Computer-aided Compliance Audit to Support Performance-based Fire Engineering Design

Johannes Dimyadi[1], Charles Clifton[2], Michael Spearpoint[3], Robert Amor[4]

[1]Dept of Computer Science, University of Auckland, New Zealand
email: jdim006@aucklanduni.ac.nz

[2]Dept of Civil Engineering, University of Auckland, New Zealand
email: c.clifton@auckland.ac.nz

[3]Dept of Civil & Natural Resources Engineering, University of Canterbury, New Zealand
email: michael.spearpoint@canterbury.ac.nz

[4]Dept of Computer Science, University of Auckland, New Zealand
email: trebor@cs.auckland.ac.nz

ABSTRACT

Computer-aided compliance auditing aims to provide an automated system to assess engineering designs against specified regulatory representations. Previous research has largely focused on prescriptive regulatory rules, which are relatively easier to audit than those pertaining to performance-based codes with qualitative criteria. There have been a few prototype implementations of rule-based compliance auditing systems that tend to represent regulatory knowledge as complex rule sets integrated into the system. The drawback of this approach is inflexibility, relatively high costs and dependency on the system programmer to modify built-in rules in response to on-going regulatory amendments.

The current research looks at representing regulatory knowledge as a library of compliant design procedures (CDP) and the associated regulatory rules, which are treated as external input components to the system. This would allow them to be managed and maintained independently by designers and regulators as appropriate experts in their respective fields.

This paper reports on the development of a computable regulatory knowledge model (RKM) that can be used in conjunction with a CDP to audit an object-based building information model (BIM) automatically. CDPs can be described graphically as workflows in the open standard Business Process Model and Notation (BPMN), which can be executed to automate the compliance audit process. A RKM representing the fire engineering performance-based verification method prescribed by the New Zealand Building Code is proposed for use by a set of CDP workflows to automatically audit the design for compliance. The potential of interfacing with simulation tools to provide some of the required input parameters is discussed.

INTRODUCTION

The regulatory compliant design of a building, regardless of whether it is based on prescriptive or performance-based codes, is usually preceded by data collection, which is often referred to as the "building code analysis", particularly in the North America. This is basically a systematic process of manually gathering detailed information about the project and the building from the architectural design and entering them onto a spreadsheet or a pre-defined form.

Once necessary data is collected, a particular design can then proceed in accordance with a set of industry standard compliant design procedures. For example, there is a standard compliant procedure for the design of means of escape from fire, and another for the design of a smoke control system, or for assessing the impact of a fire scenario on a particular occupied space, and so on. For performance-based designs, some of the steps in the procedure may involve running computer simulations, carrying out hand calculations, or even conducting laboratory tests.

While these compliant procedures may be systematic and robust, a manual design practice is labour intensive, costly and error-prone. Furthermore, paper-based design information sharing is inefficient and represents a duplication of effort, which contributes to general productivity degradation in the industry as a whole. Since the advent of personal computer, there have been many computer-aided design tools available for various aspects of building engineering design (Amor et al., 1993; Dolšak and Novak, 2011). Some of these tools have incorporated basic compliance audit functionality. However, most of them use proprietary data models and a black-box approach to design and compliance audit.

The current research investigates a unified approach to compliance audit by automating the manual compliant design procedures (CDP) and representing building design information and regulatory knowledge in open standard models that can be managed independently by engineers and regulators for use as input components to the compliance audit system.

COMPUTER-AIDED COMPLIANCE AUDIT SYSTEM

The computer-aided compliance audit system proposed in this research can be described using the following system architecture (Figure 1).

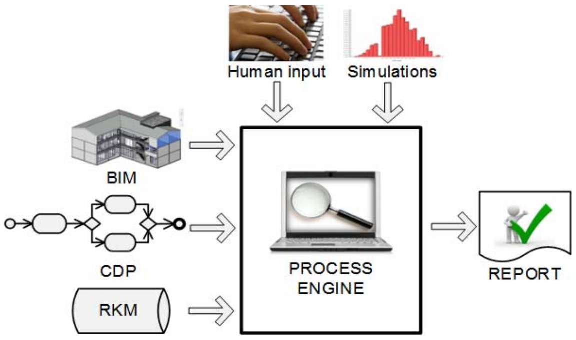

Figure 1: System Architecture

The proposed system consists of the core process engine that can process a given set of CDP in conjunction with a given design represented in the building information model (BIM) model view for auditing against a specified regulatory knowledge model (RKM). Supplementary information not available in the BIM model view may be provided by human input. If necessary, external simulations or calculations tools may be used to provide further information as may be required by the RKM. The BIM model view, CDP and the associated RKM, which represent the main input components to the compliance audit system, will be described in more details in subsequent sections of this paper.

