# Integrating the BIM Rule Language into Compliant Design Audit Processes

Johannes Dimyadi, j.dimyadi@auckland.ac.nz
*Department of Computer Science, University of Auckland, New Zealand*

Wawan Solihin, wsolihin@outlook.com
*Invicara Pte Ltd, Singapore*

Charles Eastman, mcharles.eastman@coa.geotech.ac.edu
*College of Architecture, Georgia Institute of Technology, United States*

Robert Amor, trebor@cs.auckland.ac.nz
*Department of Computer Science, University of Auckland, New Zealand*

## Abstract

Extracting the right information from Building Information Models (BIM) and Regulatory Knowledge Models (RKM) is a core activity in computer-aided compliance audit processes. Despite many research attempts, however, this has remained a challenging task. One reason is that BIM is necessarily a highly complex model. Representing all components of such a complex object as a building is a huge challenge. Equally complex is accessing existing regulatory knowledge from voluminous paper-based documents conveyed in natural language and not readily suitable for computer processing. Most regulatory knowledge can be represented as rules, but there are some that cannot be pre-defined and require human intervention and/or supplementary processes.

This paper describes how BIMRL can be integrated into a computer-aided compliance audit framework and used as a companion, or an alternative, to RKQL to better facilitate information gathering in the execution of CDP workflows

The study is based on two independent research projects recently looked at addressing these challenges. One project focused on the data management level and developed the BIM Rule Language (BIMRL) query system that enables efficient access to BIM and RKM data. The other project focused on high-level Compliant Design Procedures (CDP) by utilising an open standard BPMN (Business Process Model and Notation)-compliant executable workflow model in conjunction with a purpose-built domain-specific Regulatory Knowledge Query Language (RKQL).

To illustrate its potential application, a sample building model was chosen as a case study together with a set of performance-based compliance criteria from a compliance document of the New Zealand Building Code relating to fire safety in buildings.

**Keywords:** BIM, CDP, RKM, BIMRL, compliance audit, rule language

## 1 Introduction

Research into automated compliance audit processes in the Architecture, Engineering, Construction and Facilities Management (AEC/FM) domain has been conducted for more than half of a century. Rule-based checking has been the most common approach throughout this time starting with the work of Fenves using decision tables (Fenves, 1966). Various automated compliance audit approaches have been reviewed in the literature (Dimyadi & Amor, 2013). These range from using mark-up and Natural Language Processing (NLP) techniques to extract rules from regulatory texts to capturing regulatory knowledge using logic and language-based representations. A comprehensive review of several conventional and prototype rule-based checking systems has also been reported in the literature (Eastman et al., 2009).

In its simplest form, a conventional rule-based checking system would embed rules as part of the computer programming code used to implement the system. One common issue with conventional rule-based checking approaches is maintainability and extensibility. Building regulations evolve with new materials, performance requirements, and contexts. This is difficult to handle when rules have been hard-coded or incorporated into black-box or closed proprietary systems. This is particularly crucial for rules that are subject to regular amendments such as those in documents governing relatively new fields in a domain. In such cases, modification to the rules requires significant development effort and cost. Another challenge is that not all regulatory knowledge can be represented easily with rules; there will always be the need for supplementary human input in automated audit processes. Furthermore, automated audit for compliance with modern regulatory frameworks needs to support performance-based design where the criteria often depend on dynamic environmental factors that cannot be fully anticipated. Another aspect of the rule-based approach that has not had much attention in research is the handling of geometry and spatial query of BIM models. Work in this area includes BERA and BIMserver (Borrmann et al, 2009; Lee, 2011; Mazairac & Beetz, 2013), which are promising but they have not adequately addressed the gap between rule requirements and the availability of data in the building model.

This paper describes the contribution from two independent research projects in addressing some of the above described challenges. Specifically, this paper focuses on the integration of an efficient BIM Rule Language (BIMRL) into a compliance audit framework (ARCABIM) that treats building information, regulatory knowledge, and compliant design workflow representations as independent input components. This approach enables the input components to be maintained independently and more effectively. The framework also allows supplementary human input as well as having the ability to generate input data for external computation or simulation tools to assist performance-based compliant design (Dimyadi et al., 2016).

## 2 Requirements for Automated Compliance Audit

The main input components of an automated compliant design audit process in the AEC/FM domain are a building model or a subset of the model, a regulatory knowledge model (RKM) containing the rules, supplementary human input, and a mechanism for data exchange with supporting processes such as computational or simulation tools. The audit workflow involves activities to process these input components and report an outcome. One of the current research projects has developed a method of defining the scope of an audit process using executable compliant design procedures (CDP), which can be described in BPMN (Business Process Model and Notation) (Object Management Group, 2011). Each script-task of the BPMN 2.0-compliant CDP workflows (Figure 1) can embed computer-readable instructions to process the input components. In other words, executable CDP workflows can be used to guide the compliance audit process, particularly with respect to getting the right information from appropriate sources and using the collected information for computations. However, as the instructions must be able to process information and concepts specific to the domain, they need to be written in a domain-specific query language.

