REGULATORY KNOWLEDGE REPRESENTATION FOR AUTOMATED COMPLIANCE AUDIT OF BIM-BASED MODELS

Johannes Dimyadi, PhD Researcher, jdim006@aucklanduni.ac.nz
Robert Amor, Professor, trebor@cs.auckland.ac.nz
Department of Computer Science, University of Auckland, New Zealand

ABSTRACT

There has been significant research in the area of automated and semi-automated regulatory compliance checking in the AEC domain over the past four decades. In order to computerise the regulatory compliance checking process, we first need to have computer representations of both the building model and the regulations. We now have Building Information Modelling (BIM) as the industry standard representation for buildings, but the challenge remains to find an efficient and practical digital representation of the regulatory knowledge.

One common approach to represent regulatory knowledge for architectural and building engineering designs is to extract rules from the regulatory texts, which has proven to be quite challenging to automate. However, a popular technique used in Business Process Management (BPM) could be adapted to facilitate the representation of regulatory knowledge in the AEC domain. Additionally, the legal domain has come up with some initiatives to provide a means to share digital legislative documents and regulatory structure, e.g. Cen MetaLex, LKIF, LegalRuleML, etc., which could be useful for exchanging compliance checking data in the AEC domain.

This paper outlines a practical approach to represent regulatory knowledge in terms of industry's accepted compliant design procedures using a business process modelling technique with visual editing capabilities. The corresponding regulatory constraints, thresholds and conditional logic are treated as lookup data and rules, which could also be represented graphically. Each design procedure and its associated regulatory data would constitute the regulatory knowledge required for a particular type of design or compliance check. This paper also briefly outlines a compliance checking framework that could bring in any library of regulatory knowledge representations that are defined and maintained externally. Both the prescriptive and performance-based criteria of the New Zealand Building Code are being investigated.

Keywords: Knowledge Management; BIM; Design procedures; Regulatory knowledge

1. INTRODUCTION

In the context of the Architecture, Engineering, Construction (AEC) domains, regulatory knowledge refers to the procedures, rules and normative data that can be used to achieve a compliant design, which is an essential ingredient of an automated compliance checking system.

Over the last four decades, there has been significant research in the AEC industry on the representation of regulatory knowledge for this purpose. One of the earliest successful attempts was the use of decision tables to aid engineering design for conformance with the AISC (American Institute of Steel Construction) Specifications (Fenves et al., 1969). This was successfully implemented into a design tool and used well into the mid 1980's through a number of developments such as SASE, SICAD, SPEX (Fenves and Garrett, 1986). Throughout the 1990's there were a number of rule-based or knowledge-based expert systems being investigated around the world. There have been a number of successful prototype systems implemented in Australia, Singapore, Finland, Sweden, Norway, Germany, France and the USA. However, these systems have not yet been taken up successfully by the industry. In the absence of an automated system, the current paper-based practice of compliance checking in the industry remains inefficient, error-prone and very costly (Gallaher et al., 2004).

A number of pilot systems in the AEC industry employed hard-coded IF-THEN-ELSE rules to represent prescriptive regulatory requirements, e.g. Solibri Model Checker (Eastman et al., 2009). Some of the challenges associated with hard-coding rules include the inflexibility to modifications, the dependency on the system programmer to recode any changes, and the difficulty for code experts to ascertain correctness of program code. One of the most common approaches to capture regulatory semantics and rules is to extract them directly from the regulatory texts. This can be done manually by direct transcription or by using mark-up techniques. More recently, automated or semi-automated data extraction approaches employing artificial intelligence (AI) techniques such as Natural Language Processing (NLP) and semantic modelling have been explored (Zhang and El-Gohary, 2012a, 2012b), but it has proven to be quite challenging. Ontology-based regulatory knowledge representation on the semantic web has also been explored by researchers in conjunction with inferencing languages, e.g. SPARQL Inferencing Notation (SPIN), Semantic Web RuleML (SWRL), Rule Interchange Format (RIF) (Bouzidi et al., 2012; Pauwels et al., 2011). Regulatory texts are generally written for human interpretation, therefore it would be challenging to come up with any automated method of information extraction that would satisfy all possible scenarios. Any digital representation should be in accordance with one set of the official interpretation provided by the authority or domain experts.

