A quality evaluation framework for a CNL for agile law execution

Ilona Wilmont
HAN University of Applied Science,
P.O. Box 2217, 6802 CE, Arnhem,
The Netherlands
Radboud University Nijmegen,
P.O. Box 9010, 6500 GL, Nijmegen,
The Netherlands
ilona.wilmont@han.nl

Diederik Dulfer
Dutch Tax Administration,
John F. Kennedylaan 8,
7314 PS, Apeldoorn,
The Netherlands
dph.dulfer@belastingdienst.nl

Jan Hof
Dutch Tax Administration,
John F. Kennedylaan 8,
7314 PS, Apeldoorn,
The Netherlands
ja.hof@belastingdienst.nl

Mischa Corsius
HAN University of Applied Science,
P.O. Box 2217, 6802 CE, Arnhem,
The Netherlands
mischa.corsius@han.nl

Stijn Hoppenbrouwers
HAN University of Applied Science,
P.O. Box 2217, 6802 CE, Arnhem,
The Netherlands
Radboud University Nijmegen,
P.O. Box 9010, 6500 GL, Nijmegen,
The Netherlands
stijn.hoppenbrouwers@han.nl

Abstract

The Dutch Tax Administration has developed and exploited a CNL, RegelSpraak, to automate law execution. This CNL is meant to be comprehensible for legal specialists, IT developers and computers. However, quality assessment of rule patterns and their ability to express all relevant tax laws is currently not possible. In this study we evaluate potential quality criteria that offer the capability to evaluate rule pattern quality based on semantic expressive power, cognitive usability and functional and structural correctness. We design a quality framework based on insights from literature review, interviews and observations, which has been qualitatively operationalized. Initial results suggest that they touch on relevant variables, but that further quantitative operationalization and testing of the framework's usability in practice are needed.

1 Introduction

RegelSpraak is a Dutch controlled natural language (CNL) developed in 2008 by the Dutch Tax Administration (DTA), which has since been employed for the purpose of law execution. It is meant to be comprehensible for legal specialists, IT developers and computers alike, offering a common ground for unambiguous communication about legislation.

RegelSpraak is based on RuleSpeak (RuleSpeak, 2021) and has a strong pragmatic focus. It is currently being employed in several projects for the specification and implementation of tax rules, such as income tax, corporation tax and wage tax. For these selected purposes, RegelSpraak functions adequately. However, it is not possible to assess whether RegelSpraak, given its current state, is capable of expressing all relevant tax laws. Nevertheless, new patterns for the formulation of rules are being developed as deemed necessary. The DTA is therefore seeking to develop a framework

for evaluation of rule pattern quality which can be directly used during the process of rule pattern development rather than afterwards. In addition to rule pattern assessment, the framework must be able to assess the quality of the CNL as a whole.

This study focuses on the aspects of and perspectives on quality which may be relevant to assess in the context of a CNL aimed at automated execution of rules. We evaluate potential criteria for assessment of the understandability of RegelSpraak and the quality of a rule pattern, as well as whether additional quality measures are needed. These insights can further the development of rule patterns, as well as the tooling used to implement them.

1.1 The basic structure of RegelSpraak

Two categories of CNLs are those improving readability for human users, and those facilitating an automated semantic analysis (Tommila and Pakonen, 2014). RegelSpraak falls into both categories. The foundations of RegelSpraak lie in the Dutch version of RuleSpeak (RuleSpeak, 2021): a set of guidelines to formulate business rules in a precise yet user-friendly manner. As opposed to RuleSpeak, though, RegelSpraak has a predefined syntax and semantics. The syntax is enforced though the defined language patterns, which have a precisely defined semantics. The use of examples of the intended rule behaviors ensure that the rule analyst understands the implication of the rules specified. These language patterns facilitate the automated execution of the rules, which is a major step forward for the DTA's IT implementation process.