For a typical project, a fire design engineer would first obtain a BIM model view by using a schema that specifies the type of information to be extracted from the BIM model for the purposes of compliant design process and compliance auditing against the fire code. This can be achieved with the aid of a software client that can query the BIM model using the schema in a model server environment such as via the BIMserver (Beetz and van Berlo, 2010). The engineer would then choose an appropriate pre-published RKM model, such as that representing the regulatory framework for fire engineering design known as C/VM2, to audit the design against. Once the BIM model view is generated and an RKM model is selected, the engineer would then use one of the predefined CDPs, such as those for the fire engineering design in accordance with C/VM2, or create their own to start the automated audit process. The process typically navigates through each task of the CDP and looks up building information in the BIM model view and evaluates any constraints or rules in the RKM model. If necessary, the CDP would stop at a task and request for either a human input or additional information that can be obtained from a simulation. This general automation process is as shown Figure 2.

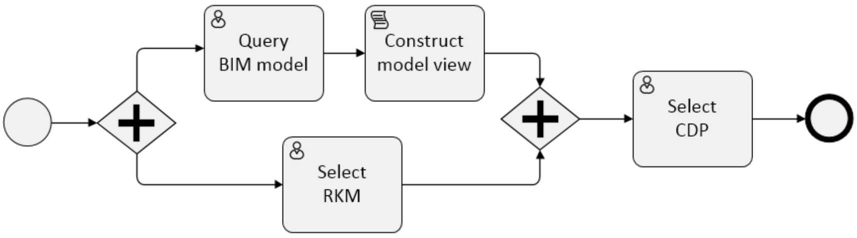

Figure 2: Computer-aided compliance audit automation workflow

BUILDING INFORMATION MODELLING (BIM)

The traditional method of documenting a building design using 2D and 3D CAD systems have shifted over the years into a new paradigm known as the building information modelling (BIM), which is an object-based and collaborative approach to design, construct and manage a building. It is also a digital representation of all the functional and physical information captured in the entire building life cycle. In a highly complex and fragmented domain such as Architectural, Engineering and Construction (AEC), the emergence of BIM technology is a major milestone towards a general productivity improvement. In particular, this has the potential of contributing towards more efficient information sharing and the ability to automate some of the compliance audit tasks.

Computer applications within the AEC domain can now take advantage of the ability to share BIM data using an open standard (ISO 16739) industry specific data model known as the Industry Foundation Classes (IFC). The latest version, IFC4, has the ability to represent over 760 specified entities (e.g. building storey, wall, door, window, etc.) and allows for extensible property sets (Liebich, 2010) to describe additional entities. A number of open standard software development toolkits are now available to extract building information from the IFC data model for different applications including compliance audits.

Instead of manually taking information off paper-based drawings and specification documents, it is now possible to conduct the predesign data collection directly on the BIM models using a computer software tool. There have been successful implementations of the data collection procedure reported by a number of researchers in this area (Clayton, 2013; Hassan M. Satti, 2007). The subset of the BIM model or the model view is effectively equivalent to the spreadsheet or the paper form that has been used traditionally for data collection, except that it now allows interoperability. A proposed BIM model view schema has been developed in this research for fire engineering compliant design using the Extensible Markup Language (XML) Schema Definition (XSD) (Figure 3). The schema can be managed graphically using a number of available commercial or public-domain software tools.

Information that can usually be extracted from the BIM model and collected in the model view includes physical building objects such as building storeys, spaces, walls, stairs, slabs, doors, windows, which are specified in the schema as either mandatory or optional elements, shown in solid or dotted boxes, respectively. Physical properties such as floor and wall areas, building object dimensions, which can usually be extracted from the BIM model, are specified in the schema as attributes. Metadata and design specific information that may not be available from the BIM model such as project information, site location, building or space usage and activity classifications, occupant type classifications, other specific design parameters can be entered manually into the model view as attributes.