Meanwhile, researchers in the legal informatics domain have come up with methods to share legislative documents and regulatory structure electronically and have implemented a number of open standard models to represent legislative information, e.g. AKOMA NTOSO, CEN MetaLex, LKIF, RuleML, and more recently LegalRuleML. The latter is being standardised by OASIS (Organisation for the Advancement of Structured Information Standards) to provide a standard rule interchange language for the legal domain (Athan et al., 2013; Vitali and Zeni, 2007). This could be useful as a medium to exchange compliance checking data with external applications. Modern techniques in Business Process Management (BPM) such as the Business Process Model and Notation (BPMN 2.0) and the use of advanced rule engines to represent the business logic could be adapted for compliance auditing purposes in the AEC domain.

2. PERFORMANCE-BASED BUILDING CODE OF NEW ZEALAND

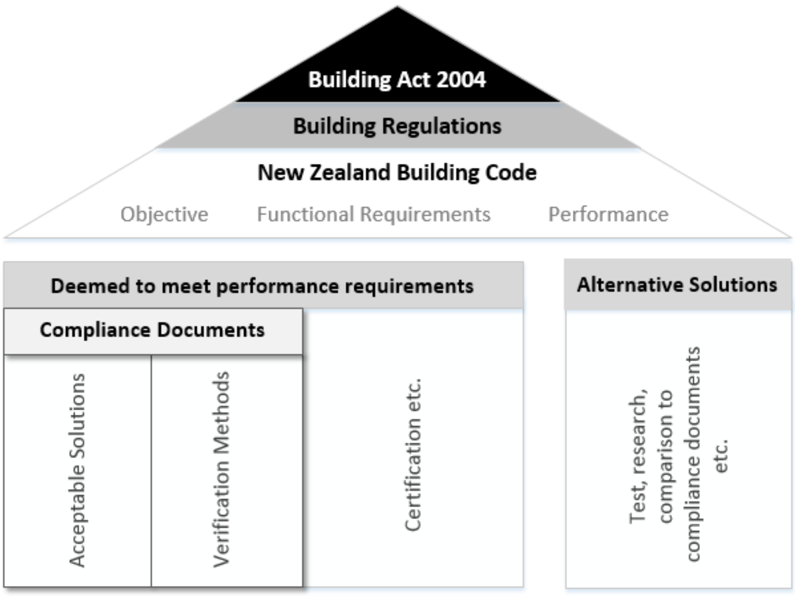

Figure 1: New Zealand Building Control Framework, adapted from (Building and Housing, 2011)

Building controls in New Zealand are legislated by a three-tier framework (Figure 1), i.e. the Building Act, which provides the overall legislative governance; Building Regulations, which are made under the Act; and the New Zealand Building Code (NZBC), which is part of the Building Regulations and sets out the mandatory objective, functional requirements and performance criteria for the construction of all new buildings and alterations to existing buildings (Building and Housing, 2007).

As with many performance-based codes in the world, NZBC provides two means of compliance, i.e. the deemed-to-satisfy methods and the Alternative Solutions. The most common approach to comply via the deemed-to-satisfy methods is by using the Compliance Documents, which comprises the Acceptable Solutions and the Verification Methods. Each Acceptable Solution document contains a set of prescribed methods of construction and restrictions on material use, and is one way to demonstrate compliance. The Verification Methods can also be used to demonstrate compliance by means of prescribed engineering calculation methods or tests. The largest set of Verification Methods provided in NZBC is that for fire safety design. This requires external computations and simulations at certain steps of the procedure to help evaluate fire scenarios and the tenability of escape routes.

3. ANATOMY OF A COMPLIANCE CHECKING FRAMEWORK

To address some of the issues found with the inflexibility and inefficiency of the current implementation of regulatory representations, a practical compliance checking framework is being investigated (Figure 2). The core functionality of the framework is the Checker, which would bring in all other input components (i.e. building model, library of design procedures, and regulatory data and rules) that are defined and managed externally. The intermediate output would be a series of violation alerts with suggestions for different design options that could achieve compliance, if applicable. The final output of the framework would be a report highlighting any unresolved violation.