Table 1 presents the types of rules RegelSpraak employs. A RegelSpraak rule always has the following format: [RESULT] IF [CONDITION(S)]. A condition compares attributes, which can have boolean or numerical values, or be dates, enumerations or roles. The result is executed as a consequence of the successful evaluation of the conditions. The results and conditions are connected using carefully composed Dutch phrases to maximize the resemblance to a natural sentence.

The following fictitious example is a RegelSpraak calculation rule. The results part is specified before the listing of the conditions.

```
Rule tax based on travel time
  Valid from 2011 to 2015

    The tax on a train journey must be
    calculated as the TAX_PERCENTAGE of
    (900 minus the travel time by train
    in minutes of the taxed journey)
    if the journey meets all of the
    following conditions:

      - the type of the travel tax is
        equal to 'gross tax based on
        travel time'
      - the travel time by train in
        minutes of the taxed journey is
        larger than or equal to 600
      - the travel time by train in
        minutes of the taxed journey is
        smaller than 900.
```

The aforementioned rule is an instantiation of the following rule pattern:

```
Rule <description>
 Valid from <year> to <year>

   The <attribute> of an <object> must
   be calculated as the <expression>

   if the <object> meets <all | one
   | at least n | at most n | none
   | exactly n> of the following
   conditions:

     - <condition>
     - <condition>.
```

1.2 Two cognitive perspectives on rule authoring

Rule authoring is a cognitively demanding task for analysts, and it is therefore important to review some cognitive process requirements. Analogies from modeling and writing research are used (Wilmont, 2020).

Early research on story writing found that writers use three strongly intertwined processes: *planning*, *sentence generation*, and *revision* (Hayes and Flower, 1986). Furthermore, writing is goal directed and goals are hierarchically organized. Skilled writers comment on their goals early in the process and continue to define subgoals after having determined the main goal. This process can also be observed in conceptual and scientific modeling (Ross et al., 1975; Sins et al., 2005). The ability to abstract the essential meaning from concrete activities or observations, as well as the ability to engage in metacognitive monitoring behavior, are important success factors in modeling (Wilmont et al., 2013). Tax rules can be viewed as a model of the legal reality, therefore the importance of these skills applies to rule authoring as well.

Following Gemino and Wand (2003), during rule authoring an analyst can take on one of two perspectives: *reader* or *writer*. The rule writer is the active role, responsible for translating knowledge into

Rule type	Description
Decision rule	Deduces a true/false result
Calculation rule	Calculates a number or amount
Time/date rule	Calculates dates
Consistency rule	Validates whether the value of an attribute is consistent when compared to the value of another attribute, or a validity check for one attribute
Assigning characteristics	Determines the value of a characteristic
Initialisation rule	Assigns a value to an attribute if the value is not being determined by a rule or an input message
Object creation rule	Creates a new object
Fact creation rule	Creates a new fact

Table 1: Types of RegelSpraak rules

rules. On the contrary, the rule reader is passive, concerned with reading the rules and constructing a mental model of the domain based on the information contained in the selected rules, RegelSpraak's grammatical structure and internalized knowledge gained from prior experiences.

This dichotomy translates perfectly to the roles assumed in rule development teams. Typically, they are made up of tax experts and rule analysts. Tax experts have highly specialized knowledge on legal contexts in which the rules are meant to function, as opposed to rule analysts who have intricate knowledge of RegelSpraak syntax and methods for rule authoring. Therefore, tax experts are primarily concerned with reviewing what the rule analysts have created. However, despite the fact that each team member is likely to operate in a dominant role, everyone will inevitably have to switch between the roles occasionally during the process of rule authoring. It is therefore essential that the quality assessment framework flexibly accommodates reflection on team progress from both the reader and writer perspectives.

