2012•2013
FACULTY OF BUSINESS ECONOMICS
Master of Management: Management Information Systems

Masterproef

Business Rules Management Systems
A practical implementation

Promotor :
Prof. dr. Koenraad VANHOOF

Fatma Maqsoom Mubarik

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

universiteit hasselt
KNOWLEDGE IN ACTION

Universiteit Hasselt | Campus Hasselt | Martelarenlaan 42 | BE-3500 Hasselt
Universiteit Hasselt | Campus Diepenbeek | Agoralaan Gebouw D | BE-3590 Diepenbeek

universiteit hasselt
KNOWLEDGE IN ACTION

2012•2013
FACULTY OF BUSINESS ECONOMICS
Master of Management: Management Information Systems

Masterproef

Business Rules Management Systems
A practical implementation

Promotor :
Prof. dr. Koenraad VANHOOF

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

universiteit hasselt
KNOWLEDGE IN ACTION

Acknowledgements

This thesis has been made possible by kind support given to me by several individuals and was a great pleasure to work them all.

First and foremost I would like to thank my promoter Prof. Koen Vanhoof for giving me the opportunity, support and guidance all the way to explore and learn the new knowledge area.

I am greatly thankful to Frank Vanhoenshoven, who has always made time for me and shared his immense theoretical and hands-on knowledge and guided me through the practical implementation of Business Rules Management System. Without his support and significant amount of devoted time practical part of this thesis was not possible.

Additional I want to express my appreciation to Etalentis B.V. management & team for bearing with me and providing all needed information.

I wish to bestow my heartiest gratitude to my loving husband Zaheer Ud Din Babar, who has provided with much love and encouraged me throughout my studies. Thanks for believing in me, it is to you that I dedicate this thesis.

Table of Contents

Preface

There has been considerably extensive research done on Business Rules Management Systems and the implementation of this intelligent system. The goal of this paper/thesis is to shed a brief light on the widely accepted frameworks and theories till current date. Additionally how these theories and frameworks can help to develop a system and we will see a practical example of developing a system or migrating from manual to automatic environment with the help of business rules management systems.

The structure of this paper/thesis is designed in such a way that technical and non-technical readers can understand it easily. Here is the brief description of each chapter included and how this paper is structured.

Section I: Introduction (Chapter 1 & 2): We will start with the introduction, history and purpose of Business Rules Management System.

Chapter 1 provides an introduction on the Business Rules Management Systems role today and who actually owns these rules IT or business people?

Chapter 2 introduces the business rules; it's brief history, what are business rules and how they can be managed and most importantly why they have a significant role in aligning IT & Business.

Section II: Technical Insight (Chapter 3): In this section we will get to know basic technical know-how of business rules management systems.

Chapter 3 describes technical insight of Business Rules Management Systems, architectures, Zachman's Framework, difference between decision models and decision table and lastly comparison of few selected business rules enabler products available in market today i.e. Open Rules, IBM Operational Decision Management System, FICSO Blaze Advisor and Drools Guvnor.

Section III: Importance of Decision Model; Theories & Frameworks (Chapter 4 & 5): This section will review James Taylor and Barbara von Halle & Larry Goldberg published work on decision management systems and it's role in developing business rules management systems.

Chapter 4 introduces decision models importance in business rules management system from James Taylor's point of view as described in his book 'Decision Management Systems: A Practical Guide to Using Business Rules and Predictive Analytics' a step-by-step getting insight of why decision models are important, how to build them and what are the enablers.

Chapter 5 prominences the book 'The Decision Model: A Business Logic Framework Linking Business and Technology' written by pioneer author and expert Barbara von Halle and Larry Goldberg. This section provides a pragmatic approach on how to rethink, view, design, execute, and govern business logic and how to implement the decision models in an ever changing businesses and technologies.

Section IV: Practical Implementation (Chapter 6): As till now we have learned a lot about the business rules management systems and here we will practice the knowledge gained in order to evaluate how the theories comes together during the practical implementation.

Chapter 6 is the core of this paper meaning in this section all the theories, frameworks, architecture and models learned and described in previous chapters will be implemented by working out a real business-operating example of salary simulation in a business setup.

Section V: Conclusion (Chapter 7): The last section will conclude the findings, feedback and areas for future research.

Chapter 7 concludes the key finding, development of business rules management systems and implementation process. In addition what lessons were learned and pinpoint the gaps, which still exist and needs to be further investigated and explored.

SECTION I INTRODUCTION

1 Introduction

With the ongoing innovation and development of tools, process, methods, technologies & ways of working in a business environment, it's quite challenging to update daily in use working tools in the company in an uninterrupted, timely and efficient manner. Business Rules Management System (BRMS) is a way to move forward to coupe with the rapid changes in the business process today.

Business rules management itself is part of the overall business process management (BPM) framework. To improve the processes and react to accelerated and ever-changing business needs & environment, companies started to automate the business processes by using software applications. As a surprising fact business rules always existed within these software applications i.e. they were buried deep inside the application in some procedural programming language. With the passage of time as the business rules matured, business users tired to manage and control these rules without working together with the IT.

Business rules are described as the externalization of business logic and automating the means of implementing and enforcing business policies, and BRMS are a means of accelerating the change process. This thesis tries to solve that mystery of what differentiates a structured logic to be classified as a business rule by looking at a real-world example of designing and implementing the 'Salary Determination Model ' in other words automation of existing manual salary determination process in a small but growing business consultancy company called Etalentis to fulfill the business needs.

Lastly, is the initial BRMS promise "*Return Control over Business Logic back to Business People!*" achieved? There has been so much advancement in the development of the new tools and how they operate etc. But how many of does really achieved the goal of giving back the control to business users? That's what I am going to find out and describe the findings in the chapter 7 of this paper.

2 Business & IT Alignment

Business rules management technologies can improve the time to market and cost of developing and maintaining software applications to support the business strategy to achieve short & long term goals.

Before we go into more details on our main topic business rules management System, I would like to provide a brief description on why business and IT alignment is important for business rules management.

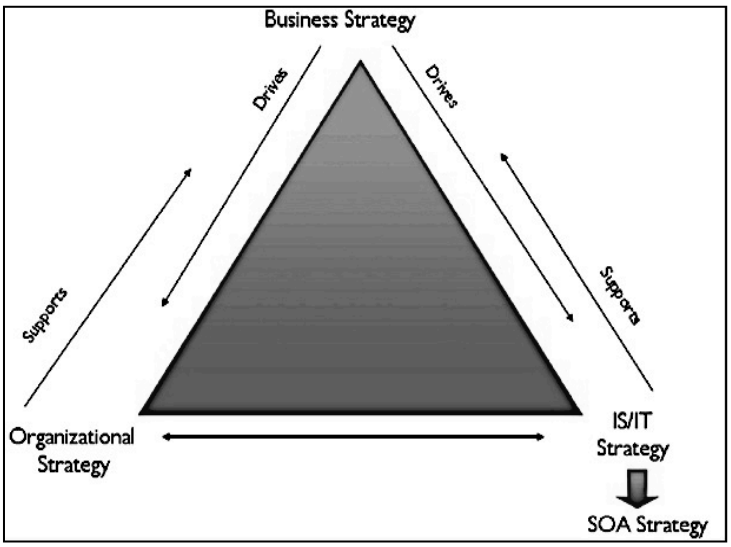

Figure 2.1: IS Strategy Triangle & SOA

Business Strategy drives all other strategies as explained in figure 2.1.

- Organizational and Information Strategy are dependent upon the Business Strategy.
- Changes in any strategy require changes in the others to maintain balance.
- Information System Strategy is also affected by the other strategies an organization uses.

Therefore, understanding the relationship between business, organization and IS/IT is important to design the IT tools to efficiently operate the businesses.

The problem in real life that we too often face is that developers expect users to understand and learn their complex language for example unified modeling language diagrams. In today's fast moving competitive environment there is not much time to learn and understand these complex languages. Instead there is a need to introduce simple language, which is understood by users and developers alike. Business process modeling approach and business rules management systems both have a crucial role in designing easily & simple to use, maintaining and improving IT tools.

The question is how to align IT practices with business need? To believe that adopting a business rules management system on its own will solve this problem would be naïve. Business rules management is only a part of the solution. To align IT with business we must also consider innovative approaches to requirements engineering

and service oriented architecture. Whilst its focus remains on business rules, business rules are about aligning IT with the business too.

2.1 Historical Background

The unpublished history of business rules exist as with the existence of the software with deeply and complex embedded codes in the software. With the evolution of the business the more formal vision of business rules arose from the dedicated experienced professionals with a goal to offer organizations best possible approach to develop business solution involving systems. Formal first definition of business rules management emerges back in 1984. Following is brief historical evolution of business rules existence:

1984 *"An explicit statement of a constraint that exists within a business's ontology."* [1]

1987 *"Specific rules (or business policies) that govern ... behavior [of the enterprise] and distinguish it from others.... These rules govern changes in the status [state] of the enterprise."*[2]

1994 *"A discrete operational business policy or practice. A business rule may be considered a user requirement that is expressed in non-procedural and non-technical form (usually textual statements).... A business rule represents a statement about business behavior."*[3]

2000 *"A statement that defines or constrains some aspect of the business ... [which is] intended to assert business structure, or to control or influence the behavior of the business. [A business rule] cannot be broken down or decomposed further into more detailed business rules.... [I]f reduced any further, there would be loss of important information about the business."*[4]

1997 *"A term, fact (type) or rule, representing a predicate."*[5]

1998 *"A directive that is intended to influence or guide business behavior. Such directives exist in support of business policy, which is formulated in response to risks, threats, or opportunities."*[6]

2000 *"An atomic piece of re-usable business logic, specified declaratively."*[7]

2002 *"... Conditions that govern a business event so that it occurs in such a way that is acceptable to the business."*[8]

2002 *"Basically, a business rule is a compact statement about an aspect of the business.... It's a constraint, in the sense that a business rule lays down what must or must not be the case. At any particular point, it should be possible to determine that the condition implied by the constraint is true in a logical sense; if not, remedial*

action is needed. This interpretation, which might be described as Boolean from a software perspective, is the main reason that the term business logic is so commonly used." [9]

2003 *" Rules build directly on terms and facts. Actually, a rule should simply add the sense of must or must not to terms and facts that have already been defined in the fact model and Concepts Catalog. In business problems involving hundreds or thousands of rules — not at all uncommon — there is no way to achieve consistency across such large numbers of rules without a common base of terms and facts."* [10]

2007 *". Represent the logic of business decisions."*[11]

2008 *". A rule that is under business jurisdiction."*[12]

2.2 What are Business Rules?

Rules are logical statements about how a system operates. Some of these rules may be expressed in the business language referring to real-world business entities, and are therefore called business rules. Business rules can represent among other things typical business situations. In recent years, the term business rule has gained an increasing amount of popularity in some communities, particularly certain information technology practitioners and BRMS vendors.

Examples of business rules include (are as follows):

- A customer will get a discount of 5% on his purchase, if he buys more than 2 items of product A
- If an offer has been sent to a customer and he/she has not responded after three days, a sales representative will call him/her back to ask whether he/she is still interested.

At first instance it seems quite easy to understand the meaning of the word ' business rules' but still there are various definitions out there to define' business rules' which at some point create misunderstanding to misguide you. So, before we go deep into the topic it's is important to understand and be clear about the meaning of 'Business rules'.

The Business Rules Group (BRG)[13], an organization of system and business analysis methodologists, has defined the definition of 'Business Rules' from business perspective and information system perspective as following;

Information System perspective: a business rule pertains to the facts of the system that are recorded as data and to the constraints on changes to the values of those facts. Or " *a business rule is a statement that defines or constraints some*

aspect of the business. It is intended to assert business structure or to control or influence the behavior of the business"

Business perspective: A business rule involves the behavior of people in the business (human activity) system i.e. " *business rule is guidance that there is an obligation concerning conduct, action, practice or procedure within a particular activity or sphere"*

Two important characteristics of a business rule are;

1 There ought to be a explicit motivation for it
2 It should have an enforcement regime stating what the consequence would be if the rule was broken

A classification scheme increases insight in what an author considers as the scope of the concept "business rule". The Business Rules Group distinguishes the following types of business rules:

1. **Structural assertions:** Which are either "terms" or "facts", in the usual meaning of the words.
2. **Action assertions:** Which test a certain condition. This kind can be split up further in a number of ways. One way is to distinguish conditions that must always be true ("integrity constraints") to keep a valid state, conditions that enable an action or trigger an event when true, and authorization rules for certain actions. Another way to categorize action assertions is by distinguishing hard (strict) and soft (guideline) action assertions.
3. **Derivations:** Which are either mathematical calculations or inferences.

In the table 2.1 the types of business rules are explained with an example to better grasp the concepts.

Structural Assertions	
Term	The Head Office
Fact	The Head Office is located in Maastricht
Action Assertions	
Constraint	A customer may order between 1 to 1000 items
Action Trigger	If the stock levels are below 50 items then replenish
Derivations	
Calculation	Profit = Revenue - Costs
Influence	If stock level are below 20, then the state is critical

Table 2.1: Types of Business Rules

As there is a lot of research going on this topic most probably some researchers may come up with other types and explanations or simply just don't agree.

Various sources mention a number of properties of business rules. As mentioned before, a problem is that some authors do not make clear whether they regard certain

aspects as inherent properties of business rules or rather guidelines for use.
Therefore, we first take a neutral point of view and describe only the properties that are most often encountered. Then an attempt is made to express the relation between business rules and knowledge.

2.3 What is Business Rules Management?

"*Business Rules Management System; a domain-independent knowledge system that includes a repository, an execution mechanism and authoring and management functionality for business rules and is invoked by an external system through a generic communications interface"*

- **WHAT:** A business rule management system (BRMS) enables organizational policies – and the operational decisions associated with those policies – to be defined, deployed, monitored and maintained separately from core application code. By externalizing business rules and providing tools to manage them, a BRMS allows business experts to define and maintain the decisions that guide systems behavior, reducing the amount of time and effort required to update production systems, and increasing the organization's ability to respond to changes in the business environment.[14]

- **WHY:** In the table 2.2 some of the main reasons are briefly explained stress the importance of the business rules management systems.

Business User Driven[15]	- Controlled by business with less IT intervention - Simulate scenarios for business and customer end users
Reduced Cycle Time	-Rapid business controlled change and publish cycle -Faster time-to-market
Automated	-Automate high volume and complex operational decisions -More efficient and less costly than current environment
Reusable	-Reuse your business policies across applications and across your extended enterprise -Reuse enterprise data across applications
Governance	-Manage business rules across the enterprise -Standardize change control, comply with regulations with complete audit trail

Table 2.2: Reasons for using Business Rules Management Systems

- **WHERE:** To provide some insight where the business rules management system can be effectively implemented. The table 2.3 provides some typical application of BRMS technology includes[16]:

Automating procedures	- Claims processing - Customer service management - Credit approval and limit management - Problem resolution - Sales
Advice & decision support	- Benefits eligibility - Sales promotions and cross selling - Credit collection strategy - Marketing strategy
Compliance	- External and legal regulations - Company policy
Planning & Scheduling	- Advertising - Timetables and meetings - Budgets - Product design and assembly
Classification	- Customers - Products & services - Risks

Table 2.3: Application of Business Rules Management System

- **How[17]:** Figure 2.2 is the visual explanation of how BRMS can link existing data source to improved business decision IT environment.

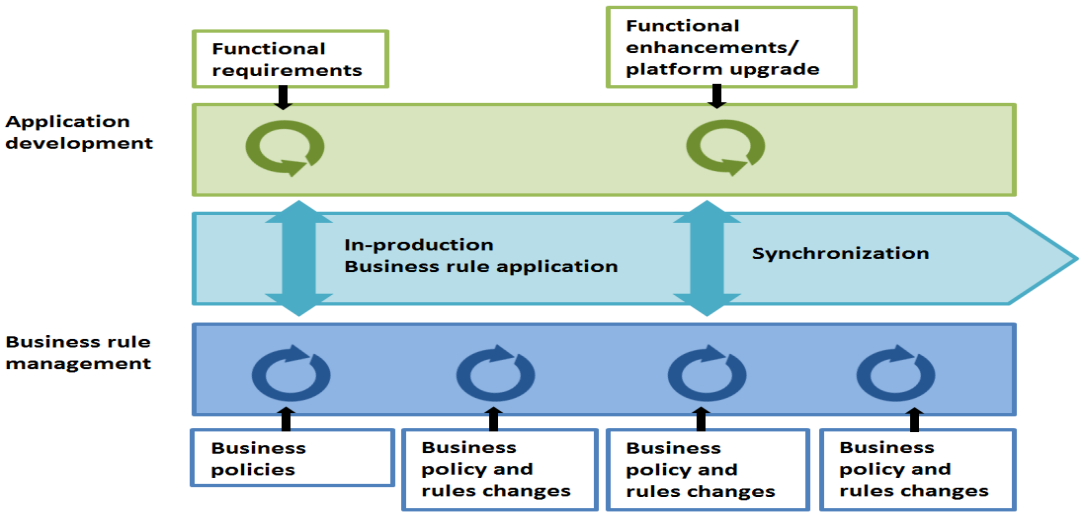

Figure 2.2: How BRMS links existing data source to business decision IT environment

The rules are stored in a repository or better in a central database. Furthermore, storing the rules in a layer separate from both applications and from the various databases that may exist in an organization gives obvious maintenance advantages. Centralizing the rules makes them more readily reusable. However, there is an opposing force: that of the need for reuse of the objects in our domain model.

If the rules (and indeed rule sets) are not encapsulated within the objects that they constrain, then those objects are incomplete and, if reused, may function incorrectly. For example, if an object "Insured" includes attributes related to a person's military services, it does not require that all business rules that deal with the insured be interested in those attributes. Such encapsulation of only the essential information in the Excel-based data types, together with live process modeling, allows for example OpenRules to complete the rule modeling cycle without leaving Excel.

2.4 Why use Business Rules Management System?

If we summarize according to various studies and research, there are top 6 reasons[18] of software development failure[19] are as following:

1- Lack of user involvement
2- Unrealistic expectations
3- Lack of executive support
4- No clear or ever changing requirements & specification
5- Incomplete requirements
6- Lack of resources

Five of the above mentioned failure reasons have nothing to do with choice of

platform, development environment, hardware of language but the communication between the developer and the user (stakeholder).

In development-centric culture of many IT development organization is that the IT developer expect the 'users' to learn and understand their language (i.e. form of UML diagrams). In today's highly competitive and fast moving environment this is quite unrealistic. Thus it is very important that the project team must develop common language, which is understood by the users and the developers, and is based on simple conceptual models of the domain written in easily understood terms. According to Ian Graham[20] business process modeling and business rules management systems both have an important role to resolve IT challenges faced today. Another challenge when the software is once is use, maintanace becomes the huge cost and in worse case scenario would lead to failure of the software. This is one of the reasons that object-oriented and component based development is so appealing. When the implementation of a data structure or function changes, these changes do not propagate to other objects. So the maintenance is localized to the changed object(s).

BRMSs can facilitate separate definition of policy from implementation and code details. In BRMSs the rules are subdivided that are into modules, encapsulated in individual objects. These encapsulate contain global or organizational policy, while rule sets that pertain to specific classes (such as clients or products) are stored within those objects.
The separated rule sets need to be maintained and kept under version control. This implies that a good BRMS will store rule sets in a repository.
Every business has rules, and business users want complete control and visibility over these rules. A BRMS decouples an application's business logic from its data access and from its flow control. Decision logic is exposed behind decision services, which support some business task. Such services are Validate Customer, Validate invoice, Check Credit History etc.

Using BRMS those service trend delegates the processing to a rule set deployed within a Rule Engine. That rule engine can be managed using a rule execution server, which provides the pooling mechanism; rule set management and execution monitoring functions.

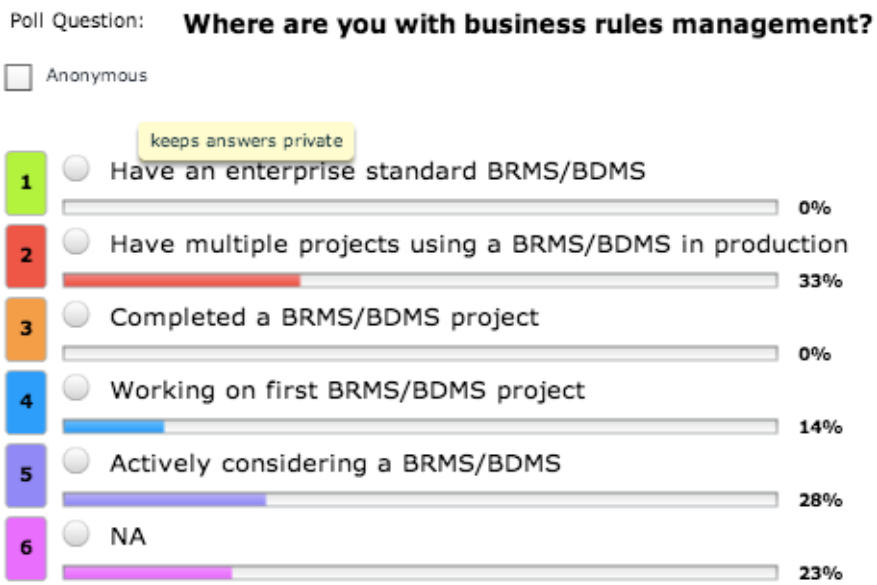

Figure 2.3:BRMS acceptance & knowledge

According to a poll (see results in Figure 2.3) done by James Taylor business people are becoming more and more aware of the benefits of business rules base management systems. Although the there are no still a lot of research and development is going on to standardize the business rules management systems but at least organization and are well informed and willing to adopt the new type to intelligent business rule based management system development.

2.5 Benefits and problems of BRMS

In this section I have briefly summarized some of the main benefits and problems of using business rules management system and put together in the table below. As we have already discussed in details in above section what is BRMS and how the is works thus see the Table 2.4 for a list of pros and cons of BRMS.

Pros	**Cons**
Better alignment and understanding between business and IT	There can be technical problems in debugging a system with thousands of rules
Greater consistency across the enterprise	Large, badly segmented rules become increasingly difficult to manage
Reusable business rules	Rules written in COBOL or 3GL programming languages are buried deep in the system code of a single application.
Simplicity and clearer auditability	Adding rules to legacy systems can be risky, given the fragility of some

	of these systems and the complexity of trying to make them perform functions they were never designed to support.
Easy and faster maintenance. Rules can be easy changed by business people without requiring IT	
Decision applications and services are developed usually by IT. But the developers can also let nontechnical business users write many of the rules themselves using graphical trees, tables or Web forms.	
Increased control over implemented decision logic for compliance and better business management	
Automated decisions save time and money. Rule can be used in many flexible ways to reach decisions that can be carried out by virtually any application or business process	
Rules are written in a simple rule language and managed separately from system code. Thus easy application development	

Table 2.4: Business Rules Management System

SECTION II: TECHNICAL INSIGHT

3 Implementing Business Rules Management System

This section describes the implementation process of a BRMS by first exploring the architecture, methods and techniques of BRMS to choose from the possibility. Next understanding the difference between the decision tables and decision models to have a clear knowledge on these elements. Lastly an analysis on BRMS tools for final implementation.

3.1 Context of BRMS implementation

This subsection describes the context of a BRMS implementation from the viewpoint of business user perspective. Keeping in mind all relevant aspects and knowledge needed to develop a business rule management base application and still providing comprehensive technical understandable concepts.

Furthermore the implementation of any software system does not stand on its own, but is done in order to accomplish a certain goal or execute a strategy. The continuous process of the development, implementation and management of changes in an organization is often seen as a cycle. The Deming-Shewhart "Plan-Do-Check-Act" quality cycle is a famous example, but many variations exist that are based on it. Methodologies for Software Engineering, Business Process Management, Enterprise Architecture and Organization Change Management can be placed within this cycle.

