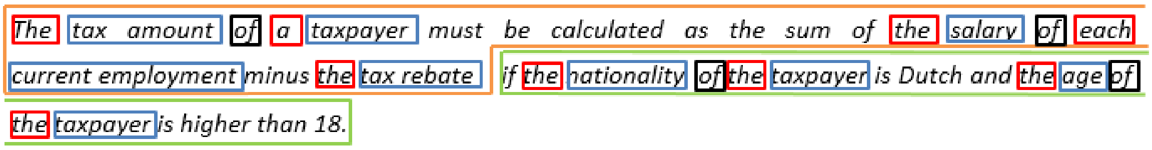
Fundamental Constructs for Derivation Business Rules



The tax amount of a taxpayer must be calculated as the sum of the salary of each current employment minus the tax rebate if the nationality of the taxpayer is Dutch and the age of the taxpayer is higher than 18.

Eline de Haan
Marco Spruit
Martijn Zoet

Technical Report UU-CS-2019-010
December 2019

Department of Information and Computing Sciences
Utrecht University, Utrecht, The Netherlands
www.cs.uu.nl

ISSN: 0924-3275

Department of Information and Computing Sciences
Utrecht University
P.O. Box 80.089
3508 TB Utrecht
The Netherlands

Fundamental Constructs for Derivation Business Rules

Eline de Haan[1], Marco Spruit[1] and Martijn Zoet[2]

[1]Dept. of Information and Computing Sciences, Utrecht University, The Netherlands

[2] Zuyd University of Applied Sciences, The Netherlands

Abstract

Derivation business rules play a 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, current languages are not precise enough and result in ambiguous business rules which cannot be interpreted by an automated information system. So, a need for a precise language to capture business rules can be identified. A specific notation form that can comply with a high level of precision, is a controlled natural language (CNL). During this research, a CNL is created which comprises: 15 fundamental constructs and an underlying formal grammar including 40 grammar rules. The created artifacts have been validated in three different rounds using sequentially 37 patterns, 252 business rules, and six business rules management systems by applying Mill's Method which indicated usefulness and completeness.

Keywords: Business Rules Management, Business Rules, Derivation Business Rules, Controlled Natural Language, Fundamental Constructs, Formal grammar

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*" (Hay & Healy, 2000). 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 (Zur Muehlen & Indulska, 2010).

The abundance of available systems and languages, and the fact that they differ to a large extent regarding their expressive power, causes two problems. The first problem 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 Zoet, Ravesteyn, and Versendaal (2011) and Zur Muehlen and Indulska (2010). Zur Muehlen and Indulska (2010) compared the representational capabilities of four different business rule languages, by mapping the fundamental elements of these languages onto the constructs of the Bunge-Wand-Weber (BWW) representation theory. The BWW representation model allows to analyze the degree to which a modeling language is capable of representing elements of the real world (Wand & Weber, 1993).

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. During this research, the aim was 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. Firstly, the literature review is presented which provides insight into different types of business rules and the specification thereof. Furthermore, the identification and creation of the artifacts is presented. After that, the applied research method to devise and validate the artifacts is explained. In addition, the data collection and analysis process are described. Subsequently, the results that derive from the analysis are presented. At the end of this paper, the conclusions of the study are provided including the contributions and limitations.

Literature

In literature, a "business rule" is defined in a variety of ways which is emphasized by a statement of Von Halle (1994) "*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*". 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) (Boyer & Mili, 2011; Caron, Vanthienen, & Baesens, 2013; do Prado Leite & Leonardi, 1998; Hay & Healy, 2000; Object Management Group, 2008, 2013; Sangers-van Cappellen, 2014; Von Halle, 2001; Wan-Kadir & Loucopoulos, 2004; Zoet, 2014). 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 evaluate facts, by means of a calculation or classification, leading to a new fact (i.e. conclusion)"* (Hay & Healy, 2000; Von Halle & Goldberg, 2009). 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, 5) Decision rules, 6) Calculation rules, and 7) Rounding rules (Boyer & Mili, 2011; Von Halle, 2001; Hay & Healy, 2000; Wan-Kadir & Loucopoulos, 2004; Morgan, 2002; Sangers-van Cappellen, 2014).

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*" (Zoet & Versendaal, 2013). When organization use such an implementation dependent language and switch to a new business rule management system, the business rules have to be re-specified to be processable by this system 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 an specific automated information system*" (Zoet & Versendaal, 2013). So, this second formalism could by applied in multiple environments addressing the problem stated above but is generally not precise enough to be directly executable by an automated information system. A promising solution to address the identified problem is the use of a controlled natural language (CNL) which is a specific notation form that can comply with a high level of precision, without being restricted to be applied by solely one automated information system (Kuhn, 2010, 2013). An additional benefit is that a CNL can resemble a natural language to a certain extend which enables humans to specify and verify the business rules (Kuhn, 2010, 2013). So, the use of a CNL can positively affect the specification and verification of business rules by humans who mostly lack knowledge about formal notations. A lot of research has been done in the field of CNLs which supports that a CNL can provide the following advantages: 1) improve communication among humans, 2) improve machine-assisted translation and reduce overall translation costs, and 3) provide an intuitive representation for formal notations which makes it easier for humans to use and understand (Aikawa, Schwartz, King, Corston-Oliver, & Lozano, 2007; Chervak, Drury, & Ouellette, 1996; Hallett, Scott, & Power, 2007; O'Brien & Roturier, 2007; Ruffino, 1982; Shubert, Spyridakis, Holmback, & Coney, 1995; Temnikova, 2010).