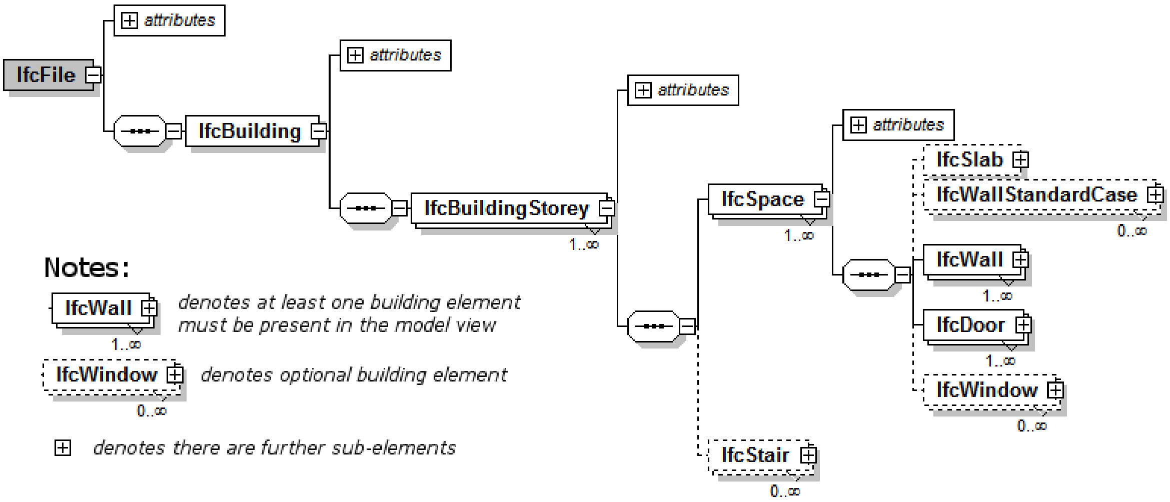

Figure 3: Proposed BIM model view schema for fire engineering compliant design

A model server such as the BIMserver can be used as a central repository of BIM models for extracting information from the BIM model and to generate a model view document in accordance with the proposed schema. Together with any supplementary data, this represents a complete set of building information necessary to conduct a particular compliance audit.

NEW ZEALAND’S PERFORMANCE-BASED CODE

The New Zealand Building Code (NZBC) is a performance-based code containing 37 technical clauses covering different aspects of the building construction and occupancy standards including stability and durability of structure, fire safety and protection, accessibility, moisture control, safety of users, services and facilities.

There are basically two means of compliance with NZBC, i.e. deemed-to-satisfy methods published by the national regulator, the Ministry of Business, Innovation, and Employment (MBIE), and Alternative Solutions. The deemed-to-satisfy method is further divided into the Acceptable Solution and the Verification Method. The Acceptable Solution is a relatively straightforward rule-based approach to demonstrate full compliance of basic or simple buildings with a set of prescriptive documents. For fire safety design, each of the Acceptable Solution documents C/AS1 to C/AS7 is applicable to one of seven risk groups.

Verification Method documents provide frameworks for undertaking engineering calculation methods or tests and sufficient detail to get consistency in the application of the method or test being undertaken, such as defining design fires, calculation methods for life safety and structural fire resistance. In the current edition of NZBC, the Verification Method for fire safety compliant design is C/VM2, Framework for Fire Safety Design (Ministry of Business Innovation and Employment, 2013). Although prescriptive in principle, the C/VM2 document incorporates industry standard design methods that allow the use of external computation and/or simulation tools to help evaluate fire design scenarios and assess the tenability condition of escape routes and structural stability, as appropriate.

Methods other than the Acceptable Solutions or Verification Methods are regarded as the Alternative Solutions. This may involve specific compliant design solutions based on calculations, tests, comparative analyses, appraisals, expert evidence, or any combination of these. A considerable amount of evidence is usually required to demonstrate compliance with the performance criteria of relevant clauses of the NZBC when using an Alternative Solution. For the fire safety design, the current expectation of the New Zealand regulator is that an Alternative Solution will not be used very often.

REGULATORY KNOWLEDGE REPRESENTATION

Regulatory documents are written in human readable natural language that is not primarily intended for computer interpretation. However, the structure of regulatory documents and the inherent knowledge can be represented in a form that allows computer processing. There are several common approaches suggested by researchers over the years to represent regulatory knowledge for computer-aided compliance audit, as follows:

1. Manually encoding regulatory rules and constraints from natural language into a set of logical statements used in a computable knowledge base (Fenves and Garrett, 1986).

2. Manually translating regulatory rules and constraints from natural language into procedural computer codes that are integrated with the compliance audit system (Mugridge et al., 1996; Nguyen and Asa, 2006). Different document mark-up techniques have been suggested to aid the translation process (Hjelseth and Nisbet, 2011)

3. Manually translating regulatory texts from the natural language to a formal language such as the Gellish English (van Renssen et al., 2007), or using artificial intelligence (AI) techniques such as natural language processing (NLP) to automatically extract regulatory knowledge from the document corpus as a formal language (Zhang and El-Gohary, 2012).

4. Manually translating regulatory rules and constraints from natural language into semantic rule and query languages that can be processed by rule or inference engines in conjunction with a domain ontology (Bouzidi et al., 2012).

5. Using data modelling techniques to capture manually encoded regulatory rules and constraints into an open standard data structure that can be maintained independently of the compliance audit system (Palmirani et al., 2011).

There have been promising results reported with some of the above approaches in recent years. However, keeping regulatory knowledge representations from becoming too complex and costly to maintain remains an active area of research. This is particularly relevant when representing highly complex regulatory documents, or where representations are integrated with the compliance audit system and reliant on the system programmer to make changes in response to on-going regulatory amendments, which can occur as frequently as once or twice yearly.