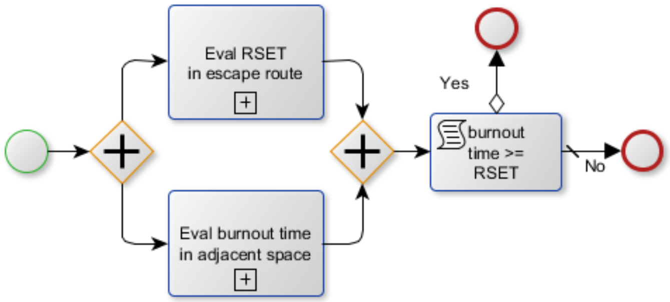


**Figure 1 An exemplar BPMN-compliant CDP workflow diagram**

Checking the IFC model in its entirety is unrealistic as it would involve using the complete set of RKMs representing all applicable regulatory documents, specifications and associated standards, which would also mean an excessively large and complex set of rules to process. For practical reasons, each rule or class of rules would typically apply to a subset of the building model data, which is also in accordance with the common building design practice where each aspect of the building is typically designed by a different discipline. The international standard concept of model subset is known as Model View Definition (MVD) (Hietanen & Lehtinen, 2014). The ISO standard for defining MVD is known as the Information Delivery Manual (IDM) (ISO 29481-1, 2010). Using this approach, an MVD for compliance audit processes can be developed accordingly. The current research has referred to this MVD as the Building Compliance Model (BCM). A further subset of BCM can also be defined for a specific design discipline. For example, a BCM that relates to the fire safety design can be defined as the Fire Compliance Model (FCM), which would contain specific requirements and information for the compliant fire safety and fire protection services design discipline. Similarly, an audit of a compliant electrical services or structural design would make use of an ECM (Electrical Compliance Model) or SCM (Structural Compliance Model), respectively, and so on.

## 3 Accessing Building Information

BIM is now commonly shared using the ISO standard Industry Foundation Classes (IFC) (ISO 16739, 2013). The IFC schema is designed for data exchange given its highly structured and verbose object model. However, its inherent tree-like hierarchical object-based structure with its select types make it difficult to perform quick and dynamic queries. There have been attempts to transform the entire IFC model into a database system with one-to-one translation to facilitate information query (Lee et al., 2012; You et al, 2004). This results in a poorly performing system because of the need to execute joins at each level in the object hierarchy. A more successful approach is that taken by the open source BIMserver (Beetz & van Berlo, 2010) that takes advantage of the smaller performance overhead with a dedicated NOSQL database, although it still uses the one-to-one mapping of the IFC schema.