1.3 Evaluating CNL quality

As RegelSpraak is constantly evolving, fast and targeted quality evaluation during development is crucial. Interestingly, earlier work on the evaluation of CNLs has focused mostly on the usability aspects, in particular *understandability* and *learnability* (Kuhn, 2009). However, during interpretation tasks it proved difficult to determine whether people truly understand the CNL or whether they follow syntactical patterns to complete the task. Kuhn (2009) demonstrated that graphical representations, which prompted people to reason about the meaning of statements, encouraged understanding. However, RegelSpraak projects may contain over 1000 rules, in which case graphical representations would not be suitable. Additionally, basic knowledge of logic helped to evaluate statements in which potentially ambiguous constructs such as an inclusive OR are used. This implies that for the user, some background knowledge is desirable.

Concerning RegelSpraak itself, a set of rules can be viewed as a model of legal reality. From this point of view, conceptual modeling quality criteria can be applied for evaluation. An early quality framework differentiated between syntactic, semantic and pragmatic quality (Lindland et al., 1994). This framework has been expanded and revised to become the SEQUAL (SEmiotic QUALity) framework for model evaluation (Krogstie et al., 2006), which provides theoretical, qualitative guidelines. It encompasses the following aspects: physical representation and availability, empirical layout and readability, syntactic requirements, semantic understandability, pragmatic impact of the activated model, socially mediated shared understanding and organizational aspects such as which parts of the model can be changed.

For our purposes the original three aspects are leading. Syntactic quality focuses on how to enhance the formal syntax, to implement error detection, error prevention and recovery. Semantic quality refers to the human interpretation of the model, the relation between one's knowledge and the model. Pragmatic quality concerns model activation; that is, how the models are interpreted by both social and technical actors. Technical interpretation demands complete models with operational semantics, whereas social interpretation requires flexibility and effective abstraction. We use the aspects from the revised SEQUAL version to guide the development of the RegelSpraak evaluation criteria.

2 Methodology

For this exploratory project, we organized our activities in three phases: 1) an exploration of potential quality criteria, 2) an investigation of the criteria in terms of user perspectives and cognitive processes, and 3) integration into a conceptual framework.

The exploration of quality criteria was done through a literature review, which included both scientific and grey literature. The domains of CNL, software quality and linguistic quality were queried. Additionally, documentation provided by the DTA reviewed.

For the exploration of user perspectives, we conducted interviews with five DTA employees who worked in varying functions. The sample consisted of one developer and four rule analysts, of whom one had a legal background and one was responsible for RegelSpraak training courses. We set up an interview guide to include the following aspects: generic linguistic quality aspects, the role of RegelSpraak in the daily workflow, difficulties and best practices when using rules and patterns, incentives for new pattern development, the documentation of design decisions and how best to teach it to new employees.

Interviews were audio-recorded with the participants' consent, and analysed directly from the audio files. We denoted timestamps and took elaborate notes whenever utterances of the following types were encountered:

- Decision-making aspects of determining rule and pattern quality,
- Underlying principles and frameworks which analysts and developers (unconsciously) use to test and validate new rules and patterns during development,
- Explicit mentions of routine actions.

Furthermore, we conducted a semi-structured, non-participant observation of a rule pattern design session, in which three rule analysts, one developer and a project leader took part. This session was held via Cisco Webex, and the researchers switched off their audio and cameras to appear as unobtrusive as possible. The session was video-recorded with the participants' consent. The following structural framework was used to analyse the video:

- Spontaneous mention of quality criteria (either implicit or explicit),
- Differences in interpretation of quality criteria,
- Mention of operational norms for quality criteria,
- Allocation of attention to different quality criteria,
- The use of the human versus the rule engine perspective in the discussion,
- The process of decision making in the context of differing opinions,
- Quality aspects which might block further pattern development progress.

The final integration phase took place in short design cycles in which DTA employees were actively involved. Significantly recurring quality criteria were deduced by triangulating the literature review, interview and observation results. The integration process focused on formulating operational criteria for each of the quality criteria, and on designing a user-friendly procedure for testing rule pattern quality in daily practice.