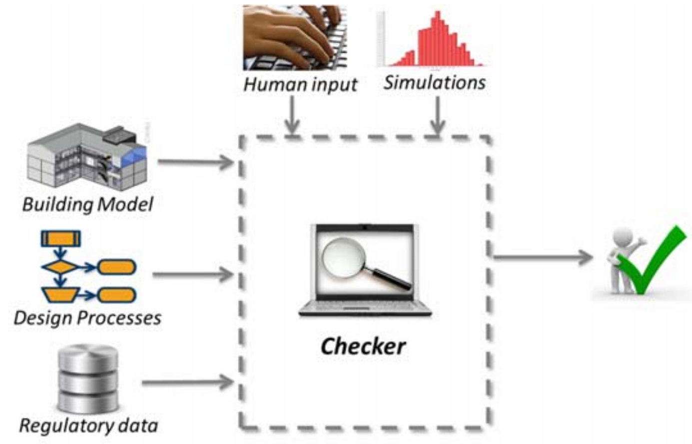

Figure 2: Anatomy of a Compliance Checking Framework

A predominant use case of the framework would be to aid the design process. Proving compliance as part of the design process would be more cost effective than for violations to be discovered during the official consent auditing stage as it would result in unscheduled design revisions that could contribute to project delays and additional costs. Furthermore, following a design procedure and focusing only on certain relevant parts of regulations would be more efficient than checking through the complete set of regulatory requirements, which cover many aspects of the design.

4. BUILDING INFORMATION REPRESENTATION

The main input component to the compliance checking framework is the Building Model, which can be created as a semantically rich functional and physical representation of the actual building using Building Information Modelling (BIM) techniques. Building information can now be exchanged using the ISO 16739 Industry Foundation Classes (IFC) open standard data model. BIM-based models can be shared between stakeholders of the project using a range of BIM management applications such as the open source BIMServer (Singh et al., 2011).

A building is a complex object to represent, which is reflected in the relatively large IFC schema (i.e. the latest version has 766 entities). However, a subset of the IFC model can be defined using a Model View Definition (MVD) that is tailored to provide a dedicated view, e.g. for compliance checking purposes (Nawari, 2011).

5. REGULATORY KNOWLEDGE REPRESENTATION

Compliant architectural and engineering design procedures (e.g. fire safety, durability, accessibility, energy efficient design, etc.) are usually based on best practice that are adopted by industry's practitioners as de-facto standards. Each procedure is really a subset of the relevant section of the building code because it only covers one particular aspect of the design, e.g. a particular type of building or occupancy. The design procedures may have steps that look up relevant sections of the building code for parameters, values, limits, constraints and thresholds. There may also be a set of conditional logic that must be evaluated as well, which may lead to different design options. These values, limits, constraints, thresholds and conditional logic can be represented as regulatory data and rules.

Each design procedure and its corresponding regulatory data and rules would then constitute the regulatory knowledge for that particular type of design or compliance audit. So, a BIM model would be audited against different sets of regulatory knowledge for different design aspects. To audit a complete building design for compliance against the entire building code, there will need to be a complete library of regulatory knowledge covering all the requirements of the building code, each with a set of design procedures and their associated regulatory data and rules pertaining to a particular design or compliance checking aspect.

5.1 Representation of Design Procedures

There are many ways to represent design procedures in computer readable forms, e.g. as hard-coded rules, programming logic, etc. Ideally, the representation should be defined and managed by the domain experts, who are generally not computer programmers. In order for this task to be managed effectively and conveniently, there needs to be a high level user-friendly front-end tool. One common approach for modelling formal business processes in other domains is the use of an internationally accepted workflow language such as BPMN 2.0, which is an OMG (Object Management Group) standard, along with a visual editor. The resultant Business Process Diagram (BPD) can be executed in a computing environment to produce a computer readable formalism such as XML (Becker and Delfmann, 2012; Governatori and Shek, 2013), which can then be used for compliance checking purposes. This practical approach could be adapted to represent design procedures in the AEC domain (Figure 3) as it lends itself well for designs that follow either the prescriptive path of the Acceptable Solutions or the Verification Methods due to their procedural nature.

