



Human-Centred Automated Reasoning for Regulatory Reporting via Knowledge-Driven Computing

Dilhan J. Thilakarathne^(✉), Newres Al Haider, and Joost Bosman

ING Tech R&D, ING Bank, Haarlerbergweg 13-19, 1101 Amsterdam, CH, The Netherlands
Dilhan.Thilakarathne@ing.com

Abstract. The rise in both the complexity and volume of regulations in the regulatory landscape have contributed to an increase in the awareness of the level of automation necessary for becoming fully compliant. Nevertheless, the question of how exactly to become fully compliant by adhering to all necessary laws and regulations remains. This paper presents a human-centred, knowledge-driven approach to regulatory reporting. A given regulation is represented in a controlled natural language form including its metadata and bindings, with the assistance of subject matter experts. This representation of a semi-formal controlled natural language translates into a self-executable formal representation via a context-free grammar. A meta reasoner with the knowledge to execute the given self-executable formal representation while generating regulatory reports including explanations on derived results has been developed. Finally, the proposed approach has been implemented as a prototype and validated in the realm of financial regulation: Money Market Statistical Reporting Regulation.

Keywords: Regulations · Compliance · RegTech · Human-centred · Reasoning · Knowledge systems · Explainability · Rule-based systems · Finance · Banking

1 Introduction and Background

The Global Financial Crisis in 2008 changed the way in which financial markets, services and institutions manage financial risk. One necessity in financial risk management is understanding the importance of fully adhering to a process of compliance with its concealed complexity [3, 5]. Compliance adherence involves a significant allocation of both time and expertise; therefore, the associated cost has led to an increase in compliance budgets due to both a rise in the complexity and volume of regulations [5]. According to Duff & Phelps, financial institutes already spend a significant amount of money during the process of becoming compliant and expect a substantial increase in future compliance costs: 24% of participants expect to spend more than 5% of their annual revenue on compliance by 2023, while 10% expect to spend more than 10% by 2023 [10]. Given the increasing complexity and volume of regulations, it is clearly necessary to provide a cost-effective approach that is easy to explain [3, 5, 10, 14].

Regulatory technology (RegTech) is the centrifugal force both in re-conceptualizing the financial regulation process and achieving principal regulatory objectives [3]. According to Arner and colleagues, RegTech is ‘*a contraction of the terms regulatory and technology, and it comprises the use of technology, particularly information technology (IT), in the context of regulatory monitoring, reporting, and compliance*’ [1, p. 373]. Given that many objectives are associated with RegTech-driven solutions in the financial domain, an appealing objective is to ensure the prudential safety and soundness of financial markets. To achieve this, the wisest approach is to enhance the transparency of the process with the support of an explainable causal reasoning methodology [8]. In this way, Subject Matter Experts (SMEs) can understand the causation behind derived conclusions (whether compliant or non-compliant) not only to gain a retrospective interpretation of precisely how compliant something is/was but also to unambiguously explain it to regulatory bodies. Furthermore, ‘*right to explanation*’ has become a social convention (e.g., GDPR); therefore, explainability in the execution of financial tasks is necessary even from the customer/user point of view [6].

Another key goal in RegTech-driven regulatory compliance solutions is reducing outlay while managing compliance obligations via more efficient compliance system automation [3, 8, 10, 14, 27]. The efficacy of regulatory compliance is associated with three perspectives: corrective, detective and preventative steps [27]. The ideal RegTech approach has the power to reduce the cost associated with each of these perspectives. In this paper, the scope of the proposed approach is limited to before-the-fact detection (under detective perspective) with a focus mainly on regulatory compliance reporting.

This paper presents a human-centred approach for regulatory reporting through the use of knowledge-driven computing techniques, that aims to achieve the objectives of being both explainable and capable of automating arbitrary compliance regulations.

In this process the regulatory text is translated into a semi-formal representation, which is labelled Controlled Natural Language (CNL), that can be easily understood by SMEs. This CNL contains a hierarchical tree structure (root, intermediate and leaf) where leaf nodes only contain atomic facts and top-level nodes represent aggregated information. This CNL representation is further enriched by meta-data which is translated into a more formal rule set where individual rules are self-executable. A self-executable rule set is interpreted via a symbolic reasoner implemented on top of actual data that is required to validate its compliant status. The interpreter provides a final status (compliant or non-compliant) which always has a detailed explanation behind its derivation, along with extra information via metadata. The proposed approach is validated using a section of Money Market Statistical Reporting (MMSR) regulation.

The rest of the paper is structured as follows: Sect. 2 covers the related work and highlights key approaches in automating the regulatory compliance process; Sect. 3 describes the approach with relevant insight; Sect. 4 presents an example based on the MMSR document which systematically explains how the proposed approach can be used in a practical way; Sect. 5 concludes and draws possible links to future work.