At this moment, merely three CNLs are found devised for the specification of business rules (i.e. RuleSpeak, SBVR Structured English, and Caron et al.(2013)). Since the first two CNLs are not strictly defined by means of a formal grammar (Kuhn, 2010, 2013), they are not precise enough to be interpreted by an information system. In addition, the latter one is only applicable for specifying process business rules. So, these current existing CNLs are not suitable to address the identified research problems. Therefore, a CNL especially focused on specifying derivation business rules is created during this research. A CNL consist of fundamental constructs (i.e. building blocks of the language) and grammar rules that restrict and/or impose the application of these fundamental constructs. Together, the fundamental constructs and grammar rules, constitute a meta-model to which the CNL has to conform. To ground the discussion, this meta-model is first presented in Figure 1 below. In the remainder of this section, each identified fundamental construct along with the grammar rules of the meta-model are explained in succession.

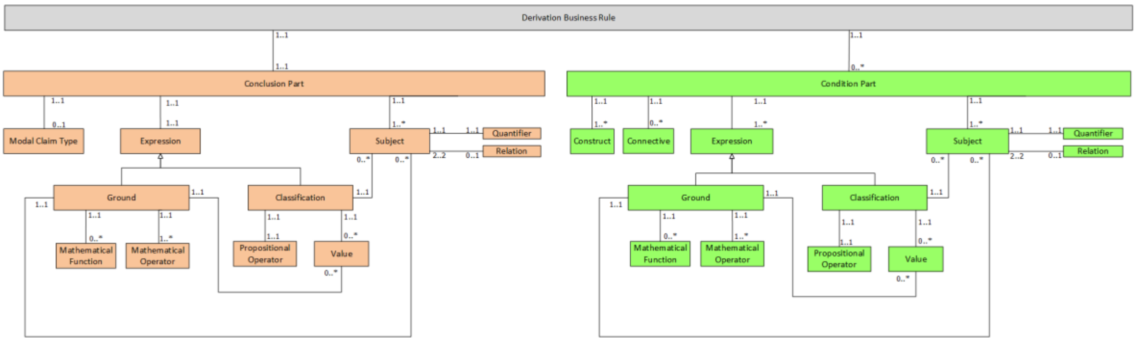
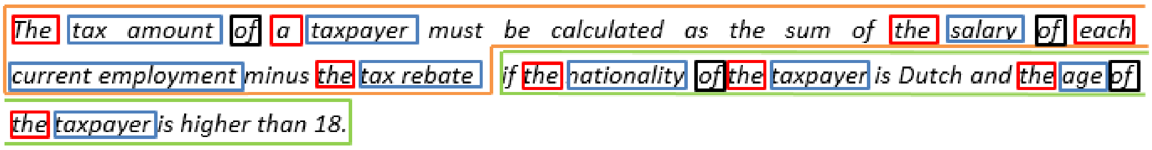


Figure 1: Meta-model

From the literature study emerged that a business rule is composed of two fundamental constructs on the highest abstraction level: the *conclusion part* and *condition part* (Von Halle & Goldberg, 2009; Zoet et al., 2011). In the example business rule of Figure 2, 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' (Von Halle & Goldberg, 2009; Zoet et al., 2011). The relationship between both fundamental constructs and the number of times they occur in one business rule can also differ per source, for example due to personal choice of the business rule modeler or the business rule language that is used (Zoet et al., 2011). Most languages allow the exclusion of a condition or the inclusion of one or multiple conditions and allow only one conclusion. Meeting these requirements ensures the creation of atomic business rules (Boyer & Mili, 2011; Von Halle & Goldberg, 2009; Zoet et al., 2011). Atomic business rules are "*business rules that cannot be further decomposed without losing meaning*" (Boyer & Mili, 2011). In contrast, other languages allow multiple conclusions. Enforcing only one conclusion part is desirable since it can provide several advantages: it eliminates ambiguity of meaning, enhances understandability, maintainability, execution efficiency and manageability, increases ease of validation and implementation, and can eventually prevent redundant or overlapping business rules (Boyer & Mili, 2011; Von Halle & Goldberg, 2009; Zoet et al., 2011). Given these advantages, the creation of atomic business rules will be enforced by the CNL with the following grammar rules: 1) A Derivation business rule consists of **exactly one** Conclusion Part, 2) A Derivation business rule consists of **zero or more** Condition part(s), 3) A Conclusion part belongs to **exactly one** Derivation business rule, and 4) A Condition Part belongs **to exactly one** Derivation business rule.