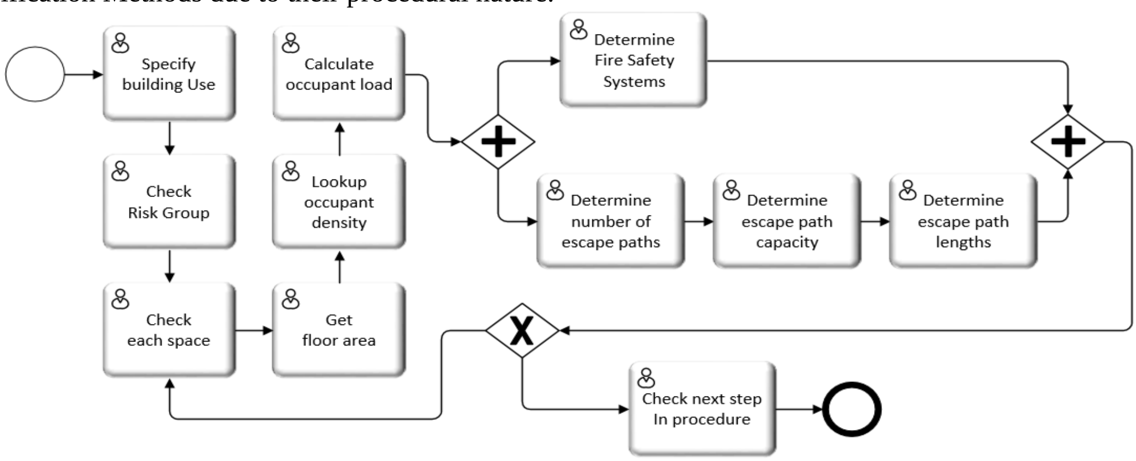

Figure 3: An example design procedure represented in BPMN 2.0

Using a graphical notation approach would provide a user with the flexibility of varying a design procedure as required to suit the design brief. At certain steps in the path of an Acceptable Solution design, each procedure may need to obtain various regulatory data and evaluate different regulatory rules. These regulatory data and rules should be represented independently but managed in conjunction with the corresponding procedures (refer to Section 5.2 for more discussions). Similarly, at various steps in the Verification Methods, the procedure may call upon external simulations or computations for inputs. See Section 6 for more discussions on the support for performance-based requirements.

5.2 Representation of Regulatory Data and Rules

Apart from providing compliant design procedures covering every aspect of compliant building designs, the building code also contains regulatory factual data such as acceptable values, thresholds, as well as conditional statements such as constraints or rules. Some examples of these regulatory data and rules are as follows:

- Values: risk groups and occupant density based on space usage type, clearance to boundaries, etc.
- Thresholds: the length of one direction escape path must not exceed 20 m
- Constraints or rules: door must swing outwards when occupant load using the door is greater than 50 persons; If occupant load in a space is greater than 50, two exits must be provided

One way to represent regulatory data is by using database lookup tables such as shown in Table 1 and Table 2. A desirable attribute is for the representation to be adaptable to frequent regulatory changes and easily accessible by the code experts who would be involved in amending the code.

Activity	**OccupantDensity (m^2/person)**
Cafeteria	1.25
Exhibition	1.4
Classroom	2
Daycare	4
Fitness	5
Office	10

Table 1: Sample values lookup table (from NZBC C/AS4 Table 1.2)

Escape path type	**Manual or no fire alarm system**	**Smoke detection system installed**
One direction escape path length	20 m	40 m
Total escape path length	50 m	100 m

Table 2: Sample thresholds lookup table (from NZBC C/AS4 Table 3.2)

Regulatory rules can be represented in a number of ways, e.g. decision tables, decision trees, hard-coded declarative logic, domain-specific rule language, etc. Again, any representation should be easy to manage and adaptable to continuing regulatory amendments. These rules should exist in different rulesets that correspond with each design procedure. To explore which method would be most suitable to represent regulatory rules for compliance checking in the AEC domain, three different approaches are compared, i.e.

1. DBMS (Database Management System) user-defined functions and look up tables
2. Rule Sets in the Solibri Model Checker (SMC)
3. Drools Rule Language (DRL) with a graphical editor.

Two exemplar regulatory rules from NZBC C/AS4 have been used for the comparison (Table 3).

C/AS4 Paragraph 1.4.2	Occupant loads shall be calculated from the occupant densities given in Table 1.2 based on the floor area of the part of the building housing the activity.
C/AS4 Paragraph 3.15.3	Doors on escape routes shall be hung to open in the direction of escape. However, this is not required if the number of occupants of spaces with egress using the door is no greater than 50.

Table 3: Example regulatory rules from NZBC C/AS4

5.2.1 DBMS approach

A general purpose DBMS can be used to represent the above simple regulatory rules via user-defined functions (Table 4) and stored procedures.

```sql
/* Determine occupant load of a space */
CREATE Function
[dbo].[CalculateSpaceOccupantLoad]
(@Activity varchar(20),
@FloorArea decimal(18,2))
Returns int
As
Begin
 Declare @SpaceOccupantLoad int
 Begin
   SET @SpaceOccupantLoad =
   CEILING(@FloorArea/(Select OccupantDensity
   From dbo.OccupantDensities
   Where Activity = @Activity))
 End
Return @SpaceOccupantLoad
End
```

```sql
/* Check door opening direction */
CREATE PROCEDURE
[dbo].[CheckDoorOpeningDirection]
(@OccupantLoad Int,
@DoorOpeningDirection bit)
AS
BEGIN
   If @OccupantLoad > 50
     and @DoorOpeningDirection = 1
   Select 'Door must open outwards'
END
```

Table 4: DBMS user-defined function and procedure in MS SQL Server to represent regulatory rules

One drawback of this approach is that it would require someone with DBMS programming knowledge to code. A dedicated web-based user interface such as shown in Figure 4 could be developed in conjunction with a domain-specific language to assist with the generation of one-off functions or rules, which would be useful as part of the ruleset maintenance, e.g. to replace an outdated rule or to add a supplementary rule in the ruleset.