2 Related Work

There have been significant contributions to regulatory compliance automation with many adopted approaches in several domains: construction [4, 9, 21], health [29], finance

[11, 12, 18, 26] and business processes (see [16]). Nevertheless, the research community is still far from fully overcoming the associated issues, though there are some key areas for further development [3, 5, 10, 16].

A known approach for automating regulatory compliance is to represent it as an (logical-) object model together with domain-specific parameters [4, 9, 17, 21]. This object model approach has been applied to many building regulations (see [17]); Malsane and colleagues have extended this generic approach to filter the rules before interpreting them [21]. In the knowledge formalisation, regulations are interpreted into a set of computer-implementable rules which are used to create an object model that can be logically validated. This representation is then used as the basis to create an Industry Foundation Class compliant data model. This model includes a hierarchical structure that helps to rationalise terminological ambiguities that exist in the semantic representation and uses the derived objects and associated relations as rules to validate compliance needs [21]. Furthermore, the data model can be converted to a machine-executable rule format: Semantic Web Rule Language (SWRL) that provides binary results [4].

SWRL together with ontologies is used for regulatory purposes in different domains as ontological informatics systems [11, 12, 29]. In the pharmaceutical domain, a key role is played by a domain-specific reasoner and a rule engine that manually parses a pharmaceutical regulation text into OWL axioms and SWRL rules [29]. The purpose of having a (OWL-) reasoner is to identify logical inconsistencies while a rule engine (e.g. Java Jess) interprets and enforces the (SWRL-) rules imposed by the SMEs. Finally these two results are combined to provide a structural incorporation to facilitate user interaction via Protégé. Instead of translating regulatory text into SWRL, different control languages such as SBVR have been used to address the same problem but are more focused on (financial-) business vocabularies as ontologies (e.g., FIRO, FIBO) together with formal reasoning and verification techniques.

Semantics of Business Vocabulary and Business Rules (SBVR) is one of the leading control languages in automating compliance applications [11, 18, 26]. Roychoudhury and colleagues presented an approach that translates legal text into SBVR via a control language called ‘Structured English’ (StEng) [26]. The purpose of introducing StEng in this approach is to partly eliminate the inherent complexity of understanding the legal text representation in SBVR by SMEs. StEng provides a format which is comparatively easier for SMEs to understand so they can easily intervene in this semi-automated task to validate and improve the transformation from legal text to an StEng [26]. Once there is StEng representation, it is translated into a regulatory model in SBVR and then further translated into formal logic that enables validation of a given data set via inferencing engines like DROOLs or DR-Prolog [25]. There is an extra step in this approach where a conceptual data model—DDL (i.e., data required for checking)—is generated from SBVR as a basis for mapping enterprise schema to the conceptual schema by a database expert, as it is required to populate the fact base in a DB in this approach.

Abdellatif and colleagues used SBVR StEng notation to translate legal text into SBVR with POS tagging and dependency parsing in NLP to automate this step [15]. Mercury, proposes a language to capture regulations for the purpose of compliance checking that consist of two main parts: Mercury StEng and Mercury Markup Language [7]. Mercury StEng is an extension of a subset of SBVR that is workable enough for

SMEs to directly represent and maintains the information available in regulatory text. This allows it to eliminate the complexity and verbosity of standard SBVR-driven logical formulation together with the introduction of Mercury Markup Language. This language contains a vocabulary and a rulebook which both help to map into FIRO ontology. Nevertheless, the main difficulty in translating legal text to SBVR is a complex semantic analysis of the English language, which is a challenging task for SMEs [7, 28].

3 Approach

A human-centred approach motivated by knowledge-driven computing techniques is used in this paper. Figure 1 presents the overall approach and highlights the end-to-end process including specific human interventions. The first step in knowledge acquisition takes place when a selected regulatory text is (manually-) translated into the proposed CNL representation. This representation is more human-oriented than machine-oriented [28] with the purpose of facilitating Subject Matter Experts (SMEs) to easily represent the given clauses.