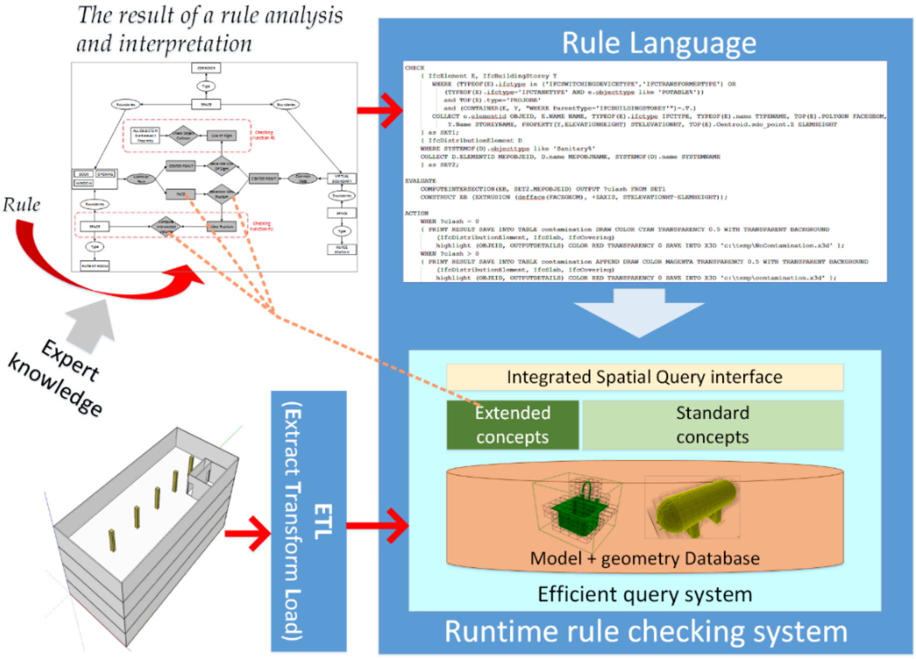
## 4 Query and Rule Languages

Query languages have been used extensively in many domains and applications to manage and extract information from databases since the 1970s starting with the development of SQUARE (Specifying Queries As Relational Expressions) and SEQUEL (Structured English Query Language), which were the basis for the ISO standard SQL (Structured Query Language) (ISO/IEC 9075-1, 2008). SQL has the flexibility of dealing with very simple to highly complex queries combining multiple sub-queries. Over the years, SQL has been extended to include spatial data and object relational features (Borrmann & Rank, 2010; Egenhofer, 1994). Other more dedicated and domain specific query languages that have emerged in recent years include SPARQL for RDF (Resource Description Framework) graphs, which are used in the semantic web technology (Berners-Lee et al, 2001), ProMQL (Product Model Query Language) that is part of the EuroSTEP Model Server (Jørgensen et al., 2008), PMQL (Partial Model Query Language) developed for accessing partial models from an IFC model server (Adachi, 2002), BIMQL (BIM Query Language) that has been incorporated as a prototype software plugin to the BIMserver (Mazairac & Beetz, 2013), RKQL (Regulatory Knowledge Query Language) as a high-level scripting language used in research on executable BPMN 2.0-compliant CDP workflows (Dimyadi et al., 2015).

Rule languages have gained significant research attention in the last few years, particularly in the legal domain for modelling the semantics of business vocabularies, facts and rules (Gordon et al, 2009) such as the SBVR (Semantics of Business Vocabulary and Rules) (Object Management Group, 2015). Other examples of rule languages include SWRL (Semantic Web Rule Language) that combines OWL (Web Ontology Language) with RuleML (Rule Markup Language) (W3C, 2004), NRL (Natural Rule Language) that provides a high-level representation of first-order logic expressions (Nentwich & James, 2011), and LegalRuleML, which is an extension of RuleML incorporating features specific to the formalisation of norms, guidelines and legal reasoning. LegalRuleML is currently being standardised by OASIS (Organisation for the Advancement of Structured Information Standards) to be the de-facto rule interchange language for the legal domain (Athan et al., 2013). There are also rule languages that are developed specifically for the AEC/FM domain, such as EQL (EXPRESS Query Language) and Tnr (Templates and Rules) for ad hoc queries on a STEP file, which is a common format for exchanging the IFC data (Huang, 1999), BERA (Building Environment Rule and Analysis) research tool has a similar approach to EQL for querying a BOM (Building Object Model), which is a simplified version of BIM with a neutral representation that is not specific to any BIM format (Lee, 2011).

## 5 BIM Rule Language (BIMRL)

BIMRL was developed to standardise and simplify building rule checking using a language approach. During the development, it was realised that the problems with rule checking involve much more than just the expressions of the rules. It has to deal with higher level semantics of the building data that the IFC building model is not readily usable for most of the rules. In order to support such semantics, an additional layer of abstraction is needed (Solihin, 2015). BIMRL approaches the problems with a suite of components (Figure 2). The components provide building blocks that support richer semantics with a mind for a practical implementation. For an ease and efficient access to the building data, BIMRL defines a simplified schema that is similar to the star-schema of Data Warehouses (Kimball & Ross, 2013) using the relational database schema. It also integrates the 3D geometry inside the database (Solihin, 2016). This approach is intended for unidirectional read-only data processing where IFC models are loaded into the simplified schema using a lossless ETL (Extract, Transform, Load) process. The rule checking language is defined in tandem with the schema allowing users to query the model using an SQL-like syntax, including complex and nested expressions, and the integrated spatial query capability. It also allows rule checking algorithms to be applied to the query results and to perform logic checks for the rule compliance. BIMRL provides a lightweight geometry engine that comes with a few standardized checking algorithms such as an intersection check, a graph traversal, and shortest path finding.


