Fundamental Constructs for Derivation Business Rules

Eline de Haan
Application Development
Dutch Tax and Customs Administration
Apeldoorn, the Netherlands
ey.de.haan@belastingdienst.nl

Martijn Zoet
Optimizing Knowledge-Intensive Business Processes
Zuyd University of Applied Sciences
Sittard, the Netherlands
martijn.zoet@zuyd.nl

***Abstract*—Due to the creation of the new Decision Model Standard, derivation business rules play an even more crucial role in organizations' daily operations. To capture these business rules, organizations can choose between a multitude of commercially and scientifically available business rule languages. However, currently no set of criteria exists to evaluate these business rule languages and underlying tools with regard to expressiveness and preciseness. So, a need for a reference framework to simplify the selection process can be identified. During this research, a set of 15 fundamental constructs is identified, required to create precise and expressive business rules, which can be used as reference framework to perform an evaluation. The identified fundamental constructs have been validated in three different rounds using sequentially 37 patterns, 252 business rules, and six business rule management systems by applying Mill's Method, which indicated usefulness and completeness.**

Keywords-Business Rules; Fundamental Constructs; Business Rule Management; Business Rule Languages; Derivation Business Rules.

I. INTRODUCTION

More and more organizations capture their business logic in the form of business rules. A business rule is defined as: "*a statement that defines or constrains some aspect of the business, intending to assert business structure or to control the behavior of the business* [1]." In the last decade, these business rules have become an increasingly valuable asset for organizations. To specify and manage this asset, a multitude of business rule languages and systems is available. For instance: RuleSpeak, The Decision Model (TDM), the Simple Rule Markup Language (SRML), the Semantic Web Rules Language (SWRL), the Production Rule Representation (PRR), the Semantics of Business Vocabulary and Business Rules (SBVR), SRL, N3, and IRL [2].

The abundance of available systems and languages, and the fact that they differ to a large extent regarding their expressive power, causes two challenges. The first challenge organizations may encounter are difficulties in selecting an appropriate business rule management system or business rule language, since no set of criteria exists, which could be used as reference point for comparison. This can for instance lead to the selection of a language with a too extensive or too low level of expressive power. A second problem can occur when a language, tailored to a particular business rule management system, is selected. In case an organization transfers to a new or additional system, the business rules have to be re-specified to comply with the specification language of that specific system, which is highly inefficient, expensive and error prone.

Research has been initiated to compare the business rule languages, since various differences between the languages exist. Examples of such studies are [3] and [2]. Zoet et al. compared the representational capabilities of four different business rule languages [3], by mapping the fundamental elements of these languages onto the constructs of the Bunge-Wand-Weber (BWW) representation theory [4].

Previous studies focused on high-level elements (e.g., thing, property) of business rule languages. This view is applicable to analyze business rule languages at a global level, but not to evaluate the details of the syntax and semantics of the languages. Other previous studies focused on creating a business rule language that could cope with a whole range of logic. Examples of such languages are LISP and PROLOG [5]. However, much of the expressive power of these languages is not even applied in practice. This is caused by different factors, for instance: unusable for business users, but more importantly, most of this expressive power is not necessary to be able to specify derivation business rules.

The aim of this research is to evaluate business rule languages from a more detailed and practical view in order to tackle the outlined problems above. This research was conducted based on the following research question: *"How can derivation business rules be specified precisely and implementation independent?"*

This paper is organized as follows. Section II presents the literature review, which provides insight into different types of business rules and the specification thereof. Section III explains the applied research method to devise and validate the envisioned artifacts. The data collection and data analysis process are described respectively in Section IV and Section V. In Section VI, the results that derive from the identification and creation of artifacts are presented. Section VII provides the conclusions of the study including the contributions, limitations and future work.