Past research has focused mainly on prescriptive code requirements that may be more readily encoded into computable rules, but as performance-based codes are gaining acceptance and popularity worldwide, researchers have started exploring computable representations of quantitative and qualitative performance criteria (Han et al., 2002; Suter and Mahdavi, 2004).

A performance-based building code sets out the qualitative or quantitative performance criteria that a completed building and its components must comply with, appropriate to the designed occupancy, throughout its intended life. To facilitate compliance for simple buildings, an accompanying set of prescriptive documents are usually provided. Satisfying prescribed requirements given by the documents is deemed to comply with the performance criteria. Prescriptive compliance documents typically contain specifications on how building objects such as spaces, doors, separation walls, stairs, etc. must be designed and constructed in order to comply. However, there are normally multiple compliant design solutions available, depending on which design parameters and options are selected.

COMPLIANT DESIGN PROCEDURES (CDP)

Compliance audit occurs at all stages in the life-cycle of a building, although it is largely part of the design activities right through to the construction period with the ultimate aim of getting the completed building approved by the Authority Having Jurisdiction (AHJ) and to the satisfaction of the client. Consequently, design procedures are driven by the need to comply with the regulatory as well as non-regulatory requirements specific to the project. A compliant design procedure (CDP) is effectively a compliance audit procedure for a particular design. For example, designing for a door in an occupied space involves auditing it against all regulatory requirements that relate to its minimum capacity, opening direction, and other features corresponding to the activity and occupant load in the space, fire separation and accessibility constraints.

A CDP is effectively a design workflow with tasks representing the design steps. For example, a simple design task to determine an occupant load in an occupied space can be described graphically in an executable workflow diagram using the open standard Business Process Model and Notation (BPMN) (Object Management Group, 2011) (Figure 4).

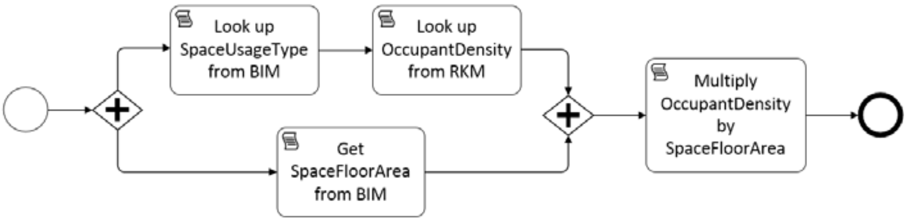

Figure 4: An exemplar CDP to determine an occupant load in an occupied space

In Figure 4, the main activities are determining the usage type of a particular space from the BIM model view as well as obtaining the floor area of the space. These two activities can take place in parallel or independently from each other. However, the task of getting the occupant density value is dependent on the usage type of the space, so it cannot take place until the usage type has been determined. Once the occupant density and the floor area have been obtained, the occupant load of the space can then be calculated by multiplying these two values together. Each task would execute a computer script to do the specified activity before moving on to the next one. Encoding guidelines using a subset of BPMN components to describe a specific set of CDPs has been discussed in a related paper (Dimyadi et al., 2014).

The ability to describe CDPs graphically in the standard BPMN workflow allows a design engineer to easily create new CDPs or modify existing CDPs to suit any regulatory amendment or custom practice procedures. BPMN workflow can be exchanged as an XML document, which is computer readable for transfer to the compliance audit process.

It is envisaged that a library of industry's acceptable standard CDPs for different design disciplines will be published by the industry's professional body such as the institution of professional engineers. For compliant fire engineering design in New Zealand, it is envisaged that a set of official CDPs representing the design procedures for using the C/VM2 document would be published by the MBIE in conjunction with the Building Research Association of New Zealand (BRANZ) and the Society of Fire Protection Engineers (SFPE) New Zealand Chapter, which is a technical group of the Institution of Professional Engineers of New Zealand (IPENZ). In the preliminary design stages, building designers or fire engineers have the option of creating their own CDPs for checking specific aspects of their design to suit their own practices.

REGULATORY KNOWLEDGE MODEL (RKM)

In the context of the current research, a Regulatory Knowledge Model (RKM) can be defined as a computable representation of a set of regulatory corpus. For example, the structure and content of the C/AS1 to C/AS7, and C/VM2 documents can be represented in a data model described using the XML schema in XSD, similar to that used for the BIM model view.

Instead of representing an entire document in one large data model, it is a good practice to modularise various common sub-models and link or include them by reference to the main model. For example, the main RKM for C/VM2 document has three sub-model schemas linked to it, i.e. RefDocs, Dictionary and Occupancy, which are common to several other documents.

