Computerising the New Zealand Building Code for Automated Compliance Audit

[1]Dimyadi, Johannes; [2]Fernando, Sudara; [3]Davies, Kathryn; [4]Amor, Robert
[1, 2, 3, 4] *Compliance Audit Systems Limited*
[3] *School of Building Construction, Unitec Institute of Technology*
[4] *School of Computer Science, University of Auckland*

jdimyadi@complianceauditsystems.com

ABSTRACT

One key ingredient in the automated compliance audit process is the availability of a computable form of normative requirements (e.g. codes and standards), which are usually written in natural language intended for human interpretation and not readily processable by machines. The predominantly 'Blackbox' approach of hardcoding these computable normative rules into a compliance audit system has been reported to be problematic and costly to maintain in response to frequent regulatory changes. The current research sets out to investigate to what extent normative texts can be represented as computable rules for automated compliance audit as well as to ease maintenance in response to changes in the source documents. A set of priority compliance documents supporting the New Zealand Building Code has been selected as the subject for a case study. This paper describes the digitisation and quality assurance process, the knowledge extraction experience, and challenges identified during the study. Furthermore, the paper explores how the legal knowledge captured by the digitised rules can be used effectively in an automated compliance audit environment. The findings from the study suggest that a semi-automated digitisation process is feasible and up to 80% of prescriptive text can be translated and encoded into the open standard *LegalRuleML*. However, only approximately 50% of these can be used directly in an automated compliance audit environment without any human intervention. The lessons learnt from the study can be used towards improving the digitisation process. Ultimately, this could in turn help to improve the natural language source text in subsequent revisions of the codes.

Keywords: Normative, Computable rules, Automated compliance audit, Building information modelling, Legalruleml

1 Introduction

1.1 Background and Motivation

The compliance audit process in the built environment has conventionally been a manual process, which is costly, error-prone and inefficient. One key obstacle in automating this process has been the inability for machines to process normative requirements currently conveyed in natural language intended for human interpretation. Over the past 40 years, there have been numerous approaches to sharing normative requirements for automated compliance audit processes (Dimyadi & Amor, 2013). A common solution has been to represent normative requirements as rules that are hard coded into a compliance audit system. This "Blackbox" approach creates a snapshot of the normative requirements in the form of static rules that may not necessarily reflect the latest amendment of the source provisions. This approach lacks the transparency to allow independent verification of the correctness of the representation and has been reported as problematic and costly to maintain.

Previous research has shown that representing normative requirements in an open standard computable form is one solution towards enabling a "digital twin" of the source document that maintains the same status of the source provisions (Dimyadi & Amor, 2017; Dimyadi, Governatori, & Amor, 2017). This provides a guarantee that the computable rules always reflect the latest amendments.

Regardless of the representation approach, however, the first step towards computerising any legal text is the knowledge extraction process and the formalisation of that knowledge into computable rules. Fully automated knowledge extraction from natural language is still an active research topic despite extensive research over the years (Voorhees, 1999; Zhang & El-Gohary, 2013). At the other end of the scale, manual knowledge extraction by a domain expert remains a reliable approach, albeit laborious and costly. In between, there have also been numerous semi-automated approaches suggested by researchers (Dragoni et al., 2016; Kiyavitskaya, Zeni, & Breaux, 2007; Strahonja, 2006; Wyner & Peters, 2011).

The current research sets out to investigate to what extent natural language normative requirements, such as those conveyed by regulations or a building code, can be computerised to support automated compliance audit processing of a given building design.

1.2 Open Standard Legal Knowledge Model (LKM)

Emerging open legal knowledge interchange standards *LegalDocML* (LDML) and *LegalRuleML* (LRML) (OASIS, 2015, 2016) have recently drawn some attention among researchers in the Architectural, Engineering, Construction (AEC) domain (Dimyadi et al., 2017; McGibbney & Kumar, 2013) as a potential de-facto standard for representing normative requirements in the domain. LDML is a standardisation of *Akoma Ntoso* (Cervone et al., 2016), a former UN project for e-Parliament services in the Pan-African context, which has been designed to represent the structure and literal content of a legal document. LRML (Athan et al., 2013) has been developed on top of the open standard RuleML (Boley, Paschke, & Shafiq, 2010) with formal features specific to norm modelling, and is intended to represent the semantic or logical content of a legal document. Together, LDML and LRML constitute LKM in the context of this research as they are complementary standards that are close coupled by means of isomorphism, in which each rule in LRML is linked to its legal source provision by a unique key in LDML.