It should be noted that the primary scope for this initial phase of the RegelSpraak evaluation project is focused on standardizing the best practices to align understandability for users and formal syntax as produced by rule analysts. A formal, computational evaluation of the resulting RegelSpraak rules is beyond the scope of this paper.

3 Results

Our multi-faceted exploration yielded insights from literature review, interviews, observations and active design cycles. The final set of quality criteria that we selected consisted of the following: Readability, Recognizability, Explainability, Usability, Completeness, Syntax, Complexity, Efficiency, Compositionality, Domain Independence, Technical Executability. These criteria were then differentiated and described from both a *fiscal* and a *computational* perspective. The resulting categorizations are shown in Section 3.3: the Fiscal perspective is shown in Table 2, and the Computational perspective in Table 3.

We continue to describe how the iterative research process resulted in the design of our framework. Firstly, the results of our quality criteria exploration are discussed, followed by the integration

with user perspectives and finally the translation to actual framework development.

3.1 Phase 1: An exploration of quality criteria

We approached the exploration of quality criteria from four perspectives: linguistics, conceptual modeling, software engineering and usability. Major themes that emerged were semantic expressiveness, cognitive usability and the functional and structural aspects associated with formal languages.

3.1.1 Semantic expressiveness

As described in Section 1.3, syntax, semantics and pragmatics form the basis for the framework. At the intersection of linguistics, conceptual modeling and usability, the ability to express the desired semantics in a rule pattern is one of the most crucial themes. Ultimately, if the model does not convey the right message to the stakeholders for whom it was created, the relevance of all other aspects of model quality may be questioned. In the case of RegelSpraak, semantic consensus between fiscal experts, rule analysts and IT developers must be achieved.

Legal texts are often formulated with precision and disambiguation in mind, yet many linguistic constructs are used to express desired nuances. Despite RegelSpraak's user-centred focus, its main goal is to automate rules, for which human texts have to be transformed to fit into a formal syntax. This requires a delicate balance between the abstraction levels of both rule patterns and rules, as they must be designed to express as many diverse meanings as possible yet not become overly generic.

Guidelines for how to incorporate maximal expressive power in a formal rule pattern can be found in Grice's maxims (Grice, 1991). In short, Grice's maxims dictate that contributions to a conversation should be maximally informative within minimal wording, contain no untruths, be precise, unambiguous and above all relevant. Whenever there is a possibility that ambiguity might occur, for instance with 'can'-patterns, they must be tested extensively against a diverse sample of concrete rules until there is no room left for interpretation. This records how fiscal concepts and norms are interpreted and applied. From this test, essential relevance and pattern complexity minimization can be determined.

Furthermore, automated rule engines can only deal with a specified vocabulary, which further emphasizes the importance of a concise vocabulary. If patterns are designed to accommodate this vocabulary and structured in such a way that chosen signalling words allow minimal freedom of interpretation, RegelSpraak forms an excellent basis for facilitating flexible, or agile, law execution.

3.1.2 Cognitive usability

Cognitive usability requirements featured prominently in the interviews. The concepts of *understandability*, *readability* and *explainability* were considered to have the highest priority, although the interview results suggest that their precise definitions and relations to each other are all but clear-cut.

Understandability is required in different forms. For one, rule analysts are taught the existing rule patterns and what they can express. It is therefore essential that the logic and abstraction associated with rule patterns are understandable. For another, in the current situation fiscal experts do not write rules. They only read what the analysts have written, and listen to analysts explaining their creations. Therefore, readability and explainability may be seen as critical understandability prerequisites.

When the interviewees were prompted to comment on a rule pattern the researchers considered poorly understandable, aspects such as sentence complexity, poor abstraction (*"too much detail in one construct", "using a summation or a basic calculation should be coverable with 'calculation', and the context should clarify how the calculation is to be performed"*), structure (*"diverse options have been written out in different places in the pattern"*) and clarity of option meanings (*"the options in this pattern raise too many questions. For example, why is 'first/last' here? Is this variable the first of a sequence?"*) were mentioned. From this, we deduce that structure and abstraction level are critical understandability requirements for the rule pattern. For the terminology used in the rule pattern, unambiguous interpretation is most important.

