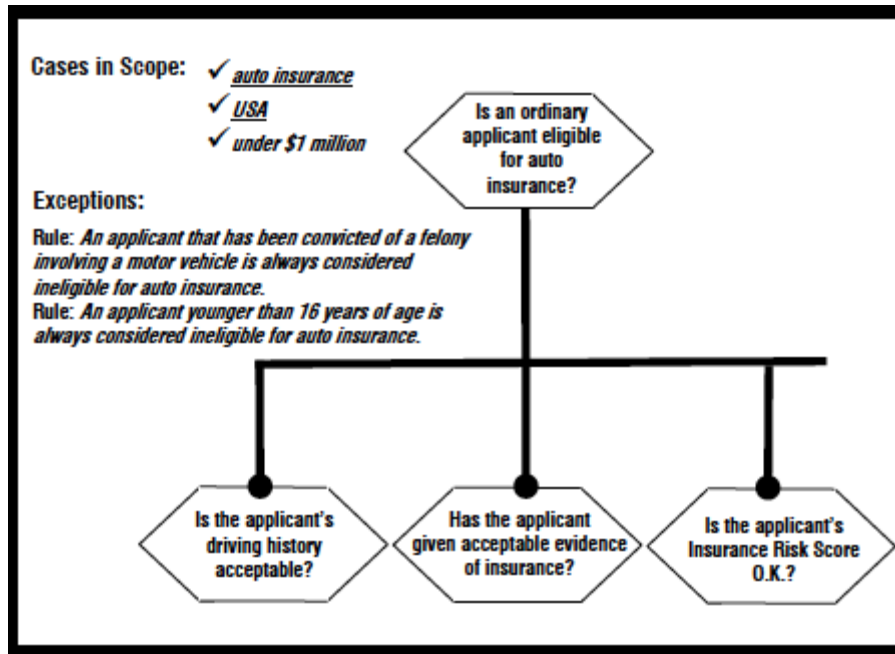
In the next figure the different levels of the Zachman Framework are visible with the link to software development.

		← Abstractions (Columns) →					
The Zachman Framework		DATA <i>What (Things)</i>	FUNCTION <i>How (Process)</i>	NETWORK <i>Where (Location)</i>	PEOPLE <i>Who (People)</i>	TIME <i>When (Time)</i>	MOTIVATION <i>Why (Motivation)</i>
↑ Perspectives (Rows) ↓	SCOPE (Contextual) <i>Planner</i>	List of things important to the business	List of processes the business performs	List of Locations in which the business operates	List of Organizations Important to the Business	List of Events Significant to the Business	List of Business Goals/Strategies
	BUSINESS MODEL (Conceptual) <i>Owner</i>	Semantic Model	Business Process Model	Business Logistics System	Work Flow Model	Master Schedule	Business Plan
	Computation-Independent Model (CIM)						
	SYSTEM MODEL (Logical) <i>Designer</i>	Logical Data Model	Application Architecture	Distributed System Architecture	Human Interface Architecture	Processing Structure	Business Rule Model
	Platform-Independent Model (PIM)						
TECHNOLOGY MODEL (Physical) <i>Builder</i>	Physical Data Model	System Design	Technology Architecture	Presentation Architecture	Control Structure	Rule Design	
Platform-Specific Model (PSM)							
DETAILED REPRESENTATIONS (Out-of-Context) <i>Sub-Contractor</i>	Data Definition	Program	Network Architecture	Security Architecture	Timing Definition	Rule Specification	
CODE							

Figure 10: How the MDA models used in software development map to the Zachman Framework.
 Source: The Zachman Framework and the OMG's Model Driven Architecture (Frankel, D., et al., 2003)

Appendix F: Question charts:

The figure below represents a question chart for car insurance. The main question is about the eligibility of an applicant for auto insurance. This question represents the decision that needs to be taken.



*Figure 11: Q-Chart for the decision Is an applicant eligible for auto insurance for USA under \$1 million? Source: *Introducing Question Charts (Q-Charts™) for Analyzing Operational Business Decisions: A New Technique for Getting at Business Rules.* (Ross, 2010)*

In the next picture the different dependencies are made clear. The consideration dependency is illustrated on the left with the solid line connection. The one on the right with the dotted line, illustrates a relevance dependency.