3.2 BRMS Architect, Methods & Techniques

There have been many various business frameworks introduced to define the role of business rules in the organization and system development. The Zachman's Framework has came on top, the reason is simple an architecture is needed for alignment, integration, change and time to market. Aren't that all we today want to achieve in our businesses, processes, procedures and systems? John Zachman's Framework for Enterprise Architecture is answer to the question. As the framework is very generic which means one can adopt it to its needs and usage. Barbara von Halle has also discussed and written quite much on Zachman's Framework and taken a step further to adopt the framework and simplify it to the needs to current systems development environment[21].The Zachman's Framework depicted as a bounded 6x6 matrix shown in Figure 3.1 with the communication interrogatives as columns and the reification transformations as rows. The Zachman's framework provides a common vocabulary and set of perspectives used for defining and describing complex

enterprise systems. It is a logical structure for classifying and organizing the elements of an enterprise that are important to both the management of the enterprise and the development of its information systems. It is a generic framework. It can be used to classify the descriptive representations of anything and therefore to analyze anything relative to its architectural composition. It is recursive. It can be used to analyze the architectural composition of itself. The analyst knows the analytical target and establishes the boundaries of the analysis and defines the architecture model. With the business rules management systems we initially want to give the control to the business analyst and this framework can help to establish the solid ground for this to happen. In a nutshell this framework is a 'logical tool of thinking' and offers a simple way of communicating. Furthermore it covers every aspect of the Enterprise context and it is independent of the tools and methodologies used for achieving targeted results.

In this information age the enterprises are becoming increasingly complex and dynamic, this leads to change on continuous bases. Thus according to Zachman Enterprise Architecture is the determinant of survival in the information age. There is still a lot to learn about the enterprise architecture and this framework is a good starting point.

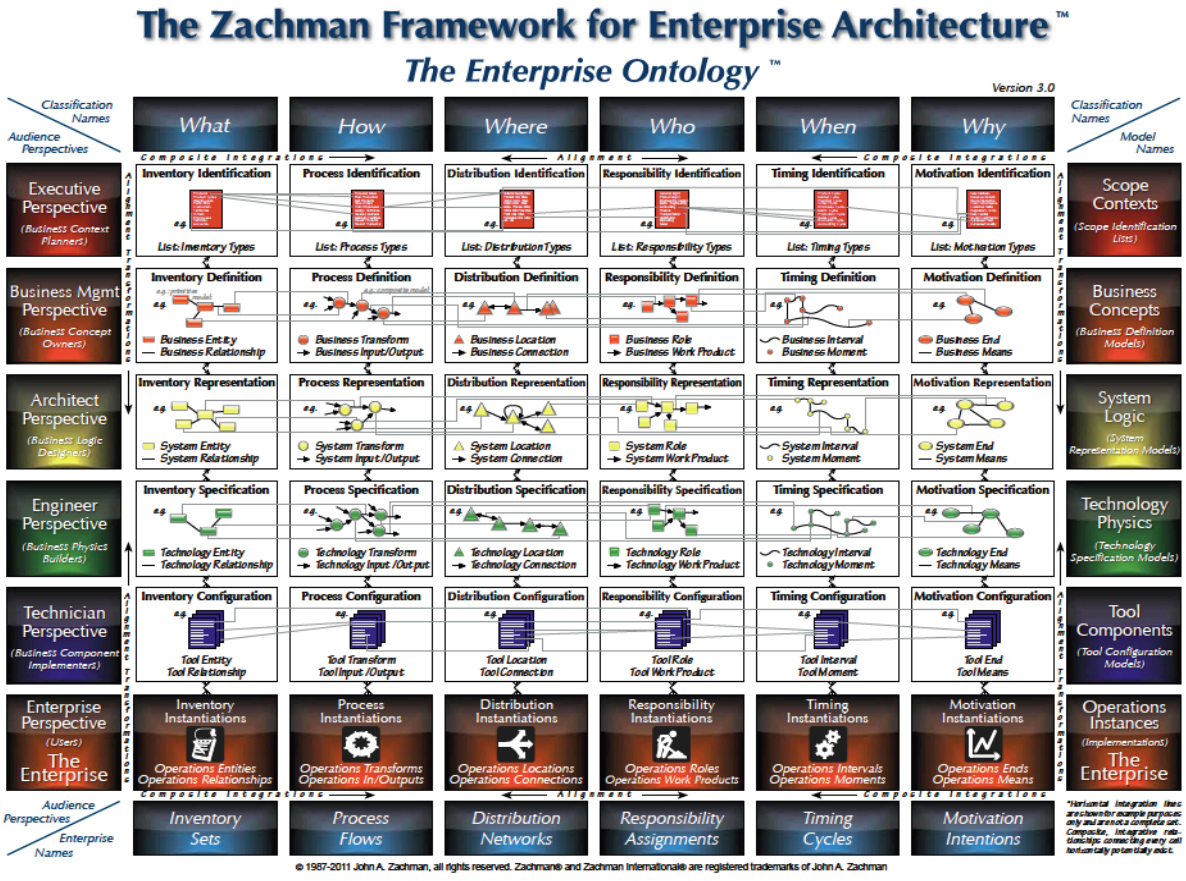

Figure 3.1 The John Zachman Framework for Enterprise Architecture

3.3 Decision model versus Decision Table

In this subsection a comparison is done on the Decision Table theory versus Decision Model theory and lastly the conclusion which approach is best.

- **Decision Table**

A decision table is a tabular representation used to describe and analyze decision situations, where the state of a number of conditions determines the execution of a set of actions.

Not just any representation, however, but one in which all-distinct situations are shown as columns in a table, such that every possible case is included in one and only one column.

Decision Table consist of three parts

- *Condition rows* lists conditions relevant to decisions
- *Action rows* that result from a given set of conditions
- *Rules* specify which actions are to be followed for a given set of conditions

Why use Decision Tables?

- Powerful visualization: Compact and structured presentation
- Both business & technical professionals understand a decision table if it is lacking technical artifacts.
- Prevention error is easy: Avoids incompleteness, inconsistency, and redundancy as it becomes visible in a decision table.
- Modular knowledge organization: Group related rules into single table and combine tables to achieve decision.
- Most Business Rule Engines (or Business Rule Management Systems) accept decision tables as a format for creating and making changes. That’s because automation of decision tables into such engines is fairly straightforward.

Decision tables have a simple and human-oriented structure. Therefore, they are an ideal medium for documentation and communication. Indeed, decision tables can be manipulated and understood very easily by the user. This gives the user some control over systems development and will increase the user's confidence in the resulting system. Decision tables offer a simple, visual aid, and they can be applied in knowledge-based systems to perform verification processes efficiently.

- **Limitations**

Although decision tables have many things, which are appreciating, but there are also some limitations. According to Barbara Von Halle & Larry Goldberg[22] below mentioned are some main shortcomings:

- Decision table in multiple formats add a level of complexity.
- Un-normalized decision table has redundant content and this lead to complex and error prone change management
- Lack of connecting tables, single unconnected decision table even when it may be possible to connect them together, this might happen when not starting with the higher business decision to support connected set of business logic structure
- Multiple actions leads to complexity
- Possibility of duplication with the same sub-action on different paths
- No differentiation of decisions and actions in Business process modeling

- **Decision Model**

Decision Model is in detailed discusses in chapter 4 and chapter 5. In this subsection will provide a brief description in comparison to decision tables.

Decision Model is always normalized; as in this way you can group logic statements in consist manners. Decision model is a complete model and not about single row or columns in a table. Decision model diagram is a representation of business decision and associated business motivation and metrics. Everything in it is connected, thus there is no stand-alone elements without reasoning.

Development of decision models start with logical business decisions and not with the table. It's more of high- level top down structural approach as compared to decision tables. This approach provides more focus on the business importance than on decision-making itself. Thus business decisions, single decision tables and even single rule family table out of model context are no longer enough anymore.

For the business side, decisions are more important and get more management attention than do decision tables. Decision models become a target for improved business thinking.

On the technology side, innovations for decision model-driven software become possible. These include advanced functionality, such as comparing whole models, validating whole models, creating different decision model views of the same business decision, and the sharing of whole decision models. There is much to expect in the near future!

- **Conclusion**

According to Jan Vanthienen of the University of Leuven view on Decision Modeling Notation[23] business processes need flexibility, meaning you cannot hard code rules and decisions in your processes as decision might change but not the processes. Rule Tasks in BPMN 2 for instance can manage how a decision should be made. There is a good case for an obvious separation of concerns with decisions and processes managed separately and collaborating to deliver business results. A decision-modeling notation would help manage this approach.

Decision tables give a good view of our decision-making approach, they are compact, can be verified and validated, and support a modular approach. Of course there is also a lot of work done on how to execute decision tables. In a decision table we separate conditions and actions and lay them out in a tabular form.

Tables must be normalized in the same way that data must be so that actions are dependent on the conditions, the whole set of conditions and nothing but those conditions. A complex problem does not require a single, simple decision table but a dependent network of decisions. Jan calls this a "decision goal network", showing which sub decisions are required to make a new decision.

For decision tables, there are a variety of methods for developing them. So, I believe combining the decision table a decision model to create a flexible, agile and efficient system there is a lot of work needs to be done. James Taylor is currently working on such a system but it's too early to say about the expected results and impact. As Barbara Von Halle says the transition from single decision tables to decision models is a fundamental leap in maturity. This leap has fascinating implications for business and technology.

3.4 BRMS Tools Analysis

There are many business rule management system development products in the markets, which allow users to develop rule-based systems. For the practical implementation of the business rules I will be doing an actual case study of a company who is looking for to automate it's salary system. Further details on the case will be discussed in chapter 6.

In order to choose the right tool I will be discussing few BRMS packages, which are available in the market today. Base on the analysis done in this section on various BRMS products a list of most common features is put together, this list might have missed an important feature but could be used as a starting point.

In general for choosing the BRMS package, it should have following features;

- Business analysts should be able to create and modify the rules;
- Uses a fully-featured repository;
- Should support backward chaining;
- Should allow the rule engine to be a component or service within larger applications;
- Should allow applications to be deployed in a service oriented architecture;
- Should focus on business rules management problems;
- Should provide good report generation facilities;
- Should provide evidence of successful commercial applications;
- Should be compatible with a component-based or service-oriented architecture; and offer commercial-standard professional support;
- Should be rete-based;

Firstly I will discuss briefly all chosen products and later conclude with what made me choose the package for practical implementation part of the case study.

3.4.1 Open Rules[24]

OpenRules is a Business Decision Management System (BDMS) available as an Open Source product. It allows subject matter experts and software developers to create, test, execute, and maintain enterprise-class decision support applications. OpenRules allows business analysts and not the programmers in control of the complex business logic of the mission critial applications. It is possible to configure your own BDMS that fits required specific needs in the best possible way.

Basically various elements are defined in tables within Excel, each labeled to define what it is, and the engine then picks up the Excel specification and executes it. OpenRules can use Excel for everything but also allows the rules to be stored in a physical rule repository and can be passed in to the execution environment as objects. Thus we can conclude that OpenRules doen NOT requires proprietary like as following:

- Excel like UI instead MS excel is directly used as rules editing environment
- No rule language,it’s built on Java

- **Architecture[25]**

Service Oriented Architecture makes it possible to integrate business process management system with business rules management systems. Figure 3.2 shows how SOA is designed in OpenRules.

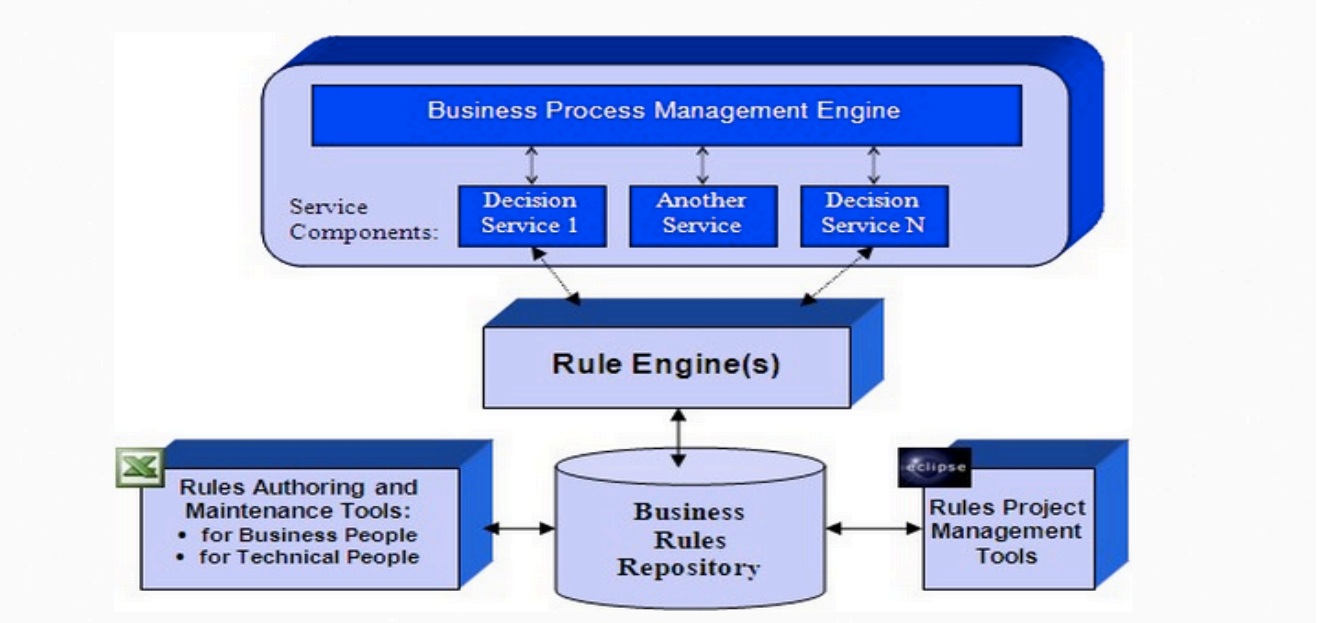

Figure 3.2 Service Oriented Architecture with OpenRules

OpenRules is basically designed in such away that without being an expert one should be able to develop web applications. Secondly as most of us are so much use to MS Excel and OpenRules allow being in their comfort zone and focusing on the actual development rather than getting familiar with the new development tools.

- **Web Applications with OpenRules**

Along with basic rule services incorporated in any Java application, it allows you to create two major types of Web applications 1-Web Services (business logic) & Rule Forms (business and presentation logic)

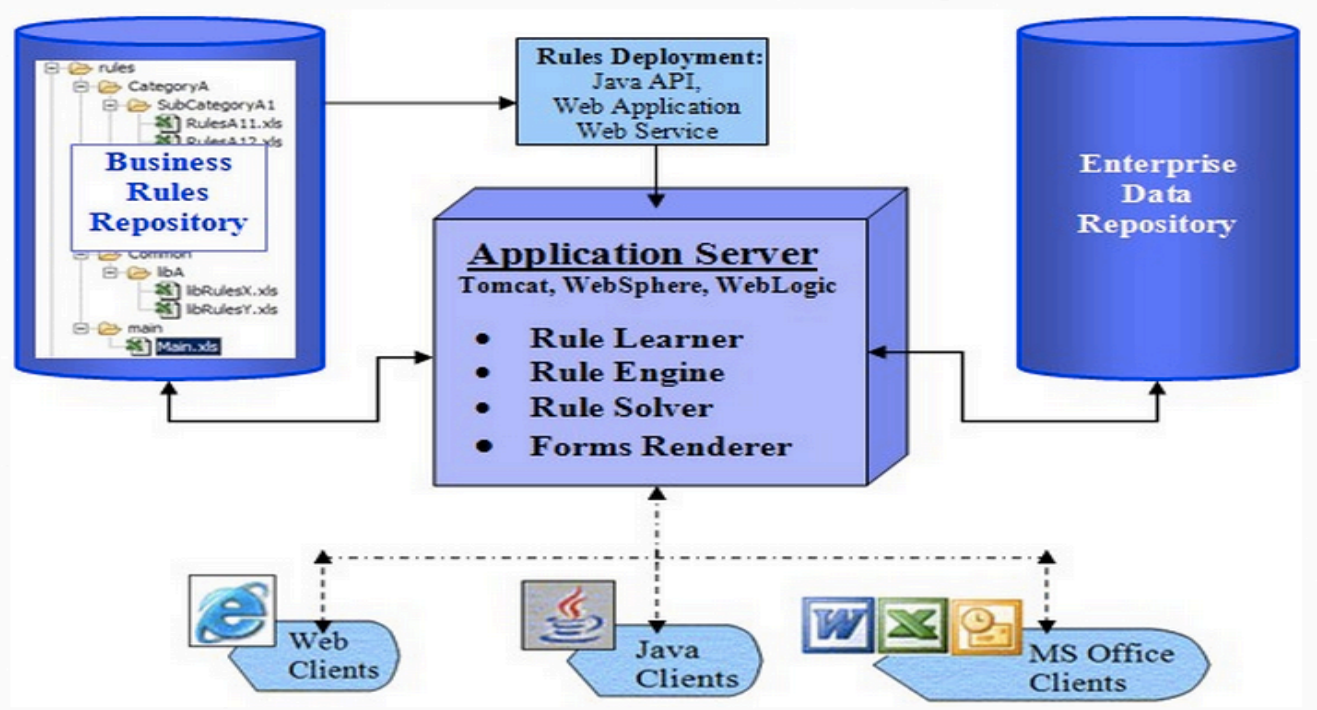

Figure 3.3 Major component of Web application with OpenRules

As mentioned earlier without being expert, OpenRules allows you to develop complex Web applications using the commonly known Excel interface, you can define your business logic in a form of Excel-based business rules and make them available as fine-grained Web services. Then, you may define your presentation logic using Excel-based web with associated interaction rules. Being deployed as a presentation-oriented Web application, it will invoke the related rule services whenever necessary. Frequently, such a service-oriented approach to web application development also involves a workflow engine that uses a publish/subscribe paradigm for automatic invocation of different web services.

Service-oriented Web Applications: With OpenRules one can define and maintain business rules using a combination of Excel, Java, and Eclipse. Next test your rules as a stand-alone application or execute them locally using a simple Java API.

Presentation-oriented Web Applications: A presentation-oriented Web application generates dynamic Web pages containing various types of markup language (HTML, XML…) in response to requests.

Web applications created with OpenRules can be deployed on any Java Application Server such as WebSphere, WebLogic or ApacheTomcat. OpenRules provides a push-button solution for cumbersome web application configuration and deployment tasks.

- **Logical and Physical Rule Repositories**

OpenRules utilizes the well-established spreadsheet concepts of workbooks, worksheets, and tables to build enterprise-level rule repositories. You can create and save the business rules in MS Excel, Google docs or OpenOffice.

Figure 3.4 & 3.5 how excel sheets can be managed with OpenRules application for authoring, maintanace and repository.

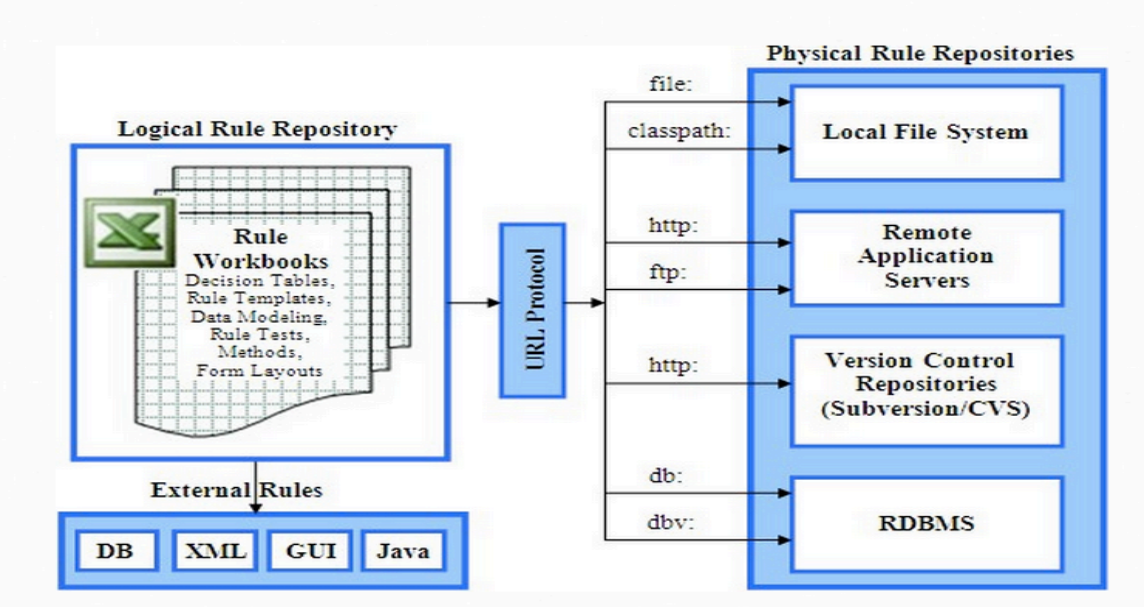

Figure 3.4 Logical and physical organization of OpenRules repositories

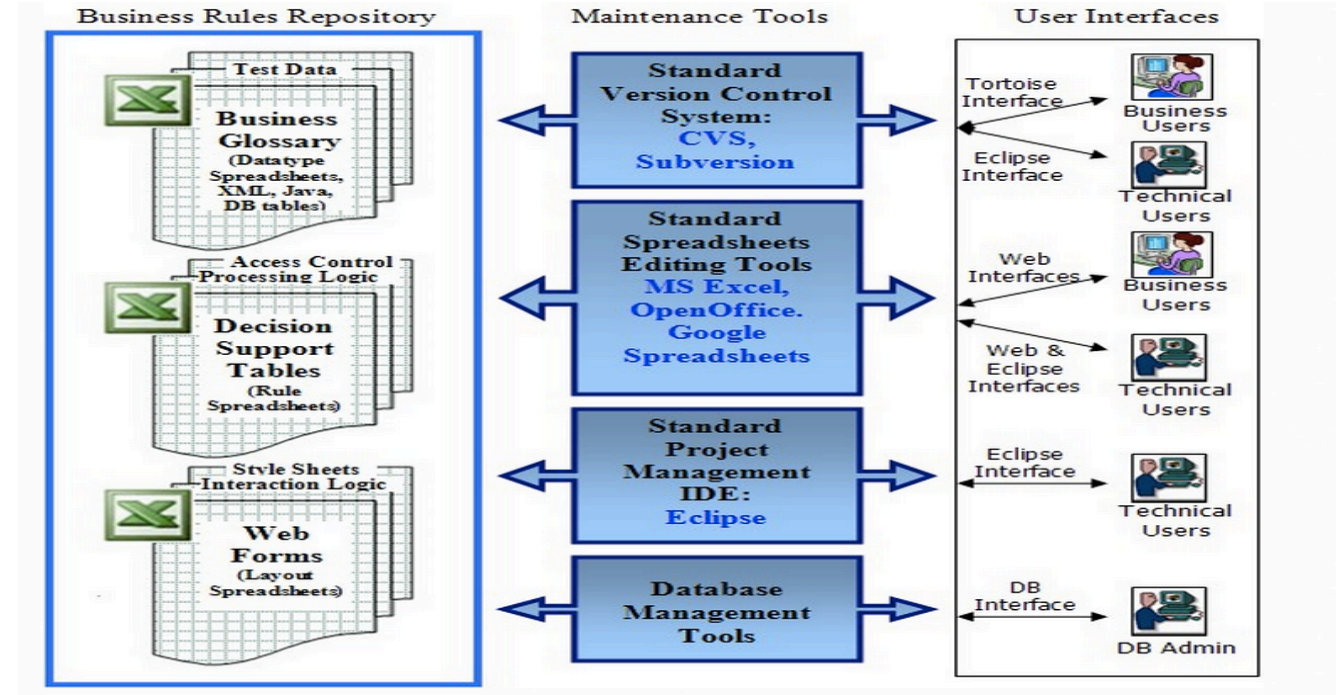

Figure 3.5 Rule authoring and maintanace tool

3.4.2 IBM Operational Decision Manager

IBM Operational Decision Manager (ODM)[26] is launched in December 2011. Prior to this ILOG and Websphere business event were available as separate product and they are combined in ODM. An overview can be seen of this application in figure 3.6.

IBM Operational Decision Manager integrates business events and business rules to automate decisions across processes and applications. IBM Operational Decision Manager built on the success of ILOG JRules consists of two component sets, which together form a complete platform for managing and executing business rules and business events.

- **IBM Decision Center**

It provides an integrated repository and management components, allowing you to govern business rules and event-based decision logic. Delivers a configurable, non-technical rule language and capabilities, allowing you to maintain decision logic.

Provides comprehensive decision governance including role-based security, custom metadata, multi-branch release management, testing and simulation, and historical reporting. Improves communication and collaboration between IT and business functions through a centralized decision management repository and shared governance controls. Accelerates the decision process and maintenance lifecycle, and evolves based on new external and internal requirements.