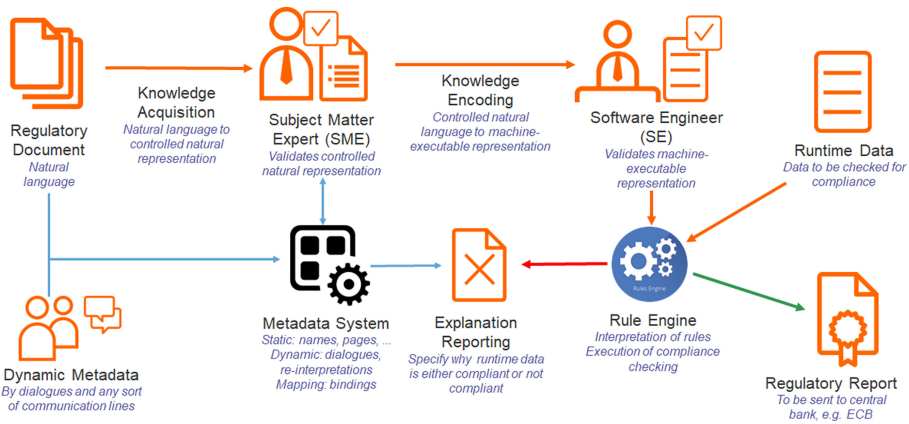


Fig. 1. Overall approach of human-centred regulatory compliance

A unique feature of this knowledge-acquisition phase is embedding metadata into the CNL. This includes not only the causal information of a clause in a regulatory text but it also includes the associated descriptive information including conversations about different possible interpretations; both of these result in more meaningful explanations in regulatory reporting. This CNL includes specific grammar and vocabulary; therefore, performing knowledge encoding (in this phase translating CNL into a Machine-Executable Representation) is achieved via a parser that translates CNLs context-free grammar (CFG) into a machine-executable representation form. This CNL contains a hierarchical tree structure (root, intermediate and leaf) where leaf nodes only contain atomic facts and inner nodes represent aggregated information.

This executable representation is validated by software engineers (SEs) with necessary interventions including the addition of correct functions from a pre-developed

regulatory library for any missing or incorrect references provided for the calculation of the atomic facts represented in the leaf nodes. A specific declarative rule engine is implemented that interprets rules represented in machine-executable form together with given runtime data to generate a regulatory report that includes statuses and explanations. These explanations are enriched with metadata binding to facilitate clearer and unambiguous elaboration of the scenario. More detailed information about each part is explained in following subsections.

3.1 Process of Regulatory Text into CNL Representation

Even though laws and regulations are supposed to be unambiguous, ambiguities are prevalent in regulatory texts due to the inherent complexity of natural languages and their underlying processes [22]. As a result of this complexity, the key step in automating the regulatory compliance process is to extract these rules into a more formal structure, to make the rules more explicit and eliminate ambiguities.

As the first step of this process, SMEs need to translate a given regulatory text into a CNL, which is a semi-formal representation of a natural text (e.g., English text). Unsurprisingly, if the CNL can facilitate a closer expressivity in line with its selected natural text to non-technical specialists while enabling accurate and efficient processing to machines, then such a CNL is an ideal representation for this task [13]. Nevertheless, such an ideal representation is difficult to obtain practically; therefore, this approach uses a human-oriented CNL which is enriched with strong readability and comprehensibility features that are essential for SMEs when they need to manually transform regulatory text into CNL text [28].

For this transformation a specific CFG-based CNL is created as in Table 1. The root term of this grammar is the ‘*specification*’ which holds a set of rules. Each rule contains a head (unique string), state, conditions (unique strings) and meta knowledge to elaborate a selected clause or phrase in the given regulatory text. Specifically, if the given conditions (in here conditions are conjunctions of individual conditions) are true then the head holds the value of the state which is a boolean value. Furthermore, conditions are constraints that should be satisfied to conclude the state of the head as in formal logic. Finally, meta knowledge consists of extra information to improve the meaning and understanding of the formation of the rule which provides clear insight into the derivation of content. As per the CFG both head and condition hold a statement, therefore we maintain a chain of rules within this specific hierarchy.

In the process of transforming regulatory text into its mapping form of CNL representation, a top-down approach is used in which the division of aggregated relationships in a compound clause contributes to dividing the full text into a set of rules. In other terms, structure of the ‘specification’ (specifically a single rule of the specification) is a tree and is composed of a root-node, intermediate-nodes and leaf-nodes. A root-node represents the aggregated goal of the regulatory document. There is commonly one single root for the document but it is possible to have many root nodes if the goals are mutually exclusive. The intermediate-nodes break the overall regulation into sub-clauses. Furthermore, each intermediate-node may further sub-divide into sub-intermediate-nodes depending on its level of aggregation and complexity. Finally, any element which does not require

Table 1. CFG for proposed CNL representation

specification	::= rule ⁺
rule	::= rule head is state if following conditions hold: { conditions } with meta knowledge { know }
head	::= statement
state	::= true false
conditions	::= condition ⁺
condition	::= statement (not statement) (statement op condition ⁺)
op	::= and or
statement	::= ^[a-zA-Z]+([a-zA-Z0-9\s]+.*)
know	::= static data { pair ⁺ } dynamic data { pair ⁺ } mapping data { mapping ⁺ }
pair	::= key : value ;
mapping	::= key : value { variable [; variable ...] }
key	::= string
value	::= string
variable	::= string

any further sub-division, is labelled as a leaf-node, which carries the real atomic computation information to derive its ultimate runtime value. In order to evaluate whether a regulation holds or not, the computation at the leaf nodes can be executed with the appropriate data, and the result of which can be aggregated up through the links with the intermediate nodes to the root.