The result of a rule analysis and interpretation
Rule
Expert knowledge
Rule Language
Integrated Spatial Query interface
Extended concepts
Standard concepts
Model + geometry Database
Efficient query system
Runtime rule checking system
ETL
(Extract Transform Load)

```
CHECK
    { IfcElement E, IfcBuildingStorey Y
      WHERE (TYPEOF(E).ifcType in ('IFCSWITCHINGDEVICETYPE','IFCTRANSFORMERTYPE') OR
         (TYPEOF(E).ifctype='IFCTANKTYPE' AND e.objecttype like 'POTABLE%'))
         and TOP(E).type='PROJOBB'
         and (CONTAINER(E, Y, "WHERE ParentType='IFCBUILDINGSTOREY'")=.T.)
      COLLECT e.elementid OBJEID, E.NAME NAME, TYPEOF(E).ifctype IFCTYPE, TYPEOF(E).name TYPENAME, TOP(E).POLYGON FACEGEOM,
         Y.Name STOREYNAME, PROPERTY(Y,ELEVATIONHEIGHT) STELEVATIONHT, TOP(E).Centroid.sdo_point.Z ELEMHEIGHT
    } as SET1;
    { IfcDistributionElement D
    WHERE SYSTEMOF(D).objecttype like 'Sanitary%'
    COLLECT D.ELEMENTID MEPOBJEID, D.name MEPOBJNAME, SYSTEMOF(D).name SYSTEMNAME
    } as SET2;

EVALUATE
    COMPUTEINTERSECTION(EB, SET2.MEPOBJEID) OUTPUT ?clash FROM SET1
    CONSTRUCT EB (EXTRUSION (defface(FACEGEOM), +ZAXIS, STELEVATIONHT-ELEMHEIGHT));

ACTION
    WHEN ?clash = 0
    { PRINT RESULT SAVE INTO TABLE contamination DRAW COLOR CYAN TRANSPARENCY 0.5 WITH TRANSPARENT BACKGROUND
        (IfcDistributionElement, IfcSlab, IfcCovering)
        highlight (OBJEID, OUTPUTDETAILS) COLOR RED TRANSPARENCY 0 SAVE INTO X3D 'c:\temp\NoContamination.x3d' };
    WHEN ?clash > 0
    { PRINT RESULT SAVE INTO TABLE contamination APPEND DRAW COLOR MAGENTA TRANSPARENCY 0.5 WITH TRANSPARENT BACKGROUND
        (IfcDistributionElement, IfcSlab, IfcCovering)
        highlight (OBJEID, OUTPUTDETAILS) COLOR RED TRANSPARENCY 0 SAVE INTO X3D 'c:\temp\contamination.x3d' };
```

**Figure 2 Components of the BIMRL rule-based checking system**

### 5.1 BIMRL rule-based Checking

By design, BIMRL is able to deal with problems associated with the rule-based checking approach by providing an efficient query functionality and a high-performance response to complex queries involving geometry and spatial operations, all in one language. This language approach opens up the possibility for a flexible query system and the concept of extensions or plugins that work similar to the stored procedures in relational database systems (RDBMS). Very complex rules can be represented as a series of chained rules (Solihin, 2016).

As a language, BIMRL takes advantage of the expressive power and flexibility of the SQL style query approach and extends it with the ability to construct transient geometry for spatial queries against objects in the building model. Using BIMRL ones are able to query the model, construct transient geometries if required, apply checking algorithms, perform logic check based on the checking results and the compliance requirements, and determine the output presentations of the checking results, both in tabular and in 3D views.

## 5.2 BIMRL Queries

There are two types of query supported by BIMRL, namely the BIMRL rule triplets and SQL pass-through uninterpreted query. The BIMRL rule triplets closely resemble that of the NRL (Natural Rule Language) rule structure and consists of three parts, as follows:

1. CHECK statement. This is the base query system traversing the BIMRL schema. It allows the selection of objects of interest. It supports built-in functions to simplify join operations within a statement. The CHECK sub-clause supports multiple selection sets.
2. EVALUATE statement. In this sub-clause, a checking function will be executed on selected objects. It also allows JOIN statements to be defined for multiple data sets generated in the CHECK sub-clause. At the same time, additional (transient) geometry can be generated.
3. ACTION statement. This sub-clause defines the action to be performed based on the outcome of the EVALUATE sub-clause. It generates results into a tabular format, in JSON, and also the geometry (both from the model and the transient geometry) for 3D visualization.

An exemplar BIMRL triplet query is shown in Listing 1. This query checks a given building model and outputs any space in the building that has the smallest dimension less than 1000 mm in any direction (length, width or depth).

**Listing 1 An exemplar BIMRL triplet query**

```
CHECK IfcSpace S, BIMRL_TOPO_FACE F
WHERE F.ElementId = S.ElementId
AND F.Type = 'BODY'
AND F.Orientation in ('BOTTOM','SIDE')
COLLECT S.ElementId SpaceId,S.Name SpaceNo,S.LongName SpaceName,
  F.Polygon Geom,PROPERTY(S,Height,BaseQuantities) SpaceHeight;
EVALUATE SmallestRectangularEdge(Geom) OUTPUT ?SmallestEdge;
ACTION WHEN ?SmallestEdge < 1000 {PRINT RESULT};
```

Executing the query on a sample building model produces a tabular output as shown in Table 1. In this case, one space is found that has a dimension of less than 1000 mm.

**Table 1 The result of executing the query in Listing 1**

| Element Id | Space Number | Space Name | PropertySet | Height | Smallest Edge |
|---|---|---|---|---|---|
| 1EBPnyWt$bJerAs_yHIWpx | 013 | Store Room | BaseQuantities | 2400. | 954.60 |

# 6 Compliance Audit Process

## 6.1 Compliance Audit Framework

A compliance audit framework (referred to as ARCABIM) has recently been developed in a research project. A unique feature of the framework is its ability to execute BPMN-compliant workflows representing compliant design procedures (CDP), which incorporate script-tasks with embedded instructions to guide the data gathering and audit process.