For both the conclusion part and condition part, many equivalent underlying fundamental constructs are found in literature and business rule pattern catalogues (Caron et al., 2013; do Prado Leite & Leonardi, 1998; Hay & Healy, 2000; Hoppenbrouwers, 2011; Morgan, 2002; Object Management Group, 2013; Sangers-van Cappellen, 2014; Von Halle, 2001; Von Halle & Goldberg, 2009; Wan-Kadir & Loucopoulos, 2004). These will be described in succession. First, the definition of a derivation business rule is taken into account again: "a*n expression that evaluates facts, by means of a calculation or classification, leading to a new fact (i.e. conclusion)*". Where facts are seen as values, pieces of information or data, that can be filled in for a specific business concept incorporated in a business rule (Von Halle & Goldberg, 2009). In general, a business concept is considered as "*a noun, a thing with an agreed-upon definition, a recognizable business entity*" (Morgan, 2002; Von Halle, 2001; Von Halle & Goldberg, 2009). With regard to business rule specification, a business concept is seen as one of the fundamental constructs. In the business rule pattern catalogues and literature, several different names are found to refer to a business concept like: term, subject, result, value, subj, property of a concept, entity, and attribute (Von Halle, 2001; Morgan, 2002; Wan-Kadir and Loucopoulos, 2004; Hoppenbrouwers, 2011; Hay & Healy, 2000, Object Management Group, 2013). For this research, the word *Subject* is chosen from all previous listed alternatives to refer to the fundamental construct. The choice to only include one fundamental construct to refer to these different levels of concepts is made in order to keep the amount of fundamental constructs of the CNL limited. In Figure 2, each subject in the conclusion and condition part is denoted by a blue border. So, both the conclusion part and condition part can consist of multiple subjects. To make the business rule meaningful, it has to include at least one subject to reason about. This leads to the following grammar rules for the CNL: 1) A Conclusion Part consists of **one or more** Subject(s), 2) A Subject belongs to **exactly one** Conclusion Part, 3) A Condition Part consists of **one or more** Subject(s) and 4) A Subject belongs to **exactly one** Condition Part.



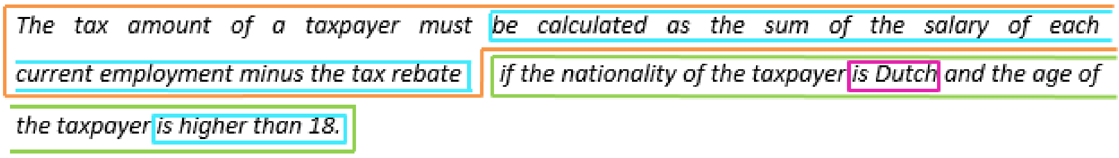
The tax amount of a taxpayer must be calculated as the sum of the salary of each current employment minus the tax rebate if the nationality of the taxpayer is Dutch and the age of the taxpayer is higher than 18.

Figure 2: Example Business Rule

Furthermore, a fundamental construct is found which denotes if the business rule involves: a specific subject (e.g. **the** subject), one subject (e.g. **a/an** subject) or more subjects (e.g. **each/every** subject). In the pattern catalogue of Morgan (2002) this fundamental construct is called a *determiner*, and in the business rule language SBVR as *keyword* (Object Management Group, 2013). For the CNL, this fundamental construct is included as *Quantifier* (see red border in Figure 2). Each quantifier has a direct association with one subject (see blue borders) and vice versa, which makes a business rule more precise and unambiguous (Object Management Group, 2013). Since precision is important for the CNL, this association is covered by the following grammar rules: 1) A Subject is associated with **exactly one** Quantifier, and 2) A Quantifier is associated with **exactly one** Subject.

In literature, also a fundamental construct emerged that specifies the relation/association between subjects called a "fact", "fact type" or the instantiation "(is) of" (Hay & Healy, 2000; Von Halle, 2001; Von Halle & Goldberg, 2009; Object Management Group, 2013). For the CNL, such a fundamental construct will also be included to be able to precisely specify the relation between exactly two subjects (i.e. binary relationship) which makes the different granularity levels between subjects clear again (e.g. entity, attribute). This fundamental construct is called *relation* for the CNL (see black borders in Figure 2), and has to adhere to the following grammar rules: 1) A Subject is associated with **zero or one** Relation, and 2) A Relation is associated with **exactly two** Subjects.

Considering the definition of a derivation business rule once again, two statements can be made: 1) facts (i.e. subjects) in a derivation business rule are evaluated by means of a calculation or classification, and 2) a new fact (i.e. subject) of a derivation business rule is either determined by a calculation or a classification. Where the calculation is called a *ground* with regard to the CNL. The distinction between a ground and classification is supported by different sources, where a ground is referred to with the words: *mathematical calculation*, *algorithm*, *comp*, and a classification with the words: *inference*, *fact*, *state*, and *type* (Hay & Healy, 2000; Morgan, 2002; Wan-Kadir & Loucopoulos, 2004; Hoppenbrouwers, 2011). Therefore, the ground and classification are seen as a separate fundamental construct of a derivation business rule. The example business rule is considered again, in which each occurring classification is depicted with a pink border and each occurring ground is depicted with a light blue border (see Figure 3):



The tax amount of a taxpayer must be calculated as the sum of the salary of each current employment minus the tax rebate if the nationality of the taxpayer is Dutch and the age of the taxpayer is higher than 18.

Figure 3: Example business rule

By means of the example in Figure 3, a few properties of the conclusion and condition part can be demonstrated. A conclusion part should include exactly one classification or ground: 1) without any of these two fundamental constructs (zero), no conclusion can be drawn about 'the tax amount of a taxpayer' and the business rule becomes meaningless, and 2) it cannot include both fundamental constructs since it is logically not possible to say that 'the tax amount' must be calculated and classified at the same time. A condition part should include one or more classifications or grounds: 1) without any of these two fundamental constructs, it becomes meaningless, 2) it can include both fundamental constructs since a condition part can cover more than one condition and each condition should include one classification or ground. Both the ground and classification fundamental construct are seen as a specific type of *expression*. To clarify if a derivation business rule comprises a calculation expression or a classification expression, 'expression' is also included as a fundamental construct of the CNL. These premises corresponds to the following grammar rules: 1) A <u>Conclusion Part</u> consists of **exactly one** <u>Expression</u>, 2) An <u>Expression</u> belongs to **exactly one** <u>Conclusion Part</u>, 3) A <u>Condition Part</u> consists of **one or more** <u>Expression(s)</u>, 4) An <u>Expression</u> belongs to **exactly one** <u>Condition Part</u>, and 5) An <u>Expression</u> **is either** a <u>Ground</u> or a <u>Classification</u>.