A unique feature of this knowledge-acquisition phase is embedding metadata into the CNL. In each leaf-node, SME has the option to include static, dynamic and mapping data (static and dynamic data can be used with all other nodes too). Static data includes direct reference information from the regulatory text such as the URL, title, version, page number, source text, etc. Dynamic data includes information that is used to finalise meaning and understanding of a selected clause/phrase with other colleagues, authorities or even legal experts (due to ambiguities and the level of clarity in the text). This data includes emails, written confirmations, approvals, references for voice data stored via phone calls, etc. Finally mapping data is used specifically as an aid for rule engine via machine-executable representation. As leaf nodes are atomic computation units, each node’s data should translate into a computable function at the machine-executable level which requires considerable parameter-binding information. These mapping data include expected function names from a pre-developed regulatory library and relevant environment variables that should be used with a function to derive a value. The completion of mapping data is optional to SME unless they have a technical background; otherwise this is completed by SEs.

3.2 Process of CNL into Machine-Executable Representation

Having a CNL representation of a given regulatory text provides many benefits beyond just an unambiguous interpretation, though it cannot be directly executed on a rule engine. Therefore, the main second step in the regulatory compliance process is to translate CNL

into a machine-readable form. This specific form should be able to directly execute on a rule engine. The rule engine, SWI Prolog-based meta interpreter [31], takes two inputs: machine-executable representation and runtime data. The structure of the representation is selected to directly encode CNL data to make this step fully-automated. Nevertheless, due to the possibility of missing information in leaf nodes—specifically mapping data—SEs should intervene in this step to validate the given library mappings and/or to inject correct mapping to derive expected results. Due to the availability of meta knowledge, this step is a comparably easy task for SEs and their scope is mainly limited to individual leaf nodes. The structure of machine-executable representation is presented in Table 2.

Table 2. Structure of machine-executable representation

```
rule(
  type(< root | intermediate | leaf >),
  rule_header(< HEADER NAME >),
  rule_conditions([
    (< ?not > < CONDITION 1 >) operator (< ?not > < CONDITION 2 >) operator
    ... (< ?not > < CONDITION n >)],)
  meta_knowledge([
    static_data([
      url(< URL >), title(< TITLE >), ver(< VERSION >), page_num(< NUM >), ... ]),
    dynamic_data([...]),
    mapping_data([
      prolog_call(< FUN NAME, VAR1, VAR2 >),
      get_in_xml_path(< KEY1 >, < VAR1 >),
      get_in_xml_path(< KEY2 >, < VAR2 >)]))]).
```

Due to the fact that the CNL version is already validated by the SME, the logical distribution of information in it is already accurate; therefore, the most important micro-step here is to validate mapping for functions in the auto-generated machine-executable representation. As a result, this approach is more of a human-centred automated process in which experts encode knowledge but the system automates the execution. This ensures that the interpretation responsibility is not taken away from the subject matter experts and SEs in the pipeline.

3.3 Rule Engine and Explanations

In a regulatory compliance system, a rule engine (also called reasoner or inference engine) plays a key role by executing formal knowledge extracted from a given regulatory text with runtime data [4, 12, 25]. Nevertheless, the best rule engine for a given domain is subjective and in most situations this is determined by the features required and computational complexity of the problem. Declarative rule engines are well-known in knowledge-based systems, especially with explanations [23, 24, 30]. Therefore, a backward reasoning rule engine is developed with SWI Prolog to draw logical conclusions based on the given rules and runtime data.

This engine is a reference implementation inspired by a meta-interpreter explained in the Art of Prolog [31] to reason and derive new facts to validate the state of compliance. Furthermore, this uses backward reasoning (starts with the desired conclusion and performs backward to find supporting facts) as the reasoning strategy and provides proof-tracing as a reasoning feature for the purpose of explainability [23, 31]. Additionally, this rule engine consists of a pre-built regulatory function library to facilitate various functionalities required to execute the rules defined in CNL (leaf node rules). This library can be extended by SEs to include missing behaviours; it is also possible to combine existing functions to implement custom functions.

An important feature of a regulatory compliance system (especially in a regulatory reporting system) is the explanation it gives [23]. Explanation is an inherent feature of rule engine and proof-tracing, which explains the steps involved in logical reasoning and is used as the insight into this system. As the system uses a backward reasoning strategy (goal to facts) it has a systematic way of exploring the search space that provides insight into how the system reaches given conclusions [31]. A rule engine collects this information in tandem with the reasoning steps and returns meaningful systematic explanation data. This systematic data is encoded as a proof tree and included together with regulatory reports to label the given runtime data as compliant or not.