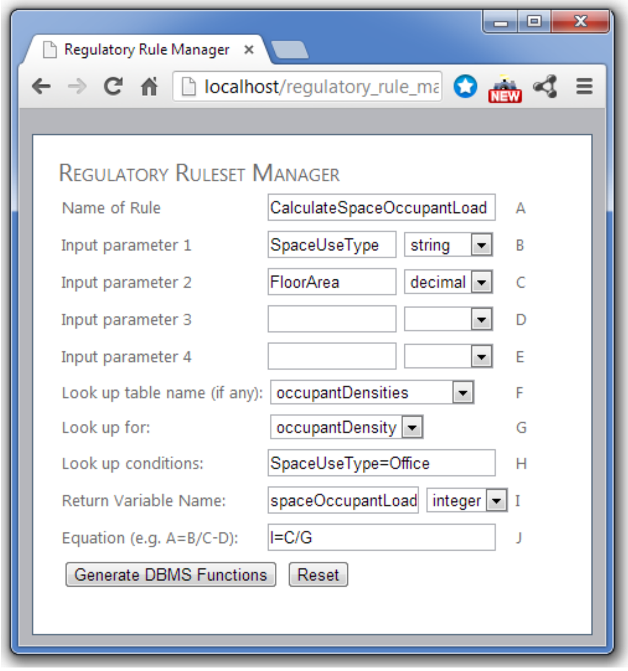

Figure 4: A user interface that could generate a DBMS function

DBMS user-defined functions and stored procedures and look up tables are suitable for a data-driven compliance checking application, however the viability of developing an automated mechanism to generate and manage the rules efficiently in DBMS would need to be further investigated.

5.2.2 SMC Ruleset

Unfortunately, it is not possible to represent the example rules using SMC, which is a popular commercial application used in the industry primarily for detecting object clashes in BIM models and for general model quality auditing. The application comes with a library of predefined rules against which many aspects of the model can be checked. However, there are only two sample rulesets that can be used for regulatory compliance checking, i.e. accessibility to ISO/DIS21542, and a generic fire egress analysis based on common fire code requirements. Although the built-in rules have parameters that can be customised to some extent, these can only be modified using the proforma attached to each rule. Furthermore, the generated rule codes are not visible to the user, so their accuracy cannot be verified.