- **IBM Decision Server**

It provides the runtime components to automate decision logic, enabling the detection of business situations and precise response based on the context of the interaction.
IBM Decision Server allows you to monitor business networks to discover events and take action based on event data patterns. Processes event data against hundreds or thousands of business rules, to determine responses, within front-end and back-end systems. Delivers high-performance runtimes to support critical decision services for both event-driven and service-oriented architectures.
Provides a unified application development environment using Eclipse-based tools.
IBM ODM It allows the creation, management, testing and governance of business rules and events and stores them in a central repository where they can be accessed by multiple individuals and software products. This central storage of the rules and events mean that they can be easily modified without having to rebuild software, and with a reduced testing cycle, and the different software products will pick up this change simultaneously.
IBM Operational Decision Management (ODM) is the exciting evolution of the traditional Business Rule Management System (BRMS). Line-of-business and IT users can work together to automate business decisions, policies, practices and regulations. The product's intuitive, elegant interface enables users to write and edit business rules in natural language, so that both business and IT people can read, understand and modify the rules. Business rules can be easily managed over time in a solution that provides best-in-class scalability, reliability and governance.

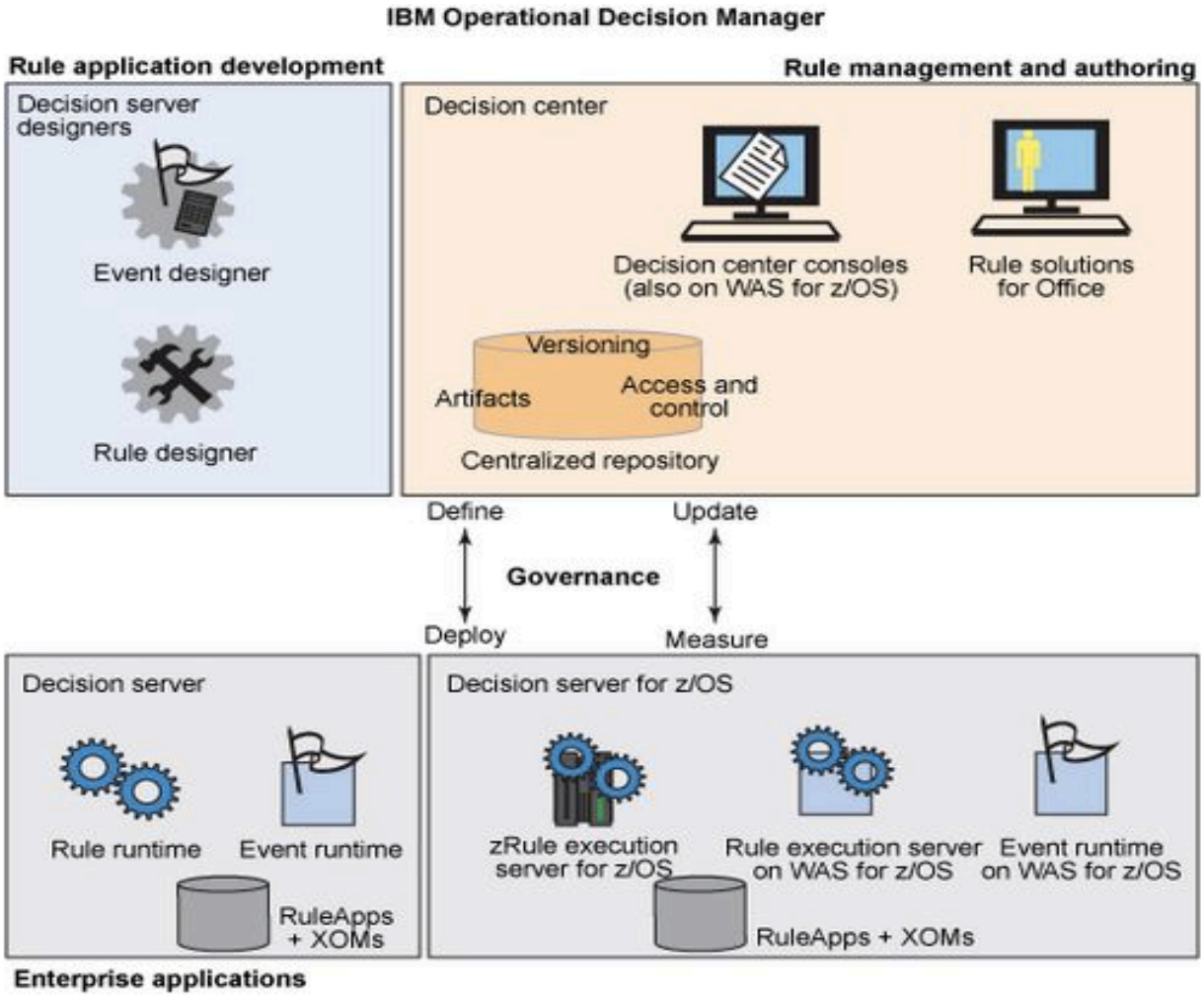

Figure 3.6: IBM Operational Decision Manager Architecture

3.4.3 FICO Blaze Advisor[27]

The role of Blaze Advisor[28] is to provide organizations with a single location from which to create, manage, and automatically execute the rules that lie behind business decision making, in order to provide consistency and the ability to rapidly manage rule change in a structured manner. These objectives are enabled through:

- A high-performance runtime environment with a choice of decision algorithms.
- The ability to create rules and rule sets to resolve complex decisions, while presenting the logic in an intuitive manner that can be understood and modified by business or IT staff.
- Deployment across a heterogeneous environment, including deployment to client-side Web forms.

- Direct integration of analytics through scorecards and other visualization mechanisms.
- Testing and change management capabilities.

Although most effectively deployed at the enterprise level, the same features make Blaze Advisor appropriate to tactical requirements where it is required to manage complex decisions in discrete projects.

This product has interactive testing facilities, making backward chaining far easier to realize, good rule analysis features and its general level of tool integration and ease of use.

A less technically inclined business analyst might have struggled more with the design interface. Therefore, Blaze Advisor is not really suitable for environments where business users need to get closely involved in the creation and maintenance of business logic unless there are development resources available to create a custom rule interface. The rules can then be presented to users in a non-technical way via generated browser pages.

Blaze Advisor is a mature product with good integration features and is capable of addressing many BRMS situations i.e.t he performance and scalability to function in a high transaction-volume environment. The central rules repository and management and execution capability provide the backbone for an enterprise-wide rules management platform that can add value to both BPM and SOA. Although it is common to view BPMSs from a technical perspective, management of Blaze Advisor believes that the focus should be on establishing the value of decision management and its role in business strategy execution. While Blaze Advisor ticks all the technology boxes (rules design, deployment, change management, automation), its business value lies in its ability to cater for both business and IT groups. It provides a range of tools that bring the two worlds closer together, plus interfaces that provide the business operation with the ability to own and manage their own rule content. This means changes can be made in a timely fashion by those who understand the nuances of the business environment, allowing closer alignment between technology and business, to the benefit of decision making. See Figure 3.7 for an overview of the architecture of Blaze Advisor.

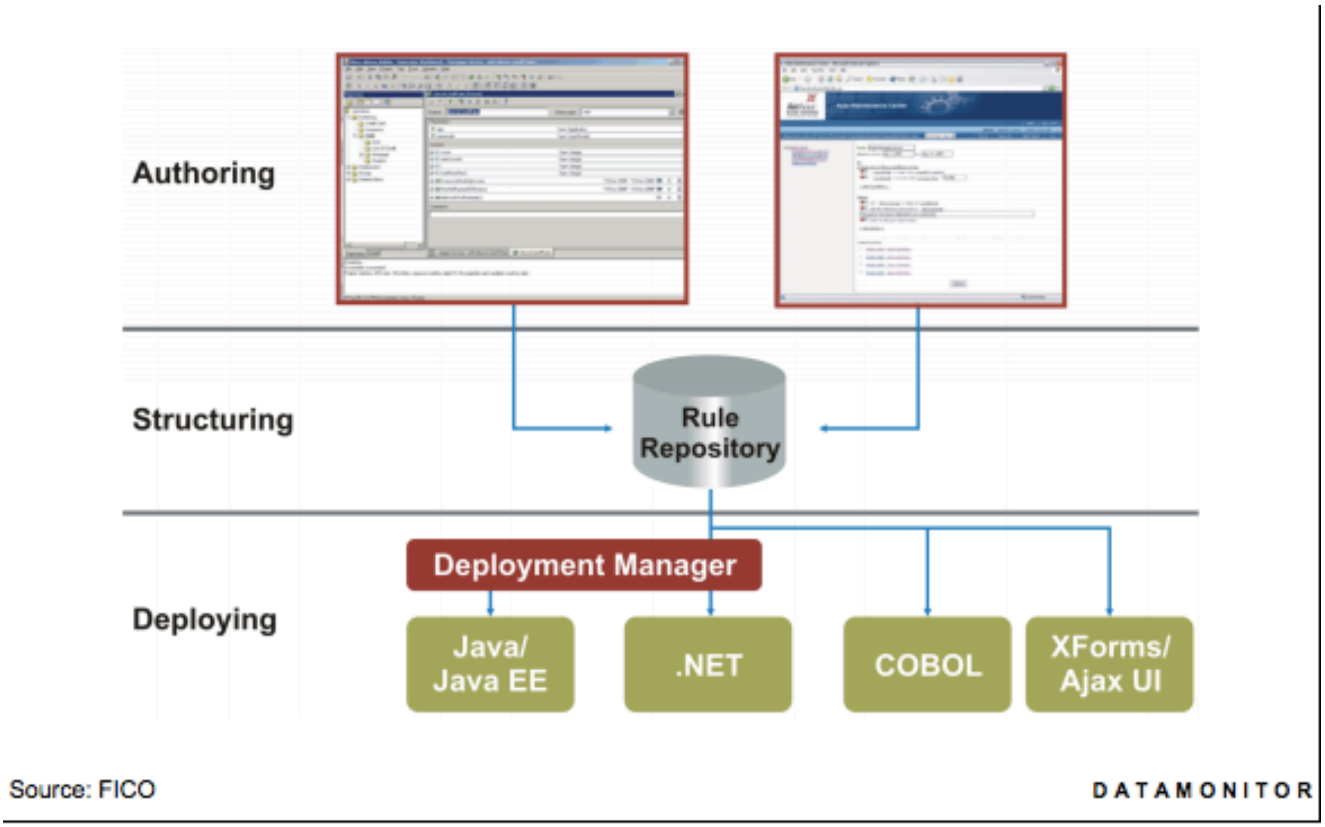

Figure 3.7: Blaze Advisor Architecture

3.4.4 Drools Guvnor[29]

Guvnor[30] is the web and network related components for managing rules with drools. This combined with the core drools engine and other tools forms the business rules manager.

One should consider Guvnor if any of the following apply:

- Need to manage versions/deployment of rules;
- Need to let multiple users of different skill levels access and edit rules;
- Don't have any existing infrastructure to manage rules;
- Have lots of "business" rules (as opposed to technical rules as part of an application);

Guvnor can be used on its own, or with the IDE tooling (often both together).

Guvnor can be "branded" and made part of your application, or it can be a central rule repository. The main roles of people who would use Guvnor are: Business Analyst, Rule expert, Developer, Administrators (rule administrators etc.).Guvnor is designed in such a way as these different roles can be accommodated, it can be controlled how much is exposed to different users in a safe fashion. See figure 3.8 for architectural overview of Drools Guvnor.

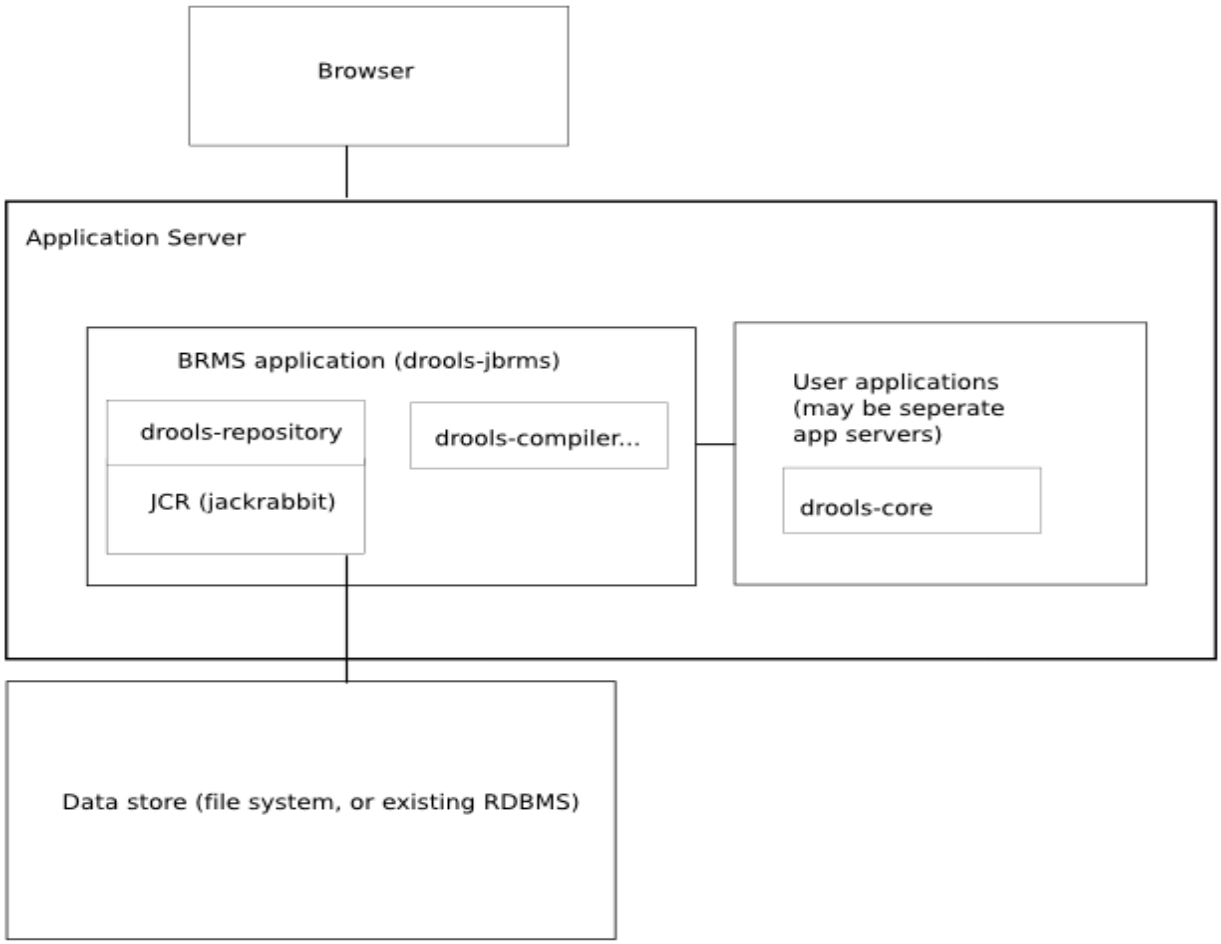

Figure 3.8: Drools Guvnor Architecture

- **Product Comparison**

As briefly described some of the currently available products in the market today. In table below all four chosen products are compared to make a choice for the practical implementation of case study, which will be explained in chapter 6 in details. An effort was made to summarize all four products in a single view, please note that it may be that not all fields are properly filled because of the fact the information was not available. Besides the comparison on the important elements of each of these product (see table 3.1) also following factors were also taken into account for making the decision of application choice;

- Ease of use and less in-depth knowledge required
- Cost of implementation
- Customer support on product related issues
- Based on the product comparison table and also on the other factor Open Rules was chosen. It's quite amazing being and open source product, the detailed information and customer support is just unbelievable.

	BRMS Products			
Evaluation Criteria	**Open Rules**	**IBM Operational Decision Manager**	**FICO Blaze Advisor**	**JBoss Drools Guvnor**
Software Platform	Java-native .NET via website	Java	Java, .NET and COBOL	Java
Facilitate for rules testing	Yes	Yes	Yes	Yes
Level of language of rules	Low	Low	High	High
Type of target organization	All	All	Large	All
Coding of rules required	No	Yes	Yes	Yes
Business analysts control business logic	Yes	Yes	-	Yes
Rules based web application	Yes	Yes	Yes	Yes
On Cloud	Yes	-	-	Not applicable
Forward/Backward chaining	Backward & Forward chaining	Forward chaining	Backward chaining	Forward & some Backward chaining
Uses Rete-base or constraint base rules engine	Rete & constraint base	Rete-Base	Rete-Base	Rete-Base
Graphical representation of rules	Yes	Yes	Yes	Yes
Class of rules base product (see appendix)	B	C	C	B

Table 3.1: Business Rules Management Systems a Product Comparison

SECTION III: IMPORTANCE OF DECISION MODEL

4 Decision Management Systems

In this section I will be reviewing the decision management systems from a practical point of view i.e. how the decision management systems can form the organization and change the way how they use to work in the old days. For reference James Taylor's book[31] on Decision management systems is used as main reference for this section.

4.1 What Are Decision Management Systems?

It is now a days common business practice for every organization to have various types of data bases, information systems, knowledge bases, reporting tools, analytic tools and so on. All these data bases, tools, applications, information systems perform various types of tasks expected but they all have limitations and most of the time the human mind is needed to make a sensible conclusion or decision based on the information extracted from all these mediums. In other words they are not intelligent enough to take over not all but partially the human responsibility of intelligent decisions making.

The question, which arises, is what are decision management systems and what is different about them. Decision management systems are intelligently designed and developed systems, which help organizations to take timely decisions to coup with the business needs.

Decision management systems can be differentiated based on following features.

- **Agile**

"Easier, quicker and cheaper to change in response to changing needs" This can be managed either by business process management or by decision management system developed based on Service Oriented Architecture (SOA). This results in end-to-end visibility i.e. clarity, accuracy and consistency. Agility focuses on improving decision making by capturing rapidly changing know how.

- **Analytic**

The decision management systems should have the ability to use the data analytically on behave of the users. More specifically manage risks by predictive analytics, reducing frauds, targeting and retaining for maximizing opportunities and focusing resources on right problem, maximizing their value. Analytic capability focuses on improving decision-making by using data to make precise and targeted decision.

- **Adaptive**

It helps organizations to test new approaches, learn about them and when to implement them. To manage more complex decisions, more objectives needs to be considered and more date on what might work so, its essentials to manage complex tradeoffs. The adaptively of the decision management system can manage this by expressing strategic objectives in terms of business rules and putting it into the system, which can determine feasible revenue management actions given all available tradeoffs.

Decision management systems are becoming powerful tools in the business environment. It can determine how organizations behave in certain situation and how it has improved the business process.

The need for decision management system is driven by many factors notable as following:

- **Changing Expectations**

Real time response has become the expected norm by the consumers. For certain type of services 24/7 response seems more realistic with systems. Global customers expect global services. Lastly self-service meaning directly interacting with the organization's system or with the user interface layered on top of the systems.

- **Changing Scale**

With the increase in the volumes of transaction have changes the way organizations used to operate. High transaction volumes must be matched to precision and it is possible with the new generation of the information systems. Data has become so big that only information systems can deal with it. Meaning not only store and manage but also analyze the data and act on it. Efficiency can be achieved i.e. more with less, fewer human interaction to process transactions. The problem is that most organizations are not willing to give up the customer intimacy or localization.

- **Changing Interaction**

The way people interact is changing rapidly. This only not includes various types of mobile devices in use but also the way people communicate i.e. using the abbreviations and gestures. It would mean that new information systems should be able to behave intelligently to mobile interaction, new ways of social interaction than human interaction, distributed interaction (e.g. customers search for cheapest product through online makes difficult for companies to use predictive analysis thus more sophisticated analysis needed) and lastly without information systems that link web of companies together, the consumer interaction will not be optimal if the personnel has not appropriate complete information at hand .

Decision management systems are very powerful tool if used optimally, the have the ability to transform the organizations. According to James Taylor there are variety of reasons, why companies are transforming:

- A market for one: For example decision management system powered by analysis makes it possible to fulfill the marketing dream of mass customization & customer centricity. By doing an intelligent analysis from customer's perspective will allow companies to provide him/her with the options of his/her interest as default e.g. Amazon's product related recommendation. Secondly this would boost one-o-one relationship based on trust and high personalization for the customers. Thirdly help to sell personalized experience like Starbucks. Thus decision management systems can deliver content aware decisions for the right person at the right place and at right time.
- Always on: With always on decision management systems it would not matter anymore how many layers of service providers were involved, they would all act as the way the brand owner wanted them to
- Breaking the ratios: Decision management systems make sure that the department resources are always focused, where they can make the difference.
- Crushing fraud: Distant outliers or match known pattern of fraud with decision management systems
- Maximizing assets: Any kind of assets value can be maximized using decision management system i.e. telecommunication networks, energy grids, factories and even airport gates.
- Maximizing revenue: The user expresses strategic objects is term of business rule's and the system determines the feasible set of revenue management actions.

- Making smart people smarter

Concluding remark decision management system should be critical element in an organization's information system portfolio.There are four principles, which addresses the characteristic capability of a decision management system.

1. Begin with a decision in mind

We are neither willing nor able to build information systems to make every decision on our behalf, but certain decision can & should be addresses by decision management system. What kind of decisions are we talking about?

- Repeatable decisions
  - Strategic decisions are not candidates for decision management systems (high risk, high value, low volume)
  - Tactical decisions are focused on management & control. They involve data & analysis so repeatable tactical decisions are candidates for decision management system (medium value, medium risk)
  - Operational decisions are most common subject for decision management system (low risk, high volume, low individual value)

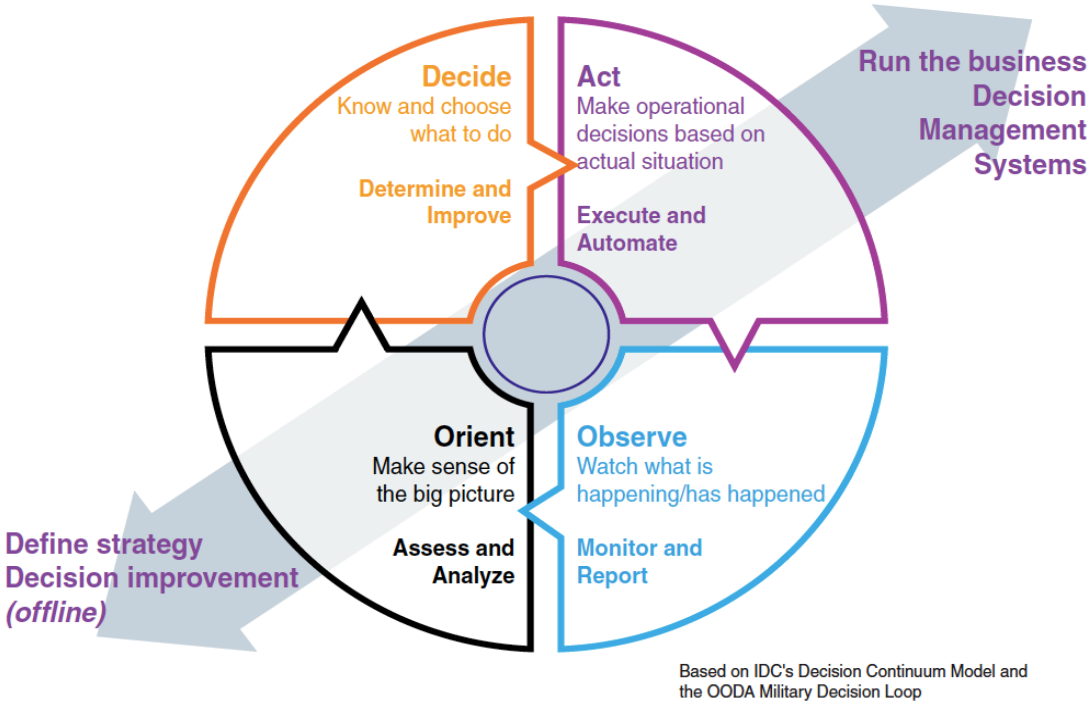

Figure 4.1:The Decision Lifecycle from strategy definition to decision automation

Figure 4.1 is a very good way to understand the relationship between strategic, tactical and operational decision making.

In order to be clear about the decision management systems and not to mix up with the decision support systems Table 4.1 distinguish between these two types of systems

	Decision Support Systems	Decision Management Systems
1	Provide information that describes the situation & perhaps **historical trends** so that humans can decide what to do & which actions to take	**Automate or recommend** the actions that should be taken based on the information that is available at the time the decision is being made
2	The policies, regulations and best practices requires the users to **remember** them or look them up separately	The policies, regulations and best practices are that determines the best actions are **embedded** in the decision management system
3	The information and insight presented is backward looking & generally **reactive**	Uses information to make predictions & aim to be **proactive**
4	Learning happens **outside** meaning users are expected to learn what works & what does not work & to apply what they learn to future decisions	Learning happens **inside,** decision management systems have experimentation or test and learn infrastructure built in so that the system itself learns what works & what not
5	Decision support systems are often desktop or **interactive** applications that execute outside the core application portfolio	Decision management systems are **integrated** into an organization's runtime environment. The make decisions for applications & service in organization enterprise application architecture

Table 4.1: Comparison between of Decision Support & Management Systems

Thus what makes decision management systems different than other systems is the focus on function, process and data.