4 Use Case on Money Market Statistical Reporting

To demonstrate this approach, a portion of the Money Market Statistical Reporting (MMSR) regulations¹ (version 3.0) relating to the unsecured market segment (one from four main areas) has been implemented. For this purpose two reference documents are used: ‘reporting instructions’ (v3.0) and ‘questions and answers’ (v3.0). A prototype system has been developed with all major components of the presented approach in conjunction with the support of Subject Matter Experts (SMEs) and SEs responsible for implementing the MMSR regulations at ING Bank Netherlands. SMEs manually translate selected text into the CNL representation. Table 3 presents a sample of CNL text generated without including the metadata that was extracted from the MMSR ‘Reporting Instructions’ document version 3.0 in page numbers 32–33.

Encoding knowledge directly from a gigantic regulation (e.g., MMSR) into its CNL form may lead to many errors unless a significant patience and attention were given. To avoid such errors, a web based prototypical tool was developed for the SMEs to interact with the encoded CNL representation. The tool helps SMEs to easily interact with CNL data to validate the correctness, accuracy and consistency of (manually-) encoded rules. The given tool includes two options to navigate the given CNL: 1) via an accordion menu (left side list view) and 2) via a deep hierarchy tree. These views provide a greater complexity decomposition to SMEs to perform their tasks and this is found to be further empowered by the attached meta data information specifically on validating the correctness of rules. Figure 2 highlights some key features of the prototypical tool.

CNL representation includes all the meta knowledge except for mapping data which is handled by SEs. SEs use machine-executable representation which is derived from

¹ https://www.ecb.europa.eu/stats/financial_markets_and_interest_rates/money_market/html/index.en.html.

Table 3. A sample CNL data from machine-executable representation

<p>Regulatory Text from MMSR ‘reporting instructions’ (v3.0) pp. 32-33: <i>“The MMSR message is sent by the reporting agents to the relevant NCB or to the ECB to report all unsecured transactions covering:</i></p> <ul style="list-style-type: none"> • <i>borrowing via unsecured deposits and call accounts, excluding current accounts, as defined in the table below, in euro with a maturity of up to and including one year (defined as transactions with a maturity date of not more than 397 days after the settlement date) by the reporting agent from financial corporations (except central banks where the transaction is not for investment purposes), general government or from non-financial corporations classified as ‘wholesale’ under the Basel III LCR framework ...”</i>
<p>Mapping CNL: rule is-an-unsecured-money-market-transaction-to-be-reported is true if following conditions hold: { is-the-transaction-either-borrowing-via-unsecured-deposit-and-call-account and (not transaction-is-a-current-account) and is-a-euro-transaction and has-the-transaction-a-maturity-of-no-more-than-one-year and is-the-transaction-with-a-valid-borrowing-entity-for-unsecured-market-segment } with meta knowledge { Source: “Money Market Statistical Reporting – Reporting Instructions for the Electronic Transactions”, Author: “ECB”, Version: “3.0”, Section: “4.1”, Page:“32”, Text:“”, ... } rule is-the-transaction-either-borrowing-via-unsecured-deposit-or-call-account is true if following conditions hold: { is-the-transaction-a-borrowing-via-unsecured-deposit or is-the-transaction-a-borrowing-via-unsecured-call-account } with meta knowledge { ... } rule is-the-transaction-a-borrowing-via-unsecured-deposit is true if following conditions hold: { is-the-deposit-redeemable-at-notice or has-the-deposit-a-maturity-of-no-more-than-one-year } with meta knowledge { ... }</p>

CNL via a given CFG and they inject all mappings for leaf nodes to easily execute them on the rule engine. As explained in Sect. 3 a rule engine has been developed with SWI Prolog which accepts machine-executable representation as an input and internally represents it as a rule set to use in regulatory compliance validation. This rule set together with given runtime data are used as knowledge by the rule engine to systematically validate whether the given data set is compliant with the rule set. In addition to providing a compliant status, the rule engine provides an explanation to prove its derivation. This explanation is enriched with metadata so that it will develop the correct awareness of validation in problematic situations.