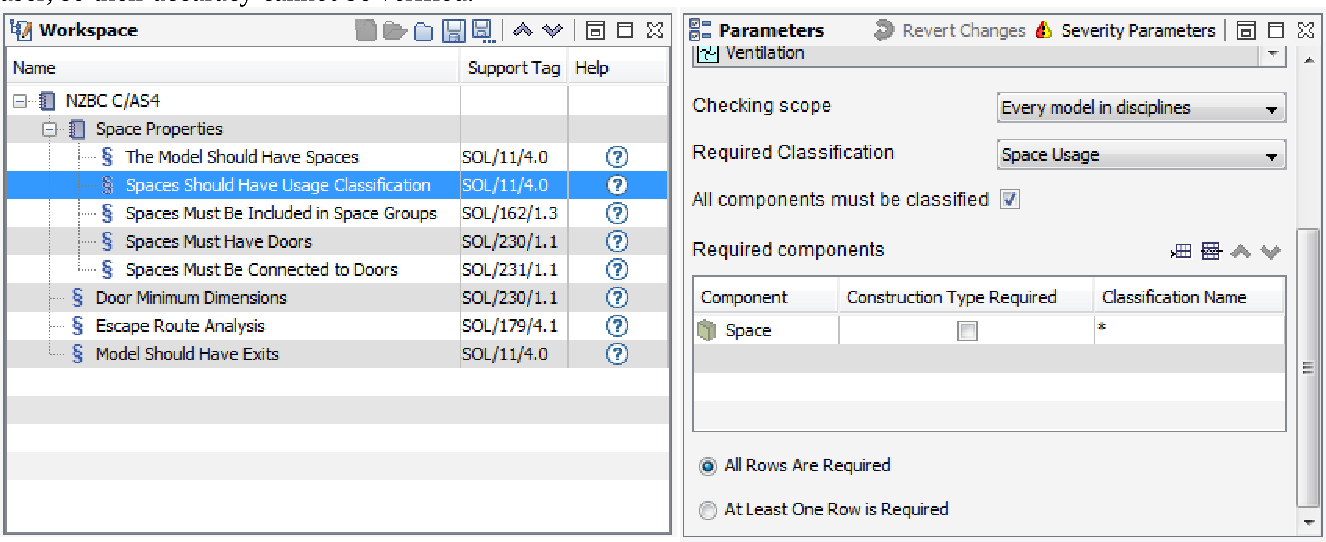

Figure 5: NZBC C/AS4 new ruleset showing a particular rule with its parameters in SMC

Individual new rules cannot be created by the user, but it is possible to bundle a selected number of built-in rules into a new ruleset for a particular compliance checking task. For this exercise, a new ruleset named NZBC C/AS4 has been compiled from the built-in rules for the purposes of checking compliance against some aspects of the means of escape provision of the Acceptable Solutions. Unfortunately, many compliance aspects specific to NZBC cannot be checked as the required rules would first need to be created by software programmers at Solibri.

A new feature provided in the latest version of SMC (version 8.1) allows the user to classify a space according to its intended use. This is useful as many regulatory requirements are based on the type of activity in the space, e.g. determining the required minimum number of escape routes. However, it would also be advantageous to include a feature for calculating the occupant load based on the occupant density associated with the activity in a space.

SMC has useful features for checking many aspects of a building design, but in order to be useful as a compliance checking system, it would need to have a rule engine that can interpret user-defined rules. At the moment, this is not yet possible.

5.2.3 Drools Rule Language (DRL)

DRL is the native rules language for Drools, which is a popular JBoss Community open source and advanced Java inference engine that has been around since 2001 and used for Business Process Management (BPM). The engine uses an improved Rete algorithm for pattern matching, which is adapted for object oriented systems such as Java (De Ley and Jacobs, 2011). DRL could be used to represent the regulatory rules (Table 5). Apart from DRL, Drools also directly supports rules generation via decision tables and domain-specific language mapping. Two other related products, i.e. JBPM and Guvnor could potentially be used to generate the rules graphically. There have also been other graphical methods for modelling rules in the business domain (Lukichev and Jarrar, 2009).

The anatomy of a DRL rule consists of two parts, namely the conditions (or WHEN) and the actions (or THEN). The conditions are a set of pattern matching or query statements in an object query language. The actions part can contain any valid Java code or related scripts.

`function int OccupantLoad(double OccupantDensity, char Activity, double FloorArea)`
`{`
`OccupantDensity = getOccupantDensity(Activity);`
`return (int)(FloorArea/OccupantDensity);`
`}`
	`rule "CheckDoorOpeningDirection"`
	`when`
	`Occupant (OccupantLoad > 50) &&`
	`ExitDoor (OpeningDirection == "in")`
	`then`
	`warning("Door must open outwards");`
	`end`

Table 5: DRL function and rule to represent the example regulatory rules.

Recent work in BPM has shown that a rule can be represented as a single Business Process Diagram (BPD) (Figure 6) (Di Bona et al., 2011). The conditions (or WHEN) and the actions (or THEN) parts of the rule can each be represented by a BPMN Sub-Process (Figure 7 and Figure 8).

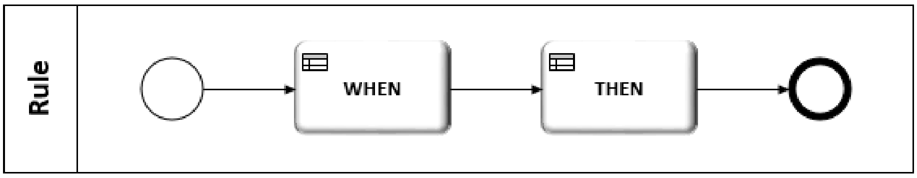

Figure 6: A rule represented by a BPD

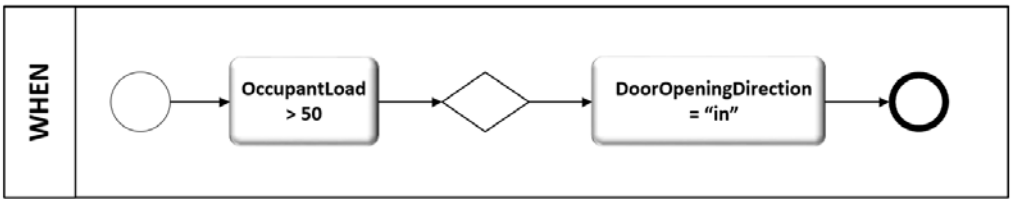

Figure 7: The Conditions (When) part of the rule represented as a BPMN Sub-Process

Figure 8: The Actions (Then) part of the rule represented as a BPMN Sub-Process

A domain-specific language would need to be developed for use in conjunction with the graphical representation to specify the conditions and assign executable actions to the individual task flows.