Figure 5 shows a graphical view of XSD data model representing the dictionary section defining specific terms used in the documents, which is common to the New Zealand's Building Code C/AS1 to C/AS7, and C/VM2 documents. Solid rectangular boxes in Figure 5 represent mandatory elements and dotted boxes represent optional elements.

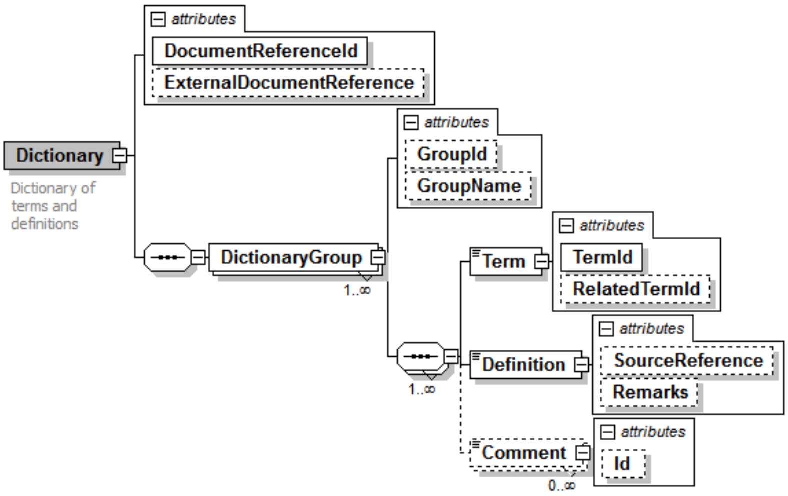

Figure 5: A high level view of the RKM schema for the common dictionary

Figure 6 is a branch of the main RKM for C/VM2 showing the representation of the RSETEquation, which is under the OccupancyMovement. Each solid box corresponds to a paragraph in the C/VM2 document.

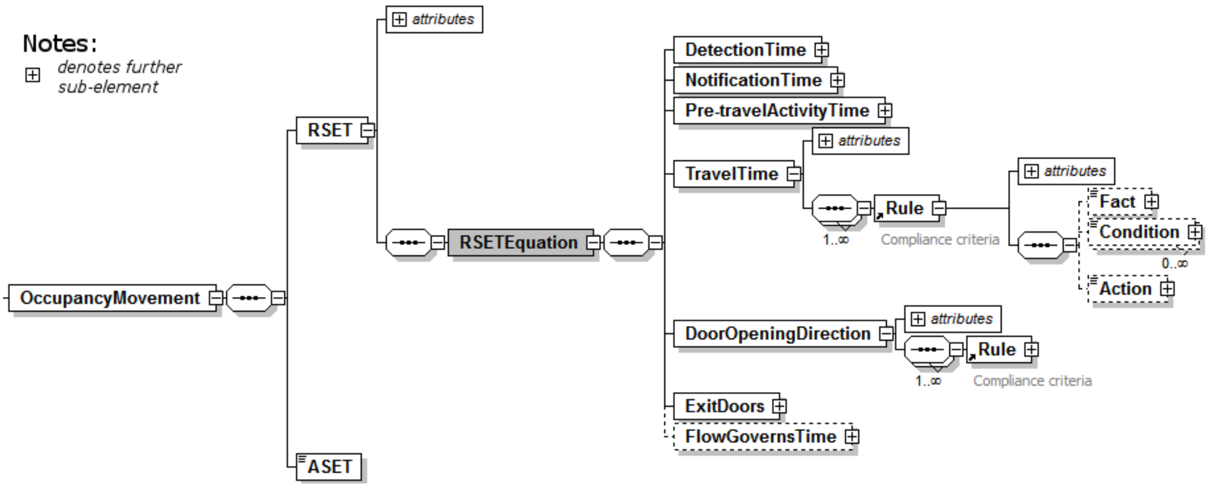

Figure 6: Part of RKM schema for C/VM2 showing rules representation

EXAMPLE RKM

Generally, there are three types of information that can be represented in the RKM, namely look-up data, mathematical equations, and rules. These represent the information referenced and extracted by script tasks of a particular CDP.

Look-up data are typically tabulated design parameters such as the occupant density that is used to determine the occupant load in a space, which depends on the intended activity of the space (Table 1).

Table 1: Occupant densities (excerpt from Table 3.1 of C/VM2)

Space activity	Occupant density (m^2/person)
Art galleries, museums	4
Classrooms	2
Offices	10
Shop spaces and pedestrian circulation areas including malls and arcades	3
Shop spaces for furniture, floor coverings, large appliances, building supplies and Manchester	10

Mathematical equations are those provided as such in the document or formalised equations derived from regulatory texts. For example, the RSET (Required Safe Egress Time) Equation in Figure 6 is given in Equation 3.1 in Paragraph 3.2 of C/VM2 as shown in Figure 7.