4.1 Validation of the Approach

The above mentioned prototypical system is validated by a team of three members who are responsible for implementing and managing the MMSR regulations at ING Bank Netherlands. They selected unsecured market segment of MMSR regulation for this experiment and three hours of workshop was conducted to train them about the features

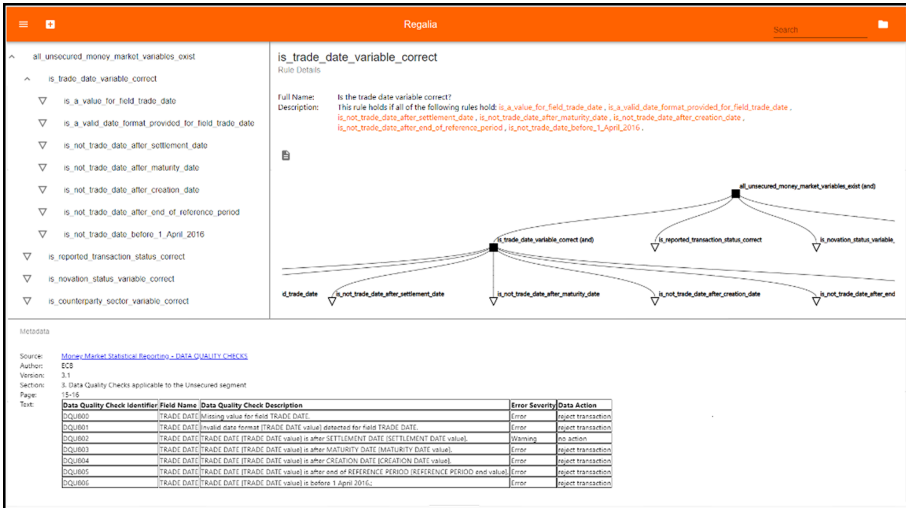


Fig. 2. User interface of developed prototype to navigate CNL data including the metadata

of the implemented prototypical system together with a description of the study’s purpose and tasks it includes. In this workshop, they achieved all the core knowledge necessary to execute the experiment: translating the unsecured market segment of MMSR regulation into its CNL representation and encoding those CNL data into the proposed machine understandable representation with leaf node bindings through the provided regulatory library.

After the workshop, they work fully independently on the given task and 23 concepts of the unsecured market segment of MMSR regulation are used where each concept includes at least three leaf nodes (with many intermediate nodes) and a sufficiently complex rule set is used. Team processes the data available in the regulation as blocks and translated them into CNL manually in incremental steps where in each step they validate the correctness, accuracy and consistency of translated information. Having successfully translated the regulatory text into CNL, the next step that executed is encoding the CNL into machine understandable representation with binding leaf node functions from a regulatory library. Finally, twenty data sets (all data sets are synthetic data) are used for this experiment. Each set has labelled information and all sets are classified with an explanation that has also been identified as accurate. The quality of the generated explanations and outputs are validated by SME, which identifies whether outputs are accurate and includes interesting information as explanations. The compliance report generated over all 20 sample data sets which were accurate to the extent of 100%. Three sample explanation outputs are provided in an external appendix with the file name: ‘SampleOutputs.pdf’². Furthermore, a significant time is saved due to the introduction of CNL when encoding regulatory text into machine-executable representation.

This being an human centred approach, as the evaluation method psychometric evaluation of an after-scenario questionnaire is used as presented by James Lewis [20].

² <https://github.com/DilhanAMS/conferences/tree/master/iea-aie2020>.

Therefore, three main questions are considered: 1) *Overall, we are satisfied with the ease of completing the tasks in this scenario*, 2) *Overall, we are satisfied with the amount of time it took to complete the tasks in this scenario* and 3) *Overall, we are satisfied with the support information when completing the tasks*. We collected the accumulative decision from the group rather than individually due to the fact that this experiment was motivated as a human centred collaborative task. We used the 7-point scale provided by James with the terms “Strongly agree” for 1 and “Strongly disagree” for 7. The team gave 2, 2, and 5 respectively for above mentioned 3 questions as a summary with some interesting feedback comments. With the above cumulative results (from 3 experts) and the feedback given, they have recognised a significant value in this approach specially in the CNL representation. They have found that CNL as an interesting representation with the feature of embedded meta data to represent rules in a less ambiguous form. Furthermore, they have stated that dynamic metadata part is very interesting in specific situations where they need to explain the reasoning behind their interpretations or decisions to an agent. Also, they have found that the time they took for this is relatively comparable with manual process, but in the feedback they have highlighted that with more experience on this approach they can reduce the required time at least by 20% (mainly due to the fact that regulatory library and limited errors in machine readable representation). The main remark for lower value on the question: support information when completing the tasks, is that the proposed prototypical tool still expect them to manually translate regulatory text into CNL and they strongly expect some degree of automation. They expect new features in the user interface such as highlight key phrases of the text together with auto suggested mapping CNL (irrespective of its accuracy).