Considering the fundamental construct *classification* into more detail, it emerged that a classification in a derivation business rule can: 1) **equate** a subject **with** another subject or a value (i.e. String, Date, Boolean, Number) – in the conclusion part, or 2) **check the consistency** between a subject and another subject or a value – in the condition part. 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 and made obligatory for the CNL. In the business rule language SBVR (Object Management Group, 2013), this fundamental construct is called *instantiation formulation* which classifies things. From reviewing multiple different business rules from different sources, it became clear that instantiations of this fundamental construct occur repeatedly. However, no general name or individual fundamental construct could be found. Therefore, an applicable name for the fundamental construct of the CNL was established: *propositional operator*. Another fundamental construct that emerged from literature and a pattern catalogue is *value* (Von Halle & Goldberg, 2009; Von Halle, 2001). Due to these sources, value is also seen as a separate fundamental construct for the CNL. Altogether, this corresponds to the following grammar rules: 1) A <u>Classification</u> consists of **exactly one** <u>Propositional Operator</u>, 2) A <u>Propositional Operator</u> belongs to **exactly one** <u>Classification</u>, 3) A <u>Classification</u> consists of **zero or more** <u>Value(s)</u>, 4) A <u>Value</u> belongs to **exactly one** <u>Classification</u>, 5) A <u>Classification</u> consists of **zero or more** <u>Subject(s)</u>, 6) A <u>Subject</u> belongs to **exactly one** <u>Classification</u>, and 7) A <u>Classification</u> consists of **at least one** <u>Value</u> or of **at least one** <u>Subject</u>.

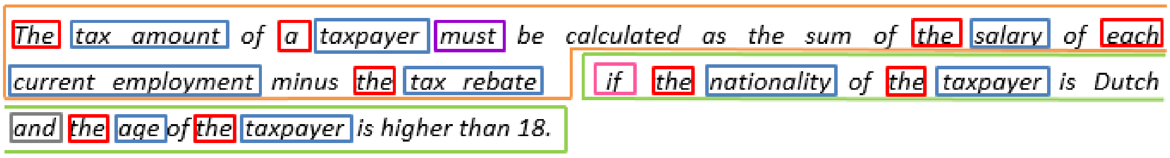
Now, the fundamental construct *ground* will be considered into more detail. A *ground* in a derivation business rule can either: 1) **equate** a subject **with** a basic ground – in the conclusion part, or 2) **compare** a subject **with** another subject, a value, or a basic ground – in the condition part. To be able to make the difference between the two *ground* options (i.e. *equate with* and *compare with*) clear, a fundamental construct will be included and made obligatory for the CNL. This fundamental construct, also covers the following instantiations: +, -, /, *. So, in case a derivation business rule equates or compares a subject with a basic ground, this fundamental construct has to be included multiple times (one to capture the equation or comparison and one or more to capture the other operators). Although instantiations of this fundamental construct are found repeatedly in existing business rule pattern catalogues and literature, no overall name or individual fundamental construct is applied. Therefore, a name that covered the intention of the fundamental construct is selected for the CNL namely *mathematical operator*. Moreover, an additional fundamental construct is incorporated in the CNL to include a mathematical function called a *mathematical function*. Altogether, this corresponds to the following grammar rules: 1) A <u>Ground</u> consists of **one or more** <u>Mathematical Operator(s)</u>, 2) A <u>Mathematical Operator</u> belongs to **exactly one** <u>Ground</u>, 3) A <u>Ground</u> consists of **zero or more** <u>Mathematical Function(s)</u>, 4) A <u>Mathematical Function</u> belongs to **exactly one** <u>Ground</u>, 5) A <u>Ground</u> consists of **zero or more** <u>Value(s)</u>, 6) A <u>Value</u> belongs to **exactly one** <u>Ground</u>, 7) A <u>Ground</u> consists of **zero or**

more Subject(s), 8) A Subject belongs to **exactly one** Ground, and 9) A Ground consists of **at least one** Subject or of **at least one** Value.

After discussing the equivalent fundamental constructs for the conclusion and condition part, now the specific fundamental constructs of each part will be considered. One fundamental construct is found in the pattern catalogues that is only applicable for the conclusion part (Caron et al., 2013; do Prado Leite & Leonardi, 1998; Hoppenbrouwers, 2011; Morgan, 2002; Sangers-van Cappellen, 2014; Von Halle, 2001; Wan-Kadir & Loucopoulos, 2004). This fundamental construct determines how the derivation business rule is imposed; the modality. In most of the patterns, modality is included by means of providing one or more specific options that can be chosen like: 'must' to impose an obligation or 'may' to impose an advice. So, these pattern catalogues did not choose a common name to refer to the fundamental construct. In contrast, the business rule language SBVR uses the word *Keyword* or *modal operator* as umbrella to define the modality options (Object Management Group, 2013). By explicitly specifying the modality of a business rule, the intention of the business rule becomes more clear for humans. However, excluding the modality will not change the logic of the business rule. When 'must' is excluded from the example business rule in Figure 4, only the representation will change from *"is calculated as"* to "*must be calculated as"*. The choice has been made to include the modality as fundamental construct for the CNL, so that business rule authors have the choice to include it to enhance the readability of the business rules for humans. To refer to this fundamental construct, the name *Modal Claim Type* is used for the CNL (see purple border in Figure 4). Taken previous premises into account, the following grammar rules are established: 1) A Conclusion part consists of **zero or one** Modal Claim Type, and 2) A Modal Claim Type belongs to **exactly one** Conclusion Part.