In the previous research, a high-level domain-specific query language referred to as RKQL (Regulatory Knowledge Query Language) has been used to demonstrate the process in a case study. The objective was to demonstrate a means of querying regulatory requirements effectively for compliance audit. BIMRL provides superior building information query capabilities and is complementary to RKQL. The ability to integrate BIMRL into executable CDP workflows for automated compliance audit has been investigated using a case study of a sample building.

## 6.2 Data Exchange Method

Both ARCABIM and BIMRL compliance process and audit engines have been developed as client/server applications, which supports data exchange with external applications on the internet using the REST (Representational State Transfer) request/response web service with the standard Hypertext Transfer Protocol (HTTP). As both applications are developed on the Microsoft's .NET framework, the standard HttpWebRequest class of the framework can be used for this purpose. The selected data-interchange format is JSON (JavaScript Object Notation), which is a popular, lightweight, human-readable format that is easy for machines to parse and generate. An exemplar

URI (Uniform Resource Identifier) for posting a query and the corresponding query result is shown in Listing 2. In this exemplar, the query unique ID is 1018.

**Listing 2 Exemplar URIs for posting BIMRL and ARCABIM queries**

```
http://query.bimrl.info/1018/
http://queryresult.arcabim.info/1018/
```

As an example, the BIMRL query given in Listing 1 can be posted by a workflow task as a JSON object (Listing 3). Once processed, the BIMRL server would post back the corresponding query result to ARCABIM using the same protocol and also in JSON.

**Listing 3 Exemplar BIMRL query in JSON**

```
{
  "CHECK" : [
    {
       "ObjectList" : [
          {
             "ObjectName" : "IfcSpace",
             "Alias" : "S"
          },
          {
             "ObjectName" : "BIMRL_TOPO_FACE",
             "Alias" : "F"
          }
       ],
       "WHERE" : "F.ElementId = S.ElementId AND F.Type = 'BODY'
                  AND F.Orientation in ('BOTTOM','SIDE')",
       "COLLECT" : [
          "S.ElementId:SpaceId",
          "S.Name:Space",
          "S.LongName:SpaceName",
          "F.Polygon:Geom",
          "PROPERTY(S,Height,BaseQuantities):SpaceHeight"
       ]
    }
   ],
  "EVALUATE" : [
    {
       "EvalExpr" : "SmallestRectangularEdge(Geom)",
       "OUTPUT" : "?SmallestEdge"
    }
  ],
  "ACTION" : [
    {
       "WHEN" : "?SmallestEdge LT 1000",
       "ActionItem" : "PRINT RESULT"
    }
  ]
}
```

For the pass-through SQL type query, the name of database and database tables are prefixed with the “bimrl_” keyword as a good security measure. The prefix is stripped off by the BIMRL server prior to processing the query.

## 7 Case study

### 7.1 Sample Building Model

An exemplar four-storey building model has been used for the purpose of this paper to illustrate the compliance audit workflow that incorporates the integrated query approach. The building is a simple open-plan office served by a single stairwell with the final exit on the ground level (Figure 3). A store room is located at the bottom of the stairwell on the ground level (Figure 5). The upper levels of the building have a typical floor layout (Figure 4).

The selected building model was initially developed by the regulatory authority in New Zealand as a test case for performance-based fire safety design for compliance with a verification document C/VM2 as one of three means of compliance with the New Zealand Building Code (MBIE, 2014).