5 Discussion and Conclusion

This paper presents a human-centred automated reasoning approach for regulatory reporting. The interpretation of legal concepts and the underlying system of rules are highly-debatable in legal philosophy [2]. This approach combines the legal expertise required to interpret the regulatory text and the reasoning techniques used in knowledge systems for inferencing. As a human-centred approach, the legal expertise required to interpret the regulatory text is achieved through the introduction of a CNL representation to Subject Matter Expert (SME). Because this CNL is more human-oriented, it can be easily followed by SME to encode legal text directly into this form. Via this approach, the system facilitates discussions of the different possible interpretations and allows them to use the final agreed-upon interpretation with supportive meta knowledge. Such interpretations can be used as part of explanations to auditors and/or authorities.

The designed CNL has a specific CFG and therefore it has encoded into a specific machine-executable representation that has been pre-identified. This executable form needs to be validated by SEs to have the correct bindings to functions in order to be executed. The results show full explanations relating to the CNL in order to be easily understood by the SMEs. Finally, the system provides the status of the given data set with explanations on why it is compliant or not.

The proposed approach has been validated through three experts responsible for MMSR regulation at ING. They have examined the approach with a sub-section selected

from the MMSR regulation report and yielded interesting results which were confirmed by SME too. This includes 20 synthetic transaction data sets with pre-labelled compliant status and the results that were obtained are 100% accurate. The main limitation that encountered was the inability of auto generating at least partly correct CNL from regulatory text. This will be a key focus on future work.

Another key part of the future work of this research is to encode the full MMSR document and validate this approach with a considerably larger transaction data set. This work mainly applies to regulatory reporting, though the proposed approach is generic enough to apply to challenges in many parts of regulatory compliance.

This approach is an extension from previous work [1] on using a CNL based approach for implementing regulations using Software Contracts. The approach presented in this paper has a more structural solution to the issue of expressing the data flow in a more explainable manner, which was a limitation in the previous work.

Many different techniques have been used to translate regulatory text into SBVR, most of which use an intermediate representation language such as StEng, SBVR StEng and Mercury StEng as presented in related work. The main strength of the approach of this paper is that it allows to express arbitrary compliance rules while using a very straightforward CNL that is both easy to express for SMEs and easy to translate into an executable process. For example, the methodologies via StEng [15, 26] the translation of English legal text to SBVR has been noted to be a challenging task for SME [7, 28]. Both the SBVR StEng notation [25, 26] and the Mercury StEng [7] contain the features of SBVR and depend on a specific pre-developed ontology in the encoding process. Nevertheless, the use of ontologies to help the encoding process is a feature [15, 19, 28] that could be used to extend the current approach.

The object model approaches have a hierarchical structure that is similar to the one in the proposed work. However, the SWRL rules that are used in these approaches have a limited expressivity (see [11, 12, 24, 29]) in relation to a full logic-programming language implementation such as SWI Prolog [23, 31]. This limits them in scenarios where arbitrarily complex compliance rules have to be used. These approaches have a good integration with Semantic Web based technologies through their object model. In future work, the meta-knowledge in the approach can be extended to make use of Semantic Web technologies, which allows for an easy integration of domain specific ontologies like FIBO and FIRO [11, 12, 29].

Another way to extend the approach is to increase the expressivity of the intermediate nodes in the tree structure by adding modal logic operators. A key feature of this methodology is using bindings with a pre-developed regulatory library. Although initially the creation of such bindings is a manual process, learning techniques could be applied to automatically help to derive this information.

Acknowledgment. We would like to thank Robbert Haring, Vaibhav Saxena, Priyanka Kumar, John Marapengopie, Tim Slingsby for useful discussions and their expertise.