One of the interviewees suggested the following: *"it would be good to provide such generic patterns with instantiated examples, but because of a combination of priority and time this unfortunately does not happen well enough"*. Research confirms that modelers use abstraction skills to actively connect generic concepts to concrete activities and objects through generalization and instantiation (Theodor-

akis et al., 1999). This is therefore another critical aspect contributing to understandability to be included in the quality evaluation framework.

3.1.3 Functional and structural evaluation

RegelSpraak has many parallels with programming languages. For functional and structural assessment, inspiration may thus be taken from existing software quality criteria, such as the ISO/IEC 25010 (International Organization for Standardization, 2017). Two perspectives are relevant: *functional quality*, which measures the extent to which the software satisfies the functional requirements for its intended use, and *structural quality*, which measures the extent to which the software satisfies non-functional requirements such as robustness, usability, maintainability, compositionality or complexity. Functional assessment must be considered during creation of rule patterns, whereas structural assessment must be performed immediately after creation. It is thereby important to assess the current state of a rule or rule pattern against the desired target state, using both objective data and user perceptions, and describe the results both in terms of qualitative descriptions and hard numbers (Paroubek et al., 2007). Examples should be used to the point of saturation to ensure the stability of the rule patterns in terms of grammar and expressiveness (Veizaga et al., 2020).

One interviewee made it particularly clear that the principle of compositionality is essential for linking the structure of a rule pattern to the way it is to be implemented in the rule engine. In the linguistic sense, it concerns the building blocks of sentences and the relations between them, and in the computational sense that a system should be designed by linking smaller, independent subsystems so that recursive reasoning about the meaning and workings of the system is possible. The rule pattern - implementation connection is made by relating the modular, atomic building blocks of a law as simple and consistently as possible to derive equally modular rule patterns from them, which can thereafter be implemented as such.

3.2 Phase 2: User perspectives and cognitive processes

In order to be able to design quality criteria, insights in the current way of rule pattern use and development are necessary. Additionally, clarity on the prospective users is needed. From the interviews and observations we gained some insights in how people *think* they use rule patterns, how they are being used in a collaborative design session and which future users they envision.

3.2.1 Prospective users

Concerning prospective users, the way in which these opinions were expressed was very much diversified. Whereas one interviewee idealized that "John Doe must be able to understand it", other interviewees were more conservative: "It is still a domain-specific language, so it is of particular importance that the 'inhabitants' of the domain, namely the rule analysts and the fiscal experts, are being served."

In the current system, rule analysts are being trained in what can be expressed with existing rule patterns. There are no fiscal experts who design rules independently. Fiscal experts are only engaged as 'rule readers'. Note that this concerns only the concrete, implementable rules, not rule patterns. Therefore, the emphasis of quality criteria for fiscal experts focuses only around understandability and readability. There thus exists a rather large divide between the rule analyst and the fiscal expert in the way they work with rules and rule patterns.

3.2.2 Rule pattern development best practices

For the way of working when designing rule patterns, best practices have already been formulated. Rule pattern design is always done in a multidisciplinary team, which undergoes a design cycle consisting of the following phases: Analysis, Specification of rule pattern, Review with designated team, Implementation, Create documentation and Use pattern in practice. The activities of explaining ideas for patterns to colleagues and paying special attention to understandability are emphasized. However, all interviewees agreed that more structure for evaluation is desired. "We don't really have a good framework for testing pattern development, so many people contribute ideas from their own backgrounds and perspectives, we miss a common framework that states what is important and why. We must all work from the same framework."