2-Be transparent & agile

Most information systems in use today are unclear & hard to change. The design must be transparent so that it is clear that the system is executing the behavior expected of it. The execution must be transparent so that it is clear how each decision was made. System must be agile so that their behavior can be changed when necessary without delay and without unnecessary expense.

3- Be predictive, not reactive

There are 3 specific ways in which decision management systems cane be given prediction. Firstly predict risk or fraud, secondly predict opportunity and lastly predict

impact of decision. The prediction of the likely impact of each action can be combined with prediction of risk & opportunity to improve quality of decision-making in decision management systems.

4- Test, learn & continuously improve

The constant change in the definition of an effective decision means that decision management systems must optimize their behavior over time, continuously refining and improving how they act.

Collect and use information to improve, when a decision management systems makes a decision it should record what decision is made as well as how and why it made it.

Simulation and modeling of best approaches, and testing them against historical data might show which approach the superior than all. This process does not stop by just finding the best approach need to be continuously repeated over time to improve the decision management approach.

4.2 How to build decision management system?

We know now why decision management systems are so important in today's business environment but the question is how we can build these decision management systems.

Below is the list of characteristics of suitable decisions, which are the core of the system.

- **Repeatable**

You can recognize a repeatable decision by the decision is made at defined times, the same information is used each time, defined set of actions, and consistent measure of success.

- **Nontrivial**

Organizations see ROI if the decision being automated has degree of complexity drivers for complexity are policies & regulations, need for domain knowledge, need to analyze large amount of data, need to select from various numbers of outcomes, need to trade off competing objectives. Nontrivial need to pass all these tasks very complex regulation or very difficult trade offs might be enough.

- **A decision taxonomy**

The most common categories are eligibility validation & calculation decisions, risk, fraud & opportunity decisions are less common but still part of most organization's decision inventory. Repeatable management decision, micro decisions.

- **Finding decisions**

There are several ways to find suitable decisions among those below are few approached

i) Top down, process centric: four approaches can be used separately or on combination to fit the decision in a process

1 Looking for explicit decision point (determine, validate, calculate, assess, Select and choose)
2 Multiple similar processes
3 Local expectations
4 Escalation and referral

ii) Event centric: Three approaches can be used separately or in combination to find the decisions in event bases systems-looking for places where event & non –event date are presented together, event streams that are coming under increasingly precise scrutiny, and examining a finite state analysis of the entire system.

iii) Analyze legacy system

Find the high change components by application portfolio analysis, change request or project log. Look for table drive code, which implement decisions.

- **Using Metrics and performance management systems**

Understanding the key decisions that are responsible for the business outcomes being monitored with KPIs and other metrics helps differentiate good decision from the bad decisions and shows which decisions to focus on when metrics or KPIs are out of acceptable bounds. Secondly map decisions and KPIs to improve the effectiveness of decision choices. Lastly analyzing past KPI based data, root cause analysis of what if analysis can also help to identify decisions.

- **Documenting Decisions**

Find answers to the questions and link them to business. Model the decision dependencies with the help of following steps;

1-Determine dependencies

2-Find dependencies

3-Find information source

4-Find know-how

5-Iterate

- **Decision Characteristics**

Some important characteristics of decision management system include, high volume repeatable decisions, timelines (how often it occurs), consistency over time, value range, Time to value and degree of freedom).

- **Prioritizing Decisions**

The enterprise-level purpose of the project is a critical driver for prioritization. Use this to identify the measures and KPIs that most closely align to the project.
Besides asking following questions could help to prioritize best decisions.
- How measurable is the decisions impact on the measures it affects?
- How big is the difference between good and bad results in terms of revenue, risk or royalty?
- How often do you make the decision?
- How much spending is committed as a result of the decision?
- How much does it cost to make a decision?
- How hard will it be to develop a decision management system for the decision?

- **Design and implement Decision Service**

Once decisions are identified and prioritized next step is to design and implement decision services. A decision service is at the core of the decision management system. Highest-level decisions should be used as decision service (i.e. decision required by process, an event or could replace a module in legacy system). Lower level decision can be used to define decision service but cannot be turned into decision service.

Following needs to be considered to build Decision Service:

- Core purpose of implementing decision service is as business rules, predictive analytical model and optimizing model. To make this all this together scaffolding is needed.
- Define Service contract, given certain set of information, possible answers will be answered by Decision Service along with supporting information. Furthermore it defines input required and output produced by the decision service. Defining test case then next step to verify the expected results.

- Turn dependency network into decision flows by defining decision-making tasks linked together with conditional branches if necessary. Decision flow is a design element thus this is the time to define which steps should go in which order.
- Build rest and learn infrastructure by examining the decision flow and identify where multiple possible approached could be applied. Important to consider here is keep track of decision-making approach for each transaction.
- Next step is to build the business rules in a decision service and it requires 3 main steps: determining the source rules(natural language statements), defining executable rules and integrating the executable rules as part of decision service.
- Implement the business rules with supported business rules management systems. There are five high level steps of developing executable rules; determine objects, selecting business rules representations, writing the rules themselves, linking to source rules, and validating & verify the rules.
- Continuous integration and deployment process ensures that every new, updated rules set works and can be used. Secondly immediately new of changed rules will be available in decision service to run in the environment.
- Build predictive analytical model by exploring and understanding data might be predictive, preparing this data, applying various mathematic analysis techniques, building and testing the model and lastly deploying it.
- Build optimization models and this can be done with following three main use cases; Optimization within a single decision, Optimization across many decisions and Optimization of future decision- making approaches.

➢ Integrate Decision Services

Once decision services are built they need to integrate with the rest of the information technology model. Following elements needs to consider while integrating the decision services; Data integration, Process integration, Event integration and Case management integration.

➢ **Monitor and Improve decisions**

After Decision Service is deployed it entered into new phase of decision analysis to monitor and improve systematically decisions. We need to do this because decision service depends on following:

- Changes to the business goals
- New regulations of policies
- Changes to the underlying data patterns

- Overall decision performance

➢ Monitoring decision analysis

Monitoring decisions requires focus on reacting to changes in business goals and metrics, new regulations & policies, changes to the underlying data, proactive changes, capture decision effectiveness data (i.e. Decision Service execution & response data and enterprise data), building decision monitoring environment and linking performance management and decision management.

➢ Determine the appropriate response

In order to determine what change is needed, the specific change drivers needs to be analyzed.

- Design impact analysis involves finding the artifacts (regulations, data, policies) that are changing and what other element in the decision service they are impacting.
- Assess decision effectiveness in terms of current performance and likely performance if the decision-making approach is not changed. It is primarily assessed by the impact of the decision made on business metrics and KPIs associated with the decision.
- Compare the existing implemented approaches i.e. A/B testing (If the current decision- making approach has no particular importance then no need to consider other approaches, A/B testing is applied) or champion-challenger testing (When the current approach cannot lightly be changes or when there is a clear case that one approach will probably be best then champion-challenger can be applied) to find out which approach is more effective for which segment etc. Determine whether multiple new approached are required
- Once the current decision performance has been determined, new business rules, new predictive analytic models and new optimization models must be designed and developed to implement the new decision making approach(es) needed.
- Keep predictive analytic model up to date by self-learning, model refreshing and champion-challenger.
- As the new decision-making approaches are identified, make sure to verify and validate the results before deploying. Confirming the impact as

expected can be performed by case testing, simulation, what of analysis and advance simulation. After the results are verified deploy the change.

4.3 What are enablers for decision management system?

During the development and deployment of the decision management system according to James Taylor there are three critical enablers (people enablers, process enablers and technology enablers) for the success of a decision manger system. In this section these enablers will be discussed briefly.

- **People Enablers**

Decision management systems like any other technology require people to use it and build it. Business people, IT people and analytic people are the building block of a 'Three legged stool' and play a critical role in the success of a decision management system.

They really need to work together and understand the value they can add together. So, far it's been a challenge for most organizations where these three groups work independently with completely different point of views. But this problem needs to be resolved and we need to bring them together for the success of the decision management system.

So how can we make these three groups work together?

- Get these people involve in starting a three way conversations. Meaning share your point of view with others, discuss, evaluate and incorporate changes as needed.
- Build collaboration skills by organizing meetings and make them work together, cross training and may be consider brown bag lunches.
- Decision discovery is critical so, it's very important that all three groups are on board to discover decisions.
- One of effective ways to get disparate groups to collaborate is to align their personal and organizational objectives, which will result in aligned business result and ultimately this is what we want to achieve with the decision management systems.
- Avoid any organizational and reporting issues and resolve existing organizational issues.
- Consider using a decision management Center of Excellence (CoE). CoE is not specific to decision management but for any strategic technology leveraging across the enterprise whether it is BPM, analytics, business rules

management, SOA or master data management. One kind of CoE will not suit all organizations so worth considering many potential dimensions (see Figure 4.2)

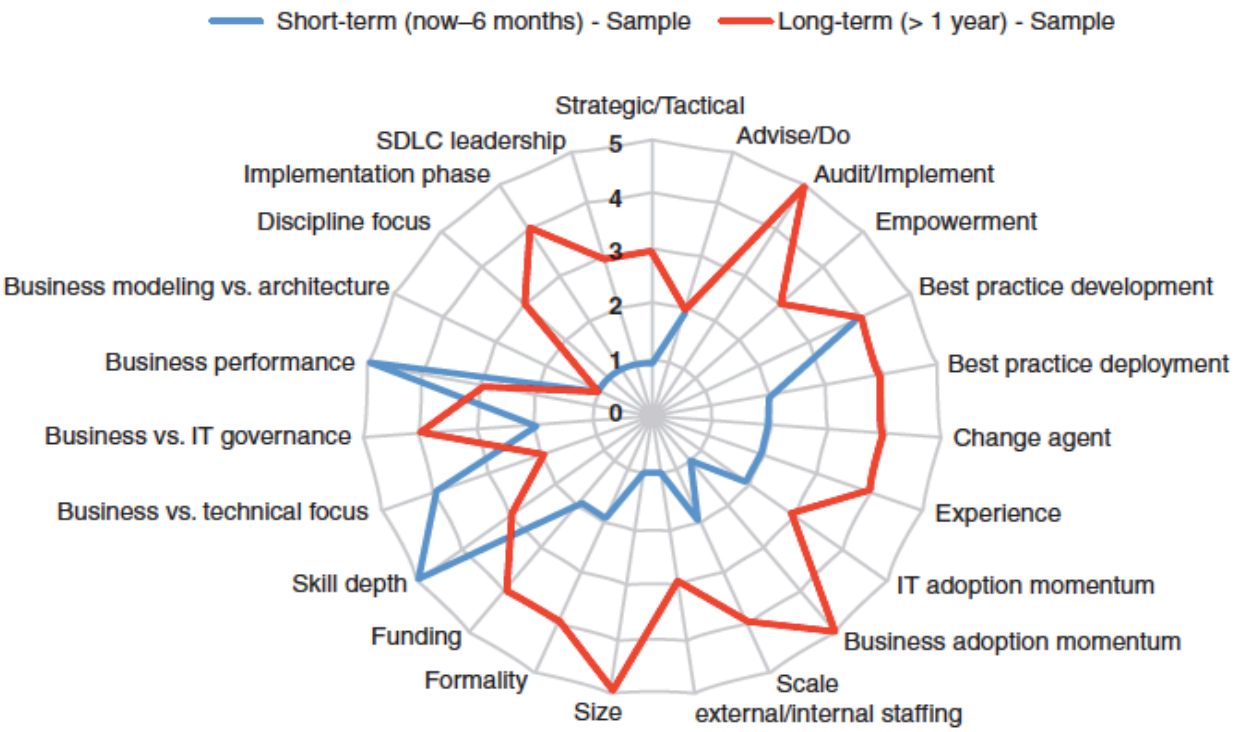

Figure 4.2: Short and long term CoE shapes

-Last but not least organizational change need to be taken seriously as with the implantation a lot things for example way of working, staff's responsibilities, change in control etc. might be effected. So, it's very critical to address this issue and invest time and money to smoothly go through the organizational change transition.

- **Process Enablers**

With any new initiatives it is useful to review existing processes and develop new one to support the decision management systems, a process for maintaining your decision inventory over time and a revised software life cycle are two key success processes.

Managing a decision inventory plays a crucial role in decision management system, as it requires up to date and accurate inventory.

Service decisions are typically deployed as part of business service layer in a Service-Oriented Architecture (SOA). When a Decision Service uses business rules management systems to support the implementation, the focus of the software development life cycle (SDLC) will be on rule harvesting, design, and implementation.

Therefore SDLC should include all the tasks, work products and guidelines to support new activities. Adopt the SDCL using an agile, incremental and iterative approach.

Decision service integration patterns can be broadly classified into three groups as following:

1- Tactical pattern are used to address an immediate business problem that has little strategic value. These patterns involve little or no analysis of reuse or common problem. Furthermore these are often one-off solutions that will have to be discarded or greatly modified as new business issue occurs and solution needs change.
2- Strategic pattern focus on long-term strategic objectives and take a longer view. Such pattern involves evolving a decision service so it can be used in multiple processes. Using decision service to continuously improve processes with root cause analysis and making business processes adaptive by embedding decision service are two other strategic patterns.
3- Incremental patterns bridge the two ends of spectrum by applying in a localized way and can be evolved through later analysis. Typical pattern include externalization and automating decisions embedded in existing or legacy applications and automating/assisting decisions in the workflow to improve.

- **A Culture of Experimentation**

Experimentation is an essential element and one of top technology enablers in decision management systems. It's important especially for operational/micro level decisions, as they are core of the decision management system and best candidate for experimentation.

IT experimentation is not what is currently embedded in the legacy systems nor IT to build such systems. With decision management systems it might be possible but IT needs to change it's mindset to achieve this.

From business perspective experimenting is a way to balance the short term to long term. Making test and learn normal approach business will be important but not easy one.

The controlled group approach where a group of selected users are treated following legacy approach and rest with the new approach. A control is formed ensuring that the old approach continued to be used as decision management system. To establish

the increased effectiveness of the system you need baseline; that's what control group can provide.

Experimentation is valuable and you need to carefully design the experiments. Following points needs to be considered while designing experiments; Comparison, randomization, replication, blocking, orthogonality and factorial experimentation.

- **Moving to Fact-Based Decision**

Fact based decision making requires a focus on the decision making approach as well as the results. It also creates environment necessary for experimentation to flourish, establishing the value of improving decision making approach and not just making good decisions right now. It also requires a broader statistical awareness and policy of presenting data and decisions as set. It does not dismiss the value of the expert but complement it and builds on a data strategy foundation.

- **The Observe, Orient, Decide, Act (OODA) Loop**

Observing both the detailed results and the overall behavior of the organization is essential. Any observation must be interpreted before they can be used. Observe-Orient represent control part of the loop. Thinking from observe-orient,, can result in new strategic and tactical objective. When applying the OODA loop to decision management systems, the Decide part can be mapped to the determination of appropriate decision making approach. Actually running the decision management system and determining which action to take in each situation is the Act part of the OODA loop.

- **Technology Enablers**

There are quite few numbers of technologies, which enable decision management systems. Following is the brief description of them all.

- **Business Rules Management System**

As being main research topic of my thesis you can find more details in chapter 1-3. I will only point out few details from James Taylor's perspective.

Business rules can be executed by BRMS is several different ways; sequentially, inferentially and designed approach. All three approached have pros and cons and many BRMS support more than one approach.

BRMS system requires business rules repository, design tools, rules management interfaces, validation and verification tools, testing tools, business simulation tools, deployment tools and high performance business rules engine (see Figure 4.3).

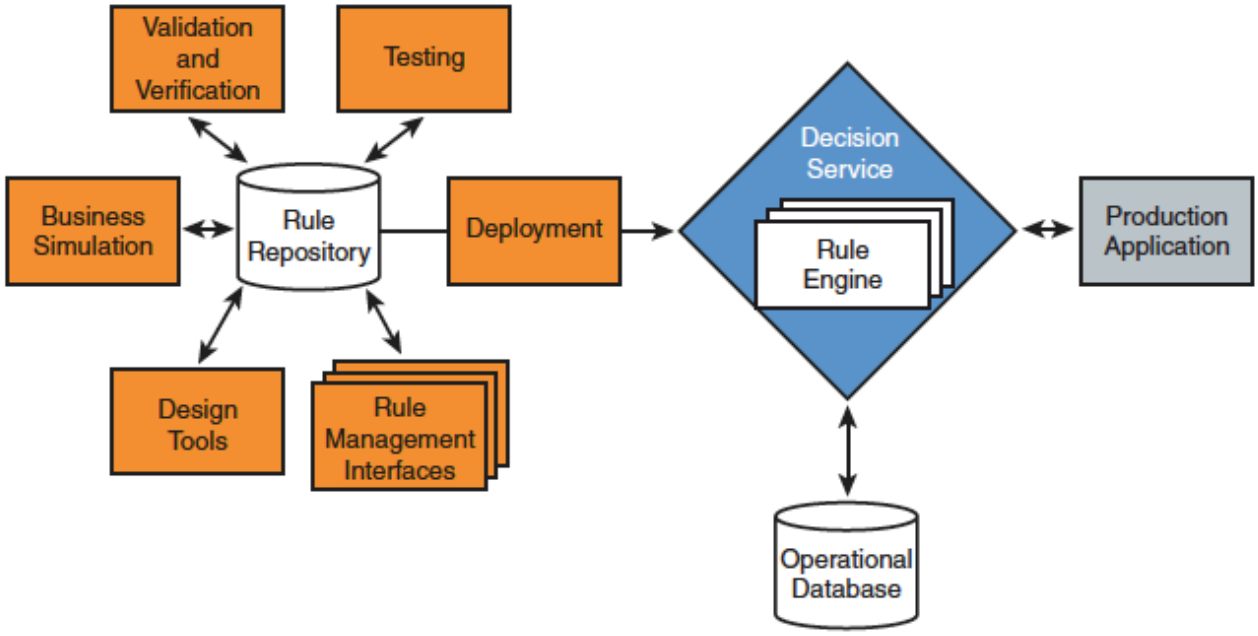

Figure 4.3: Components of a Business Rules Management System

➢ Predictive Analytics Workbenches

A predictive analytics or data-mining workbench is a set of software components designed to enable the analysis of a set of data sources to determine the mathematical relationships within that data and the produce a predictive analytic model that embodies these relationships. It provides a set of capabilities that enable the user to connect to data, prepare the data for modeling, visualize the data, builds predictive and statistical models, test predictive analytic models against hold out data, assess business impact of models, deploy models into production and manage deployed models (see Figure 4.4).

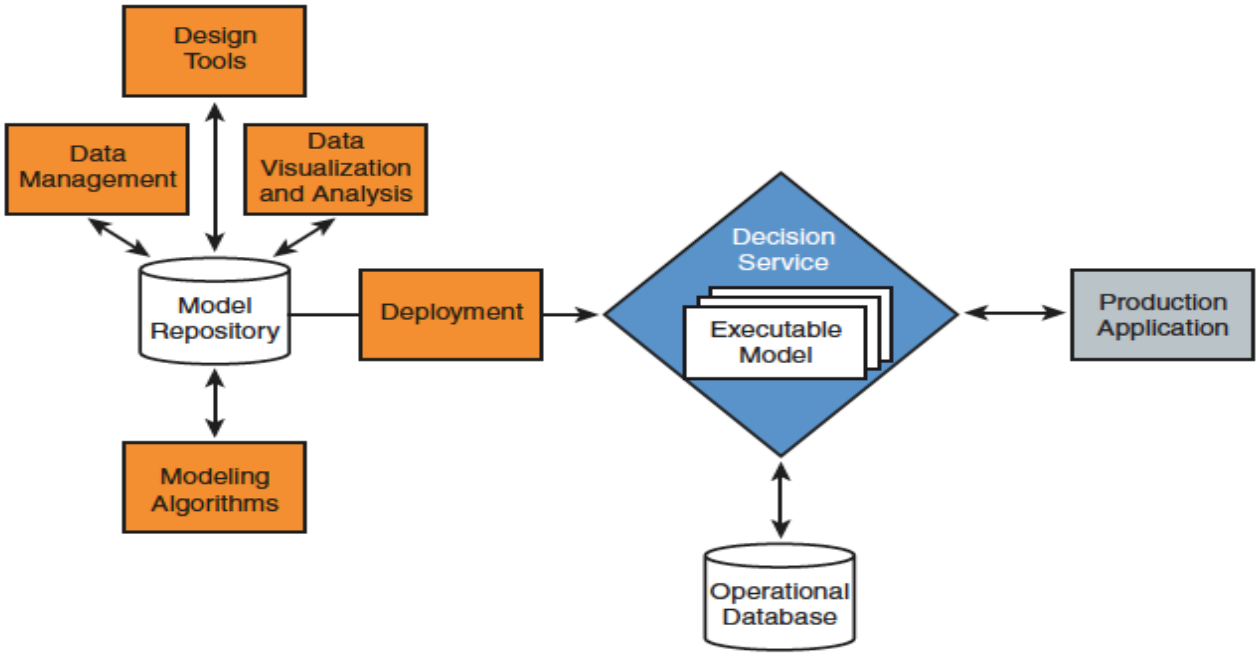

Figure 4.4: Components of a predictive analytics workbench

It support different modeling techniques such as Rule induction, decision tree, linear regression, logistical regression, clustering, K-means, affinity analysis, nearest neighbor, neural networks and genetic algorithms.

These techniques can be used to build predictive analytic models, clustering models, association models and statistical models.

- **Optimization Systems**

Optimization can be used to rune the threshold or parameters of business rules in an existing decision service. An optimization system has a number of components such as A solver, modeling language, design tools and interactive user interface.

- **Pre-configured Decision Management Systems**

Pre-configured decision management systems are targeted for 'out of the box' implementation solution oriented audience. Simplicity, one integrated user interface, lower IT requirements, software as service option, implementation focus, easier to deploy updates and content are the pros of such systems. On the other hand these systems are less flexible, few options outside marketing or fraud, poor integration with existing predictive analytic/business rules and limited cross-silo leaning are main problems with these systems. For example a Customer Relationship management system (CRM), which allows the execution or/and analytics both in the same system.

➢ Data Infrastructure

Data has critical importance to decision management system. It is the basis for predictive analytic models that are embedded in decision management systems, as well as analysis of decision effectiveness. These decision management systems must be integrated with data infrastructure. Following five aspects of data infrastructure are important for decision management systems.

1- **Operational databases** must be integrated with decision management systems
2- **Data warehouse** contains information on performance and often represent the only cross-silo information store available for analysis.
3- **Analytics data marts** are increasingly common for solution-centric analysis.
4- **In-database analytics** can involve in-database creation of predictive analytics models, in-database execution of predictive analytic models or both.
5- **Big data** platform can ensure that web-site logs, call details records, social media, call center and customer service email, and more all feed into these decisions.

➢ A Service-Oriented Platform

Decision management systems can be built and deployed in any architectural framework, most are deployed using service oriented architect (SOA) and are integrated with business process management systems, event processing systems or both.

The use of SOA is a perfect fit for decision service, as it allows decision service to be deployed as loosely coupled, coherent, well behaved services that mesh easily with other services.

➢ Business Process Management Systems (BPMS)

A business process management system (BPMS) allows the definition and execution of all the necessary tasks to execute a business process and to fulfill a business need.

Business process management systems are made up from software components that aid the following:

- Defining and managing the tasks in a business process
- Integrating multiple systems into a single process flow
- Executing both human and system tasks to achieve a business outcome

- Monitoring and reporting on process execution

4.4 Summary

There is huge opportunity for organizations they can develop, agile, analytical decisions management systems and the organizations for better. The technology they need is business rules management system, predictive analytic workbenches and optimization systems. Decision management systems are powerful tools and can add value to the organizations effectiveness. Adopting decision management system is ongoing development process of an organization. As there are always more decisions to find and automate, new opportunities, new rule, new regulations & policies, changing business condition and much more. Thus this is not a one-time effort but a permanent change in how to think about the decision management system and how you can adopt and stay ahead from the rest by being the leading organization in effectiveness and continuous improvement of the processes.

James Taylor is currently working on an application, which could set a platform for the beginner and built the decision management systems according to organizations need. This is still work in process in chapter 6; I will discuss in more details on this cloud base application.