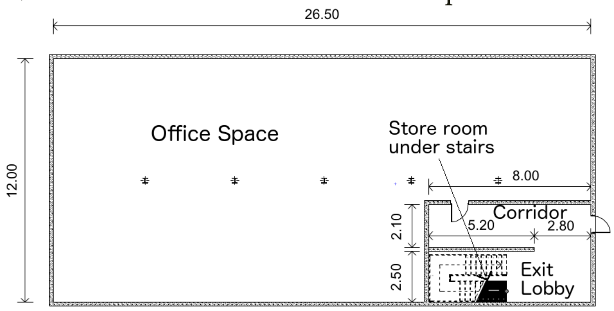


**Figure 3 Ground level layout plan**

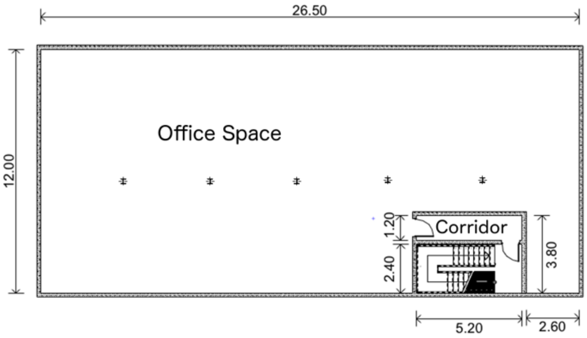


**Figure 4 Typical upper levels layout plan**

**Figure 5 The "Unknown Threat" fire scenario**

## 7.2 An Exemplar Compliance Audit Scenario

Fire evacuation design is an important aspect in a compliant building design. The exemplar building has a single means of escape for the occupants via the stairwell and through the exit lobby and the final exit door on the ground level. As there is no alternative escape routes, a common scenario in performance-based fire safety design is to assess if a fire in any normally-unoccupied space, such as the store room, would pose a risk to an adjacent occupied space or evacuation route and potentially hinder the fire evacuation. This is referred to as the "Unknown Threat" fire scenario in the C/VM2 verification document of the New Zealand Building Code (MBIE, 2014). In this case, a fire in the store room may pose an unknown threat to the main evacuation route for the building and needs to be assessed and designed to mitigate the risk. One method of complying with this scenario is to either install an automatic fire sprinkler system or confine the fire within the store room with a fire-rated enclosure for the entire duration of the evacuation. The fire-rating of the enclosure must be based on the following criteria:

1. If there is no fire detection system installed in the store room, the fire-rating of the enclosure must be designed to withstand a "full burnout fire", which can be calculated using a specific method known as the "Time Equivalence" formula (MBIE, 2014).
2. Given a fire detection system installed in the store room, an enclosure with 60 minutes fire-rating is considered adequate for compliance. Alternatively, it may also be demonstrated by calculations that the fire can be confined within the room for at least the entire duration of evacuation.

There are several queries and rules involved in verifying compliance with this scenario that can be processed by BIMRL:

1. Is the store room protected by an automatic fire sprinkler system?
2. Has a fire detection (with smoke or heat sensors) system been installed in the store room?
3. Get geometrical properties of the store room to enable full burnout fire calculations

The CDP workflow appropriate to assess compliance with this scenario is given in Figure 6. In this workflow, the presence of a fire sprinkler system installation in the store room is checked. If present, then compliance is established. Otherwise, further checking is required to assess whether or not the fire would be confined within the store room for the entire duration of the evacuation. The time required for evacuation is usually calculated using a fire evacuation simulation tool, which ARCABIM can interface with by generating the required input data for the tool and receiving the calculation result. As mentioned above, the time required for fire to break out of the enclosure can be determined from the Time Equivalence formula (MBIE, 2014).

The task to determine whether the fire can be confined within the room (Figure 6) is a sub-process of the workflow that calls an external process (Figure 7).

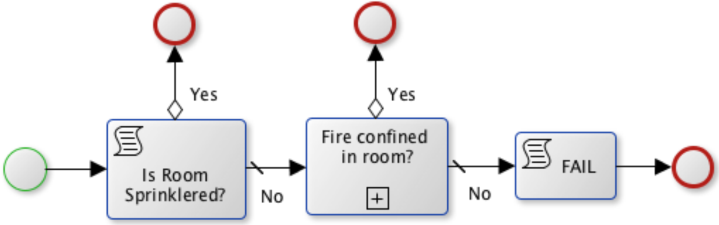


**Figure 6 Exemplar CDP workflow to check the "Unknown Threat" fire scenario**

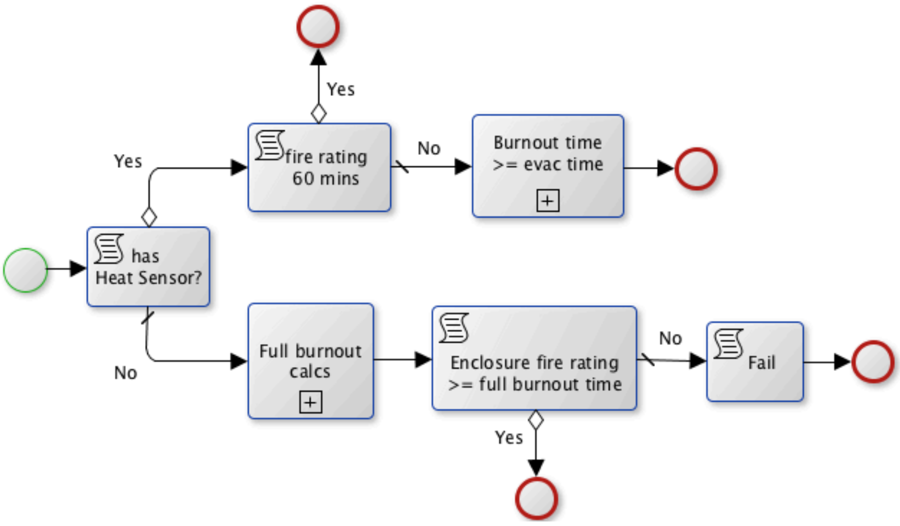


**Figure 7 Exemplar CDP workflow to check if fire can be confined within the room**

The BIMRL query to get a list of spaces that have an automatic fire sprinkler system installed is given in Listing 4. To check if a specific room has an automatic fire sprinklers system, the condition of the query can be modified by specifying the *ElementId* property of the *IfcSpace*. The output of the query can be filtered by the room ID to get information for a specific room.