 ISBN: 978-1-61208-542-5

II. LITERATURE

In literature, a "business rule" is defined in a variety of ways, which is emphasized by a statement of Von Halle "*depending on whom you ask, business rules may encompass some or all relationship verbs, mathematical calculations, inference rules, step-by-step instructions, database constraints, business goals and policies, and business definitions* [6]." Furthermore, not one commonly accepted way to classify business rules exists. From literature, ten different classification schemes to classify business rules emerged, which each cover several business rule categories (types) [1][7][8][9][10][11][12]. Among the ten classification schemes, different names are used to refer to either similar or dissimilar business rule categories.

To delimit this research, the focus will lie on one specific type of business rules namely **derivation business rules**. A derivation business rule can be defined as: *"an expression that evaluates facts, by means of a calculation or classification, leading to a new fact (i.e., conclusion) [1][13]."* To position the type of business rule on which this research focuses, derivation business rules, this type is compared to the categories included in the ten found classification schemes. This comparison showed that derivation business rules correspond to the following categories of the found classification schemes: 1) Inference rules, 2) Computation rules, 3) Derivation rules, 4) Classification rules, 5) Decision rules, 6) Calculation rules, and 7) Rounding rules [1][7][8][11][12][14].

Besides the fact that different business rule definitions and categories exist, also many different business rule notation forms are available to specify derivation business rules. At the highest abstraction level, two main formalism types can be identified: implementation dependent and implementation independent languages. The first type is defined as "*an implementation dependent language is a language that complies to a specific software formalism, has a delimited predefined expressiveness, and is tailored to be interpreted by a particular information system* [15]." Examples of implementation dependent languages are LISP and Haskell, but also the languages used by specific business rule (management) systems, such as Corticon or Be Informed. When organizations use such an implementation dependent language and switch to a new business rule management system, the business rules must be re-specified in order for this system to process them, which is highly inefficient, expensive and error prone. In contrast, an implementation independent language is considered as: "*a language that complies with a certain level of naturalness but has a delimited predefined expressiveness and is not tailored to be applicable for a specific automated information system* [15]." So, this second formalism could by applied in multiple environments addressing the disadvantages of a dependent language but is generally not **precise** enough to be directly executable by an automated information system.

A solution for this problem can be found by investigating which fundamental constructs (i.e., building blocks of a language) are necessary to specify a precise derivation business rule. Similar studies are performed in different research fields concerning fundamental constructs. For example, Moody created a checklist comprising a defined set of criteria to determine if a language can be easily understood by people [27]. Furthermore, Van der Aalst created a list of patterns to check if business process management systems could handle different types of process elements [28]. Like in the previous studies, our goal is not to create a new language. However, the focus will lie on the identification of the minimal set of constructs a language needs to contain to be able to precisely specify business rules found in practice. When this minimal set of fundamental constructs is used as reference point to select a language, it should be made clear that not all these constructs have to be included in the language. In some cases, these constructs are already available as property in the business rule (management) system. For example, in tools like Be Informed and Berkely Bridge the relationship between constructs cannot be expressed by means of the language but only by the use of a system property.

III. RESEARCH METHOD

The purpose of this research is to identify the fundamental constructs that characterize a precise and transformable derivation business rule. The premise of this research is that the identified set of fundamental constructs is good enough when most common business rules in practice can be captured. To accomplish this goal, a research approach is needed that can identify 1) the fundamental constructs applied in business rules and 2) the similarities and dissimilarities between fundamental constructs applied in business rules.

Both requirements can be met by applying grounded theory. The purpose of grounded theory is to *"explain with the fewest possible concepts, and with the greatest possible scope, as much variation as possible in the behavior and problem under study [16]."* Grounded theory identifies differences and similarities by applying eighteen coding families. However, this does not provide a structured comparison of the identified situational factors across cases. Therefore, an additional technique is needed to compare the differences between a variety of business rules. A technique specifically engineered to inspect cases for similarities and differences is ordinal comparison based on Mill's method of agreement and difference [17]. Mill's methods are used to draw conclusions about causal relationships by analyzing the data (i.e., effects) and find common denominators (i.e., causes) [18]. With regard to this research, the common denominators correspond to the required fundamental constructs found in each case to be able to specify precise derivation business rules.