In addition, two fundamental constructs are found that are only applicable for the condition part. One of these fundamental constructs is used to indicate a condition part. Most pattern catalogues only include specific instantiations for this fundamental construct and no overall name. For instance: if, unless, only if (Morgan, 2002; Wan-Kadir & Loucrouplos, 2004; Von Halle, 2001; Caron et al., 2013). Solely Hoppenbrouwers (2011) provides an overall name namely *keywords*, which covers: if, when and only if. In computer science, the above provided instantiations are commonly referred to as *constructs* (2015). Therefore, the word 'construct' is adopted as name for the fundamental construct of the CNL (see pink border in Figure 4). Since a derivation business rule can include several condition parts, the *construct* is made obligatory. In this way, a clear separation between condition parts can be ensured by using the CNL. This corresponds to the following grammar rules: 1) A Condition Part consists of **one or more** Construct(s), and 2) A Construct belongs to **exactly one** Condition Part.

The provided example business rule in Figure 4 includes two condition parts, namely: 1) the nationality of the taxpayer is Dutch, and 2) the age of the taxpayer is higher than 18. In this case, both conditions have to be met (i.e. be true) which becomes clear by means of the word 'and' (see grey border in Figure 4). However, sometimes only one condition has to be met, or a few of them, or maximal one. So in case a derivation business rule includes more than one condition part, the connection between these conditions has to be made clear. In the reviewed business rules catalogues, this is done in different ways. For example, Morgan (2002) applies one fundamental construct: *at least <m> [and not more than <n>] of the following is true*. Von Halle (2001) uses specific instantiations to connect conditions in a binary way (i.e. AND, OR). The SBVR language uses the fundamental construct called *logical operation* (Object Management Group, 2013). Since the fundamental construct connects two or more condition parts, the choice has been made to call it a *connective* with regard to the CNL. This fundamental construct is only necessary when more than one condition part is included in the business rule. This results in the following grammar rules: 1) A Condition Part consists of **zero or more** Connective(s), 2) A Connective belongs to **exactly one** Condition Part, and 3) A Connective must be included to connect **two or more** Conditions.



The tax amount of a taxpayer must be calculated as the sum of the salary of each current employment minus the tax rebate if the nationality of the taxpayer is Dutch and the age of the taxpayer is higher than 18.

Figure 4: Example business rule

In this section, all fundamental constructs of the CNL which are necessary to specify a precise derivation business rule are indicated and described along with the corresponding grammar rules. Here, these fifteen fundamental constructs are listed again for clarity reasons: 1) Conclusion part, 2) Condition part, 3) Modal Claim Type, 4) Construct, 5) Connective, 6) Expression, 7) Subject, 8) Quantifier, 9) Relation, 10) Ground, 11) Classification, 12) Propositional Operator, 13) Value, 14) Mathematical Operator, and 15) Mathematical Function. This set of fundamental constructs could be used as reference point to assess and compare the precision level of business rules languages, addressing the first mentioned research problem.

Research Method

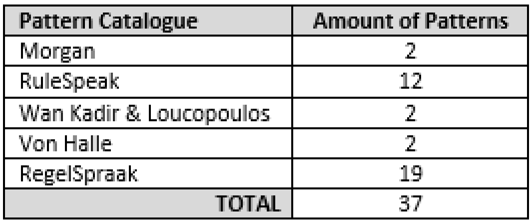
The purpose of this research is to create a CNL to specify a set of derivation business rules once, and which allows automatic transformation of the business rule set to be applicable for multiple business rules management systems. To accomplish this goal, a literature study has been performed in order to identify the fundamental constructs and grammar rules as described in previous section. According to Hevner, March, Park, and Ram (2004), different quality attributes can be appointed to evaluate the devised artifacts of a design-science research like: consistency, reliability, usability, etc. Considering the CNL as one of the artifacts of this design-science research, the usefulness and completeness criteria are taken into account. To validate both criteria with regard to the created CNL, a research approach is needed to identify the similarities and differences between the fundamental constructs and grammar rules of the CNL and: 1) existing business rule patterns, 2) existing business rules, and 3) business rules management systems. A research approach that is especially applicable to investigate the similarities and differences across cases is nominal comparison (Mahoney, 1999). Methods that are based on nominal comparison are Mill's methods (Mahoney, 1999; Mill, 1906). Therefore one of the five Mill's methods (Mill, 1906), the "Joint Method of Agreement and Difference", is adopted as overall data analysis method for the validation of the CNL (i.e. grammar rules and fundamental constructs). In general, 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) (Mill, 1906). With regard to this research, the common denominators correspond to the required fundamental constructs and their relations (i.e. grammar rules) found in each case to be able to specify specific derivation business rules.

Data Collection

During the data collection process, three different data sets are collected. The first data set comprised existing business rule patterns, these were collected in order to investigate if all the building blocks of existing patterns could be captured with the identified fundamental constructs of the CNL. For the second data set, existing business rules were gathered to analyze if specific instantiations could be specified by means of the fundamental constructs. The last data set, including the implementation documentation of business rules management systems, was applied to examine the applicability of the fundamental constructs in an implementation dependent environment.