**Listing 4 An exemplar BIMRL query to get spaces with an automatic fire sprinkler system**

```
CHECK IfcSpace SC
WHERE CONTAINS(SC,USEGEOMETRY).ElementType='IFCFLOWTERMINAL'
AND CONTAINS(SC,USEGEOMETRY).Name like 'Sprinkler%'
COLLECT SC.ElementId SpaceId,SC.Name SpaceName, SC.LongName SpaceLName,
  COUNT(unique CO.ElementID) SprinklerNo
GROUP BY SC.ELementId,SC.Name,SC.LongName;
EVALUATE Nothing();
ACTION PRINT RESULT;
```

The BIMRL query to get a list of spaces in the building model where heat sensors are present (component of the fire detection system) is given in Listing 5. Similarly, the output of the query can be filtered using the room ID to get information for a specific room.

**Listing 5 An exemplar BIMRL query to get spaces with heat sensors**

```
CHECK IfcSpace SC, IfcDistributionControlElement DC
WHERE CONTAINS(SC,DC,USEGEOMETRY)=.T.
AND TypeOf(DC).IfcType = 'IFCSENSORTYPE'
AND TypeOf(DC).PredefinedType = 'HEATSENSOR'
COLLECT SC.ElementId SpaceId,SC.Name SpaceName, SC.LongName SpaceLName,
  COUNT(unique DC.ElementID) SPRINKLERNO
GROUP BY SC.ELementId, SC.Name, SC.LongName;
EVALUATE Nothing();
ACTION PRINT RESULT;
```

Other BIMRL queries that are necessary for the calculation of full burnout fire of the store room, for example, would include the following:

1. Get the floor area, perimeter length, and average height of the room
2. Get the inside surface area of the room enclosure
3. Get vertical opening areas, e.g. windows

Another use case for BIMRL is to assist ARCABIM in interfacing with external computational tools. For example, one variable that can be included in the input data generated for an evacuation simulation is the indication of the likely evacuation path. BIMRL can provide this information as shown in Figure 8.

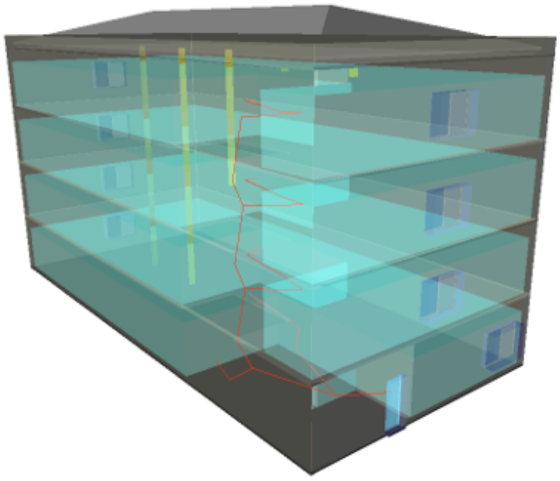


**Figure 8 A likely escape path (shown in red) in the building as identified by BIMRL**

## 8 Discussions and Summary

This paper has described how an external rule-based query language (BIMRL) can be integrated into an automated compliance audit framework (ARCABIM) using standard REST-based web services and the JSON data-interchange format. BIMRL has powerful and efficient BIM querying capabilities and can be used as an alternative query language in conjunction with the executable BPMN-compliant CDP workflows. The ability to use different domain-specific query languages with the CDP workflows is particularly beneficial as it opens up more opportunities to interface with external tools. For example, using BIMRL to generate a likely evacuation path provides an advantage for ARCABIM to use the route as the basis for generating input data for an external fire evacuation simulation tools.

Some of the challenges anticipated with this type of integration include possible performance costs due to the dependency on the network speed, band-width constraints, etc. Other performance improvement and optimisations effort worth investigating include the use of different data exchange options such as reading JSON as a stream instead of a string, utilising data compression, using an alternative JSON serialiser, and so on. The security aspects of exchanging database queries using open standard protocols for this type of applications may also need further evaluations.

## References

Adachi, Y. (2002). Overview of IFC model server framework. In *Proceedings of the European Conference on Product and Process Modelling (ECPPM)*. pp. 367–373.

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.

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

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American*, *284*(5), pp. 29–37.

Borrmann, A., & Rank, E. (2010). Query Support for BIMs using Semantic and Spatial Conditions. *Handbook of Research on Building Information Modeling and Construction Informatics*, pp. 405–450. doi:10.4018/978-1-60566-928-1.ch018

Borrmann, A., Schraufstetter, S., & Rank, E. (2009). Implementing Metric Operators of a Spatial Query Language for 3D Building Models: Octree and B-Rep Approaches. *Journal of Computing in Civil Engineering*, *23*(1), pp. 34–46. doi:10.1061/(ASCE)0887-3801(2009)23:1(34)

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

Dimyadi, J., Clifton, C., Spearpoint, M., & Amor, R. (2016). Computerizing Regulatory Knowledge for Building Engineering Design. *Journal of Computing in Civil Engineering*, *C4016001*, pp. 1–13. doi:10.1061/(ASCE)CP.1943-5487.0000572

Dimyadi, J., Pauwels, P., Spearpoint, M., Clifton, C., & Amor, R. (2015). Querying a Regulatory Model for Compliant Building Design Audit. *Proc. of the 32nd CIB W78 Conference 2015, 27th-29th October 2015, Eindhoven, The Netherlands*, pp. 139–148.