 ISBN: 978-1-61208-542-5

IV. DATA COLLECTION

Three rounds of data collection were performed. The first data set comprised existing business rule patterns, these were collected in order to identify the first set of fundamental constructs. For the second data set, existing business rules were gathered to analyze if the identified fundamental constructs could cover the business rules or additional constructs were needed. For the third data set, business rules were collected, which were implemented in a specific business rule management system, to examine the applicability of the identified fundamental constructs in an implementation dependent environment.

To select the data sets, one overall practical selection criterion was applied namely *site/document access* to be able to use the data for this research. In contrast, the applied theoretical selection criteria differed per data set. For the first data set, one theoretical criterion was taken into account, which meant that solely business rule patterns focused on specifying derivation business rules were included. Based on this criterion, 37 patterns from the following five current existing business rule pattern catalogues were selected: [8][11][12][14][19]. Table I shows the amount of collected patterns per catalogue.

TABLE I. AMOUNT OF PATTERN COLLECTED PER CATALOGUE.

Pattern Catalogue	Amount of Patterns
Morgan	2
RuleSpeak	12
Wan Kadir & Loucopoulos	2
Von Halle	2
RegelSpraak	19
TOTAL	37

For the second data set, one theoretical selection criterion was applied: only instantiations of derivation business rules were eligible. By adhering to this criterion, 252 derivation business rules were randomly selected from the following eleven different business rule cases originating from both literature and practice: [8][11][12][14][19][20][21][22][23][24][25]. This sampling strategy is followed in order to cover a wide range of domains where business rules are utilized. Table II lists the amount of selected business rules per case.

With regard to the third data set, two theoretical criteria were applied. The first theoretical criterion to select the business rule management systems implied that the documentation of each system covered the implementation of the same business rule set (i.e., use case). The second theoretical selection criterion corresponded to the fact that the business rule set had to comprise derivation business rules. As result, implementation documentation including 69 derivation business rules was collected of the following six business rule management systems: 1) Blueriq, 2) Corticon, 3) IBM ODM, 4) Sapiens, 5) OpenRules, and 6) OpenL Tablets.

TABLE II. AMOUNT OF BUSINESS RULES COLLECTED PER CASE.

Business Rule Case	Amount of Business Rules
Morgan	4
RuleSpeak	11
Wan Kadir & Loucopoulos	4
Von Halle	9
RegelSpraak	19
WereWolf	16
Diabetic Patient Monitoring	6
Patient Therapy	12
Tax Return	32
Au pair	1
UServ Product Derby	138
TOTAL	252

V. DATA ANALYSIS

The data analysis comprised three different validation rounds. For each validation round, the same coding procedure and scheme were applied. The coding procedure was established together with a second researcher and based on the Joint Method of Agreement and Difference of [18]. Due to space limitations, only an excerpt of the coding scheme is shown in Figure 1 and Figure 2 including two example business rules from the second validation round.

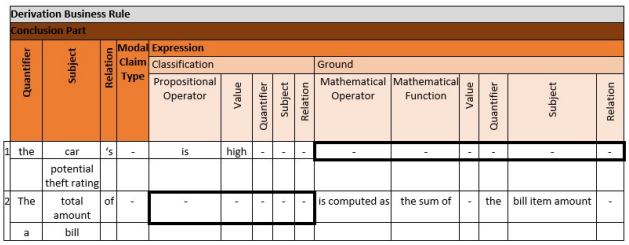

Figure 1. Example mapping of Business Rules on Conclusion Part.

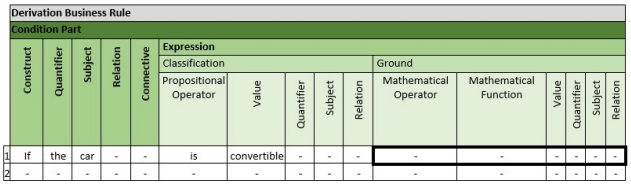

Figure 2. Example mapping of Business Rules on Condition Part.