1.3 New Zealand Building Code (NZBC)

The New Zealand Building Code (NZBC) is part of the Building Regulations made under and in accordance with the primary legislation for the AEC domain in New Zealand, the Building Act 2004. The NZBC is a performance-based code, which specifies how a building is required to perform in its intended use but does not define how this performance is to be achieved. In order to provide practical information on how the requirements can be met, the NZBC is supported by a set of documents that are either Acceptable Solutions, which set out technical specifications for construction systems, materials or methods; or Verification Methods, based on industry-established calculation methods, laboratory tests or in-situ tests for building components or systems. These documents provide prescriptive approaches to meeting the performance requirements of the NZBC, and designs that demonstrate compliance with them must be accepted by the building consent authority (BCA).

The NZBC is divided into clauses, each with associated acceptable solutions and verification methods, that relate to particular technical aspects of building design and construction, including stability (B series documents), protection from fire (C series documents), access (D series documents), moisture (E series documents), safety (F series documents), services and facilities (G series documents), and energy efficiency (H series documents).

2 The Computerisation Process

2.1 *General Process*

In the context of this research, computerisation pertains to the process of digitising a legal document by capturing its structure and literal content as well as the semantics of its normative texts by translating and formalising them into a set of computable rules, which are then encoded into an open standard format. It also extends to the process of enabling access to the computable rules by a specific application, such as in an automated compliance audit environment.

Figure 1: The Digitisation Process

As depicted in Figure 1, the digitisation process starts with document preparation where the structure and literal content of each document is captured. This is followed by the knowledge extraction step where the intent and semantics of the normative text are formalised into rules.

The rules are then encoded into the open standard LRML format, which is then made available for access and query by any software system that supports the LRML standard. To provide a white-box solution to automated compliance audit, the LRML version of the NZBC must be owned and maintained by the official government body or a certified third-party responsible for the upkeep of the paper-based source documents. The intention is for the digital version to be updated at the same time and alongside the source documents in response to any amendment. Any system can then request an authentication from the remote repository hosting the documents to access the digital content on demand.

A guidelines document was written at the outset to set out the standard and conventions to be used for each step of the digitisation process. A number of software tools (written in Python and Swift programming languages) were also developed to automate and manage some of the tasks involved in the process.

2.2 *Document Preparation*

New Zealand legislation, regulations including NZBC Acceptable Solutions and Verification Methods, and some normative standards are published online as PDF documents. The first step in the document preparation process is using a software tool (such as the Adobe Acrobat) to extract the content of a PDF document into plain text, which is then formatted into an intermediate XML data structure (Figure 2) that will enable mapping to the LDML schema. The main objective of this initial step is to ensure the structure of the document is captured as accurately as possible.

```
<section key="9.0" title="Disposal to Soak Pit">
    <paragraph key="9.0.1">
        <p>1 Where the collected surface water is to be discharged tc
        without causing damage or nuisance to neighbouring property,
        <commentary title="COMMENT:">Means of demonstrating the suita
        surface water to a soak pit may also require a resource manag
    </paragraph>
    <paragraph key="9.0.2">
        <p>2 Field testing of soakage shall be carried out as follows
        <ol>
            <li key="9.0.2.a">Bore test holes of 100 mm to 150 mm dia
            hole then this depth shall be taken as the depth of the s
            <li key="9.0.2.b">Fill the hole with water and maintain f
            a short time).</li>
            <li key="9.0.2.c">Fill the hole with water to within 750
            greater than 30 minutes, until the hole is almost empty,
            <li key="9.0.2.d">Plot the drop in water level against ti
            curve. If there is a marked decrease in soakage rate as t
            average can be adopted.</li>
        </ol>
    </paragraph>
    <paragraph key="9.0.3">
        <p>3 The soak pit shall be designed utilising soakage and stc
        overflowing. The rainfall intensity used in the design of the
```

Figure 2: Structural and Literal Content of Document

One of the software tools developed can be used to take this intermediate XML representation of the document as input and to generate a spreadsheet proforma with pre-populated text paragraphs and their corresponding rule IDs. This process also splits complex paragraphs into more manageable sentences to facilitate knowledge extraction by the domain expert. Rules are related by their rule IDs and can therefore be automatically grouped together at the end of the digitisation process.