The most important insights emerging from the work session observation was that several quality criteria such as relevance, consistency, uniformity, flexibility and non-ambiguity were already spontaneously mentioned, and participants also mentioned afterwards that these were not consciously

planned or organized. The human usability perspective featured most prominently, although some operational criteria are discussed. The questions a rule pattern design raises leads the discussion rather than criteria. Additionally, the explicit differentiation between rules and rule pattern was not made, they were being used interchangeably which sometimes led to confusion. Finally, this observation also gives valuable insights in the dynamic, flexible workflow of daily practice, which is an important criterion to keep in mind when designing the evaluation framework to keep it usable.

3.3 Phase 3: Developing a quality assessment framework

Based on the input from the literature review, interviews and observations we have been able to deduce a set of quality criteria for the evaluation of RegelSpraak rules and rule patterns. The main challenge that remains is to operationalize them to the extent that they are precise yet workable.

3.3.1 Differentiating perspectives

As mentioned above, our set of quality criteria consists of the following: Readability, Recognizability, Explainability, Usability, Completeness, Syntax, Complexity, Efficiency, Compositionality, Domain Independence, Technical Executability.

1. Human - Machine After reviewing the set of quality criteria, we came to the conclusion that most can be interpreted from two perspectives: the human and the machine. The human perspective encompasses how fiscal experts and rule analysts work with the rules and patterns, whereas the machine perspective embodies the computational implementation and execution of rules. Readability, Recognizability and Explainability are uniquely human, but the other criteria can be operationalized from multiple perspectives.

2. Read - Write On top of the human - machine differentiation, we included the read - write distinction within each perspective, as mentioned in Section 1.2. From the human perspective, fiscal experts read rules and rule analysts write rules. From the computational perspective, the rule engine reads the rules as input and writes them to executable RegelSpraak rules as output.

3. Rule - Rule Pattern The next critical step was to differentiate explicitly between the two abstraction levels of concrete, implementable rules, and abstract rule patterns still to be instantiated. We had previously observed that the two were being used interchangeably in practice, but for precise operationalization a clear distinction is necessary.

Revision: Fiscal - Computational However, during the consecutive iteration it became clear that simply distinguishing between a human and a computational perspective was too simple. For example, when defining a criterion such as Complexity, even from a computational perspective human developers are responsible for delivering the right level of complexity to the machine. Therefore, what was initially the 'Human' perspective became the 'Fiscal' perspective, and 'Machine' changed to 'Computational'.

Additionally, it became clear that applying the Read - Write and the Rule - Rule Pattern distinctions to the computational perspective resulted in an artificial description that did not correspond with what could be tested in practice. In particular, the implementation and execution of instantiated rules could be seen as a result of successful rule pattern instantiation, and therefore not truly relevant. Therefore, we unified the Computational perspective to focus on the Rule pattern level and drop the Read - Write distinction. The core focus then becomes the implementability of the rule pattern, whereby 'implementable' is viewed from the usability perspective of the UX designer as creating a usable interface for the rule engine, and from the perspective of the developer as delivering efficient and complete code.

The final differentiation of perspectives and categorization of criteria within the Fiscal perspective is shown in Table 2, and the Computational perspective in Table 3.

3.3.2 Operationalizing qualitative norms

Tables 2 and 3 show that certain criteria are unique to a perspective, whereas others are needed from different perspectives. The most pervasive one is Complexity. One could argue that within the Fiscal - Read perspective, complexity is integrated in all criteria, since an overload of complexity hinders readability, recognizability and understandability. For the Fiscal - Write and Computational perspectives, however, complexity needs to be specifically defined. We began the operationalization with qualitatively formulated norms, touching on core variables. From the Fiscal perspective, Complexity of a rule or rule pattern encompasses the following four

	Fiscal - Rule Pattern	Fiscal - Rule
Read	Recognizability Readability Explainability	Recognizability Readability Explainability
Write	Complexity Completeness Usability Syntax Compositionality Domain Independence	Complexity Completeness Usability Syntax

Table 2: The Fiscal perspective.