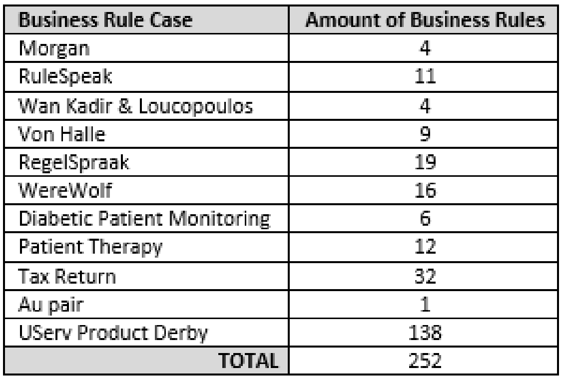
To select the data sets, one overall practical selection criterion was applied namely *site/document access* to be able to use the data for this research. In contrast, the applied theoretical selection criteria differed per data set. For the first data set, business rule patterns were gathered by taken one theoretical criterion 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: 1) Morgan (2002), 2) RuleSpeak of Hoppenbrouwers (2011), 3) Wan-Kadir and Loucopoulos (2004), 4) Von Halle (2001), and 5) RegelSpraak of Sangers-van Cappellen (2014). Table 1 shows the amount of collected patterns per catalogue.

For the second data set, business rules were collected based on one theoretical selection criterion: 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: 1) Morgan (2002), 2) RuleSpeak of Hoppenbrouwers (2011), 3) Wan-Kadir and Loucopoulos (2004), 4) Von Halle (2001), 5) RegelSpraak of Sangers-van Cappellen (2014), 6) WereWolf (DM Community, 2015), 7) Diabetic Patient Monitoring (Parish, 2014), 8) Patient Therapy (Feldman, 2014), 9) Tax Return (Feldman, 2011), 10) Au Pair (Anonymous Dutch government organization), and 11) UServ Product Derby (BBC, 2004). This sampling strategy is followed in order to cover a wide range of domains where business rules are applied. Table 2 lists the amount of selected business rules per case.



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

Table 1: Figures Data Collection Round 1



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

Table 2: Figures Data Collection Round 2

With regard to the third data set, two theoretical criteria were applied. The first theoretical criterion to select the business rules 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 comprised derivation business rules. As result, implementation documentation of the following six BRM systems was collected: 1) Blueriq, 2) Corticon, 3) IBM ODM, 4) Sapiens, 5) OpenRules, and 6) OpenL Tablets.

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 Mill (1906). Due to space limitations, only an excerpt of the coding scheme is shown in Table 3 and Table 4 including two example business rules from the second validation round. 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 nr. 1 in both tables) 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 Table 3) and "if the car is convertible" to the condition part (see Table 4). 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".

	Derivation Business Rule														
	Conclusion Part														
	Quantifier	Subject	Relation	Modal Claim Type	Expression										
					Classification					Ground					
					Propositional Operator	Value	Quantifier	Subject	Relation	Mathematical Operator	Mathematical Function	Value	Quantifier	Subject	Relation
1	the	car	‘s	-	is	high	-	-	-	-	-	-	-	-	-
		potential theft rating													
2	The	total amount	of	-	-	-	-	-	-	is computed as	the sum of	-	the	bill item amount	-
	a	bill													

Table 3: Coding scheme of the conclusion part including two examples Business Rules

	Derivation Business Rule															
	Condition Part															
	Construct	Quantifier	Subject	Relation	Connective	Expression										
						Classification					Ground					
						Propositional Operator	Value	Quantifier	Subject	Relation	Mathematical Operator	Mathematical Function	Value	Quantifier	Subject	Relation
1	If	the	car	-	-	is	convertible	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 4: Coding scheme of the condition part including two example Business Rules

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 (Mays & Pope, 1995). 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 of the implemented version of the business rules from the implementation documentation of the six selected BRM 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.

Results

This section presents the results of the three validation rounds. From the first and second round emerged that all 37 patterns and 252 business rules could be coded, but some cells remained empty as indicated with a dash in Table 3 and Table 4. Since these empty cells could indicate that the establishment of the fundamental constructs for the CNL was incorrect (e.g. superfluous constructs), these cells were investigated further from which two overall reasons emerged. Firstly, in case a pattern or business rule comprised a ground, all cells of the fundamental constructs related to a classification remained empty (see thick border in example business rule 2 in Table 3). The other way around, mapping a classification business rule or pattern resulted in empty cells for the fundamental constructs related to a ground (see example business rule 1 in Table 3). In addition, some patterns or business rules solely represented the conclusion part omitting the condition part and vice versa. As a result, all fundamental constructs associated with one of these main parts remained empty (see example business rule 2 in Table 4). This first finding indicates high cohesiveness between specific fundamental constructs, providing substantiation for the established grammar rules of the CNL. Furthermore, this finding provides a logical explanation for the fact that many cells remained empty and contradicts the assumption that these fundamental constructs are superfluous. The second overall finding was that individual fundamental constructs remained empty. Therefore, these were analyzed further by calculating the amount of empty cells per fundamental construct for both the first and second mapping round (see Table 5).