The coding scheme is split up for readability reasons into two separate tables, where the orange and green cells contain the fundamental constructs and the white cells the data item parts (i.e., business rule parts). The first example business rule (see row no. 1 in Fig. 1 and Fig. 2) corresponds to the coding of the following derivation business rule of the UServ Product Derby case: *"The car's potential theft rating is high if the car is convertible."* To code this business rule, the conclusion and condition part were identified first where "the car's potential theft rating is

high" corresponds to the conclusion part (see Figure 1) and "if the car is convertible" to the condition part (see Figure 2). Subsequently, the conclusion and condition part were disassembled in smaller parts, which were matched onto the fundamental constructs of the coding scheme. For example, the fundamental construct *Quantifier* is two times included as "the" and three instantiations of the fundamental construct *Subject* are identified namely "car", "potential theft rating" and "car".

Although the same coding procedure and scheme were applied for every validation round, some differences can be appointed between the three rounds with regard to the process. During the first round, one researcher coded the 37 collected business rule patterns. In case the researcher was not certain about the coding of particular parts, a second researcher was consulted and the coding process was continued. Subsequently, this second researcher coded a few randomly selected business rule patterns, which were compared with the coded variant of the first researcher. Any discrepancies were discussed until agreement was reached. For the second round, three researcher were involved namely the two researcher of the first round and one additional researcher. This additional researcher acted as reliability coder since the outcome of the coding could be influenced by the mindset and convention of the researcher after the first round. Involving a reliability coder could reduce this effect and could enhance the reliability of the results [26]. So, the 252 selected business rules were coded by both the first researcher and the reliability coder applying the same coding procedure. Besides the use of this coding procedure, the first researcher coded and explained a few example business rules to the reliability coder in advance to ensure that the coding was performed in exactly the same way. After both mappings were conducted, the results were compared and the differences were discussed among all three researchers until agreement was reached again. Prior to the third validation round, a few data items were coded together by the two researchers of the first round. Then, the entire coding procedure of the implemented version of business rules from the implementation documentation of the six selected business rule management systems was completed by the first researcher. Same as applied for the first round, the second researcher was consulted when obscurities emerged. Finally, the second researcher randomly validated a few coded data items. Any anomalies were discussed until agreement was reached, after which the third coding round was finalized.

VI. RESULTS

In this section, the 15 fundamental constructs that are identified to specify a precise and implementation independent derivation business rule are described, which are: conclusion part, condition part, subject, quantifier, relation, expression, classification, value, propositional operator, ground, mathematical operator, mathematical function, modal claim type, construct and connective.

A derivation business rule is composed of two fundamental constructs on the highest abstraction level: the conclusion part and condition part. In the example business rule of Figure 3, the conclusion part is denoted by an orange border and the condition part by a green border. In literature, the conclusion part is also referred to as 'conclusion assertion' or 'then-part', and the condition part as 'if-part' or 'when-part' [3][13]. The conclusion part and condition part are further specified with specific underlying fundamental constructs, which are described in the remainder of this section.

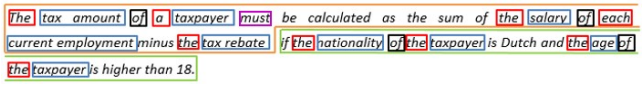

Figure 3. Example business rule indicating the fundamental constructs.

A. Subject

The *subject* is the most fundamental part of a business rule. A *subject* is "*a noun, a thing with an agreed-upon definition, a recognizable business entity* [13][14]." It refers to the business entity on which a conclusion is drawn, as well as the condition(s) that should be applied to reach this conclusion. In the example business rule, subjects are denoted by a blue border, for example: tax amount, taxpayer, and salary (see Fig. 3). In the business rule pattern catalogues and literature, several different names are found to refer to a *subject* like: term, subject, result, value, subj, property of a concept, entity, and attribute [8][11][14].