Computational - Rule Pattern
Complexity
Completeness
Usability
Compositionality
Efficiency

Table 3: The Computational perspective.

elements: Number of variables, Number of variable types, Number of relations between variables, and Order. In case of an instantiated rule, it can be made up of several pieces of rule pattern. Therefore it is important to minimize for each rule the scope and thereby the number of rule patterns or pattern parts that have to be used to express the meaning of the rule. From the Computational perspective, however, the non-functional requirements of Performance efficiency and Maintainability were the critical variables. The three qualitative definitions of Complexity are as follows:

Complexity 'Fiscal - Write - Rule Pattern' A rule pattern must be divided into sub-patterns if the following complexity conditions cannot be met:

1. Is the number of variables minimized?
2. Is the number of operators minimized?
3. Are the types of values a variable can take minimized?
4. Is the number of relations between variables minimized?
5. Is the scope of the pattern minimized?
6. Are the variables presented in a logical order in which consecutive variables build on prior variables to maximize readability, usability and domain independence?

Complexity 'Fiscal - Write - Rule' A rule must be divided into sub-rules or enumerations if the following complexity conditions cannot be met:

1. Does the number of legal concepts represented in the rule remain such that connected sentence parts remain readable, recognizable and understandable?
2. Is the number of operators in the rule minimized?
3. Is the number of legal concepts in the rule minimized?
4. Is the number of relations between legal concepts and (parts of) rule patterns minimized?
5. Is the scope of the rule minimized?
6. Have the legal concepts been defined on the highest possible level of abstraction such that the semantics are still representative of the law?
7. Is legal maintenance of rules easy to perform?

Complexity 'Computational' Verify whether the complexity of the code to be implemented is such that performance efficiency and maintainability are guaranteed:

1. Is the number of functions needed to implement the rule pattern minimized?
2. Is the number of steps needed to go from the first to the final step in the function minimized?
3. Do the functions have a modular structure in relation to each other?
4. Is there legacy code that is no longer functional?
5. Which critical consequences may happen if a rule pattern fails? How many steps are needed to deduce this?
6. Which non-critical consequences may happen if a rule pattern fails? How many steps are needed to deduce this?
7. How does error handling in the rule pattern implementations take place?

8. Can rule pattern maintenance be efficiently and accurately performed?

4 Discussion and future work

Given the current framework for relevant quality criteria and the positioning of these within clearly delineated perspectives, a rough standard has been created to improve insights in the quality of Regel-Spraak rule patterns and stimulating critical discussions.

As the study progressed, the question arose to which extent rule pattern quality could be studied separately from the process of rule creation in practice. One of the best ways to test the meaning of a rule pattern remains to instantiate it with existing legal procedures. This led us to further emphasize the importance of making the Rule - Rule pattern distinction, and the Read - Write distinction on top of it. A usable framework with impact will therefore always have to combine analytic evaluation based on abstract quality criteria with user evaluation based on actual implementation tests, preferably in an iterative, flexible fashion.

We have not yet been able to test the criteria in the practice of rule pattern design. However, an initial qualitative review of the criteria suggested that the content indeed touches on relevant variables. The most important next steps are to further refine the best practice cycle the DTA team has already set up to incorporate the evaluative criteria, to focus specifically on how to implement them so that they are usable within the flexible dynamic of design sessions, and to determine the level of quantitative operationalization needed. All these factors need to be incorporated in practical usability testing. Further quantitative validation of our design choices to demonstrate that they are representative of legal constructs need be performed, for example using text mining or surveying a large population of legal professionals.

For usability in practice, the core procedure boils down to awareness of within which predefined perspective a task at hand can be classified, to take note of the criteria defined within that perspective, and to use the operational qualitative criteria to evaluate the result, within the dynamic team discussion. A trade-off will have to be made between the purposes of stimulating critical discussion about rule pattern proposals, and actually measuring quality in a more quantitative sense in relation to a predefined quality standard. It will be interesting to study what numerical constraints on understandability may be useful, such as incorporating the known limits of human working memory (3-5 items) (McCutchen, 1996; Ricker et al., 2010) to constrain the number of nested elements, operators and parameters.