The pre-populated proforma prepared for each document is then distributed to the domain expert team for the knowledge extraction exercise.

2.3 *Knowledge Extraction*

The knowledge extraction exercise involved manually capturing the logic (condition expressions and conclusions) inherent in individual normative text paragraphs and sentences and identifying atoms, their relationships, and logical operators. The outcome was highly dependent on the level of expertise and experience of individual domain experts undertaking the work.

Apart from text paragraphs, NZBC Acceptable Solutions and Verification Methods also contain many tables, graphs, and illustrated provisions, as well as explicit and implicit mathematical expressions. Some of these forms of normative requirements, particularly illustrations, are often not easily formalised into rules (see Section 4 for some discussions on the challenges). Tabulated provisions, however, can be encoded into LRML semi-automatically in most cases by means of pre-populated and proforma-based specifications.

Atoms (entities, attributes, and relationships) and logical operators extracted from each sentence were entered into their respective places on the Knowledge Capture Profoma (Figure 3) along with logical expressions and deontic operators (obligation, prohibition, permission) to form one or more logical statements for each rule. Atoms and operators were also added to a centralised LKM data dictionary, which had been developed to align with the buildingSMART Data Dictionary (bsDD), formerly known as the International Framework for Dictionaries (IFD) (ISO, 2007). Additional parameters such as mathematical functions and intermediate variables were introduced into each rule as necessary to convey the intent of the provision and to facilitate computability of the rule.

As mentioned above, tabulated and some illustrated provisions can be generated into LRML rules semi-automatically through a machine-readable table schema proforma (Figure 4) , which incorporated input and output components and other conditional parameters specification.

RuleId	if | then	deontic		neg	fun	rel	var	operator	val
1.3.2	if		(						
				not		activity	building	is	household u
			or			riskGroup	building	equal	SH1
			)						
			and		count	step	accessRoute	equal	1
			and			purpose	step	is	threshold we
	then	prohibition				height	step	greaterThan	20 mm
1.4	if						building	has	buvo:accessI
	then	obligation				heightClearance	accessRoute	complyWith	table
			and			key	table	equal	t1.1
			and			heightClearance	accessRoute	complyWith	figure

Figure 3: Knowledge Capture Proforma

table	6.1		
input	1 and 2		
output	3 and 4		
1	2	3	4
stair.type equal	stair.pitch lessThanEqual	stair.riserHeight lessThanEqual	stair.treadDepth lessThanEqual
service	47 deg	220 mm	220 mm
minorPrivate	47 deg	220 mm	220 mm
secondaryPrivat	41 deg	200 mm	250 mm

Figure 4: Table Schema Proforma

2.4 *Validation, Formalisation and Encoding*

Each completed knowledge capture proforma was subject to quality assurance peer-review by another member of the team. The main objective is to review that the logical content of each sentence has been captured correctly and in accordance with the guidelines. More importantly, this step also ensures that the standard and conventions as specified in the guidelines are followed consistently throughout the process.

Once the knowledge capture proforma has passed the peer-review process, it is then subject to validation by the LKM Data Processor, which is a dedicated software application developed independently for managing LKM.

The LKM Data Processor also takes prepopulated table schema proformas related to a document and generate the corresponding LRML rules.

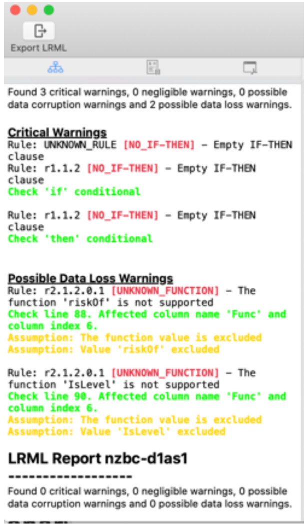

Figure 5: Validation Check Warnings

The data processing by the LKM Data Processor involves checking the content of the proforma for unknown or unidentified atoms or operators, incorrect logical expressions and syntax. A list of warnings is given as part of the validation (Figure 5).