5 The Decision Model

5.1 What is a Decision Model?

Inspiration of the Decision Models comes from Dr. E. F. Codd's revolutionary 'Rational Model' (1970), which changed the way of thinking about managing and leveraging data store in databases. The basic ideology behind this model was the inherent nature of data itself and nothing more.

According to Barbara Von Halle[32] the decision model structure is based on business logic, has it's own existence, independent of how it's executed, where in business it's executed and whether or not its execution is implemented in automated systems.

Because of decision model's nature it made possible for non-technical business people to interact intuitively with their own business logic. This leads to natural business governance over business direction and agility. Business logic has recognizable structure that is not the same as other structures for various different assets.

Here are five interesting characteristics of the Decision Model:

1- It define a technology independent way of organizing an important intangible business intellectual logic.

2- Decision models possess simple structure, declarative nature and optimal integrity.

3- It's easily implemented in technology and transcends current and future technology.

4- Decision model is neither a language nor a grammar, yet they can be defined in the model.

5- Decision model addresses an important unsolved problem; how to effectively manage business rules and business logic.

So what is a decision model? It is an intellectual template for perceiving, organizing and managing the business logic behind a business decision. Business logic is a mean by which a business derives a conclusion from facts. To make business logic tangible, common practice is to translate the business thinking into visible, communicable form, which often is a set of business rules or business statements.

The decision models has following principles:

- Structural Simplicity
- Declarative Nature
- Optimal Integrity

Thus decision model is not simply a list of business rules or business statements. Rather it is model representing a structural design of the logic embodied by those statements.

Capturing business logic, from conditions to conclusions, and refining it until it is atomic, precise, unambiguous and aligned with business objectives is what the decision model and its principles are all about.

Decision Model and its principles are needs at least two different kinds of diagrams; the Rule Family table and the decision model diagram.

➢ The Decision Model Diagram

The Decision Model diagram depicts only the Decision Model's structure and not the detailed content of its Rule Families.

➢ Rule Family

Decision Model can be created with "Rule Family". Simple Rule Family has 3 main elements 'Rule Pattern', 'Conditions' and 'Conclusion'. At first glance Rule Families of a Decision Model appear to be nothing more than familiar decision tables, but it's not. Rule Families is a comparison with the traditional notion of decision tables. A common definition of a decision table is " a table of all contingencies and the actions to be taken for each" Rule Family is a decision model differs from a decision table in two important ways.

1-Each Rule Family adheres to a full set of principles whereas a traditional decision table does not do so.

2-Each Rule Family is related to other Rule Families but Rule Family relationships are managed carefully.

➢ Normalization of The Decisions Model

Normalization principles applied to a decision model ensure that it meets a certain level of integrity. In a nutshell Decision Model normalization is a philosophy for analyzing and decomposing Decision Model structures into a different set of structures that are technology independent, process independent and more desirable than the un-normalized structure.

There are three basic forms of normalization.

1- First normal form for the Decision Model simple means that in terms of the tabular picture of a Rule Family, each Rule Family row cannot be decomposed into more than one row reaching the conclusion.

2- Second normal form for the Decision Model is to eliminate functional dependencies (i.e. inferential dependencies) involving only part of the condition key of a Rule Pattern.
3- Third normal form for the Decision Model is to eliminate functional dependencies among conditions.

Business logic best targeted for a Decision Model s o a purely business nature, involve evaluation of facts leading to a conclusion, represents one business decision, is not easily represented by another means, and is subject to changes. These considerations lead to a model that is high and measurable value to the business.

5.2 The Business Value of Decision Models

The Decision Model gives form, function, and a tangible visual representation to business logic. Business leaders can use it to expose and manage the business logic as an instrument of business agility both in normal changing times as well as in crises.

The three characteristics that are important to a business decision's business value are:

1-Each business has an operative context. The operative context is the complexity of the business environment in which the business decision is made. Basically it determines how a Decision Model solves a business problem or opportunity.

2-Each business decision varies in its volume-based economic impact. Each instance of a business decision has an economic impact on business. It provides insight into the financial value of the Decision Model.

3-Each business decision varies in the complexity of its business logic. The business logic complexity assists in understanding the cost of developing and managing a Decision Model.

The overall economic impact of a business decision to the enterprise is the product of its volume and its individual economic value. Categorizing a business decision by the complexity of its business logic means understanding the quantity of rule families, conditions, conclusions, and relationships among the Rule families. The business logic complexity assists in understanding the cost of developing and managing its Decision Model. For more details please see Table 5.1, 5.2 and 5.3 in the Appendix

In table 5.4 shows the clear distinguishes business process from business decision.

Business Process	*Business Decision*
• Procedural in nature	• Declarative in nature
• Consists of tasks connected by sequence	• Consists of Rule Families connected by inferential relationships (all independent of sequence)
• Is all about how (step-by-step sequence to carry out work)	• Is all about what is to be concluded (the logic leading from conditions to conclusion)
• Improvements in business process aim for increased work efficiency	• Improvements in a business decision aim for smarter business logic
• Represented best in a procedural business process model	• Represented best in a declarative Decision Model

Table 5.4 Important Distinctions between Business Processes and Business Decision

Following are the disadvantages to burying decisions (business logic) in Business process.

1- Forces unnecessary sequence and constraints on business logic
2- Makes changes to business process and business logic difficult
3- Adds un-meaningful complexity to business logic and business process
4- Fails to deliver a visual representation of all business logic
5- Makes governance of business process and business logic difficult to manage
6- Results in business logic and business processes that are not reusable
7- Compromises SOA

- **Business Decision Management**

The practice of managing smart, agile decisions is called Business Decision Management (BDM) and it's also referred as Enterprise Decision Management.
Business motivation, business metrics and business logic are the three elements, which interact to create smart business systems and that is the goal of BDM.

5.3 The Technical Insight of Decision Models

Service Oriented Architecture (SOA) is applied to the construction of automated systems. According to Rosen service is " a discrete unit of business functionality that

is made available through a service contract". The service contract specifies all interaction between the service consumer and service provider. This includes service interface, interface documents, service policies, quality of service and performance.
SOA is a next generation of architectural approach for computer systems. This statement is supported by three trends; major software platforms are moving towards SOA, major application vendors are re-engineering products into discrete service and lastly there is increase in offering software as service (SaaS).

- **Why is SOA important?**

SOA represent a better, more modular, flexible way to build enterprise solutions for business process. It has the potential to deliver significant business value (consistency, commonality, Modularity, Decoupling and manageability).There are four main layers of architecture, business process (business oriented layer), business services (service oriented layer), integration services (service oriented layer) and operational resources (resource oriented layer). See Appendix figure 5.1.

Service inventory is the basis of SOA. Thus it is important how well inventory is structured. Also most important consideration is the separation of concerns and three concerns deserving of separation and associates each with a service role.. These are task services (for procedural action), decision services (for business logic) and entity services (for data access). At the fine-grained level, there may be one service for each service for each process task, one for each access to a specific information group and one for each decision model. Thus decision model in the business process layer distinguishes between decision tasks and process task and business service layer serves as the basis for defining decision service. SOA combined with Decision model allows separation of business logic from other aspects in a mane that other architecture do not.

Research has shown that the number of IT projects failed[33] is quite huge. The most occurring reason for project failure is poor or incomplete specification. Thus defining the requirements in a right way is very crucial. Defining the requirement for software development is divided into two broad categories as following.

1- Functional requirements: The system performs to directly execute the mission of the software. These include process steps, calculation, data manipulations, or reporting activities.

2- Nonfunctional requirements: Are constraints on the system, these constraints includes system performance, reliability, maintainability or ease of use, cost, and other allover characteristics.

Requirements (see Appendix 5.5 for more details on requirements) are expressed in various ways (i.e. Textual statements, business use cases, prototypes, models and code). The integration of all requirements- related models, including the decision model, together with textual requirements statements, and test cases, form a complete collection of requirements. The decision model connects to other model types through common connection point (see Appendix 5.6 for details) and metadata. The decision model simplifies business use cases and business process model.

5.4 Architecture of Decision Models

John Zachman's Enterprise Architecture framework was briefly introduced in section 3.2. Here Barbara Von Halle has highlighted the role of Enterprise Architecture framework in the construction of the decision model. The techniques of the enterprise architecture lead to decomposition of the enterprise, identification of where business logic currently guides the business, and a means for determining where decision models should be deployed. The purpose of the decision model is to serve as the foundation for automatic or assisted deployment of business logic into digitized business systems.

The goal of enterprise architecture then is to define an operating business by decomposing it into its basic components, determine how they relate to each other, and define standards and guidelines by which they work alone an in unison.

It is a complete decomposition of an enterprise into its constituent interrogative perspectives (i.e. who, what, where, how, when, and why) and audience perspective (i.e. planner, owner, designer, builder, subcontractor and operator).

As shown in Figure 5.2 is the modified version of Zachman framework by positioning decision model in it. Only three out of six columns of Zachman Framework are included in this figure to illustrate connections from the Decision model to artifacts in those columns. Although there are connections from the decision model to artifacts in other columns which are not included but only more interesting columns for WHAT (data) and HOW (function) are included. See Appendix 5.7 for brief explanations of this modified model. More details on the framework can be found in the book " The Decisions Model" chapter 14 by Barbara Von Halle.

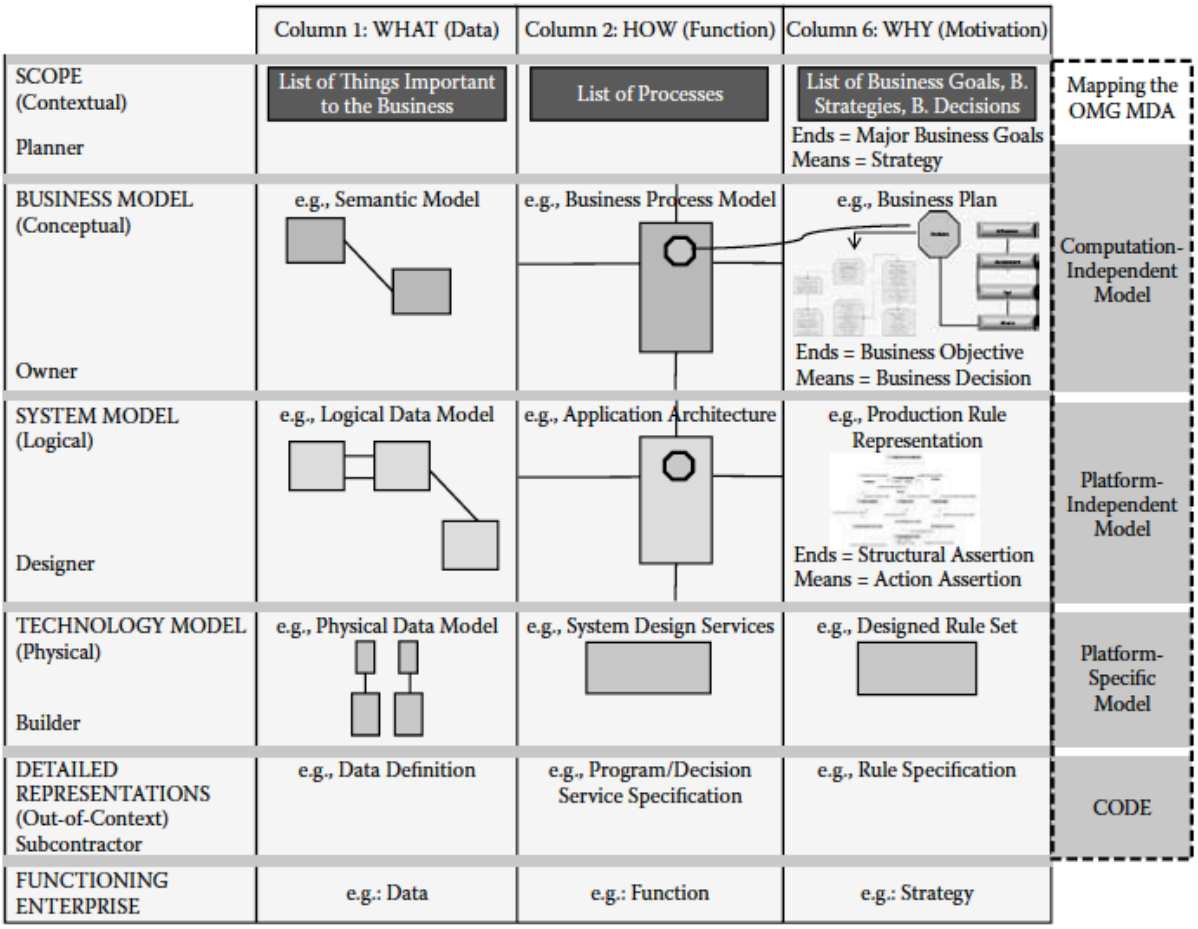

Figure 5.2 The Decision Model in the Zachman Framework

Prior to Zachman Framework the traditional approach of system development was to build each system to meet only particular needs of a specific business area or function, with little to non-thought on larger business scope. The Zachman framework emphasizes that business functions are integrated system that is optimized when engineered as whole. The business logic of an enterprise is connected to relevant views (audience perspective) and domains (interrogative perspectives) of the enterprise, such as artifacts of a business plan, business processes, and enterprises ontology.

If we could summarize Decision Model provides the ideal representation of a business decision for the implementation of a Decision Service. And also there is opportunity for finer grained management of the business governance than is provided by an encapsulated decision service. Different department may govern each Rule Family in a Decision Model.

Furthermore the structure of the Decision Model is implemented in Business Rule Management Systems, because production rule specification allows a close relationship to the rows within a Rule Family. This leads to the highest conformance between the business logic statements and its technical implementation.

Last but not least Service Oriented Architecture as introduced under section 5.3 is also plays and important role in the implementation of the decision models. SOA has evolved a lot and currently is able to introduce flexible application in use today. It is quickly becoming the architecture of choice for commercial software products and for custom application development. Yet SOA remain difficult to master because success with SOA relies not on technology but on architecture and design.

SOA provides a modern implementation approach to Decision Models that integrates decisions into overall enterprise and make them readily available for business process. SOA gets to the ”what” not the “how” of services. In order to create meaningful business process the architecture needs to specify several important characteristics, including how services;

- Have similar size, shape, form, function and other characteristics
- Conform to enterprise standards, support enterprise goal & strategy
- Communicate at Technical and semantic level

BPM without SOA is useful for building application, but difficult to extend to the enterprise. SOA without BPM is useful for creating reusable and consistent services but lacks the ability to turn those services into agile, competitive enterprise. These two are integrated through a layered architecture that consists of the following:

- Business processes and enterprise data (documents)
- Business services and consolidated data (semantic data)
- Integrated services and integration data (transformations)
- Enterprise resources existing systems and operational data (online data)

Thus everything is interconnected Decision Model, Business logic, Business rules, Rule Family, Service Oriented Architecture (SOA) and Zachman Framework and yet different in importance. So understanding all these concepts is important to implement a system, which is robust and agile at the same time.

5.5 Standards & Practices of Decision Models

The publication of standards and specification allow us to benefit from the expertise of the bodies that establish them in a commonly acceptable manner. Decision Model plays a role in various practices from business management to system development. Following is list of practices, which benefit from Decision Model.

- **Business Planning**

To meet the stakeholders goals organizations create business plan to influence or direct the organization to achieve its goals. The Business Motivation Model (BMM) is an Object Management Group (OMG) specification addressing the practice of business

planning. Originally, Business Rule Group (BRG) introduced it. The purpose of the BMM is to provide tools to support the creation of business plans for business enterprises. In other words it's a Meta model of business planning. Decision Model is a set of business rules that implement Course of Action. The decision model is also a means by which the Course of Action controls processes. Thus decision model is complementary to BMM. The relationship between Decision Model and Business Motivation Model can be seen in Appendix Figure 5.4.

- **Business Process Modeling**

Business process modeling approaches include symbols for representing different kinds of activities and various options for sequencing them.

Business Process Modeling Notation (BPMN) is a de facto standard in business process modeling by OMG. The decision model offers a significant evolution of business process modeling and BPMS in following ways;

- It supports the clear distinction and separation between the procedural nature of process tasks and the declarative nature of decision tasks.
- Decision model represent a uniform structure by which business logic is organized into a larger more complex fully declarative and normalized collection of business logic.
- The connection between a BPM and a decision model as an anchor point called business decision and not a link to individual business rules (or business logic statements).

Unified Modeling Language (UML) Specification is an OMG specification for a graphical notation for defining, visualizing and documenting software systems. To leverage the decision model a use case step should be designated as " decision step" and assigned the name of the Decision model that connects to that use case step. This will provide the opportunity to separate the decision logic form the use case description and manage it separately.

- **Information (Fact Type) Modeling**

Information model are used for a range of purposes the most popular purpose being the foundation of database design. The most common types of information models are fact model, data model and business object model. For decision models that will be automated (in a BRMS or not), a formal information model (usually object oriented or

data oriented or both) will be needed for technically oriented purpose and mush for business purpose.

➢ Business Logic and Business Rule Modeling

The practice of business logic and business rule modeling needs both perspective to come together and needs a common mechanism for creating a normalized structure of business logic statements and business rules. The decision model is not concerned with how such expressions are articulated but rather how they should be grouped in a rigorous way. Following are some industry standards currently in use;

Semantics of Business Vocabulary and Rules Specification (SBVR) :SBVR provides a context for the meaning of a business vocabulary and its rules from the business perspective. SBVR is focuses on the formal expression of the business rules statement having a comprehensive set of grammar for the expression of such a statements. The decision model provides visible structural organization of business logic statements whereas SBVR provides a means for expressing each Rule Family instance as natural language statements compliant with the SBVR specification. Conversely SBVR statements can serve as input to populating Rule Families.

Decision Trees and similar representations are not part of the Decision Model because they imply perhaps an unnecessary sequence of execution of conditions based on the arrows in such a graph.

Production Rules*:* Work is proceeding on standards for business rule expression that can be used across multiple platforms is called Production Rules. The transform o Decision Model logic into production rule standards is dependent on the finalization of those standards, and significant technical work that will then need to be completed to create those transforms.

➢ Information Systems Development Approaches

In this subsection some important elements of the " The Decision Model' book are summarized more details can be found on chapter 16 of the book by Barbara Von Halle. The development of Decision Models, specifically for business people changes the development if information system development process. As in some cases business people can and will lead the business process and decision modeling activities at least to some extend to ensure the accuracy of the business logic and the

business process. If the Decision Model is easy to create, interpret and maintain, it more likely that business people will take control of the business decision.

Below is the list of opportunities Information System Development

- Business people are more easily involved in creating and revising Decision Models;
- Decision Models can be delivered earlier n the development cycle or even before the systems development cycle;
- Decision Models can be delivered in increments, enabling agile development;
- Business process models and use case models are simples;
- Decision models lead to reusable decision services in SOA catalogs;
- Utilization of important resources is more effective;
- Earlier test script can be written before programming begins;
- Earlier object models can be created before programming begins and even Decision models are populated;
- Increased quality in business logic leads to fewer errors;

➢ STEP Business Decision Methodology approach

STEP is acronym for

- Separate the business decisions and their business logic from all other aspects, including process.

- Trace the business logic from business motivation, to code and to manual processes.

- Express the business logic in a form understandable by all stakeholders

- Position the business logic for change

Figure 5.5 in the appendix gives and overview of five high level tasks of STEP. Like any other IT is also reluctant to change and probably does not want a totally new methodology. Learning new methodologies is costly, time consuming and change is difficult.. But the tasks behind STEP are easily adapted to an organizations existing methodology. So, sometime less change are needed and in less time as well.

➢ The Decision Model and System Transformation Methods

Figure 5.6 in Appendix gives a brief overview of the system transformation fundamentals and processes. It's not easy to find the business logic from hard coded programs. Decision model makes it easy for all audiences to understand the business logic. The decision model is a deliverables that can achieve transformation form the know level existing codes to high level architecture and it can provide significant improvement of efficiency over other methods of doing so. The Decision Model provides the means of achieving a high value architectural result from System

Transformation at a cost that is potentially no greater that which would have been paid for a lower value deliverable.

SECTION IV: PRACTICAL IMPRLEMENTATION

6 Case Study

In this chapter we are going to practice the theories, frameworks, architectures and models learned and introduced in previous chapters. The purpose of this practical implementation is not to prove anything but the learning process of developing and implementing the decision models and business rules management systems.

Here I would try to explain briefly the development process of each step and point out important things to consider while working with the various tools and processes. In addition to that the learning experience shared here is basically from a non-technical user/developer point of view.

6.1 Background

In this section a brief introduction about the company will be provided for, which we are going to develop and implement the business rules management system. This will give an idea about the operative perspective of the company although, from business rules management system development perspective it is less irrelevant as for implementing the business rules management system it does not really matter from which industry or business field or operation it belong as far as there is a solution available in the business rules development.

ETALENTIS is engineering and Business Consultancy Company active is 4 main technical knowledge disciplines as followings:

- Buildings and techniques
- Energy and Infrastructure
- Engineering
- Maintenance (Continuous improvement)

Etalentis manages projects throughout the full project cycle, from concept (scope) on design and engineering to construction and maintenance. Clients are located both in the Industrial Sector, Trade & Services, as well in the public sector.

6.2 Problem Description

With the ongoing growth of the business Etalentis is actively hiring employees to realize various projects. Depending on the nature of the project Etalentis works with freelancer and by hiring permanent employees. So, during the contract negotiation phase of hiring a freelancer or an employee remuneration package needs to be created. Currently manual process is required to obtain the required information. The existing process is not only complex; time consuming but also the input information

needed is not centralized. Which not only makes the information less reliable but also chances of making error is a major risk.

Thus the ultimate goal of this case study is how BRMS can help to design the system and implement in such a way that in future change can be easily managed with the BRMS tool.

Goal: The new process will help Etalentis to automate the remuneration package simulation model to speed up the process and get efficient and accurate remuneration package comparison analysis model. Furthermore it will help to generate various types of reports to do the cost benefit analysis, project budget estimation etc.

6.3 Methodology

If we want to achieve a goal or result we typically look for a starting point by defining what do we want to achieve? How are we going to get there? What is currently available and what changes needs to be implemented? Based on the available facts and figures we plan and design the blueprint of achieving our goal or result. In a similar fashion to get a head start we need to thoroughly dig up and collect all relevant information, which is crucial to develop the system. Decision models are crucial at this stage to design-structured goal that we want to achieve. So, we need to map the decision involved in process and their importance and relevance to other decisions needed in the process.

In a nut shell a modified version of Zachman's framework which also tries to incorporate what, why, when, where, who and how but not everything will be covered in the decision modeling step but the rest will covered in the later steps of the process. It is simply tool for thinking; a simple way of communication, which covers all aspects of Enterprise context and it is independent of the tools and methodologies used for getting end results.

The question how to develop the decision model we have described it in details in chapter 5. James Taylor made it very easy for IT and most importantly non-technical business people users to understand how to develop the decision models. Recently the author has introduced a cloud based beta version of decision[34] modeler application. The application is still underdevelopment, and several bodies are working together for example to standardized notations for this application. The repositories in decisionsfirst.net models allow you to link all the information and knowledge sources into the modelers to find the information at one place. Although it might not be that much of added value for a small project but for a larger projects where multiple users

needs to work together they can easily collaborate and find the related information right at the model.

In our case this application is able to serve its basic purpose of developing the decision model as shown in figure 6.1.

This model gives a clear picture of what we want to achieve and what are the critical elements in the decision model. The ultimate goal that we want to achieve here is to determine the expected salary of potential new hire based on the employment status; as a freelancer or as a full time company employee. Down the way there are some sub-decisions (Determine insurance expense, Determine Travel Expense, Determine social benefits, Determine base salary and Determine work related benefits), which are categorized to keep thing simple and structured. As it shown in the figure 6.1 this model is the graphical representation of the salary system, it does not emphasis on the process it self, in a sense that in which order the decision needs to be taken but rather what is needs to included in the decision process. The decisions are shown as rectangular in the figure and the oval shapes are the variables, which basically are the rules, which together helps to take an action in order to get to a decision.