References

A. Gemino and Y. Wand. 2003. Evaluating Modeling Techniques Based on Models of Learning. *Communications of the ACM*, 46(10):79–84.

Paul Grice. 1991. *Studies in the Way of Words*. Harvard University Press.

J.R. Hayes and L.S. Flower. 1986. Writing research and the writer. *American Psychologist*, 41(10):1106–1113.

International Organization for Standardization. 2017. ISO/IEC 25010:2011. https://www.iso.org/standard/35733.html.

John Krogstie, Guttorm Sindre, and Havard Jorgensen. 2006. Process models representing knowledge for action: A revised quality framework. *European Journal of Information Systems*, 15(1):91–102.

Tobias Kuhn. 2009. How to Evaluate Controlled Natural Languages. In *Workshop on Controlled Natural Language*, Marettimo, Italy.

Odd Ivar Lindland, Guttorm Sindre, and Arne Solvberg. 1994. Understanding quality in conceptual modeling. *Software, IEEE*, 11(2):42–49.

Deborah McCutchen. 1996. A capacity theory of writing: Working memory in composition. *Educational Psychology Review*, 8(3):299–325.

Patrick Paroubek, Stéphane Chaudiron, and Lynette Hirschman. 2007. Principles of Evaluation in Natural Language Processing. In *Traitement Automatique Des Langues*, volume 48, pages 7–31.

Timothy J. Ricker, Angela M. AuBuchon, and Nelson Cowan. 2010. Working memory. *Wiley Interdisciplinary Reviews: Cognitive Science*, 1(4):573–585.

D.T. Ross, J.B. Goodenough, and C. A. Irvine. 1975. Software Engineering: Process, Principles, and Goals. *Computer*, 8(5):17–27.

RuleSpeak. 2021. RuleSpeak® —— Let the Business People Speak Rules! http://www.rulespeak.com/en/.

Patrick H. M. Sins, Elwin R. Savelsbergh, and Wouter R. van Joolingen. 2005. The Difficult Process of Scientific Modelling: An analysis of novices' reasoning during computer-based modelling. *International Journal of Science Education*, 27(14):1695–1721.

M. Theodorakis, A. Analyti, P. Constantopoulos, and N. Spyratos. 1999. Contextualization as an Abstraction Mechanism for Conceptual Modelling. In *Conceptual Modeling - ER '99, 18th International Conference on Conceptual Modeling, Paris, France, November, 15-18, 1999 Proceedings*, volume 1728 of *Lecture Notes in Computer Science*, pages 475–489. Springer Berlin / Heidelberg.

T. Tommila and A. Pakonen. 2014. Controlled natural language requirements in the design and analysis of safety critical I&C systems. Technical Report VTT-R-01067-14, VTT Technical Research Centre of Finland.

Alvaro Veizaga, Mauricio Alferez, Damiano Torre, Mehrdad Sabetzadeh, and Lionel Briand. 2020. On systematically building a controlled natural language for functional requirements.

I. Wilmont, S. Hengeveld, E. Barendsen, and S. J. B. A. Hoppenbrouwers. 2013. Cognitive Mechanisms of Modelling: How Do People Do It? In *Conceptual Modeling - 32th International Conference, ER 2013, Hong-Kong, China, November 11-13, 2013, Proceedings*, volume 8217 of *Lecture Notes in Computer Science*, pages 74–87. Springer Berlin Heidelberg.

Ilona Wilmont. 2020. *Cognitive Aspects of Conceptual Modelling: Exploring Abstract Reasoning and Executive Control in Modelling Practice*. PhD thesis, Radboud University Nijmegen.