The validation process may be repeated until all the errors and warnings are corrected or attended to on the source proformas. At the end of the validation process, the LKM Data Processor would encode the content of the proformas into valid LRML rule statements (Figure 6) as well as grouping related rules together by their rule IDs into associations. Some of the calculations performed as

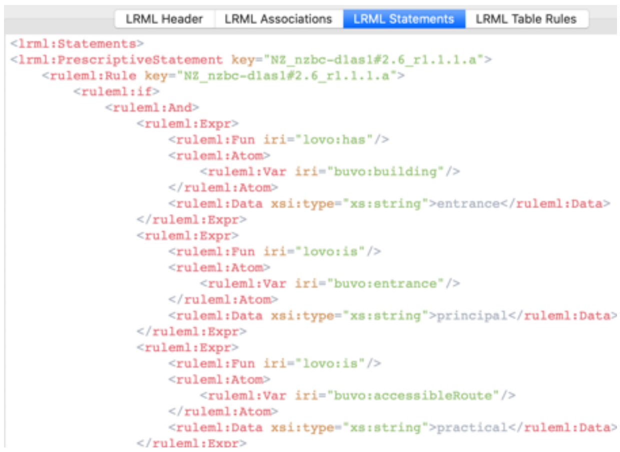

Figure 6: LRML Rule Statements

part of the formalisation process include identifying and processing the correct data types and units of measurement in the atoms, which would support the computability of the rules in the application environment.

The LKM Data Processor can also extract relevant metadata from the LDML file to construct a valid header for the LRML file.

Additionally, the LKM Data Processor supports two-way interoperability between LRML and the source proformas. The tool can also be used to manage LKM versioning and temporal characteristics of the LRML rules.

3 Case Study

3.1 *Scope*

The focus for this case study was on the prescriptive step-by-step normative requirements and processes stipulated by the Acceptable Solutions, and 20 of them from clauses B, C, E, and G of the NZBC were selected for digitisation. In addition, two Verification Methods related to G clause of the NZBC were also included, as their content was considered essential in the priority consenting environment where an automated compliance audit process will take place.

3.2 *Resources and Roles*

A team of six experts was involved in digitisation, mainly dedicated to the document preparation and knowledge extraction process. Members of the team brought expertise in a variety of domains, including knowledge engineering and computational logic, data science, fire engineering, architectural design, construction management, and software engineering. They were assigned different roles throughout the process, such as document preparation, analysis, formalisation and translation, review and verification, and software development.

4 Experience and Challenges

4.1 *Document Preparation Experience*

Most of the documents selected for digitisation had a similar hierarchical structure. However, there were some variations in the hierarchy and naming convention between some documents that caused issues when generating the knowledge capture proforma using the software tool. This was handled by scanning all of the documents and incorporated all document structure variations into a schema specification (Figure 7).

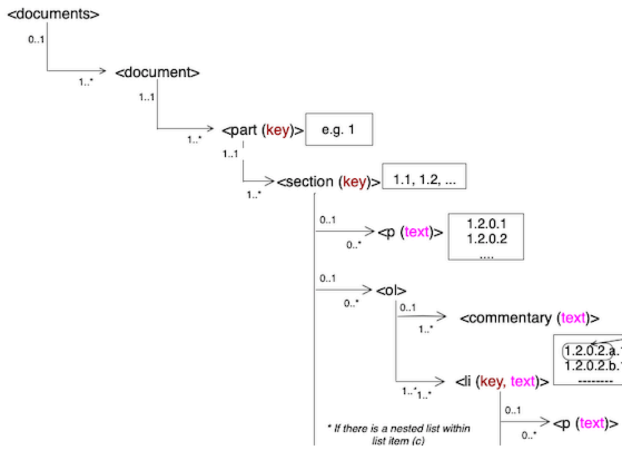

Figure 7: Document Schema for Knowledge Capture Proforma

Apart from variations in the structure of the source documents, there were also variations introduced as part of the initial document preparation from PDF to plain text and manual formatting into XML. This had to be managed iteratively by correcting any errors and eliminating unexpected elements discovered by the software tool.