B. Quantifier

The subject indicates which business entity is applied, the *quantifier* indicates how many or which specific instantiation of the business entity must be applied. This can for example be a specific subject (i.e., **the** subject), one subject (e.g., **a/an** subject) or more subjects (e.g., **each/every** subject). In the example business rule, each quantifier is denoted by a red border (see Fig. 3). In the pattern catalogue of Morgan, this fundamental construct is called a *determiner* [14], and in the business rule language SBVR a *keyword* [10].

C. Relation

In the majority of studied business rules, multiple subjects were present. See for example the business rule in Fig. 3, which includes the subjects "tax amount" and "tax payer". The purpose of this business rule is to conclude something about the combination of both, namely the "tax amount of the tax payer." The question that arises is if this combination must be seen as one subject or as two individual subjects. In practice, both solutions to this problem can be recognized. However, the choice to include subject as a single fundamental construct in the identified set to refer to both "concepts (i.e., entities)" and "properties of concepts (i.e., attributes)", can have a disadvantage. Although it keeps the amount of fundamental constructs limited, it can also make the business rule ambiguous. Therefore, some practitioners choose to add an additional fundamental construct, which addresses this disadvantage. This fundamental construct

 ISBN: 978-1-61208-542-5

specifies the relation between subjects. By means of this relation, the different granularity levels between subjects can be made clear again. Since it is necessary to be able to precisely specify the relation between subjects in a business rule, ensuring an unambiguous and precise business rule, the fundamental construct *relation* is added. This relation is shown by means of a black border in Fig. 3.

D. Expression

Taking the definition of a derivation business rule into account , "a*n expression that evaluates facts, by means of a calculation or classification, leading to a new fact (i.e., conclusion)* [1][13]", both the calculation and classification fundamental construct are seen as a specific type of *expression*. Considering this definition, two statements can be made: 1) facts in a derivation business rule are evaluated by means of a calculation or classification, and 2) a new fact (i.e., conclusion) of a derivation business rule is either determined by a *calculation* or a *classification*. Both the *calculation* and the *classification* are seen as a separate fundamental construct of a derivation business rule, where the calculation is called a *ground*. The fundamental construct is called a *ground* since it has several underlying fundamental constructs, therefore names like computation or calculation are considered as too narrow.

E. Classification

A specific type of expression is the *classification*. On the one hand, in the conclusion part a classification can equate a subject with another subject or a value. For example: "*Food Intake Risk Points of the patient must be equated to 2."* In this example, the subject Food Intake Risk Points is equated to the value 2. On the other hand, in the condition part a classification can check the consistency between a subject and another subject or a value. For example: "*If Solid Intake of the patient is equal to 5 days."* In this case, the subject Solid Intake is compared to the value 5.

F. Propositional Operator

To be able to make the difference between the two classification options (i.e., equate with or check the consistency) clear, a fundamental construct is included. This fundamental construct is called a *propositional operator*, which is underlined in the example business rules above.

G. Value

A fundamental construct that emerged from the coding exercise is *value*. Von Halle and Goldberg refer to a value by the word 'fact' or 'fact value' [13]. The fundamental construct *value* is added to distinguish between constants and variables. Where value is a constant and subjects are used to denote variables.

H. Ground

The second type of expression is the *ground*. On the one hand, in the conclusion part a *ground* can equate a subject with a basic ground. For example: "*Malnutrition Risk Points of the patient must be computed as Weight Loss Risk Points of the patient + Body Mass Index Risk Points of the patient."* On the other hand, in the condition part a ground can compare a subject with another subject, a value, or a basic ground. For example: "*IF Weight Loss of the patient is less than 5%.*"

I. Mathematical Operator and Mathematical Function

To be able to make the difference between the two ground options (i.e., equate with and compare with), a fundamental construct is included. This fundamental construct is called a mathematical operator, which is underlined in the business rules above. In addition to *mathematical operators*, also more sophisticated calculations have to be made. For example: sum, median or cosines. These are called mathematical functions. Since business rule management systems make a difference between the two, both fundamental constructs are included.