Eastman, C., Lee, J., Jeong, Y., & Lee, J. (2009). Automatic rule-based checking of building designs. *Automation in Construction*, *18*(8), pp. 1011–1033. doi:10.1016/j.autcon.2009.07.002

Egenhofer, M. J. (1994). Spatial SQL: A Query and Presentation Language. *IEEE Transactions on Knowledge and Data Engineering*, *6*(1), pp. 86–95. doi:10.1109/69.273029

Fenves, S. J. (1966). Tabular Decision Logic for Structural Design. *Journal of the Structural Division*, *92*(6), pp. 473–490.

Gordon, T., Governatori, G., & Rotolo, A. (2009). Rules and norms: Requirements for rule interchange languages in the legal domain. In G. Governatori, J. Hall, & A. Paschke (Eds.), *Rule Representation, Interchange and Reasoning on the Web*. pp. 282–296. Springer.

Hietanen, J., & Lehtinen, S. (2014). MVD AND SIMPLEBIM (whitepaper). Datacubist Oy.

Huang, L. (1999). *EXPRESS Query Language and Templates and Rules: Two Languages for Advanced Software System Integrations*. Dissertation. Ohio University.

ISO 16739. (2013). *ISO 16739:2013 Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries*. Geneva, Switzerland: International Organization for Standardization.

ISO 29481-1. (2010). *ISO 29481-1: 2010 Building information modelling - Information delivery manual*. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC 9075-1. (2008). *ISO/IEC 9075-1:2008 Information technology - Database languages - SQL - Part 1: Framework (SQL/Framework)*. Geneva, Switzerland: International Organisation for Standardisation.

Jørgensen, K. a., Skauge, J., Christiansson, P., Svidt, K., Sørensen, K. B., & Mitchell, J. (2008). Use of IFC Model Servers-Modelling Collaboration Possibilities in Practice. *Differences*, (May). Retrieved from http://vbn.aau.dk/files/14804119/ReportIfcModelServer-Final.pdf

Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. 3rd Editio. John Wiley & Sons, Inc.

Lee, G., Jeong, J., Won, J., Cho, C., You, S., Ham, S., & Kang, H. (2012). Query Performance of the IFC Model Server Using an Object-Relational Database (ORDB) Approach and a Traditional Relation Database (RDM) Approach. *Journal of Computing in Civil Engineering*, *Mar/Apr*(28), pp. 210–222. doi:10.1061/(ASCE)CP.1943-5487.0000256

Lee, J. K. (2011). *Building Environment Rule and Analysis (BERA) Language and Its Application for Evaluating Building Circulation and Spatial Program*. Dissertation. Georgia Institute of Technology. Retrieved from http://hdl.handle.net/1853/39482

Mazairac, W., & Beetz, J. (2013). BIMQL – An open query language for building information models. *Advanced Engineering Informatics*, *27*(4), pp. 444–456. doi:10.1016/j.aei.2013.06.001

MBIE. (2014). *C/VM2 Verification Method: Framework for Fire Safety Design For New Zealand Building Code Clauses C1-C6 Protection from Fire*. 4th ed. Wellington, New Zealand: The Ministry of Business, Innovation and Employment.

Nentwich, C., & James, R. (2011). NRL: The Natural Rule Language. Retrieved July 2, 2016, from http://nrl.sourceforge.net

Object Management Group. (2011). Business Process Model and Notation (BPMN), *version 2*, pp. 1–538. Retrieved from http://www.omg.org/spec/BPMN/2.0/

Object Management Group. (2015). *Semantics of Business Vocabulary and Business Rules (SBVR)*. 1.3 ed. Object Management Group. Retrieved from http://www.omg.org/spec/SBVR/1.3/PDF

Solihin, W., & Eastman, C. (2015). *Classification of rules for automated BIM rule checking development. Automation in Construction*, 53(0), pp. 69-82. doi: http://dx.doi.org/10.1016/j.autcon.2015.03.003

Solihin, W. (2016). *A Simplified BIM Data Representation Using a Relational Database Schema for an Efficient Rule Checking System and Its Associated Rule Checking Language*. Dissertation. Georgia Institute of Technology.

W3C. (2004). SWRL: A Semantic Web Rule Language Combining OWL and RuleML. Retrieved July 4, 2016, from https://www.w3.org/Submission/SWRL/

You, S.-J., Yang, D., & Eastman, C. M. (2004). Relational DB Implementation of STEP based product model. In *Proceedings of the 16th CIB World Building Congress*. pp. 1–18.