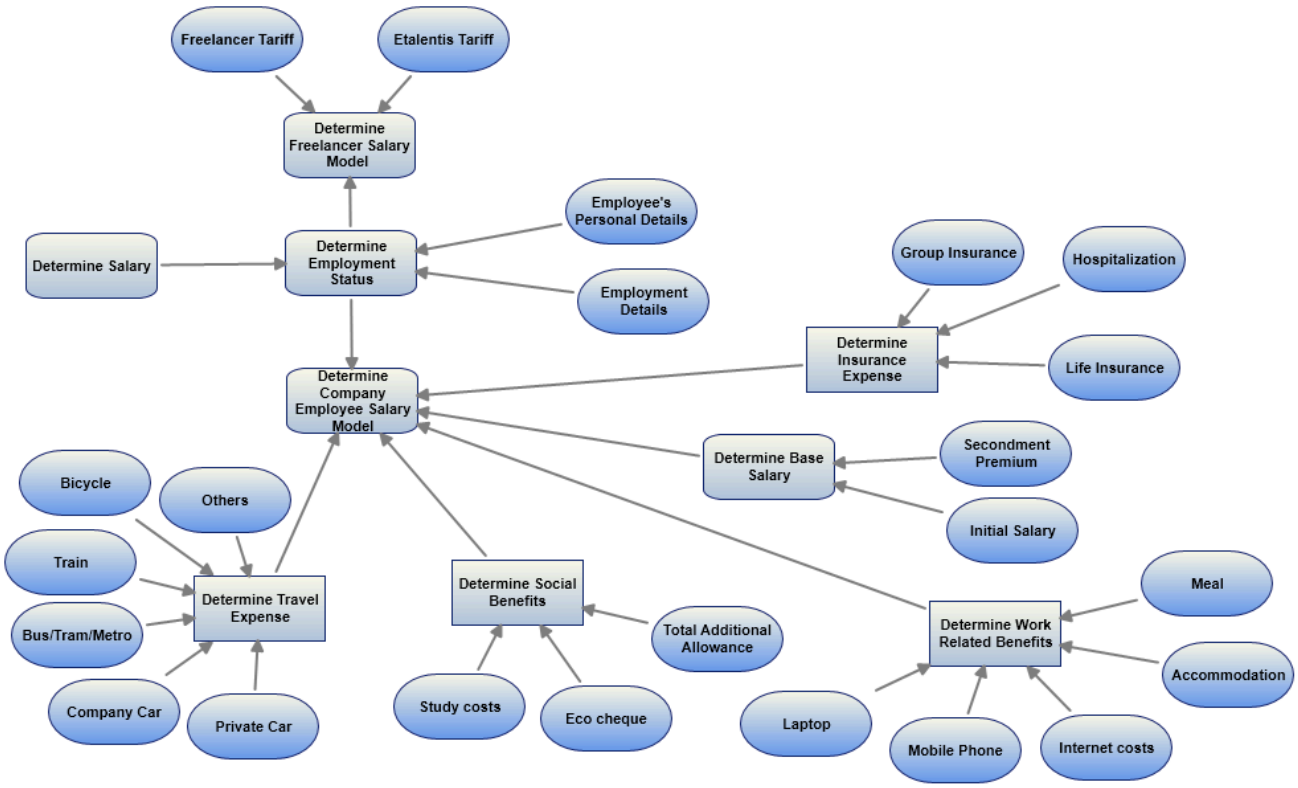

Figure 6.1: Salary Determination Decision Model

We have applied top-down development approach. It means we started with the decision and not with the rules or data. Creating the decision model does not

necessarily required technical knowledge or background, but it does required deep understanding of the process and the importance of the variables involved the model. As the decision model is going to set the base for the business rules. Furthermore it works as a framework for big data analytics to improve rules. Thus what's crucial here is to identify important decision and prioritize them accordingly.

Decision DetermineSalary	
Decisions	**Execute Decision Tables**
EmployeeDetails	:= EmployeeDetails()
EmploymentStatus	:= EmploymentStatus()
AemploymentDetails	:= AemploymentDetails()
BemploymentDetails	:= BemploymentDetails()
Freelancer	:= Freelancer()
BaseSalary	:= BaseSalary()
SecondmentPremium	:= SecondmentPremium()
Laptopallowance	:= Laptopallowance()
MobilePhoneallowance	:= MobilePhoneallowance()
AccommodationAllowance	:= AccommodationAllowance()
MealAllowance	:= MealAllowance()
Internetallowance	:= Internetallowance()
StudyAllowance	:= StudyAllowance()
ECOCheque	:= ECOCheque()
TotalAdditionalAllowance	:= TotalAdditionalAllowance()
HealthInsurance	:= HealthInsurance()
LifeInsurance	:= LifeInsurance()
GroupInsurance	:= GroupInsurance()
Bicycleexpenses	:= Bicycleexpenses()
Train	:= Train()
BusTramMetro	:= BusTramMetro()
PrivateCar	:= PrivateCar()
AcompanyCar	:= AcompanyCar()
BcompanyCar	:= BcompanyCar()
DcompanyCar	:= DcompanyCar()
EcompanyCar	:= EcompanyCar()
CcompanyCar	:= CcompanyCar()
OtherTransport	:= OtherTransport()

Table 6.1: Decision Table

As shown in Table 6.1 the list of decisions order in which they need to be taken to get the final salary proposal. Actually table 6.1 is the sequenced list of decisions in the model. The special sign in second column as shown := and () is necessary to inform OpenRules that decision tables are actually executable Java functions contrary to the decision names in the first column.

6.4.BRMS implementation tool

In chapter 3 we have analyses few Business Rules Management System products currently available in the market. Based on the comparison and the requirement we

have chosen OpenRules as our selected business rules management system development tool. The main reason for choosing this tool is as following:

- It uses MS excel sheets to create the rules which makes is very easy for a non IT person like myself to do the development
- It's an open source application but with a great, helpful and friendly online customer service
- The features offered are compatible in comparison to other expensive products

OpenRules requires the following software:

- Java SE JDK 1.5 or higher
- Apache Ant 1.6 or higher
- MS Excel or OpenOffice or Google Docs (for rules and form editing only)
- Eclipse SDK (optional, Java file development tool)

6.6.1 Knowledge Representation

Often a knowledge system uses one or more of the following knowledge representation formats: objects, formal logic, production rules and structured language. With OpenRules it's not that complicated basically you need to know how to use MS excel sheet and for creating the rules repository the format you need is define the conditions for each business concepts and the action which will result from these conditional rules.

Secondly may be most importantly as the assumption is the developer is a non-IT business analyst, which would mean one does not deep understanding of the IT concepts. Back to the topic on representing the knowledge in the excel sheet.

One of the important thing to consider for developing the decision table, glossary, data sheet or any other table which you might include for rule execution is make sure to check the spelling mistakes, lower or upper case words, where the spacing between two words is allowed and where it is not. Secondly make sure to use the same structure of the variables wherever they are repeated in the tables. As a small error can cause trouble with OpenRules engine execution.

To start the development of the decision table it might be handy to start creating the glossary. You might wonder why we need the glossary well while defining decision tables; we freely introduced different decision variables assuming that they are somehow defined. The business glossary is a special OpenRules table that actually defines all decision variables. But as what is suggested here is start with the glossary below are some reasons to why define the glossary first before creating the decision tables:

1- Gives a complete overview of the list of decision tables needs to be created.

2- You can already structure the format i.e. in which order the process is going to take place but that not mandatory.
3- You can already detect which of the attribute needs will be repeated in the decision tables and you can already have an idea how to deal with them during development of the table.

May be above mentioned reasons are not good enough to prove the point but for me it was handy to get back on track and avoid making things complicated. Table 6.2 would give you an idea how the glossary is set up for salary determination model. For complete glossary list please see table 6.2 in the appendix.

Variable Name	Business Concept	Attribute
Freelancer hourly tariff	Freelancer	FreelancerTariff
Etalentis hourly tariff		EtalentisTariff
Etalentis Margin		EtalentisMargin
Freelancer Salary per hour rate		FreelancerPay
Etalentis Customer Charge Rate		EtalentisCustumerRate
Employee first name	EmployeeDetails	FirstName
Employee last name		LastName
Employee date of birth		DOB
Employee gender		Gender
Employee age		Age
Employee place of birth		POB
Employee nationality		Nationality
Employee marital status		MaritalStatus
Employee home address		HomeAdress
Employee phone number		PhoneNo
Employee mobile number		MobileNo
Employee email		Email
Employee Validation Error		EmployeeValidationerror

Table 6.2:Glossary Salary Determination

For creating the decision tables it sounds quite easy that each row must contain certain applicable rules and it might results in an action or a conclusion. Commonly there is only one resulted action or conclusion but for some cases it is allowed to get various actions or conclusions per row. For example in our case study we have few instances see Table 6.3 where more than one 'conclusion' was possible.

DecisionTable1 Bicycleexpenses										
Condition		Condition		Conclusion			Conclusion			Conclusion
Travel KM per bicycle		Bicycle per KM allowance		Bicycle Travel Expense Per day			Bicycle Travel Expense Per Month			Bicycle Travel Expense Per Year
>	€0	<	€0	is	::= (getReal("Travel KM per bicycle") * getReal("Bicycle per KM allowance"))	is	::= (getReal("Bicycle Travel Expense Per day") * 365 / 12)	is	::= (getReal("Bicycle Travel Expense Per day") * 365)	

Table 6.3:Bicycle expense rules set

Next step is for each business concepts in the glossary you need to create a decision table, which include all the attributes related to the business concepts and each row might need a validation or conclusion cell. It's up to you how you would like to create

the decision tables, as you might to create separate file for each business concept or a single file with multiple business concept sheets in it.

DecisionTable1 Employment Details									
Condition		Condition		Condition		Condition		Conclusion	
Employee function Name		Employee function description		Employee work location		Project customer name		Employment Validation error	
is	True	is	True	is	home	is	True	is	False
is	True	is	True	is	office	is	True	is	False
is	True	is	True	is	other	is	True	is	False
is	False	is	False	is	False	is	False	is	True

Table 6.4: Employment Details (A)

DecisionTable1 Employment Details							
Condition		Condition		Condition		Conclusion	
Employee function Nname		Employee work location		Employee branch office location		Employment Validation Error	
is	True	is	office	is	Hasselt	is	False
is	True	is	office	is	Aartselaar	is	False
is	False	is	office	is	other	is	False
is	False	is	False	is	False	is	True

Table 6.5: Employment Details (B)

We take here few examples from the case table 6.4 is where employee's work related details are validated and if the employee's work location office is selected table 6.5 becomes active to verify the branch office location.

DecisionTable1 Private car									
Condition		Condition		Conclusion		Conclusion		Conclusion	
Travel KM per private car		Allowance per KM private car		Private Car Travel Expense per day		Private Car Travel Expense Per Month		Private Car Travel Expense Per Year	
>	0	>	€0	is	::= (getReal("Travel KM per private car")*getReal(" Allowance per KM private car"))	is	::= (getReal("Private Car Travel Expense per day")*365/12)	is	::= (getReal("Private Car Travel Expense per day")*365)

Table 6.6: Private Car allowance for company employees

We look at another example here how the private car allowance is determined for a company employee. As you can see that three outcomes are possible so three columns of conclusion are added to get the required results. As you can see these conclusions contain calculation formula, which requires a real value so to make them executable we need to add the special signs to mark them so that they can be easily recognized by the system. This is very sensitive work even a small error can cause the trouble for the execution.

The business logic expressed in the decision tables should not depend on the implementation of the objects these rules are dealing with. For example, if a rule says: "If employee age is less than 29 then no group insurance allowance is added"

the only thing this business rule should "know" about the object “employee” is the fact that it has a property “age” and this property has a type that support a comparison operator “<” with an integer. It is a question of configuration whether the Employee is a Java class or an XML file or a DB table from a legacy system. Similarly, if a form has an input field "Employee’s Age", the form should be able to accept a user's input into this field and automatically convert it into the proper object associated with this field independently of how this object was implemented. Thus, OpenRules supports data source independent business rules (decision tables) and web forms.

Data source independent mainly means that the Decision Service does not need to follow the corporate database structure or data formats, but that it operates on its own data structure (in this case the java files).

Once all the rules based decision table were created. The next step was to create the decision service which indicates which of the attribute would act as input and as output during the implementation process. Decision Service is designed by defining the inputs and outputs (i.e. attributes in the decision tables are defined as input or output).

6.6.2 System Development

In this section we will get into some technical aspect of the development. We need to create Java files for each business concept. There are two ways to create the Java files.

- With text editor (notepad++ has a parser that can identify java codes). But keep in mind creating the java file with text editor means everything needs to be manually added in the file.
- Using the Java tool Eclipse[35] can be very handy it allows automating part of the development (i.e. getters and setters generation).

For our case we used Eclipse to create the Java files.

Here is an example for one of business concept ‘Freelancer’. The important thing to consider here is using the correct title for the attributes based on how it is designed in the decision tables. For example ‘FreelancerTariff’ and ‘Etalentis Tariff’ is an input by the user which is a digit so we used java type ‘Int’ and ‘EtalentisMargin’, ’FreelancerPay’ and EtalentisCustomerRate’ is an formula which results in a digit number so we used java type ‘ double’. For more details on java type and how to create the files read the OpenRules user manual.

```
package Freelancer;

public class Freelancer {
```

```
	int	freelancerTariff;
	int	etalentisTariff;
	double etalentisMargin;
	double freelancerPay;
	double etalentisCustumerRate;
	public int getFreelancerTariff() {
		return freelancerTariff;
	}
	public void setFreelancerTariff(int freelancerTariff) {
		this.freelancerTariff = freelancerTariff;
	}
	public int getEtalentisTariff() {
		return etalentisTariff;
	}
	public void setEtalentisTariff(int etalentisTariff) {
		this.etalentisTariff = etalentisTariff;
	}
	public double getEtalentisMargin() {
		return etalentisMargin;
	}
	public void setEtalentisMargin(double etalentisMargin) {
		this.etalentisMargin = etalentisMargin;
	}
	public double getFreelancerPay() {
		return freelancerPay;
	}
	public void setFreelancerPay(double freelancerPay) {
		this.freelancerPay = freelancerPay;
	}
	public double getEtalentisCustumerRate() {
		return etalentisCustumerRate;
	}
	public void setEtalentisCustumerRate(double etalentisCustumerRate) {
		this.etalentisCustumerRate = etalentisCustumerRate;
	}


}
```

Besides creating the Java files for all business concepts. We need to create a Main Java file as shown below to activate the decision.xls. OpenRules provides a special API for decision execution using the Java class **"**Decision**".** The following demonstrates the use of this API for our case study. Package name 'freelancer' is not the same as business concept 'Freelancer' this was just a name of the package to bundle all the java files but has no specific relation or impact on 'Freelancer' business concept of decision.

```
package freelancer;
/*
```

```
 * Developed by OpenRules Inc.
 */

import com.openrules.ruleengine.Decision;

public class Main {

	public static void main(String[] args) {

		String fileName = "file:rules/Decision.xls";

		Decision decision = new Decision("Determine",fileName);

		decision.execute();

	}
```

6.6.3 Rules Execution

In order to make Java files executable we included another tab 'Environment' in our decision.xls (see table 6.1). Table 6.7 shows the complete view of the Environment table.

include	../../openrules.config/DecisionTemplates.xls
import.java	Freelancer.*
include.path	include
include	<Glossary.xls>
	<Data.xls>
	<Employee details.xls>
	<A-Employment details.xls>
	<B-Employment details.xls>
	<Employment status.xls>
	<Freelancer.xls>
	<Base Salary.xls>
	<Secondment premium.xls>
	<Life Insurance.xls>
	<Group Insurance.xls>
	<Health Insurance.xls>
	<Internet Allowance.xls>
	<Laptop Allowance.xls>
	<Meal Allowance.xls>
	<Mobile phone allowance.xls>
	<Accomodation.xls>
	<ECO cheque.xls>
	<Total Additional Allowance.xls>
	<Study allowance.xls>
	<Tram-Bus-Metro.xls>
	<Train.xls>
	<Private Car.xls>
	<Other Transport.xls>
	<Bicycle expenses.xls>
	<A-Company Car.xls>
	<B-Company Car.xls>
	<C-Company Car.xls>

	<D-Company Car.xls>
	<E-Company Car.xls>
import.static	com.openrules.tools.Methods

Table 6.7: Environment

OpenRules allows you to externalize business logic into xls-files. However, these files can still use objects and methods defined in your Java environment.

For example, in the salary determination case all rule tables deal with the Java object Appl defined in the Java package Freelancer.

Therefore, the proper Environment table inside file decision.xls (table 6.8) contains a property "import.java" with the value "Freelancer.*":

6.5 Implementation

OpenRules allows business analysts to do Rule Harvesting by defining business terms and facts without worrying about their implementation in Java, C#, or XML. It also provides the ability to test the business rules in a pre-integrated mode. To do standalone rule testing, a designer of rules and forms specifies it's own test data as Excel tables and creates instances of objects of these types passing them to the rule tables. Table 6.8 shows a partial test data to give you an idea how it needs to set up in the excel sheet.

Data EmployeeDetails

FirstName	LastName	DOB	Gender	Age	POB	Nationality	MaritalStatus
Employee first name	Employee last name	Employee date of birth	Employee gender	Employee age	Employee place of birth	Employee nationality	Employee marital status
Arjan	Mulder	02/01/1976	Male		The Netherlands	Dutch	Married
Sabrina	Claes	05/04/1983	Female		Belgium	Belgian	Single

Data EmployeeStatus

StatusFreelancer	StatusCompanyEmployee	EmployeeStatusValidationerror
Employment status is freelancer	Employment status is company employee	Employee Status Validation Error
Freelancer		False
	Company Employee	False

Data Freelancer

FreelancerTariff	EtalentisTariff	EtalentisMargin	FreelancerPay	EtalentisCustumerRate
Freelancer hourly tariff	Etalentis hourly tariff	Etalentis Margin	Freelancer Salary per hour rate	Etalentis Customer Charge Rate
30	40			

Data EmployementDetails

Function	FunctionDes	WorkLocation	BranchOffice	ProjectCus	EmploymentValidationerror
Employee function name	Employee function description	Employee work location	Employee branch office location	Project customer name	Employement Validation Error
Project Manager	Plan, Manage and implement technical projects	Office	Hasselt	Test-ABC	False

Table: 6.8: Test Data in pre-integrated mode

If you wonder why we need test data the reason why a business or even a technical specialist may need data modeling abilities without knowing complex software

development techniques. In accordance with the SOA principle of loosely coupled services, rule services have to specify what they actually need from the objects defined in an external environment. Data file needs to also included Environment (table 6.7) otherwise the system would not be able to test the data.

Figure 6.2 shows how we have structured the data to run the pre-integrated test.

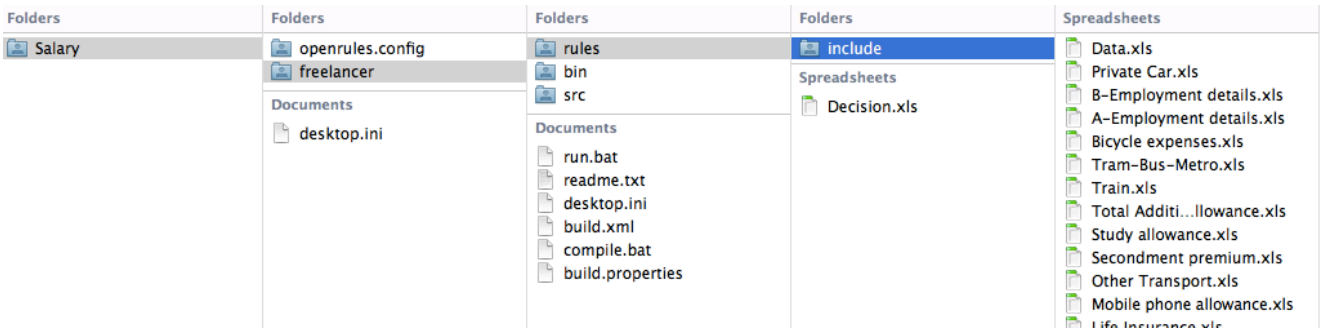

Figure 6.2: Structure of Salary Determination model execution

Once the data sources were structured as shown in figure 6.2 we run the complie.bat file and resulted in successful build as shown in figure 6.3.1 & 6.3.2.

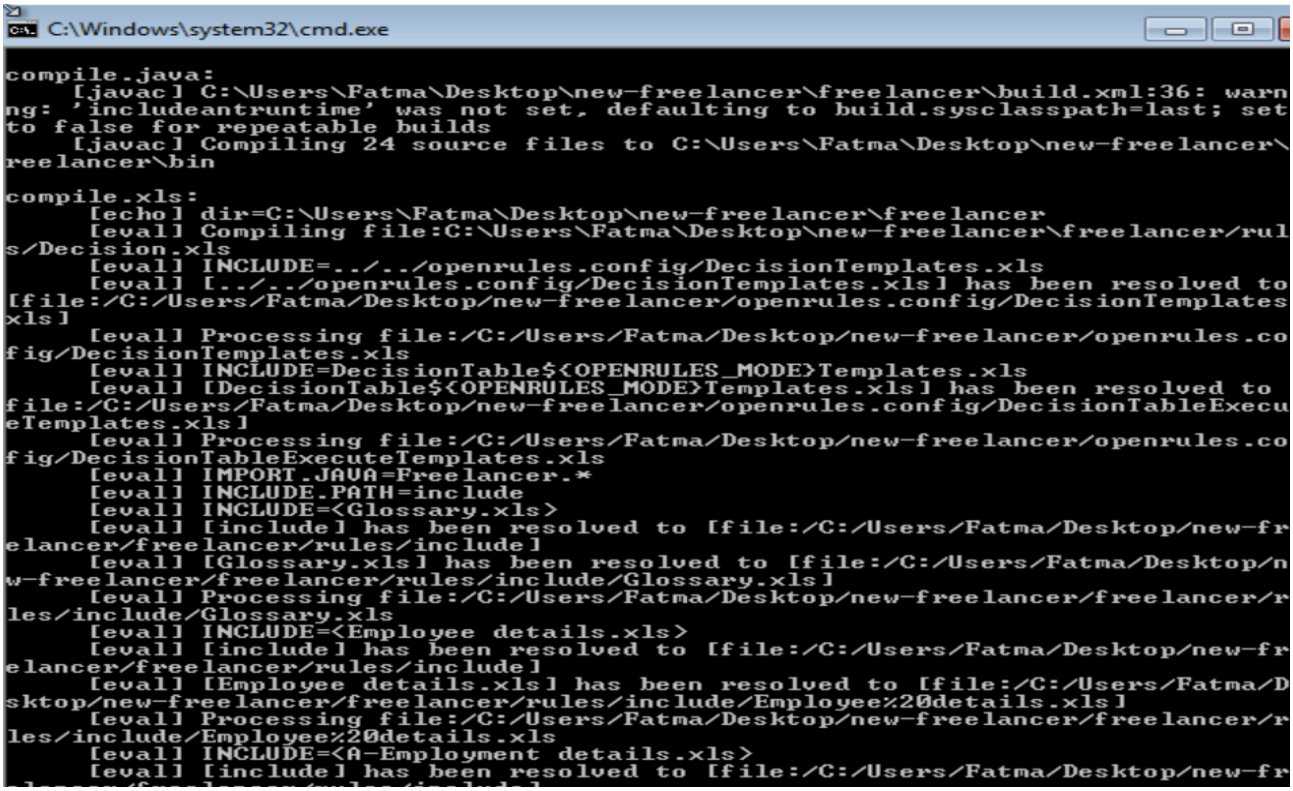

Figure 6.3.1 Compile.bat results

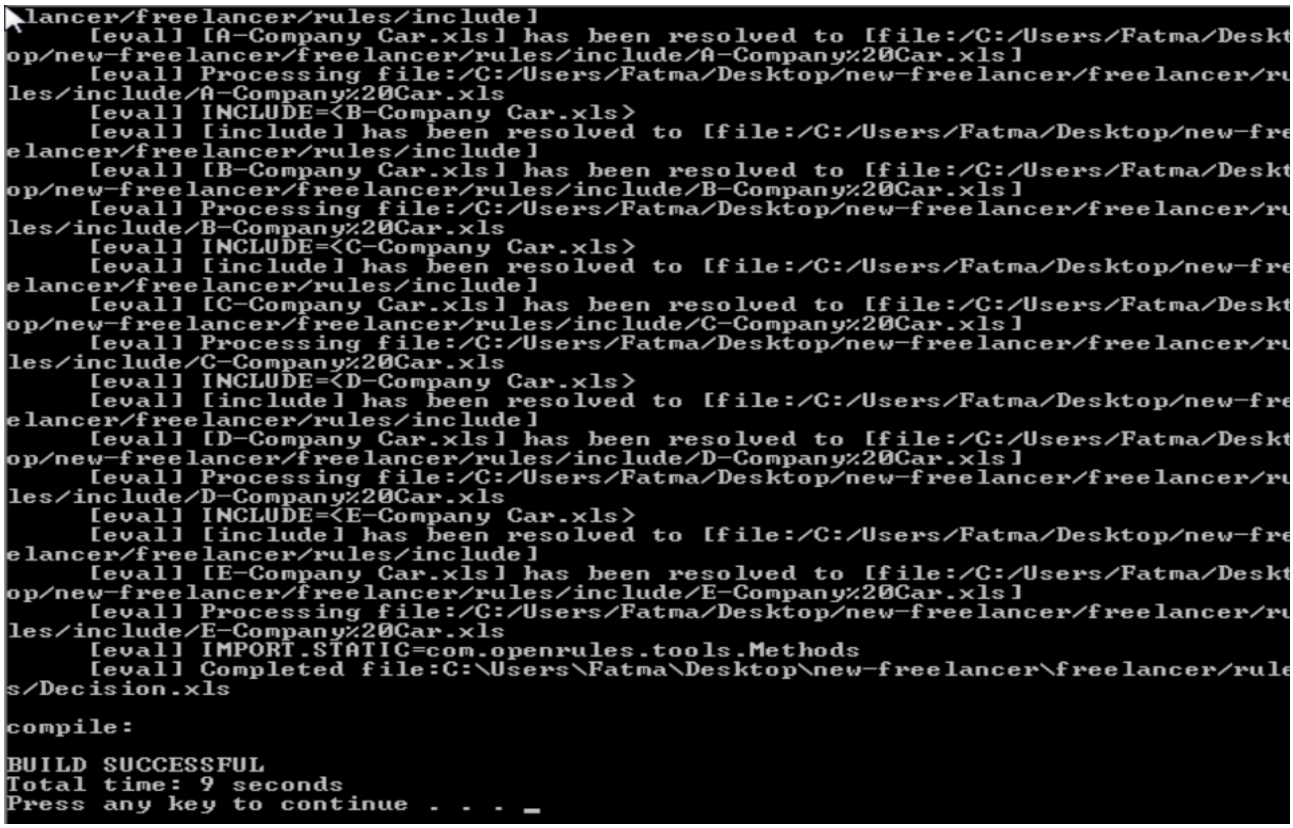

Figure 6.3.2: Compile.bat results

After compilation we run the run.bat file and got successful output as shown in figure 6.4.1 & 6.4.2. These results show that the execution of rules was successful in

OpenRules engine.