Original RSET Equation

$$RSET = \left(t_d + t_n + t_{pre}\right) + \left(t_{trav}\ or\ t_{flow}\right)$$

where:

t_d = fire detection time
t_n = time from detection to notification of the occupants
t_{pre} = time from notification until evacuation begins
t_{trav} = time spent moving towards a place of safety
t_{flow} = time spent in congestion controlled by flow characteristics

Figure 7: Original RSET Equation

This can be formalised as two different equations, i.e. RSET = (TD+TN+TPRE+TTRAV) and RSET = (TD+TN+TPRE+TFLOW), where TFLOW is only applicable for movement of people through a doorway or stairway in which congestion is likely to occur.

The formalised RSET Equation can be represented in the RKM as shown in Figure 8.

```
<RSETEquation>
    <Rule Id="ID_3.2.R1">
            <Fact Id="ID_3.2.R1.F1"><![CDATA[ RSET = (TD+TN+TPRE+TTRAV) ]]></Fact>
            <Condition Id="ID_3.2.R1.F1.C1"><![CDATA[ (ESCPATH = DOOR) OR (ESCPATH = STAIR) ]]></Condition>
            <Action Id="ID_3.2.R1.F1.C1.A1"><![CDATA[ RSET = (TD+TN+TPRE+TFLOW) ]]></Action>
    </Rule>
</RSETEquation>
```

Figure 8: Representation of RSET in RKM

The components of the equation, i.e. TD, TN, TPRE, TTRAV, TFLOW, are represented in different parts of the RKM and are to be evaluated separately. Once evaluated, RSET can simply be determined by a summation of all these components.

Another example is the method of estimating the walking speed of a person for the purposes of calculating TTRAV, which is given as Equation 3.2 in C/VM2 (left pane of Figure 9).

Original Equation	**Formalised Equation**
$S = k - akD$ where: S = horizontal walking speed (m/s) D= space occupant density (persons/m^2) k = 1.4 for horizontal travel a = 0.266	$S = 1.4 - (0.3724/D)$ If $S > 1.2$ then $S = 1.2$ Rule ID_3.2.4.R1 where: S = horizontal walking speed (m/s) D= space occupant density (m^2/person), from Table 1

Figure 9: Original and Formalised Equation 3.2 of C/VM2

The regulatory text in the document further specifies that the maximum speed used to calculate the travel time for evacuation shall be 1.2 m/s. Also, the occupant density used in the original equation is the inverse of that given in Table 1. Therefore, the equation can be formalised by substituting the variables with the constants for horizontal travel and inverting the occupant density variable, which is shown on the right pane of Figure 9.

TTRAV is the evacuation travel time, which can be determined from TTRAV = (LTRAV/S). The default formalised equation to determine LTRAV is (SpaceLength + SpaceWidth). However, the regulatory provision allows LTRAV to be determined from the actual measured length of travel paths around objects such as furniture in the space. So, LTRAV can also be a manual input. The provision of Paragraph 3.2.4 of C/VM2 on Travel Time is represented as rules in the RKM for C/VM2 as shown in Figure 10.

```
<TravelTime DocumentReferenceId=”3.2.4”>
```

Formalised rule for S (Table 7)

```
    <Rule Id=”ID_3.2.4.R1”>
        <Fact Id=”ID_3.2.4.R1.F1”><![CDATA[ S = 1.4 – (0.3724/D) ]]></Fact>
        <Condition Id=”ID_3.2.4.R1.F1.C1”><![CDATA[ S >= 1.2 ]]></Condition>
        <Action Id=”ID_3.2.4.R1.F1.C1.A1”><![CDATA[ S = 1.2 ]]></Action>
    </Rule>
```

Formalised rules for LTRAV & TTRAV

```
    <Rule Id=”ID_3.2.4.R2”>
        <Fact Id=”ID_3.2.4.R2.F1”><![CDATA[ LTRAV = (SpaceLength + SpaceWidth) ]]></Fact>
        <Condition Id=”ID_3.2.4.R2.F1.C1”><![CDATA[ COUNT(SpaceFurnitures)>0 ]]></Condition>
        <Action Id=”ID_3.2.4.R2.F1.C1.A1”><![CDATA[ LTRAV = INPUT(MeasuredLength) ]]></Action>
    </Rule>
    <Rule Id=”ID_3.2.4.R3”>
        <Fact Id=”ID_3.2.4.R3.F1”><![CDATA[ TTRAV = (LTRAV/S) ]]></Fact>
    </Rule>
</TravelTime>
```