References

1. Al Haider, N., et al.: Solving financial regulatory compliance using software contracts. Presented at the 22nd International Conference on Applications of Declarative Programming and Knowledge Management (2019)
2. Araszkievicz, M.: Analogy, similarity and factors. In: Proceedings of the 13th International Conference on Artificial Intelligence and Law - ICAIL 2011, Pittsburgh, Pennsylvania, pp. 101–105. ACM Press (2011)
3. Arner, D.W., et al.: FinTech, RegTech and the reconceptualization of financial regulation. *Northwest. J. Int. Law Bus.* **37**(3), 371–414 (2016)
4. Beach, T.H., et al.: A rule-based semantic approach for automated regulatory compliance in the construction sector. *Expert Syst. Appl.* **42**(12), 5219–5231 (2015)
5. Butler, T., O’Brien, L.: Understanding RegTech for digital regulatory compliance. In: Lynn, T., Mooney, J.G., Rosati, P., Cummins, M. (eds.) *Disrupting Finance*. PSDBET, pp. 85–102. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-02330-0_6
6. Casey, B., et al.: Rethinking explainable machines: the GDPR’s “right to explanation” debate and the rise of algorithmic audits in enterprise. *Berkeley Technol. Law J.* **34**, 145–189 (2019)
7. Ceci, M., et al.: Making sense of regulations with SBVR. In: *RuleML* (2016)
8. Chartis Research, IBM: AI in RegTech: A Quiet Upheaval (2018)
9. Dimyadi, J., Amor, R.: Regulatory knowledge representation for automated compliance audit of BIM-based models (2013)
10. Duff, Phelps: *Global Regulatory Outlook 2018: Viewpoint 2018 Opinions on Global Financial Services Regulation and Industry Developments for the Year Ahead* (2018)
11. Elgammal, A., Butler, T.: Towards a framework for semantically-enabled compliance management in financial services. In: Toumani, F., et al. (eds.) *ICSOC 2014*. LNCS, vol. 8954, pp. 171–184. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22885-3_15
12. Ford, R., et al.: Automating financial regulatory compliance using ontology+rules and sunflower. In: Proceedings of the 12th International Conference on Semantic Systems, pp. 113–120. ACM, New York (2016)
13. Fuchs, N.E., Schwitter, R.: Specifying logic programs in controlled natural language. In: *Workshop on Computational Logic for Natural Language Processing*, Edinburgh, p. 16 (1995)
14. Gurkan, C.: Banks’ interactions with listed non-financial firms as a determinant of corporate governance in banking: an agency theory analysis. In: Díaz Díaz, B., Idowu, S.O., Molyneux, P. (eds.) *Corporate Governance in Banking and Investor Protection*. CSEG, pp. 21–35. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-70007-6_2
15. Haj, A., Balouki, Y., Gadi, T.: Automatic extraction of SBVR based business vocabulary from natural language business rules. In: Khoukhi, F., Bahaj, M., Ezziyyani, M. (eds.) *AIT2S 2018*. LNNS, vol. 66, pp. 170–182. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11914-0_19
16. Hashmi, M., et al.: Are we done with business process compliance: state of the art and challenges ahead. *Knowl. Inf. Syst.* **57**(1), 79–133 (2018)
17. Ismail, A.S., et al.: A review on BIM-based automated code compliance checking system. In: *2017 International Conference on Research and Innovation in Information Systems (ICRIIS)*, Langkawi, Malaysia, pp. 1–6. IEEE (2017)
18. Kholkar, D., et al.: Towards automated generation of regulation rule bases using MDA. In: *Proceedings of the 5th International Conference on Model-Driven Engineering and Software Development*, Porto, Portugal, pp. 617–628. SCITEPRESS - Science and Technology Publications (2017)
19. Kuhn, T.: A survey and classification of controlled natural languages. *Comput. Linguist.* **40**(1), 121–170 (2014)

20. Lewis, J.R.: Psychometric evaluation of an after-scenario questionnaire for computer usability studies: the ASQ. *SIGCHI Bull.* **23**(1), 78–81 (1990)
21. Malsane, S., et al.: Development of an object model for automated compliance checking. *Autom. Constr.* **49**, 51–58 (2015)
22. Massey, A.K., et al.: Identifying and classifying ambiguity for regulatory requirements. Presented at the August 2014
23. Merritt, D.: *Building Expert Systems in Prolog*. Springer, New York (2012). <https://doi.org/10.1007/978-1-4613-8911-8>
24. Rattanasawad, T., et al.: A comparative study of rule-based inference engines for the semantic web. *IEICE Trans. Inf. Syst.* **E101.D**(1), 82–89 (2018)
25. Roychoudhury, S., Sunkle, S., Choudhary, N., Kholkar, D., Kulkarni, V.: A case study on modeling and validating financial regulations using (semi-) automated compliance framework. In: Buchmann, R.A., Karagiannis, D., Kirikova, M. (eds.) *PoEM 2018*. LNBIP, vol. 335, pp. 288–302. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02302-7_18
26. Roychoudhury, S., et al.: From natural language to SBVR model authoring using structured english for compliance checking. In: 2017 IEEE 21st International Enterprise Distributed Object Computing Conference (EDOC), pp. 73–78 (2017)
27. Sadiq, S., Governatori, G.: Managing regulatory compliance in business processes. In: vom Brocke, J., Rosemann, M. (eds.) *Handbook on Business Process Management 2*. IHIS, pp. 265–288. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-642-45103-4_11
28. Schwitter, R.: Controlled natural languages for knowledge representation. In: 23rd International Conference on Computational Linguistics, Beijing, China, pp. 1113–1121 (2010)
29. Sesen, M.B., et al.: An ontological framework for automated regulatory compliance in pharmaceutical manufacturing. *Comput. Chem. Eng.* **34**(7), 1155–1169 (2010)
30. Stefik, M.: *Introduction to Knowledge Systems*. Morgan Kaufmann Publishers, San Francisco (1995)
31. Sterling, L., Shapiro, E.Y.: *The Art of Prolog: Advanced Programming Techniques*. MIT Press, Cambridge (1994)