J. Modal Claim Type

The fundamental construct *modal claim type* is only applicable for the conclusion part and not for the condition part. This fundamental construct determines how the derivation business rule is imposed. In other words, this fundamental construct defines the modality of the business rule. Examples of these modality options, which occurred during the coding exercise, are: "must" to formulate an obligation or "may" to formulate a permission. In the example business rule of Fig. 3, the modality is denoted by a purple border. By explicitly specifying the modality of a business rule, the intention of the business rule becomes clearer for humans. However, excluding the modality will not change the logic of the business rule. When 'must' is excluded from the example business rule (see Fig. 3), only the representation will change.

K. Construct

The fundamental construct called *construct* is used to indicate a condition part of the business rule, which is repeatedly found in business rule catalogues or languages. Most pattern catalogues only include specific instantiations for this fundamental construct and no overall name is given. For instance, Morgan includes the instantiations 'if or unless' to indicate the condition part [14]. Solely the RuleSpeak pattern catalogue of Hoppenbrouwers provides an overall name for such instantiations namely keywords, which covers the following three: if, when and only if [19]. In computer science, or more specifically with regard to programming languages, the above provided instantiations are commonly referred to as constructs.

L. Connective

In some business rules more than one condition is included, for example: "*IF Age of the patient is equal to 18 AND Liquid Intake of the patient is more than 1 day*." In these cases, the connection between these conditions has to be made clear. Does only one condition has to be met, or a few of them, or maximal one. To indicate the relation between the conditions, the fundamental construct *connective* is added.

 ISBN: 978-1-61208-542-5

VII. DISCUSSION AND CONCLUSION

This research investigated the fundamental constructs of derivation business rules with the purpose of developing a reference framework to evaluate existing business rule languages and business rule management systems. To accomplish this goal, a grounded theory study was executed to derive the minimal set of fundamental constructs needed to define a precise and implementation independent business rule that can be transformed (automatically) to an implementation dependent business rule. The analysis revealed 15 fundamental constructs that are required to do so. Although the three performed validation rounds and the amount of used input data for each round are considered as sufficient (i.e., 37 patterns, 252 business rules, and 6 systems), the size of each data set could be increased for further research to enhance the generalization of the results even further. We believe that this work represents a further step in research on business rule management. Future research will focus on the formulation of patterns including the fundamental constructs. The patterns can be applied to consistently and unambiguously formulate business rules and evaluate the expressiveness of business rule management systems.

REFERENCES

[1] D. Hay, and K. Healy, *Defining Business Rules: What are they really?* (Revision 1.3). [Online]. Available from http://www.businessrulesgroup.org/first_paper/BRG-whatisBR_3ed.pdf: the Business Rules Group 2016-05-15

[2] M. zur Muehlen, and M. Indulska, "Modeling Languages for Business Processes and Business Rules: A Representational Analysis," Information Systems (35:4), pp. 379-390, 2010.

[3] M. Zoet, P. Ravesteyn, and J. Versendaal, "A Structured Analysis of Business Rules Representation Languages: Defining a Normalization Form." Proc. of the Twentieth Australasian Conference on Information Systems (ACIS 2013), Sydney, pp. 1-10.

[4] Y. Wand, and R. Weber, "On the ontological expressiveness of information systems analysis and design grammars," Information Systems Journal (3:4), pp. 217-237, 1993.

[5] D. Warren, L. Pereira, and F. Pereira, "Prolog-the language and its implementation compared with Lisp." ACM SIGART Bulletin (12:64), pp. 109-115, 1977.