Figure 10: Representation of Formalised Equation 3.2 of C/VM2 in XML

A rule is generally a formalised mathematical equation derived from regulatory texts. Each rule in the RKM consists of Fact, Condition, and Action. Mathematical equations without conditions are represented simply as a Fact within a rule. The Fact is the default outcome when the Condition is evaluated to FALSE. Otherwise, the Action is the outcome of the evaluation. A rule can return an "OK" or a calculated value to be used for further processing. For example, Paragraph 3.2.6 of the C/VM2 document specifies that "*Doors on escape routes shall be hung open in the direction of escape…*", but *"…need not apply where the number of occupants…using the door is no greater than 50"*. This can be translated into the following rules as represented in the RKM (Figure 11).

```xml
<DoorOpeningDirection DocumentReferenceId=”3.2.6”>
    <Rule Id="ID_3.2.6.R1">
            <Fact Id="ID_3.2.6.R1.F1"><![CDATA[ (DoorOpeningDirection=IN) ]]></Fact>
            <Condition Id="ID_3.2.6.R1.F1.C1"><![CDATA[ (OccupantLoad<=50) ]]></Condition>
            <Action Id="ID_3.2.6.R1.F1.C1.A1"><![CDATA[ Return('OK') ]]></Action>
    </Rule>
    <Rule Id="ID_3.2.6.R2">
            <Fact Id="ID_3.2.6.R2.F1"><![CDATA[ (DoorOpeningDirection=OUT) ]]></Fact>
            <Action Id="ID_3.2.6.R2.F1.A1"><![CDATA[ Return('OK') ]]></Action>
    </Rule>
    <Rule Id="ID_3.2.6.R3">
            <Fact Id="ID_3.2.6.R3.F1"><![CDATA[ (COUNT(SlidingDoors)>0) ]]></Fact>
            <Condition Id="ID_3.2.6.R3.F1.C1"><![CDATA[ (OccupantLoad<=20) ]]></Condition>
            <Action Id="ID_3.2.6.R3.F1.C1.A1"><![CDATA[ Return('OK') ]]></Action>
    </Rule>
</DoorOpeningDirection>
```

Figure 11: Rule representation of DoorOpeningDirection in XML

The numbering convention used in RKM strictly follows the numbering system used in the regulatory document, but prefixed or suffixed with additional sub-numbering appropriate to each representation. For rules, each Action clause has an ID that refers back to its Condition and in turn refers back to the Fact and Rule within which it applies. For example, TravelTime is Paragraph 3.2.4 and a sub-element of Part 3 Movement of People in the C/VM2 document. So, TravelTime is given the DocumentReferenceID attribute of "3.2.4" and all subsequent rules associated with this topic are prefixed with "3.2.4" (Figure 10). Similarly, DoorOpeningDirection has the DocumentReferenceID attribute of "3.2.6" and all associated rules are given the prefix of "3.2.6" (Figure 11). This numbering system is also referred to by each script task in the CDP, so that the correct information can be extracted and appropriate rules evaluated. Furthermore, having a matching set of referencing would greatly facilitate managing revisions of the RKM in response to regulatory amendments.

It is envisaged that there will be a companion RKM published together with each technical clause of the paper-based building code.

INTERFACING WITH SIMULATION MODELS

Mathematical equations and rules can be processed relatively straightforward by the compliance audit process engine. However, some of the performance-based criteria requires external tools to help evaluate. To assist with this kind of evaluation, it is possible to automate the generation of input files that can be used by a number of simulation models developed in New Zealand and used by fire engineers, such as B-RISK (Wade et al., 2013) and EvacuatioNZ (Spearpoint and Xiang, 2011). This can be achieved by collecting a set of data from the BIM model view together with some supplementary human input and generating the input files to conform to the input schema of the simulation tool. Then, by using the output schemas of these tools, it will be possible to extract the results of the simulation to support regulatory compliance decisions.

CONCLUSION

This paper has outlined the main ingredients necessary for automating the performance-based fire engineering design process. As discussed, the initial design data collection or the building code analysis process can be automated by querying the BIM model using an input schema in a model server environment and generate a model view containing all the information required for a particular compliance audit.

Data models representing BIM and RKM are developed in this research using the XSD graphical representation of the XML Schema. The compliance audit process can be automated by using the graphical open standard BPMN to describe CDPs, which ensures that each step of the design process complies with the relevant regulatory constraints and rules. The BPMN is computer readable and suitable to guide the automated compliance audit process using a dedicated compliance audit process engine, which is being developed as part of this research. A typical performance-based fire engineering design compliance audit workflow would start with a fire engineer selecting a particular RKM to audit against, and then choosing a CDP from the associated library of CDPs to execute. During an execution, the process may pause for supplementary human input or request additional information that may only be obtained using external simulation tools. In this case, an input file may be created for running selected external simulations and the results used for further processing.

ACKNOWLEDGEMENT

This research receives funding assistance from the New Zealand Building Research Levy.