Fundamental Constructs		Amount of empty cells Round 1	Amount of empty cells Round 2
Quantifier	(Conclusion Part)	0	67
Subject	(Conclusion Part)	0	0
Relation	(Conclusion Part)	2	101
Modal Claim Type	(Conclusion Part)	3	119

Propositional Operator	(Conclusion Part - Classification)	0	0
Value	(Conclusion Part - Classification)	4	8
Quantifier	(Conclusion Part - Classification)	3	72
Subject	(Conclusion Part - Classification)	2	70
Relation	(Conclusion Part - Classification)	7	77
Mathematical Operator	(Conclusion Part - Ground)	0	0
Mathematical Function	(Conclusion Part - Ground)	1	47
Value	(Conclusion Part - Ground)	4	23
Quantifier	(Conclusion Part - Ground)	2	52
Subject	(Conclusion Part - Ground)	2	6
Relation	(Conclusion Part - Ground)	5	68
Construct	(Condition Part)	0	0
Quantifier	(Condition Part)	0	109
Subject	(Condition Part)	0	0
Relation	(Condition Part)	0	124
Connective	(Condition Part)	1	93
Propositional Operator	(Condition Part - Classification)	0	0
Value	(Condition Part - Classification)	0	11
Quantifier	(Condition Part - Classification)	0	152
Subject	(Condition Part - Classification)	0	149
Relation	(Condition Part - Classification)	0	160
Mathematical Operator	(Condition Part - Ground)	0	0
Mathematical Function	(Condition Part - Ground)	0	48
Value	(Condition Part - Ground)	0	4
Quantifier	(Condition Part - Ground)	0	47
Subject	(Condition Part - Ground)	0	43
Relation	(Condition Part - Ground)	0	47

Table 5: Amount of Empty Cells per Fundamental Construct

Considering Table 5 above, the fundamental constructs Quantifier, Relation and Modal Claim Type remained empty several times. However, both the Quantifier and Relation are significant to retain for the CNL be able to specify precise and unambiguous derivation business rules. In addition, the Modal Claim Type can be included to enhance the understandability of business rules. By explicitly specifying the modality of a business rule, the intention becomes more clear for humans. For instance, does the business rule impose a requirement (i.e. must) or does the business rule impose an advice (i.e. may). However, including or excluding the modality will not change the logic of a business rule. When 'must' is included in the example business rule 2 in Table 3, only the representation will change from "*is computed as*" to "*must be computed as"*. Therefore, these three fundamental constructs are considered as necessary and stay included in the CNL. In both the conclusion and condition part, the following fundamental constructs related to a classification remained empty more often: Value, Quantifier, Subject and Relation. This can be explained by the fact that two types of classifications exist which are mutual exclusive, since it is impossible to classify 'something' as a Value and a Subject at the same time. When a pattern was targeted at classifying 'something' as a Value, the cells of the Quantifier, Subject and Relation stayed empty. In addition, the fundamental constructs Mathematical Function, Value, Quantifier, Subject, and Relation remained also empty a few times with regard to a ground. A possible explanation for this finding is that these data items only specified a very simple or specific calculation. Lastly, the Connective remained empty which can be explained by the fact that these data items only included the specification of one condition. Although these fundamental constructs remained empty, they appear in practice as can be concluded from the other data for which the cells did not remain empty. Therefore, they also stay included in the CNL.

Regarding the third validation round, the coding showed that it was possible to capture the implemented version of the business rules with the fundamental constructs of the CNL. However, the coding also indicated that not every Business Rules Management System incorporated and supported all the 15 fundamental constructs.

In summary, the three validation rounds showed that every data item could be coded ensuring that no fundamental constructs were lacking. Furthermore, a logical explanation could be found when a cell of a fundamental construct remained empty but most importantly every fundamental construct was filled at least once. Therefore, it can be concluded that the CNL complies with the *usefulness* and *completeness* criteria. In addition, the seven fundamental constructs that appeared to be zero times empty for both rounds (see white cells in Table 5) correspond to fundamental constructs that are made obligatory for the CNL. This latter finding provides a justification for the establishment of the grammar rules related to these fundamental constructs. Altogether, the coding process showed which fundamental constructs occurred in each data set and it showed the number of similarities and differences between the cases. So, the application of the Mill's method provided a further substantiation for retaining the identified fundamental constructs and grammar rules in the CNL besides the already found support from literature.

Discussion & Conclusion

This study provides further insight into the fundamental constructs of derivation business rules, which can be used for the comparison and evaluation of business rule languages and systems for example with respect to their expressive power. In this way, this research can support organizations in the decision making process for selecting an applicable business rule language and/ or system. Furthermore, a scientific contribution can be identified by adding a new type of CNL to the scientific knowledge base.

Besides the contributions, also a number of limitations can be appointed concerning the results of this research. A first limitation that can be identified is related to the validation of the research. McGrath (1981) calls validation 'the triangle of evil': 1) maximal measurement precision, 2) maximal focus on realism of research context, and 3) maximal focus on generalizability. This research focused on the first two aspects, by conducting experiments on case study data. The third aspect generalizability cannot be ensured due to the sample sizes. Although 37 patterns, 252 business rules, and 6 systems are used for the validation, the size of each data set could be increased for further research to enhance the generalization of the results even further. A second limitation corresponds to the fact that it is not certain whether the co-founding variables are eliminated. Since the majority of the coding process was conducted by solely one researcher, there is a possibility that the internal validity is threatened by the so called 'instrumentation threat'. In this case, the researcher is considered as the measurement device which could gain experience gradually traversing the three validation rounds. To address this possible threat, already one researcher and one reliability coder were involved during the validation process. However, it is recommended to involve more reliability coders for further research.

In conclusion, this research investigated how a set of derivation business rules could be specified once, and subsequently could be transformed into various implementation dependent languages based on the following research question: *"How can derivation business rules be specified precisely and implementation independent?"* To achieve this goal, a CNL is created which consists of a set of 15 fundamental constructs, and an underlying formal grammar including 40 grammar rules. The gained knowledge and resulting artifacts of this research can serve as basis for future research to further improve or extend the created CNL or to investigate if these artifacts are applicable for other types of business rules besides derivation business rules.