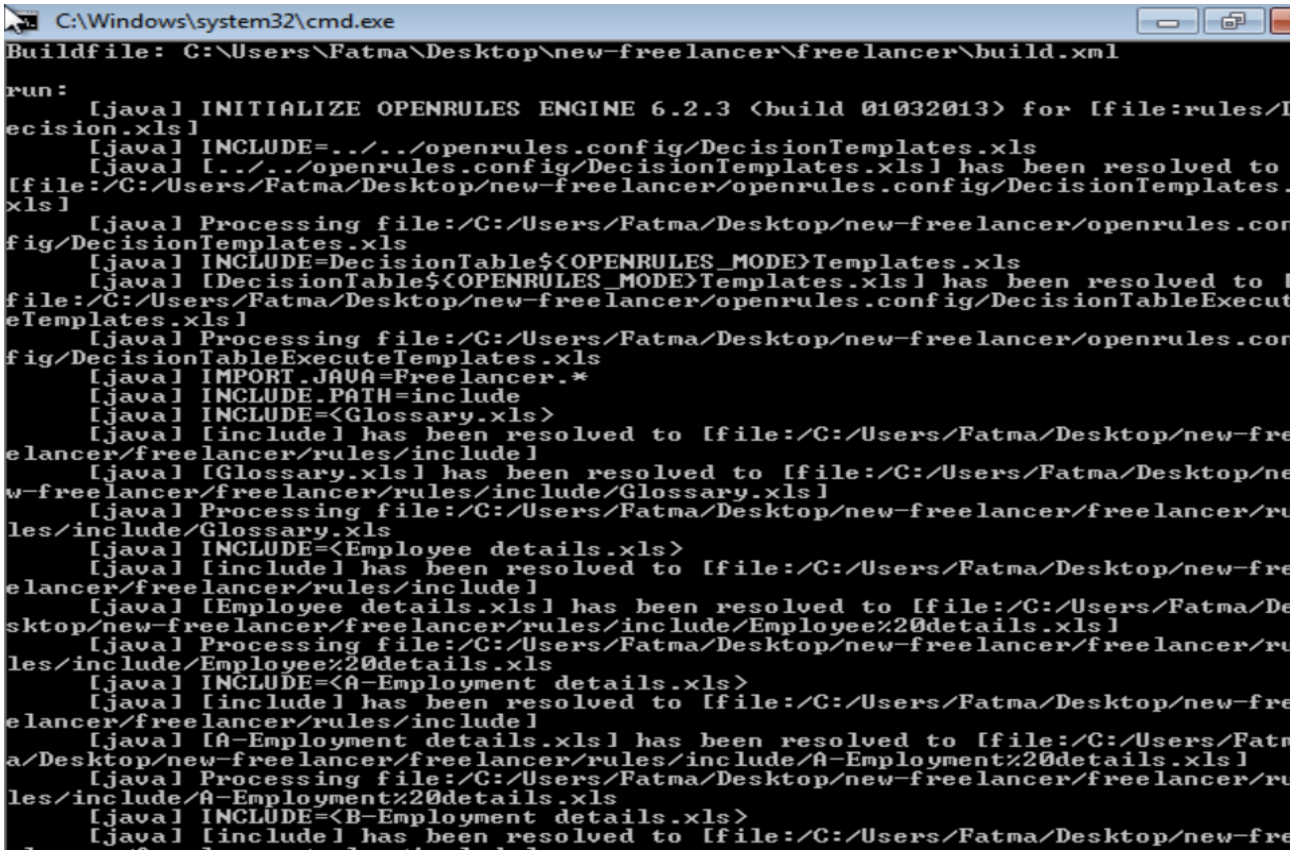

Figure 6.4.1: run.bat results

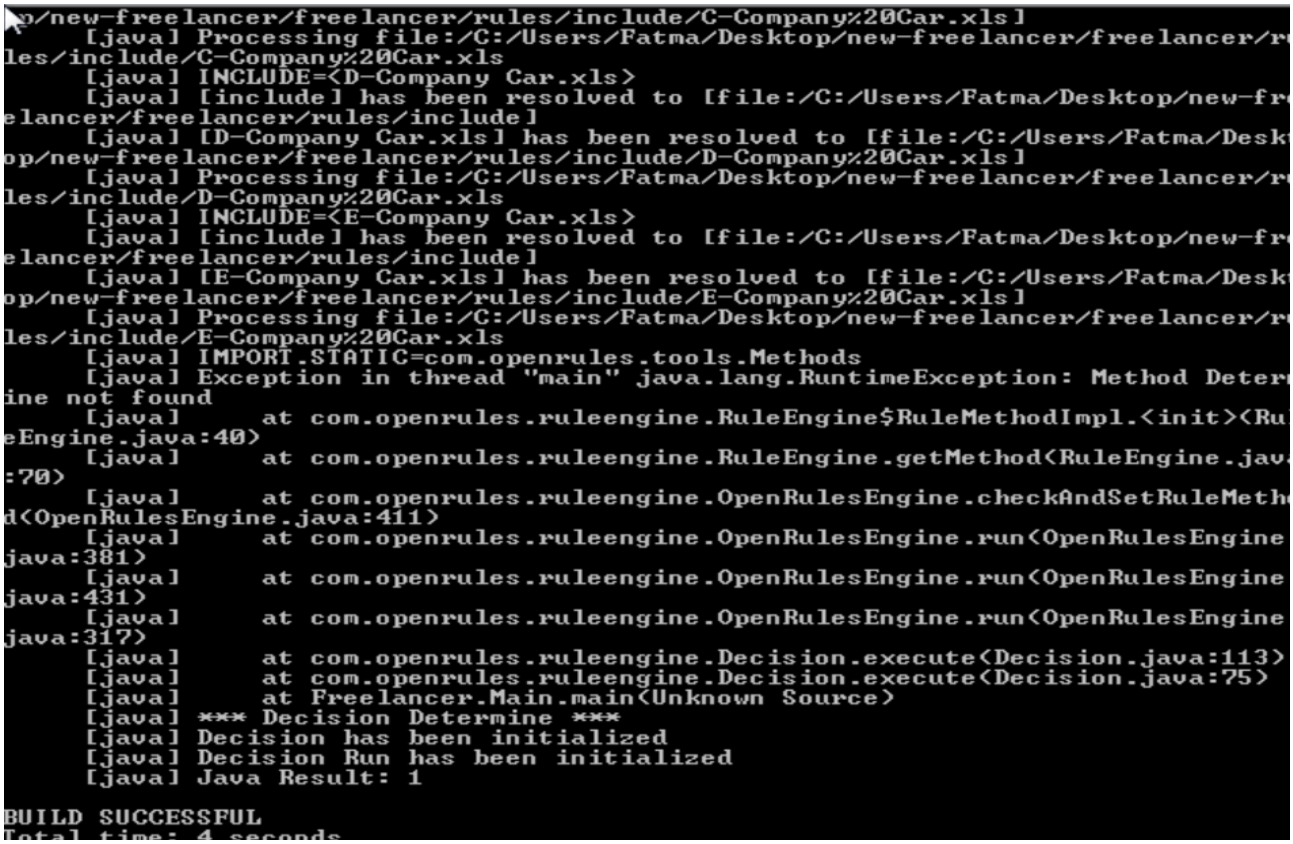

Figure 6.4.2:run.bat results

Due to time limitation it was not possible to build the codes for the web-based interface to run the salary determination model and test the results with the real data. The code, which needed to be developed, would have looked something like as following;

```
public class Main {

public static void main(String[] args) {
    String fileName = "file:rules/main/Decision.xls";
    OpenRulesEngine engine = new OpenRulesEngine(fileName);
    Decision decision = new Decision("Salary Determination",engine);
    etalentisTariff freelancer = (etalentisTariff) engine.run("getFreelancer");
    engine.log("=== INPUT:\n" + freelancer);
    decision.put("freelancer",freelancer);
    decision.execute();
    engine.log("=== OUTPUT:\n" +  freelancer);
}
}
```

Making the code work is very cumbersome activity for a non-technical developer. But once you get hands on experience it becomes easier for the next time. Making the business rules work in the OpenRules Engine is already major work done. So, we can more or less say that the implementation of business rules was successful in our case although as being non-technical developer, it took quite some time to get familiar with the application and how to set up the rules set in the excel sheet itself.

Lastly, the claim giving back the control to business analyst is still debatable at this point in time. Although, the situation is very hopeful and promising, with so much research and experimentation going on we might be able to proof the claim without any doubt.

SECTION V: CONCLUSION

7 Concluding Remarks

In the previous chapters we thoroughly explained the theoretical, technical and methodological aspects of Business Rules Management Systems. Additionally, we have placed them in perspective by looking at their history, applications and organizational context. This chapter concisely sums up our findings and addresses a few unsolved problems.

7.1 Conclusion & Future Research

The theories, models and frameworks framework that we discussed in this thesis from various authors it is without any doubt safe to say that organizations are in need for an intelligent systems, which can take care of repeatable decisions to be more effective and efficient in their day to day businesses regardless of the size of the organization or field they belong to. The question, which was explored in this thesis was that the Business Rules Management Systems can resolve this issue if the organizations understand the value of these intelligent systems. Indeed more and more organizations are becoming aware of the importance and role of Business Rules Management Systems (BRMS), which could result in reshaping the current way of working (in positive sense of process improvements) in many organizations.

The next thing as the processes are developed and implemented by mostly business people, they know how BRMS can help to shape the business processes or at least can be improved. Most organizations want processes controlled by the business people and that is also validated in figure 7.1 a poll results where many business entities took part. So, we are clear we need a system, which is intelligent and also gives the authority to business people. The good news is that the BRMS are designed for business people to establish the business rules and processes. This is what many of the BRMS also claim that a non-technical business user (more specifically business analyst can work with the development tool.

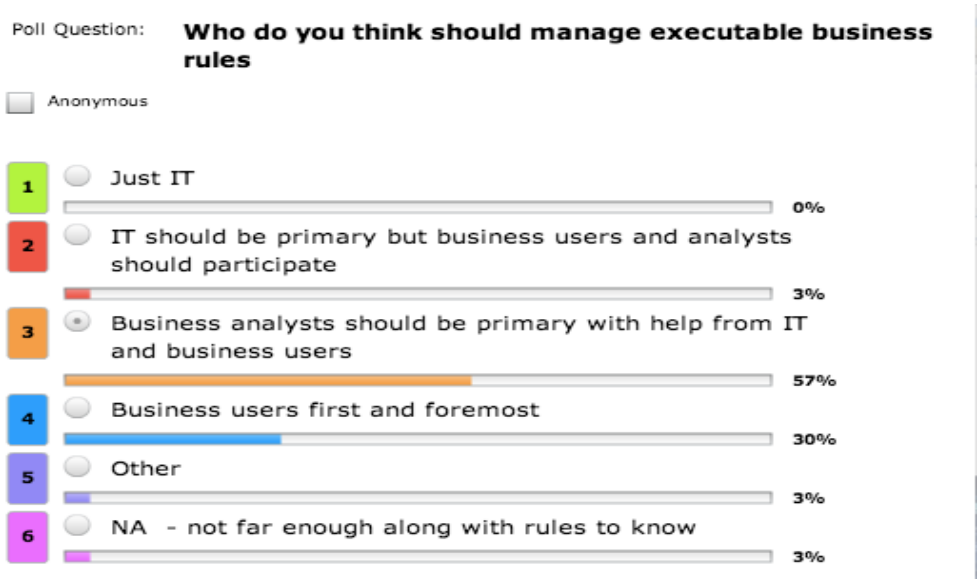

Figure 7.1: Poll results (James Taylor's session on Decision Models)

This thesis has discussed various elements related to the BRMS and from theoretical point of view, working with the BRMS is way to go in future for business people to design their processes. In a nut shell theory supporting the BRMS is fascinating.
As mentioned in the introduction that the goal of this thesis is to find out how practical it is for the business people to actually work with the BRMS. Based on the experience of working on the salary determination case study I find following points worth of mentioning or lets say debatable.

- The tools, which are available today, are designed in many different ways and some of them are way too complex for a non-technical user. Even OpenRules as being simple and easy tool one needed basic knowledge of the development and working with simple IT concept e.g. creating Java files etc.

- Work in process tools could discourage the user for example the decisionfirst.net is a very promising tools, which is putting effort to get a single tool, which allows the users to do end to end development from developing the decision model to creating business rules. But the problem I see is the advertisement of this tool should be limited as currently the tool is still in development and many users might be discouraged to adopt to this new way of developing applications/systems. Secondly currently this tool is trying to link OpenRules in the decisionsfirst.net environment but it is more providing the links to the rules.xls but currently they are not completely integrated from the development point of view.

- Organizations need to create an environment by working closely together with IT and business people in the development process of an application/system. Based on the current BRMS based development structure, it can be seen an opportunity to bring the two extreme business entities together. For example IT people can be flexible to think that they can offload a part of the development process to business people and in the same way business people can offload the IT part of the development to IT people. But what is important here is the cooperation and understanding between these two operatives of an organization. This seems to be more of an organizational problem than an IT problem but it has significant impact on the organization. The processes can be strategic for an organization so it is crucial that they are robust, flexible and easy to work. As it may sound very easy but it requires a huge input and effort to make it work. Thus based on their related field experience and knowledge IT and business people can make this happen. We might be surprised to see huge impact on our current process improvements.

- **Suggestion For Further Research**

From my point of view there is a strong need for standardization of notations and may be format of creating e.g. business rules etc. It might be a little extreme for demanding but for example how MS office works has set some standards, which were adopted by the later developers (i.e. Openoffice, Google docs. etc.) of this popular daily global working tool. It is very exciting to think that currently BRMS gives you all the freedom to work as you want but then standard can help to provide a global structure regardless of the tool you use. The purpose of BRMS is basically making it easy for the development regardless of the technical background. So, I suppose there is need that sets the basic standards accordingly.

Another area that needs attention is the currently available tools, which should allow the developer to prevent from making error or at least can detect them during the development process. As for example OpenRules allows MS Excel based, which is in a way good to use a familiar tool but it is quite easy to make minor mistakes. What I am suggesting here is that we are trying to build the intelligent systems with the Business Rules Management System and the developer should be able to develop such systems efficiently and focus on the advancement of Business Rules Management Systems rather than the usual development hectic.

In a nut shell the prime focus should be that more and more businesses adopt the business rules management bases systems for improving the processes in their organization and not that, which business rules management system development tool to use and how to work with the tool should not be the primary concern. But unfortunately based on my own experience it took me quite some time to do some research on the tools and how they operate and it was confusing to decide on the development tool as at the end, what we all look for is simplicity and ease of use. Thus it does not really matter how many features the tool has but how many useful features it has and how easy they are to use.

Appendix

Business Rules Management System Product Classification

(Class A includes products for which business people represent the main audience. They are independent of a particular development environment and perceive business rules management from a business perspective. They provide specialized services for the rules acquisition, including identification of those business artifacts that allow placing business rules in a certain context. The offer of such products is relatively small, their main draw-back being that they don't support rules implementation. Typical products for this class are Rule Xpress (Rule Arts) and Rule Track BRS (Business Rule Solutions). **Class B** includes products intended primarily for developers, especially analysts and software architects. Their role is to assist the development of rule-based applications, by providing facilities for rules acquisition, formalization, modeling and, especially, for implementation. In most cases, they don't offer support for non-technical users, such as high-level specification languages. Many products on the market belong to this class: In Rule (In Rule Technology), Visual Rules (Innovations Software Technology), Usoft (Ness Technologies), Versata BRMS (Versata) etc. Even though, technically speaking, they represent only parts of a BRMS, rule engines correspond to a technology that can be used in software development independently of any other products. Several rule engines must be noted: open source rules engines such as OpenRules and Drools for Java and NxBRE or Drools. NET for the .NET platform, but also the rule engine included in Microsoft technology, called Windows Workflow Foundation.

Class C has the widest audience and is open to all categories of persons involved in the development of knowledge-based applications. Being built on the paradigm of expert systems, they allow the creation of intelligent applications based on knowledge acquisition. They are versatile products that have powerful inference engine and provides a wide range of facilities for rules management, both at business level and at software system level. Blaze Advisor FICO (FICO) and ILOG Rules (IBM) are recognized as market leaders in this segment.)[36]

Operative Context	**Characteristics of Inputs**	**Typical Characteristics of Corresponding Decision Models**	**Usefulness of the Corresponding Decision Model**
Simple	Known-Knowns	-Known fact type -Known fact values -Related to business process models	-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	-known fact type(not always) -Unknown fact values - More than one possible solution for the same input -Related to complex event processing	-Identifies unknown fact types and values -Ascertain areas needing specific expertise -Delivers business logic shared from experts to non-experts -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	-Unknown fact type -Unknown fact values -Unclear solutions for some or all inputs	-Aids in developing solutions -Delivers business logic in a form amenable to iterative development -May reduce complex decisions

		-Related to fluid circumstances	to complicated or simple ones
Chaotic	Unknowables	-Any of the above -Related to crisis - Applies to islands of order within chaos	-Aids in delivering order where order can be envisioned

Table 5.1 The usefulness of the Decision Model Based on Operative Context

Management Level	**Typical Characteristics and Economic Impact of Corresponding Business Decisions**	**Business Value of the Corresponding Decision Model**
Strategic	-Low volume -High individual economic value -Spans simple to chaotic operative context	-Delivers the same value as its operative context; see table 5.1
Tactical	-Medium value -Medium individual economic value -Medium collective economic value -Spans simple to chaotic operative context	-Delivers the same value as its operative context; see table 5.2
Operational	-High volume -Small individual economic value -High collective economic value -Mostly operates in the simple operative context	-Provides a management approach for the small decision that add up to high economic impact -Provides a starting point for high-volume operational decisions o be automated -Provides traceability to business objectives and metrics-especially relating to business decisions that result in fines, reprocessing, and

		reclaiming of money - Serves to provide competitive advantage and agility

Table 5.2 The Value of the Decision Model Based on Management Level and Volume-Based Economic Impact

Measure of business logic complexity	**Characteristics of Corresponding Decision Model**	**Business Value of corresponding Decision Model**
Simple Complexity	Low quantity of: -Fact type (columns) -Business logic (rows) -Rules Families (tables) -Inferential relationships (connections) -Fact types known -Fact values known	-Delivers the same value as Decision Models operating in the simple context - For Decision Models that are very simple in business logic complexity, the main value is in communicating and standardization on the business logic
Medium Complexity	Medium quantity of: -Fact type (columns) -Business logic (rows) -Rules Families (tables) -Inferential relationships (connections) -Fact types known -Fact values known or require human expertise	-Delivers the same value as Decision Models operating in the complicated context -Serves as mechanism for discovering, gaining clarity, and standardizing business logic with medium complexity, ensuring the greatest degree of simplicity of the business logic as possible -Serves a s a basis for determining which parts of the Decision Model can be automated and which ought to be handled by humans
High Complexity	High quantity of:	-Delivers the same value as Decision Models

	-Fact type (columns) -Business logic (rows) -Rules Families (tables) -Inferential relationships (connections) -Fact types unknown -Fact values unknown or require human expertise	operating in the complex and chaotic context -Serves as mechanism for discovering, gaining clarity, and standardizing business logic with high complexity, ensuring the greatest degree of simplicity of the business logic as possible -Highlights the value and competitive advantage of strategic human expertise

Table 5.3 The value of the Decision Model based on Complexity of Business logic

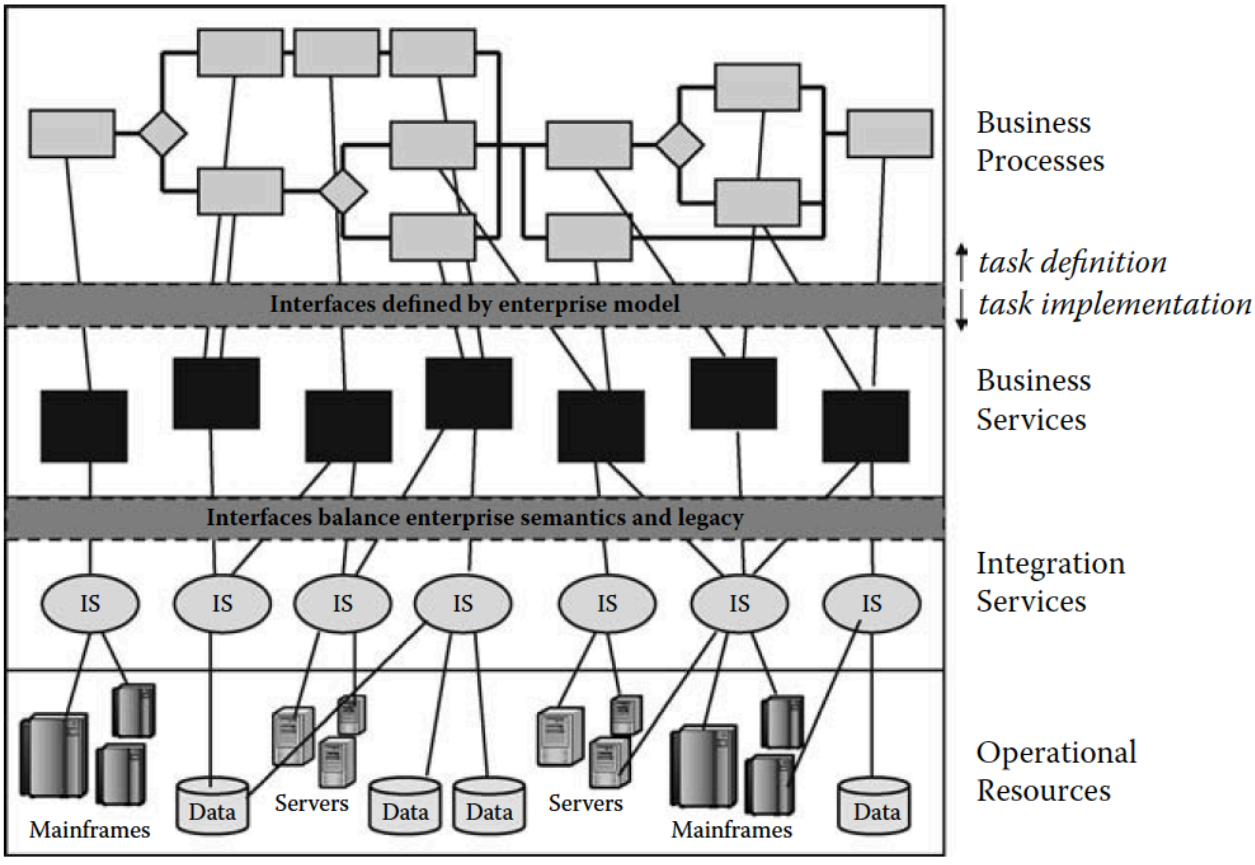

Figure 5.3: Layered SOA architecture. Source: Rosen et al. (2008)

Characteristic	*Explanation*
Cohesive	The requirement addresses one and only one thing.
Complete	The requirement is fully stated in one place with no missing information.
Consistent	The requirement does not contradict any other requirement and is fully consistent with all authoritative external documentation.
Relevant	The requirement is related to a business rationale (or business motivation) that is relevant to the mission of the software.
Correct	The requirement meets all or part of a business need as authoritatively stated by stakeholders.
Current	The requirement has not been made obsolete by the passage of time.
Externally observable	The requirement specifies a characteristic of the product that is externally observable or experienced by the user. "Requirements" that specify internal architecture, design, implementation, or testing decisions are properly constraints, and should be clearly articulated in the Constraints section of the Requirements document.
Feasible	The requirement can be implemented within the constraints of the project.
Unambiguous	The requirement is concisely stated without recourse to technical jargon, acronyms (unless defined elsewhere in the Requirements document), or other esoteric verbiage. It expresses objective facts, not subjective opinions. It is subject to one and only one interpretation. Vague subjects, adjectives, prepositions, verbs, and subjective phrases are avoided. Negative statements and compound statements are prohibited.
Mandatory	The requirement represents a stakeholder-defined characteristic the absence of which will result in a deficiency that cannot be ameliorated.
Verifiable	The implementation of the requirement can be determined through one of four possible methods: inspection, analysis, demonstration, or test.