REFERENCES

Amor, R., Hosking, J., and Donn, M. (1993). Integrating Design Tools for Total Building Evaluation. *Building and Environment*, *28*(4), pp. 475–482. Elsevier.

Beetz, J., and van Berlo, L. (2010). bimserver.org - An Open Source IFC Model Server. In *Proceedings of CIB W78 International Conference,* pp. 1–8. Cairo, Egypt.

Bouzidi, K. R., Fies, B., Faron-Zucker, C., Zarli, A., and Thanh, N. Le. (2012). Semantic Web Approach to Ease Regulation Compliance Checking in Construction Industry. *Future Internet*, *4*(3), pp. 830–851. Multidisciplinary Digital Publishing Institute (MDPI). doi:10.3390/fi4030830

Clayton, M. (2013). Automated Plan Review for Building Code Compliance Using BIM. In *Proceedings of 20th International Workshop: Intelligent Computing in Engineering (EG-ICE 2013),* pp. 1–10. Vienna, Austria. Retrieved from http://smartreview.biz/smartreview-tr001.pdf

Dimyadi, J., Clifton, C., Spearpoint, M., and Amor, R. (2014). Regulatory Knowledge Encoding Guidelines for Automated Compliance Audit of Building Engineering Design. In *Proceedings of the ICCCBE/CIB W78*. Orlando, Florida, USA.

Dolšak, B., and Novak, M. (2011). Intelligent decision support for structural design analysis. *Advanced Engineering Informatics*, *25*(2), pp. 330–340. doi:10.1016/j.aei.2010.11.001

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

Han, C., Law, K. H., Latombe, J.-C., and Kunz, J. C. (2002). A performance-based approach to wheelchair accessible route analysis. *Advanced Engineering Informatics, 16*(1), pp. 53–71. Elsevier. doi:10.1016/S1474-0346(01)00003-9

Hassan M. Satti. (2007). Automation of building code analysis: Characteristics, Relationships, Generation of Properties and Structure of Various Models. In *Proceedings of the 3rd International Conference of the Arab Society for Computer Aided Architectural Design (ASCAAD 2007)*, pp. 607–622. Alexandria, Egypt.

Hjelseth, E., and Nisbet, N. (2011). Capturing normative constraints by use of the semantic mark-up (RASE) methodology. In *Proceedings of CIB W78-W102 Conference*, pp. 1–10. Sophia Antipolis, France.

Liebich, T. (2010). Unveiling IFC2x4 - The next generation of OPENBIM. In *Proceedings of CIB W78 Conference*, pp. 124–131. Cairo, Egypt.

Ministry of Business Innovation and Employment. (2013). *C/VM2 Verification Method: Framework for Fire Safety Design For New Zealand Building Code Clauses C1-C6 Protection from Fire*, Errata2 ed., pp. 1–65. The Ministry of Business, Innovation and Employment. ISBN 9780478381641.

Mugridge, R., Hosking, J., and Amor, R. (1996). *Adding a Code Conformance Tool to an Integrated Building Design Environment*. Auckland: Auckland Uniservices Limited.

Nguyen, T. H., and Asa, E. (2006). Integrating building code compliance checking into a 3D CAD system. In *Proceedings of the Joint International Conference on Computing and Decision Making in Civil and Building Engineering*, pp. 646–655. Montreal, Canada.

Object Management Group. (2011). *Business Process Model and Notation (BPMN)*, pp. 1–538. Object Management Group. .

Palmirani, M., Governatori, G., and Rotolo, A. (2011). LegalRuleML: XML-based rules and norms. *Rule-Based Modeling and Computing on the Semantic Web*, pp. 298–312.

Spearpoint, M., and Xiang, X. (2011). Calculating Evacuation Times from Lecture Theatre Type Rooms Using a Network Model. In *Fire Safety Science*, Vol. 10, pp. 599–612. Maryland, USA. doi:10.3801/IAFSS.FSS.10-599

Suter, G., and Mahdavi, A. (2004). Elements of a representation framework for performance-based design. *Building and Environment, 39*(8), pp. 969–988. doi:10.1016/j.buildenv.2004.01.021

Van Renssen, A., Vermaas, P. E., and Zwart, S. D. (2007). A Taxonomy of Functions in Gellish English. In J.-C. Bocquet (Ed.), *Proceedings of ICED 2007, the 16th International Conference on Engineering Design*, pp. 1–10. Paris, France.

Wade, C., Baker, G., Frank, K., Robbins, A., Harrison, R., Spearpoint, M., and Fleischmann, C. (2013). *B-Risk User Guide and Technical Manual. BRANZ Study Report 282*, Vol. 282, pp. 1–38. Judgeford, Porirua City, New Zealand: BRANZ Limited.

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