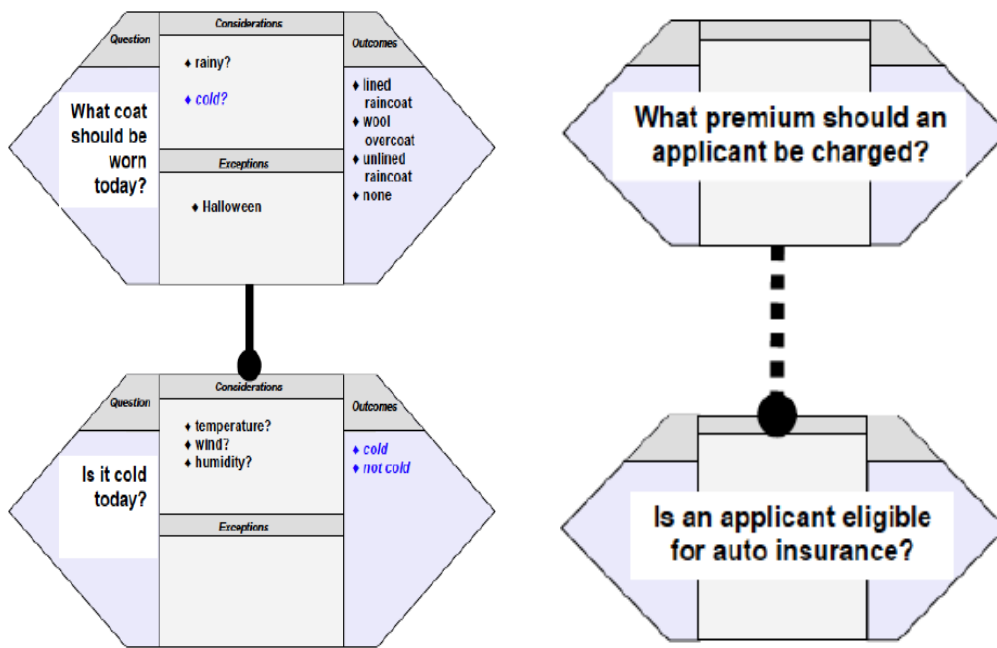


Figure 12: two different dependencies. Source: *Introducing Question Charts (Q-Charts™) for Analyzing Operational Business Decisions: A New Technique for Getting at Business Rules.* (Ross, 2010)

Appendix G: Decisions:

Contexts:

Decision have different natures dependable on the context. The Decision Model will have a different purpose with these various contexts. The contexts, with their influence by The Decision Model, described in chapter 8 are visible in the following table.

<i>Operative Context</i>	<i>Characteristics of Inputs</i>	<i>Typical Characteristics of Corresponding Decision Models</i>	<i>Usefulness of the Corresponding Decision Model</i>
Simple	Known knowns	<ul style="list-style-type: none"> • Known fact types • Known fact values • Related to business process models 	<ul style="list-style-type: none"> • Delivers and deploys agreed-upon and shared business logic • Ensures complete and accurate business logic • Is measurable against business objectives • Enhances agility in business processes
Complicated	Known unknowns	<ul style="list-style-type: none"> • Known fact types (not always) • Unknown fact values • More than one possible solution for the same input • Related to complex event processing 	<ul style="list-style-type: none"> • Identifies unknown fact types and values • Ascertains areas needing specific expertise • Delivers business logic shared from experts to nonexperts • Provides clarity to available solutions and simplifies comparative analysis • May reduce complicated business decisions to simple ones • Expedites changes when events dramatically change business conditions
Complex	Unknown unknowns	<ul style="list-style-type: none"> • Unknown fact types • Unknown fact values • Unclear solutions for some or all inputs • Related to fluid circumstances 	<ul style="list-style-type: none"> • Aids in developing solutions • Delivers business logic in a form amenable to iterative development • May reduce complex decisions to complicated or simple ones

Table 1: Usefulness of the Decision Model in different operative contexts. Source: The Decision Model a business logic framework linking business and technology (Barbara von Halle & Larry Goldberg, 2009)

<i>Operative Context</i>	<i>Characteristics of Inputs</i>	<i>Typical Characteristics of Corresponding Decision Models</i>	<i>Usefulness of the Corresponding Decision Model</i>
Chaotic	Unknowables	<ul style="list-style-type: none"> • Any of the above • Related to crisis • Applies to islands of order within chaos 	<ul style="list-style-type: none"> • Aids in delivering order where order can be envisioned

Table 1 (continued): Usefulness of the Decision Model in different operative contexts. Source: The Decision Model a business logic framework linking business and technology (Barbara von Halle & Larry Goldberg, 2009)

Categories:

There are different kinds of decisions like described in chapter 8. The figure for complexity, volume and impact of a decision is in on the next page. The criteria are based on the value of a decision and the volume that such a decision takes in the business. This criteria is represented by the y-axis. The complexity of decisions is another aspect, which is represented on the x-axis.