4.2 *Knowledge Extraction Experience*

Starting the knowledge extraction process through an expert-driven process brought its own challenges. Members of the team brought different levels of expertise and experience and came from different background. The diversity of expertise and background coupled with the subjective nature of natural language interpretation resulted in many different ways knowledge could be extracted from a text. Two divergent approaches were taken to address this issue. To explore the extent of the variation, several of the Acceptable Standards were given to multiple people to translate, and the resulting translations were compared and shared amongst the team. Preferred translations were identified and disseminated as templates for specific types of text. In other cases, each Acceptable Standard was given to a different individual to translate so that a single approach could be used throughout a text. In practice, an amalgam of both approaches was used on most documents, with members of the team collaborating to address gaps or determine appropriate methodologies.

A related issue was the standardisation of atoms and logical operators. The framework used for these was based on the glossaries provided in the NZBC and the bsDD, but there were many necessary terms and expressions that were not included in either. This was addressed iteratively through the use of an interim centralised data dictionary that was developed over the course of the process. Each Acceptable Solution brought its own definition of terms that was straightforward enough to negotiate, but one challenge was the use of similar operators that held different relationships within different contexts. For example, a structural member required to be *under* another has a different relationship to a drain that is *under* a building, despite the use of the same expression. The dictionaries of vocabulary were developed iteratively with agreed expressions and more efficient methodologies of defining expressions being disseminated to the team. Completed translations were reviewed and updated as better approaches were developed.

The complexity of many of the statements within the Acceptable Solutions caused many challenges. Clauses can include multiple sub-clauses, be cross referenced to other clauses, refer to illustrations or tables, refer to external documents such as standards or other Acceptable Solutions or Verification Methods, or involve multiple factors or relationships within one clause. Resolving some of these complex clauses became a multi-level problem, with questions of how complex statements should be allowed to get in the translation process, and how many statements to combine with combinations of AND and OR in order to define a rule. Although an early principle was to keep rules as simple as possible, in some cases it became cumbersome to divide complex clauses into individual rules, and so more convoluted combinations of AND and OR became necessary, which may impact on the computability of the rule in an application.

Computations introduced another situation where rules often became complex. The options in cases where a computation was introduced in an Acceptable Solution or Verification Method are to either encode the calculation process directly into a rule, or to define a function that fulfils the calculation, and refer to the function in the rule. The former was generally preferred, but there were exceptions and hence it was addressed on a case by case basis. Where a calculation is initially used it has tended to be represented as a rule, wherever possible. However, when it is repeated throughout a document it becomes for efficient to define the function separately. This allows a function to be defined once and used across different Acceptable Solutions and Verification Methods and requires subsequent review of previously digitised documents to follow a standard process.

Spatial and temporal computations and other more complex analyses often require external support such as third-party simulation or other tools. Other situations require human judgement or analysis and cannot be completely encoded into rules. All of these situations are accepted in this approach, with the computable aspects identified and coded for automated compliance checking, but additional resources such as tools or human expertise can be called on where needed.

4.3 *Challenges*

Despite the prescriptive intent of the Acceptable Solutions, some of the language used is imprecise and relies on human judgement. While this is a necessary element in many situations, ambiguity is introduced when specifications or minimum standards are stated but include the caveat “where practicable”, for example.

Challenges that arose were not all a result of the process of translation from natural language to a digital structure. The systematic process of breaking down the many rules highlighted a number of ambiguities or contradictions within the Acceptable Solutions themselves. In some cases, requirements to meet an Acceptable Solution for one clause of the NZBC contradicted requirements of an Acceptable Solution for another. Other issues included cross-references to out-of-date or inappropriate standards documents, incorrect or inconsistent terminology, and unclear or ambiguous writing within the documents.

5 Discussions and Conclusion

A case study to digitise a set of compliance documents from the New Zealand Building Code has been presented. The entire process and challenges experienced by the team undertaking the project have been described.

The entire digitisation project took 6 months to encode 10,729 rules. This represents approximately 80% of all normative text contained in the selected Acceptable Solutions and Verification Methods documents.

Some of the LRML rules produced as part of this study have been tested in a prototype workflow-model-driven automated compliance audit system in conjunction with a given building model. The findings from preliminary tests suggest that only half of the rules can be used without supplementary human input. The main reason was the inadequate level of details available in the building model used in the test. Another case study has been scheduled to investigate to what extent additional information will be needed to assist the automated compliance audit using these rules.