5.3 Library of Regulatory Knowledge Representations

NZBC has 35 technical clauses, e.g. Structure, Durability, Protection from Fire, Access Routes, Warning Systems, etc. Each clause has a companion compliance document with its own set of requirements, which makes an extensive library of regulatory knowledge.

To check compliance against all of the technical clauses, a library of individual design procedures corresponding to each technical clause would be required. A design procedure should be limited to a reasonable size having a manageable number of steps to be practical. Large design procedures could be divided into a number of smaller sub-procedures. Ideally, each set of regulatory knowledge would be represented by two BPDs that are managed independently, one for the design procedures and the other for the regulatory data.

6. SUPPORT FOR PERFORMANCE-BASED REQUIREMENTS

For the Verification Method path of compliance with NZBC, external computations and simulations may be required to help fill in some of the input parameters. This is particularly true with fire safety assessment where ten fire scenarios would need to be evaluated. Some of these scenarios can be evaluated by using spreadsheet calculations, but others would require full scale CFD simulations. The procedures can also be represented using the BPMN visual notation (Figure 9). Which external applications to use to provide the input required by the procedures depends on which applications can satisfactorily meet the input/output specification. The process of automatically tendering for a suitable application is yet to be investigated.

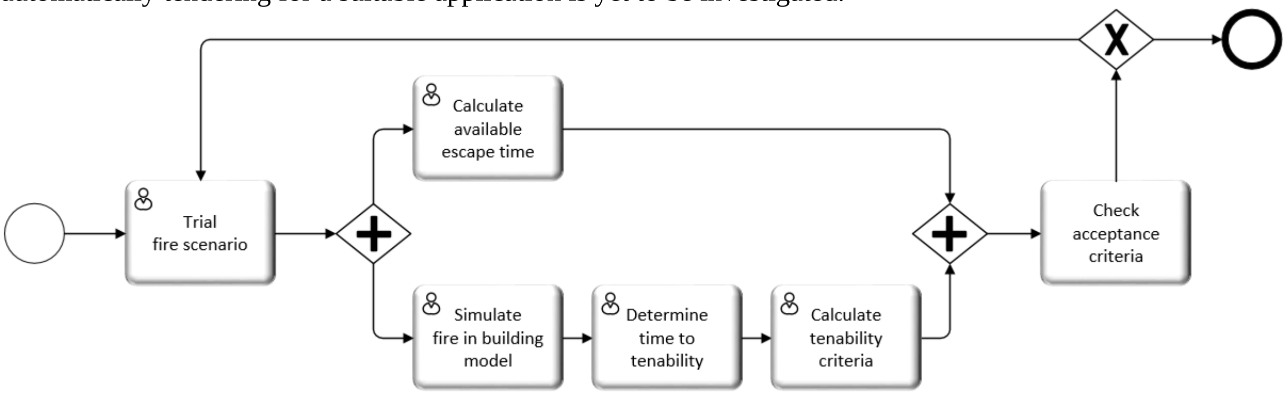

Figure 9: Example Verification Method assessment requiring external simulations

7. CONCLUSION AND FUTHER WORK

As illustrated in this paper, regulatory knowledge for compliance checking purposes can be represented in two parts, i.e. as the industry de-facto standard design processes as well as regulatory data and rules. The key is to make it possible for the domain experts, who may not have the necessary computer programming skills, to manage the representations themselves. This can be achieved by using a visual notation such as BPMN 2.0 to describe the design procedures, and creating a user-friendly editor to help generate the regulatory data and rules that can be represented using an efficient and practical inferencing language. The latter is still being investigated.

This paper compares three approaches to represent regulatory data, i.e. a general-purpose DMBS functions, rulesets in Solibri Model Checker (SMC), and an open-source rule language DRL that could be used with a graphical editor. Firstly, SMC offers a predefined set of rules with limited customisable parameters that can be managed as rulesets in the ruleset manager. The parameters editing for each rule is restricted to a pre-set proforma and the resulting rule codes are hidden from the users, which makes them difficult to verify. More importantly, individual SMC rules cannot be created by the user making it rather limited in its application. On the other hand, while manual coding of rules in DBMS or DRL require a specific knowledge in the language, they could be generated using a dedicated user interface. The possibility of using a graphical tool such as BPMN to generate the rules and the direct support for domain specific language mapping certainly give DRL a distinct advantage, which will be subject to further investigation.