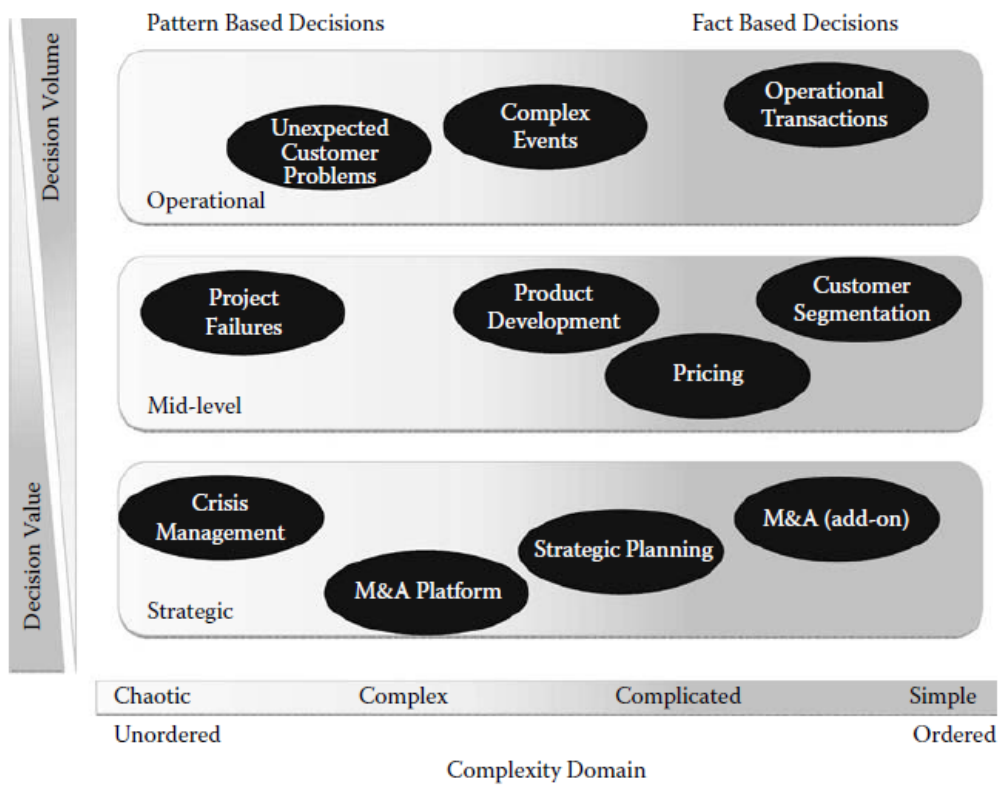


Figure 13: categorizing decisions. Source: *The Decision Model a business logic framework linking business and technology* (Barbara von Halle & Larry Goldberg, 2009)

Information quality decisions:

The rules associated with Information quality can also be modeled within The Decision Model. The different data quality dimensions are visible in the following table.

Data Quality Dimension	Definition	Repository of Data Quality
1. Completeness	<p>The determination that no additional data is needed to determine whether a Single Family Loan is eligible for settlement.</p> <p>Completeness applies to fact types that are always required, those that are sometimes required (depending on</p>	<p>The Glossary will indicate, for each fact type, whether it is <i>always</i> required or always optional for the general context of a business process (applies to all circumstances regardless of</p>

	circumstances), and those whose population is irrelevant.	process context) <u>Decision Model Views</u> will indicate the logic by which a fact type is <i>sometimes required</i> or <i>irrelevant</i> (i.e., those circumstances are tested in the logic)
2. Data Type	The determination that a fact value conforms to the predefined data type for its fact type. For example; date, integer, string, Boolean, etc.	<u>The Glossary</u> will indicate, for each fact type, its data type
3. Domain Value	The determination that a fact value falls within the fact type's valid range.	<u>The Glossary</u> will indicate, for each fact type, its valid values <u>Decision Model Views</u> will indicate the logic by which the domain for a fact type is further restricted for a specific process context to be compliant with Policy
4. Consistency	The determination that a fact value makes business sense in the context of related fact type/s fact values	<u>Decision Model Views</u> will provide the logic to determine consistency of values for related fact types.
5. Reasonableness	The determination that a fact value conforms to predefined reasonability limits.	<u>Decision Model Views</u> will provide the logic to determine whether a fact value is within sensible limits.

6. Accuracy	<p>The determination that a fact value (regardless of the source) approaches its true value. Accuracy is typically classified into the following categories:</p> <p>Accuracy to authoritative source: A measure of the degree to which data agrees with an original, acknowledged authoritative source of data about a real world object or event, such as a form, document, or unaltered electronic data from inside or outside the organization.</p> <p>Accuracy to real world data (reality): A characteristic of information quality measuring the degree to which a data value (or set of data values) correctly represents the attributes of the real-world object such as past or real time experience.</p>	<p><u>Decision Model Views</u> will indicate the logic by which a fact value is determined to be as accurate as is necessary.</p>
-------------	--	---

Table 2: Data Quality framework. Source: Better, Faster, Cheaper Part II - The Decision Model Meets Data Quality Head On (Perdersen, 2011)

Auteursrechtelijke overeenkomst

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:

Externalization of the decision process from BPM to optimize business processes

Richting: **Master of Management-Management Information Systems**

Jaar: **2011**

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Niet tegenstaand deze toekenning van het auteursrecht aan de Universiteit Hasselt behoud ik als auteur het recht om de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij te reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

Ik bevestig dat de eindverhandeling mijn origineel werk is, en dat ik het recht heb om de rechten te verlenen die in deze overeenkomst worden beschreven. Ik verklaar tevens dat de eindverhandeling, naar mijn weten, het auteursrecht van anderen niet overtreedt.

Ik verklaar tevens dat ik voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen heb verkregen zodat ik deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal mij als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze overeenkomst.

Voor akkoord,

Biesemans, Glen

Datum: **2/06/2011**