Future work includes combining spatial and temporal operations with LRML to extend its capabilities in resolving geometry-related and geometry-dependent operations.

Acknowledgement

This research received funding from the National Science Challenge (NSC 11) - Building Better Homes, Towns, and Cities (BBHTC).

References

Athan, T., Boley, H., Governatori, G., Palmirani, M., Pasckhe, A., & Wyner, A. (2013). OASIS LegalRuleML. In *The International Conference on Artificial Intelligence and Law*. Rome, Italy.

Boley, H., Paschke, A., & Shafiq, O. (2010). RuleML 1.0: The overarching specification of web rules. In J. J. Alferes, L. Bertossi, G. Governatori, P. Fodor, & D. Roman (Eds.), *International Workshop on Rules and Rule Markup Languages for the Semantic Web (RuleML 2010)* (Vol. 6403, pp. 162–178). Washington, D.C., USA: Springer. doi:10.1007/978-3-642-16289-3-15

Cervone, L., Di Lorio, A., Palmirani, M., & Vitali, F. (2016). Akoma Ntoso 3.0 Schema. *The Akoma Ntoso 3.0 - WD17*. Retrieved from http://www.akomantoso.org/release-notes/akoma-ntoso-3-0-schema/

Dimyadi, J., & Amor, R. (2013). Automated Building Code Compliance Checking - Where is it at? In *19th International CIB World Building Congress (CIB WBC 2013)* (pp. 172–185). Brisbane, Australia: Queensland Institute of Technology.

Dimyadi, J., & Amor, R. (2017). *Automating conventional compliance audit processes*. *IFIP Advances in Information and Communication Technology* (Vol. 517). doi:10.1007/978-3-319-72905-3_29

Dimyadi, J., Governatori, G., & Amor, R. (2017). Evaluating legaldocml and legalruleml as a standard for sharing normative information in the aec/fm domain. In F. Bosche, I. Brilakis, & R. Sacks (Eds.), *Proceedings of the Joint Conference on Computing in Construction (JC3)* (Vol. I, pp. 637–644). Heraklion, Greece: Heriot-Watt University, Edinburgh, UK. doi:10.24928/JC3-2017/0012

Dragoni, M., Villata, S., Rizzi, W., & Governatori, G. (2016). Combining NLP Approaches for Rule Extraction from Legal Documents. In *Proceedings of the workshop on "Mining and Reasoning with Legal Texts" collocated at the 29th International Conference on Legal Knowledge and Information Systems* (pp. 1–13).

ISO. (2007). *ISO 12006-3:2007 Building Construction - Organization of Information about Construction Works - Part 3: Framework For Object-Oriented Information* (1st ed.). Geneva, Switzerland.

Kiyavitskaya, N., Zeni, N., & Breaux, T. D. D. (2007). Extracting rights and obligations from regulations: toward a tool-supported process. In *ASE* (Vol. 1, pp. 3–6). Atlanta, Giorgia, USA.

McGibbney, L. J., & Kumar, B. (2013). A comparative study to determine a suitable representational data model for UK building regulations. *Journal of Information Technology in Construction*, *18*(February), 20–38. Retrieved from https://gcu.elsevierpure.com/ws/portalfiles/portal/1136085/ITConPaper_Jan2013.pdf

OASIS. (2015). LegalRuleML Technical Committee Charter. Retrieved from http://www.oasis-open.org/committees/legalruleml/charter.php

OASIS. (2016). OASIS LegalDocML TC. *OASIS*. Retrieved from https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=legaldocml

Strahonja, V. (2006). Modeling Legislation by using UML State Machine Diagrams. *2006 Canadian Conference on Electrical and Computer Engineering*, (May), 624–627.

doi:10.1109/CCECE.2006.277286

Voorhees, E. M. (1999). Natural language processing for information retrieval. *Information Extraction*, 1–17.

Wyner, A., & Peters, W. (2011). On rule extraction from regulations. In *Proceedings of JURIX* (pp. 1–10). Retrieved from http://wyner.info/research/Papers/WynerPetersJURIX2011.pdf

Zhang, J., & El-Gohary, N. M. (2013). Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking. *Journal of Computing in Civil Engineering*, 1–39. doi:10.1061/(ASCE)CP.1943-5487.0000346