Source: Wikipedia, 2008 (amended by the authors).

Table 5.5 The Characteristics of Good Functional Requirements

Model	*Model Connection Point*	*Equivalent Decision Model Connection Point*	*Shared Metadata*
Business Motivation Model	Business Rule (or Decision)	Business Logic Statement (or Decision)	Business Policy, Business Objective, Business Tactic
Business Process Model	Decision Task, or Business Rule Task	Decision	Decision, Object Model, Data Model, Actor (Stakeholder)
Use Case	Decision Step	Decision	Decision, Actor (Stakeholder)
Logical Data Model	Entity or Attribute	Fact Type	Glossary Alias, Description
Object Model	Entity or Attribute	Fact Type	Glossary Alias, Description
Fact Model	Fact Type	Fact Type	Operator, Operand

Table 5.6 Connection points between models

Decision Model Tool Supports:			
Zachman Row	*Storage of Decision Model Aspects*	*Links to the Same Column*	*Links to the Same Row*
1	Lists of business decisions	From the Row 1 business decisions to the corresponding Row 2 Decision Models and corresponding business planning models (e.g., OMG's BMM)	From the Column 6 business decisions to the corresponding Column 1 Fact Types for the conclusions and corresponding Column 2 Processes guided by the business decisions
2	Computation-Independent Decision Models (projectwide, enterprisewide, comparisons, inheritance, versioning, test cases, natural language representations, etc.)	From the Row 2 Decision Models to the Row 3 Platform-Independent Models and assignments to decision services	From the Column 6 Decision Models to the corresponding Column 1 Semantic Models (e.g., fact model, business object model, conceptual data model) and corresponding places in the Column 2 Business Process Models (via Decision anchor point)
3	Platform-Independent Models (in Platform-Independent Languages)	From the Row 3 Platform-Independent Models to corresponding Platform-Specific Models	From the Column 6 Platform-Independent Models to the corresponding Column 1 Logical Data Models or class diagrams and corresponding Column 2 Application Architectures
4	Platform-Specific Model for target technologies	From the Row 4 Platform-Specific Models to corresponding Business Logic Specifications	From the Column 6 Platform-Specific Models to the corresponding Column 1 Physical Data Models, class diagrams, component structure diagrams, and corresponding Column 2 System Design Services

Decision Model Tool Supports:			
Zachman Row	*Storage of Decision Model Aspects*	*Links to the Same Column*	*Links to the Same Row*
5	Business Logic Specifications	From the Row 5 Business Logic Specifications to the corresponding Operating Code	From the Column 5 Business Logic Specifications to the corresponding Column 1 Data Definitions, class specifications, and to the corresponding Column 2 Program/Decision Support Specifications

Table 5.7 Zachman Framework & Capability Decision Model Tool

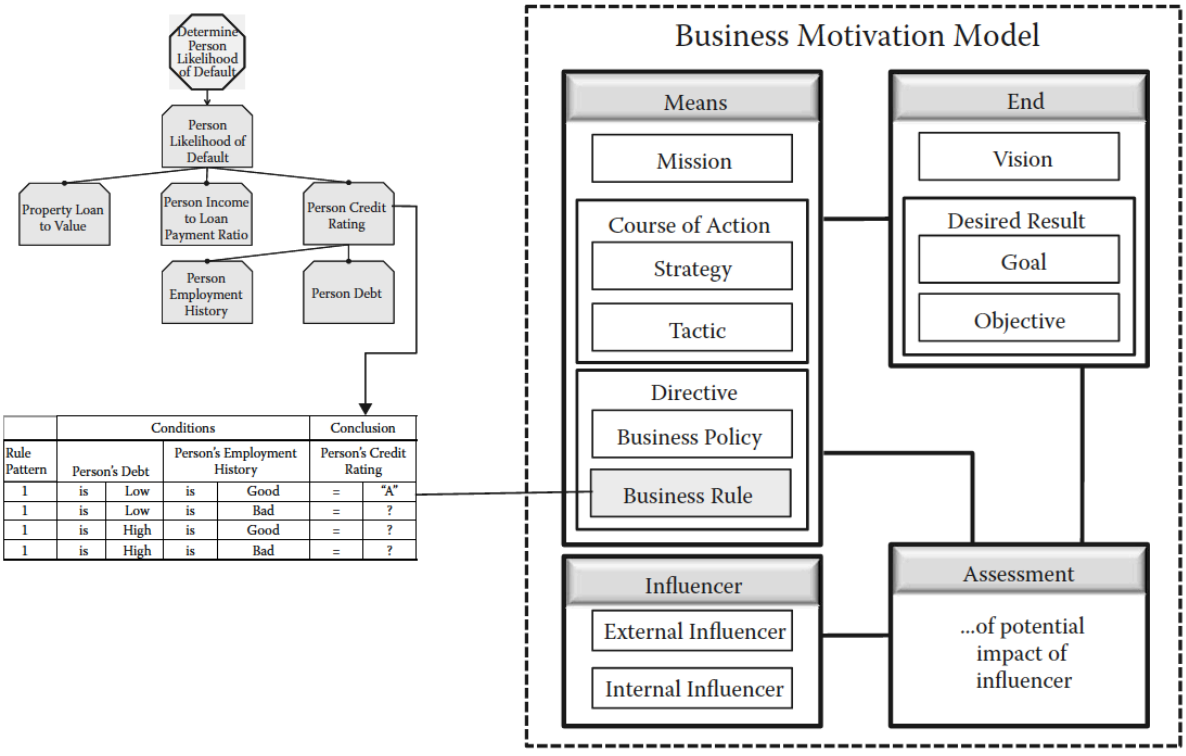

Figure 5.4: Connection between Decision Model & Business Motivation Model without changing BMM

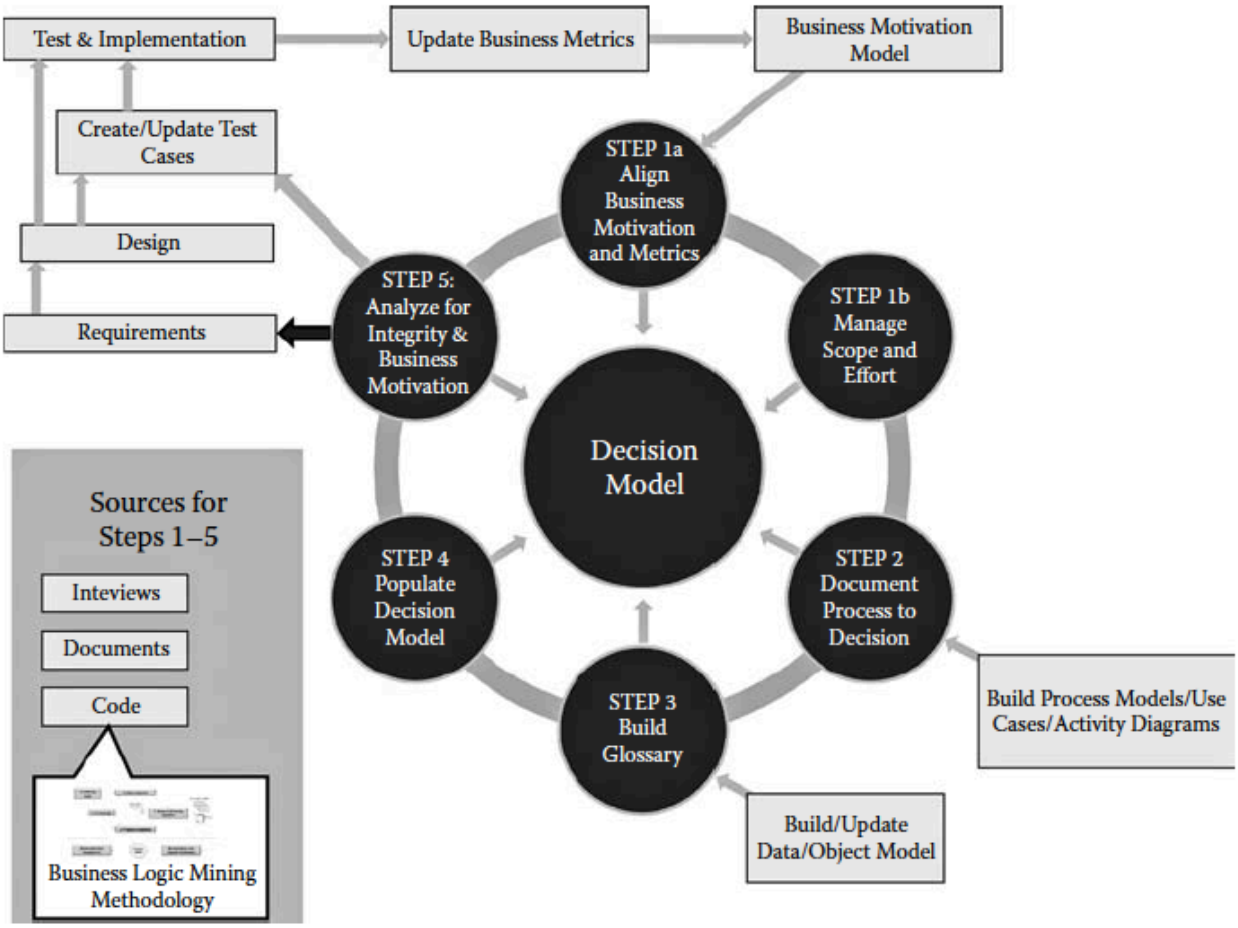

Figure 5.5: Five high level of STEP tasks

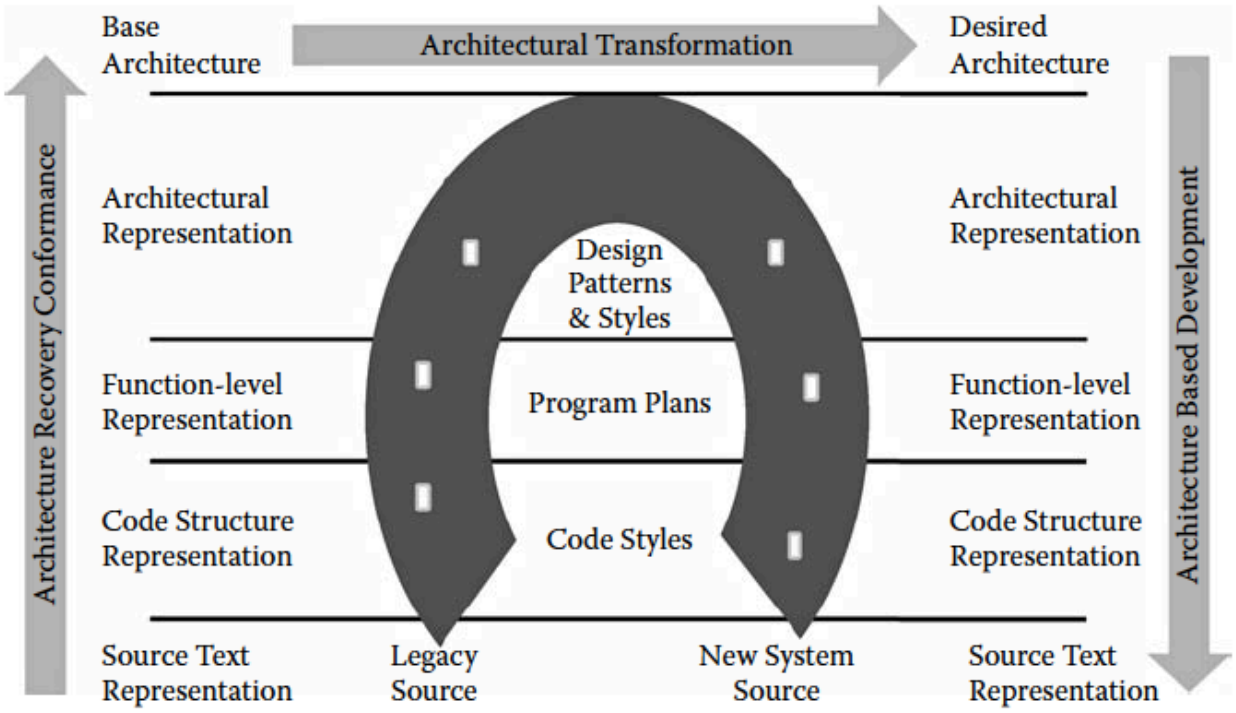

Figure 5.6: The System Transformation Horseshoe

Variable Name	Business Concept	Attribute
Freelancer hourly tariff	Freelancer	FreelancerTariff
Etalentis hourly tariff		EtalentisTariff
Etalentis Margin		EtalentisMargin
Freelancer Salary per hour rate		FreelancerPay
Etalentis Customer Charge Rate		EtalentisCustumerRate
Employee first name	EmployeeDetails	FirstName
Employee last name		LastName
Employee date of birth		DOB
Employee gender		Gender
Employee age		Age
Employee place of birth		POB
Employee nationality		Nationality
Employee marital status		MaritalStatus
Employee home address		HomeAddress
Employee phone number		PhoneNo
Employee mobile number		MobileNo
Employee email		Email
Validation Error		Validationerror
Employee function name	EmploymentDetails	Function
Employee function description		FunctionDes
Employee work location		WorkLocation
Employee branch office location		BranchOffice
Project customer name		ProjectCus
Validation Error		Validationerror

Employment status is freelancer	EmploymentStatus	StatusFreelancer
Employment status is company employee		StatusCompanyEmployee
Validation Error		Validationerror
Base Salary	BaseSalary	Basesalary
Employment percentage		EmploymentPerc
Salary occurrence		Occurrennce
RSZ costs on base salary		RSZPercsalary
Number of working days		NoWorkingdays
Final Base Salary		FinalBaseSalary
Final Etalentis Base Salary		FinalEtalentisBaseSalary
Secondment payment	SecondmentPremium	SecondmentPayment
Employment Percentage		EmploymentPerc
Final Secondment Payment		FinalSecondmentPayment
Laptop allowance	Laptopallowance	LaptopAllowance
Employer contribution percentage		EmpConLap
Laptop allowance employee		LaptopAllowanceemployee
Laptop allowance Etalentis costs		LaptopallowanceEtalentiscosts
Mobile phone allowance	MobilePhoneallowance	MobilephoneAllowance
Employer contribution on mobile allowance		EmpConMob
Mobile phone allowance Employee		MobilePhoneAllowance Employee
Mobile phone allowance Etalentis costs		MobilePhoneAllowance EtalentisCosts
Internet allowance	Internetallowance	Internetallowance
Employer contribution Percentage		EmpConInt
Internet Allowance Employee		InternetAllowanceEmployee
Internet Allowance Etalentis costs		InternetAllowanceEtalentisCosts
Accommodation Allowance	AccomodationAllowance	AccommodationAllowance
Validate		Validate
Study Allowance by user	StudyAllowance	StudyAllowance
Validate		Validate
ECO Cheque	ECOCheque	Ecocheque
ECO amount by user		EcoamountUser
Occurrence ECO cheque		EcoOccurance
Calculate		Calculate
Additional Allowance by user	TotalAdditionalAllowance	AddAllowance
Additional Allowance Description		AddAllowancedesc
Validate		Validate
Meal allowance	MealAllowance	MealAllowance
Validate		Validate
Health Insurance Allowance by user	HealthInsurance	HealthinsurallowanceUser
Occurrence Health Insurance		OccurrenceH
Health insurance allowance		HealthInsuAllowance

Life insurance employee	LifeInsurance	LifeInsurance
Validation Error		Validationerror
Group Insurance allowance	GroupInsurance	GroupInsurance
Group insurance allowance by user		GroupInsuranceUser
Occurrence group insurance		OccurrenceG
RSZ Percentage on group insurance		RSZPercGroupIns
Group Insurance Employee		GroupInsuranceEmployee
Group Insurance Etalentis Cost		GroupInsuranceEtalentisCost
Travel KM per bicycle	Bicycleexpenses	TravelkmB
Bicycle per KM allowance		KMallowB
Bicycle Travel Expense Per day		BicyclePerDay
Bicycle Travel Expense Per Month		BicyclePerMonth
Bicycle Travel Expense Per Year		BicyclePerYear
Travel KM per train	Train	TravelkmT
Travel occurrence by Train		OccuranceT
Train allowance based on occurrence		TrainAllowanceOccurrence
Travel KM per BusTramMetro	BusTramMetro	TravelkmBTM
Travel occurrence by bustrammetro		OccurrenceBTM
Tram/Bus/Metro allowance based on occurrence		BTMAllowanceOccurrence
Travel KM per Private Car	PrivateCar	TravelkmPrivCar
Allowance per KM private car		AllowPriCar
Private Car Travel Expense per day		PrivateCarPerDay
Private Car Travel Expense Per Month		PrivateCarPerMonth
Private Car Travel Expense Per Year		PrivateCarPerYear
Travel KM per company car	CompanyCar	TravelKMComCar
Vehicle type		Typevehicle
KM Model type vehicle		KMvehicle
Fuel Card		FuelCard
Company Car allowance without tax		ComCarallowance
Fuel card commission Exclusive tax		Fuelcardcom
Garage cost		Garagecost
Tax Percentage		Taxpercentage
Garage Cost Allowance		Garagecostallowance
Fixed Garage costs		Fixedgaragecosts
User Garage costs		Usergaragecosts
Total Company Car Allowance		TotalCompanyCarAllowance
Travel KM per other transport	OtherTransport	TravelkmOther
Allowance per other transport		AllowOtherT

Other Transport Travel Expense Per day		OtherTransportPerDay
Other Transport Travel Expense Per Month		OtherTransportPerMonth
Other Transport Travel Expense Per Year		OtherTransportPerYear

Table 6.2: Glossary Salary Determination

Bibliography

[1] Daniel S. Appleton, "Business Rules: The Missing Link," Datamation, October 15, 1984, pp. 145-150

[2] Ronald G. Ross, Entity Modeling: Techniques and Application, Database Research Group, Inc., 1987

[3] Ronald G. Ross, The Business Rule Book (First Edition), Business Rule Solutions, LLC. 1994

[4] "Defining Business Rules ~ What Are They Really?" (Formerly known as the "GUIDE Business Rules Project Report," 1995), edited by David C. Hay and Keri Anderson Healy, Business Rules Group, (Third Edition.), July 2000

[5] Ronald G. Ross, The Business Rule Book (Second Edition), Business Rule Solutions, LLC.1997

[6] Business Rules Group, 1998

[7] Ronald G. Ross and Gladys S. W. Lam, Capturing Business Rules, 2000

[8] Barbara von Halle, Business Rules Applied: Building Better Systems Using the Business Rule Approach, Wiley Computer Publishing, 2002

[9] Tony Morgan, Business Rules and Information Systems, Addison-Wesley, 2002

[10] Ronald G. Ross, Principles of the Business Rules Approach, Addison-Wesley, 2003

[11] James Taylor & Neil Raden, Smart (Enough) Systems, Prentice Hall, 2007

[12] Semantics of Business Vocabulary and Business Rules (SBVR), OMG, 2008

[13] http://www.businessrulesgroup.org/defnbrg.shtml

[14] http://www-01.ibm.com/software/websphere/products/business-rule-management/whatis/

[15] http://www.brsolutions.com/SmartBRMS_Overview.pdf

[16] Business Rules Management and Service Oriented Architecture A Pattern Language by Ian Graham 2007

[17] http://pic.dhe.ibm.com/infocenter/brjrules/v7r0/index.jsp?topic=%2Filog.rules.jrules.doc%2FContent%2FBusiness_Rules%2FDocumentation%2F_pubskel%2FJRules%2Fps_JRules_Global246.html

[18] http://info.psu.edu.sa/psu/cis/biq/se501/a/a1/MajorCausesofSoftwareProjectFailures.pdf

[19] Why Projects Fail, from the Perspective of Service Science by Ronald Stanley & Lorna Uden

[20] Business Rules Management and Service Oriented Architecture: A Pattern Language By Ian Graham

[21] Six Simple Steps for Accelerating and Perfecting Requirements: A Framework, a New Model, and Visualization By Larry Goldberg and Barbara von Halle, April 26, 2010

[22] http://www.modernanalyst.com/Resources/Articles/tabid/115/articleType/ArticleView/articleId/2376/More-on-Decision-Tables-and-The-Decision-Model-in-Practice.aspx#Comments

[23] http://www.omg.org/news/meetings/tc/agendas/va/DMN_pdf/Vincent_SainteMarie_vanThienen.pdf

[24] http://openrules.com/overview.htm

[25] http://openrules.com/architecture.htm

[26] http://www-03.ibm.com/software/products/us/en/odm/

[27] http://www.fico.com/en/Products/DMTools/Pages/default.aspx

[28] http://www.fico.com/en/products/dmtools/pages/fico-blaze-advisor-system.aspx

[29] http://docs.jboss.org/drools/release/6.0.0.Beta5/drools-guvnor-docs/html_single/index.html

[30] http://www.jboss.org/drools/drools-guvnor.html

[31] Decision Management Systems (A Practical Guide to Using Business Rules and Predictive Analytics) by James Taylor

[32] The Decision Model: A Business Logic Framework Linking Business and Technology (IT Management) by Barbara von Halle & Larry Goldberg

[33] http://www.ambysoft.com/surveys/success2011.html

[34] http://decisionsfirst.cloudapp.net

[35] http://www.stanford.edu/class/cs108/JavaTools/eclipse-guide/

[36] http://revistaie.ase.ro/content/52/07%20-%20Andreescu,%20Mircea.pdf

Other References
http://www.agileitarchitecture.com/2008/06/soa-and-brms.html

http://www.ibm.com/developerworks/bpm/bpmjournal/1206_boyer/1206_boyer.html

http://www-01.ibm.com/software/websphere/products/business-rule-management/

http://www-01.ibm.com/software/websphere/ilog/

http://nl.redhat.com/products/jbossenterprisemiddleware/business-rules/

http://www.microsoft.com/biztalk/en/us/business-rule-framework.aspx

http://www.modernanalyst.com/Resources/Articles/tabid/115/articleType/ArticleView/articleId/2516/categoryId/90/How-to-Introduce-The-Decision-Model-Successfully.aspx

http://www.modernanalyst.com/Resources/Articles/tabid/115/articleType/ArticleView/articleId/2376/More-on-Decision-Tables-and-The-Decision-Model-in-Practice.aspx#Comments

http://mitiq.mit.edu/IQIS/Documents/CDOIQS_201177/Papers/05_02_7A-2_von_Halle_et_al.pdf

http://www.econ.kuleuven.be/prologa/Product_Information/modeling_knowledge.htm

http://www.cs.uu.nl/docs/vakken/mdks/lectures/lecture4a.pdf

Auteursrechtelijke overeenkomst

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:
**Business Rules Management Systems
A practical implementation**

Richting: **Master of Management-Management Information Systems**
Jaar: **2013**

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,

Mubarik, Fatma Maqsoom

Datum: **22/08/2013**