[6] B. Von Halle, "Back to Business Rule Basics." Database Programming & Design, pp. 15-18, 1994.

[7] J. Boyer, and H. Mili, Agile Business Rules Development: Process, Architecture and JRules Examples. Heidelberg: Springer, 2011.

[8] B. Von Halle, Business Rules Applied: Building Better Systems Using the Business Rules Approach. New York: Wiley, 2001.

[9] F. Caron, J. Vanthienen, and B. Baesens, "Comprehensive Rule-Based Compliance Checking and Risk Management with Process Mining." Decision Support Systems (54:3), pp. 1357-1369, 2013.

[10] Object Management Group. *Semantics of Business Vocabulary and Business Rules (SBVR) (v1.2)*. [Online]. Available from: http://www.omg.org/spec/SBVR/1.2/ 2016-04-04.

[11] W. Wan-Kadir, and P. Loucopoulos, "Relating Evolving Business Rules to Software Design." Journal of Systems Architecture (50:7), pp. 367-382, 2004.

[12] G. Sangers-van Cappellen, "RegelSpraak (v3.2)." Internal publication of The Dutch Taxation Office, Belastingdienst, 2014.

[13] B. Von Halle and L. Goldberg, The Decision Model: A Business Logic Framework Linking Business and Technology. London: CRC Press, 2009.

[14] T. Morgan, "Business Rules and Information Systems: Aligning It with Business Goals," London: Addision-Wesley, 2002.

[15] M. Zoet, "Methods and Concepts for Business Rules Management," Utrecht: Hogeschool Utrecht, 2014.

[16] B. Glaser, "Theoretical Sensitivity: Advances in the Methodology of Grounded Theory," Mill Valley, CA: Sociology Press, 1978.

[17] J. Mahoney, "Nominal, Ordinal, and Narrative Appraisal in Macrocausal Analysis." American Journal of Sociology (104:4), p.p. 1154-1196, 1999.

[18] J. Mill, A system of Logic, London: Longmans Green, 1906.

[19] S. Hoppenbrouwers, "RuleSpeak grammar." Internal Publication of the Radboud University, Radboud University Nijmegen, 2011.

[20] B. Bouvier. *Decision Model: Determine the Risk of Meeting a Werewolf*. [Online]. Available from: https://dmcommunity.wordpress.com/ 2014-09-09

[21] Business Rules Forum. *Version 2 UServ Product Derby Case Study* [Online]. Available from: http://ai.ia.agh.edu.pl/wiki/_media/hekate:2005_product_derby.pdf 2015-01-04

[22] M. Parish. *Rule Modeling Case Study Generic Diabetic Monitoring (v1.0)*. [Online]. Available from: https://dmcommunity.wordpress.com/case-studies/#DiabeticPatientMonitoring 2015-01-04

[23] J. Feldman. *Preparing a Tax Return (Release 6.1)*. [Online]. Available from: http://openrules.cm/pdf/Tutorial.Dialog1040EZ.pdf: OpenRules 2015-12-12

[24] J. Feldman. *Determine Patient Therapy (Release 6.3.0)*. [Online]. Available from: http://openrules.com/pdf/Tutorial.DecisionPatientTherapy.pdf : OpenRules 2015-12-12

[25] P. Klaasen, "Internal Rules to Determine proffesion of Au Pair" Internal publication of Anonymous Dutch government organization, 2014.

[26] N. Mays, and C. Pope, "Qualitative Research: Rigour and Qualitative Research." BMJ (311:6997), pp. 109-112, 1995.

[27] D. Moody, *"The "physics" of notations: toward a scientific basis for constructing visual notations in software engineering. IEEE Transactions on Software Engineering,"* (35:6), pp. 756-779, 2009.

[28] W. Van der Aalst, A. Ter Hofstede, B. Kiepuszewski, B. and P. Barros. Workflow Patterns. "Distributed and parallel databases" (14:1), pp. 5-51, 2013.

 ISBN: 978-1-61208-542-5