ACKNOWLEDGMENT

The first author gratefully acknowledges assistance from the Building Research Levy for his research.

REFERENCES

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

Becker, J., and Delfmann, P. (2012). Generalizability and Applicability of Model-Based Business Process Compliance-Checking Approaches–A State-of-the-Art Analysis and Research Roadmap. *BuR Business Research*, *5*, 221–247. Retrieved from http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2185374

Bouzidi, K. R., Fies, B., Faron-Zucker, C., Zarli, A., and Thanh, N. L. (2012). Semantic Web Approach to Ease Regulation Compliance Checking in Construction Industry. *Future Internet*, *4*, 830–851. doi:10.3390/fi4030830

Building, D. of, and Housing. (2007). *Building for the 21st Century - Report on the Review of the Building Code* (pp. 1–86). Department of Building and Housing.

Building, D. of, and Housing. (2011). *New Zealand Building Code Handbook* (3rd Ed., p. 202). Wellington, New Zealand: Ministry of Business, Innovation & Employment.

De Ley, E., and Jacobs, D. (2011). Rules-based analysis with JBoss Drools: adding intelligence to automation. In *Proceedings of ICALEPCS 2011* (pp. 790–793). Grenoble, France: Software Technology Evolution.

Di Bona, D., Lo Re, G., Aiello, G., Tamburo, A., and Alessi, M. (2011). A Methodology for Graphical Modeling of Business Rules. *UKSim 5th European Symposium on Computer Modeling and Simulation*, 102–106. doi:10.1109/EMS.2011.68

Eastman, C., Lee, J., Jeong, Y., and Lee, J. (2009). Automatic rule-based checking of building designs. *Automation in Construction*, *18*, 1011–1033. doi:10.1016/j.autcon.2009.07.002

Fenves, S. J., and Garrett, J. H. (1986). Knowledge based standards processing. *Artificial Intelligence in Engineering*, *1*, 3–14. doi:10.1016/0954-1810(86)90029-4

Fenves, S. J., Gaylord, E. H., and Goel, S. K. (1969). *Decision Table Formulation of the 1969 AISC Specification* (pp. 1–167). University of Illinois Engineering Experiment Station. College of Engineering. University of Illinois at Urbana-Champaign, USA.

Gallaher, M., O'Connor, A., Dettbarn, J., and Gilday, L. (2004). *Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry* (pp. 1–210). Report. Gaithersburg, Maryland: US Department of Commerce.

Governatori, G., and Shek, S. (2013). Regorous: A Business Process Compliance Checker. In *International Conference on Artificial Intelligence and Law*. Rome, Italy.

Lukichev, S., and Jarrar, M. (2009). Graphical Notations for Rule Modeling. In *Handbook of Research on Emerging Rule-based Languages and Technologies* (pp. 76–98). Idea Group Inc. Retrieved from http://www.jarrar.info/publications/HRER2008_Ver12.pdf

Nawari, N. (2011). A framework for automating codes conformance in structural domain. *Journal of Computer and Information Technology*.

Pauwels, P., Van Deursen, D., Verstraeten, R., De Roo, J., De Meyer, R., Walle, R. Van de, and Van Campenhout, J. (2011). A semantic rule checking environment for building performance checking. *Automation in Construction*, *20*, 506–518. doi:10.1016/j.autcon.2010.11.017

Singh, V., Gu, N., and Wang, X. (2011). A theoretical framework of a BIM-based multi-disciplinary collaboration platform. *Automation in Construction*, *20*, 134–144. doi:10.1016/j.autcon.2010.09.011

Vitali, F., and Zeni, F. (2007). Towards a country-independent data format: the Akoma Ntoso experience. In *Proceedings of the V legislative XML workshop* (pp. 67–86). Florence, Italy.

Zhang, J., and El-Gohary, N. (2012a). Extraction of Construction Regulatory Requirements from Textual Documents Using Natural Language Processing Techniques. *Journal of Computing in Civil Engineering*, 453–460.

Zhang, J., and El-Gohary, N. (2012b). Automated regulatory information extraction from building codes leveraging syntactic and semantic information. In *ASCE Construction Research Congress* (pp. 622–632). West Lafayette, Indiana, USA.