References

Aikawa, T., Schwartz, L., King, R., Corston-Oliver, M., & Lozano, C. (2007). Impact of Controlled Language on Translation Quality and Post-editing in a Statistical Machine Translation Environment. *Proceedings of the MT Summit XI*, Copenhagen, 1-7.

Boyer, J., & Mili, H. (2011). *Agile Business Rules Development: Process, Architecture and JRules Examples*. Heidelberg: Springer.

Caron, F., Vanthienen, J., & Baesens, B. (2013). Comprehensive Rule-Based Compliance Checking and Risk Management with Process Mining. *Decision Support Systems, 54*(3), 1357-1369.

Chervak, S., Drury, C. G., & Ouellette, J. P. (1996). Field Evaluation of Simplified English for Aircraft Workcards. *Proceedings of the Tenth FAA/AAM Meeting on Human Factors in Aviation Maintenance and Inspection*, Alexandria, Virginia, 1-16.

DM Community. (2015). *Challenges.* from https://dmcommunity.wordpress.com/

do Prado Leite, J. C. S., & Leonardi, M. C. (1998). Business Rules as Organizational Policies. *Proceedings of the Ninth International Workshop on Software Specification and Design*, Ise-Shima, 68-76.

Hallett, C., Scott, D., & Power, R. (2007). Composing Questions through Conceptual Authoring. *Computational Linguistics, 33*(1), 105-133.

Hay, D., & Healy, K. (2000). *Defining Business Rules: What are they really?* (Revision 1.3). http://www.businessrulesgroup.org/first_paper/BRG-whatisBR_3ed.pdf: the Business Rules Group.

Hevner, A., March, S., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *28*(1), 75-105.

Hoppenbrouwers, S. (2011). *RuleSpeak grammar*. Radboud University Nijmegen.

HP. (2015). *IF Construct.* from h21007.www2.hp.com/portal/download/files/unprot/fortran/docs/lrm/lrm0143.htm

Kuhn, T. (2010). *Controlled English for Knowledge Representation*. Doctoral Thesis. Zurich: University of Zurich.

Kuhn, T. (2013). A Survey and Classification of Controlled Natural Languages. *Computational Linguistics*, *40*(1), 1-50.

Mahoney, J. (1999). Nominal, Ordinal, and Narrative Appraisal in Macrocausal Analysis. *American Journal of Sociology*, *104*(4), 1154-1196.

Mays, N., & Pope, C. (1995). Qualitative Research: Rigour and Qualitative Research. *BMJ, 311*(6997), 109-112.

McGrath, J. E. (1981). Dilemmatics: The Study of Research Choices and Dilemmas. *American Behavioral Scientist*, *25*(2), 179-210.

Mill, J. (1906). *A system of Logic*. London: Longmans Green.

Morgan, T. (2002). *Business Rules and Information Systems: Aligning IT with Business Goals*. London: Addision-Wesley.

O’Brien, S., & Roturier, J. (2007). How Portable are Controlled Language Rules? A Comparison of Two Empirical MT Studies. *Proceedings of the MT Summit XI*, Copenhagen, 345-352.

Object Management Group. (2008). *Semantics of Business Vocabulary and Business Rules (SBVR)* (v1.0). http://www.omg.org/spec/SBVR/1.0/PDF: Object Management Group.

Object Management Group. (2013). *Semantics of Business Vocabulary and Business Rules (SBVR)* (v1.2). http://www.omg.org/spec/SBVR/1.2/: Object Management Group.

Ruffino, J. R. (1982). Coping with Machine Translation. In V. Lawson (Ed.), *Practical Experience of Machine Translation* (pp. 57-60). Amsterdam: North-Holland Publishing Company.

Sangers-van Cappellen, G. (2014). *RegelSpraak* (v3.2). Belastingdienst.

Shubert, S., Spyridakis, J. H., Holmback, H. K., & Coney, M. B. (1995). The Comprehensibility of Simplified English in Procedures. *Journal of Technical Writing and Communication, 25*(4), 347-369.

Temnikova, I. (2010). Cognitive Evaluation Approach for a Controlled Language Post--Editing Experiment. *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, Malta, 3485-3490.

Von Halle, B. (1994). Back to Business Rule Basics. *Database Programming & Design*, 15-18.

Von Halle, B. (2001). *Business Rules Applied: Building Better Systems Using the Business Rules Approach*. New York: Wiley.

Von Halle, B., & Goldberg, L. (2009). *The Decision Model: A Business Logic Framework Linking Business and Technology*. London: CRC Press.

Wan-Kadir, W., & Loucopoulos, P. (2004). Relating Evolving Business Rules to Software Design. *Journal of Systems Architecture, 50*(7), 367-382.

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

Zoet, M. (2014). *Methods and Concepts for Business Rules Management*. Doctoral Thesis. Utrecht: Universiteit Utrecht.

Zoet, M., Ravesteyn, P., & Versendaal, J. (2011). A Structured Analysis of Business Rules Representation Languages: Defining a Normalization Form. *Proceedings of the Twentieth Australasian Conference on Information Systems*, Sydney, 1-10.

Zoet, M., & Versendaal, J. (2013). Business Rules Management Solutions Problem Space: Situational Factors. *Proceedings of the Seventeenth Pacific Asia Conference on Information Systems*, Jeju, 1-13.

Zur Muehlen, M., & Indulska, M. (2010). Modeling languages for business processes and business rules: A representational analysis. *Information Systems, 35*(4), 379-390.