A Review on BIM-Based Automated Code Compliance Checking System

Aimi Sara Ismail
Department of Quantity Surveying,
Faculty of Built Environment,
Universiti Teknologi Malaysia,
Johor, Malaysia. 81310
aimisara.ismail@gmail.com

Kherun Nita Ali
Department of Quantity Surveying,
Faculty of Built Environment,
Universiti Teknologi Malaysia,
Johor, Malaysia. 81310
b-kherun@utm.my

Noorminshah A. Iahad
Department of Information System,
Faculty of Computing,
Universiti Teknologi Malaysia,
Johor, Malaysia. 81310
minshah@utm.my

Abstract—**The checking process of the building design with building regulations or codes is a time-consuming and error prone process. The automation of code compliance checking process has been explored since 1960s and becoming more vigorous with the emergence of Building Information Modeling (BIM). A neutral format, Industry Foundation Classes (IFC) which is developed under BIM platform had led to better collaboration and information exchange throughout BIM software and development of the checking system. Generally, the structure of an automated system consists of rule interpretation stage, building model preparation stage, rule execution stage and rule reporting stage. Rule interpretation stage is the most vital and complex stage. In interpreting the rules, various techniques have been investigated and applied to employ comprehensive and accurate checking process on the building designs which are mainly in IFC format. The most common techniques used are by utilizing existing software such as Solibri Model Checker, establishing a plug-in application, adopting object-based approach, logical approach and ontological approach. This paper provides a highlight and review on previous studies which successfully employed the appropriate techniques in interpreting the rules for checking purposes. This will subsequently assist future research on this area and create more astounding code compliance checking system.**

Keywords—Building Information Modeling (BIM); Industry Foundation Classes (IFC); automated code compliance checking system; rule interpretation

I. INTRODUCTION

Building Information Modelling, or simply known as BIM, is the most recent technology used in construction industry. BIM can be defined as "a digital representation of a building, an object-oriented three-dimensional model, or a repository of project information to facilitate interoperability and exchange of information with related software applications" [1]. Collaboration between construction parties and information exchange is possible with the development of a neutral format known as Industry Foundation Classes (IFC) in BIM software [2]. This has led to automatic parametric generation of designs and automated checking of the designs [3].

Automated checking of designs, or is also referred to as automated code compliance checking is a process or system which solely assess the design through its objects, attributes, and relations without modifying the design [4]. Prior to the development on this area, code compliance checking by Designers and Local Authorities was carried out manually, and still is, in many countries. As the complexity of design is becoming higher with the emergence of BIM [3] and a large number of complex building codes to be checked [5], manual code compliance checking process is highly time-consuming and leads to many errors [4][6].

Therefore, an automated code compliance checking process is vital in BIM environment. In this transformed process, an organized structure is required to develop the checking system or tool. This structure comprises of four stages – rule interpretation; building model preparation; rule execution; and rule reporting as shown in Fig. 1 [4]. Each stage carries the utmost importance as it holds different responsibilities and functions. In this paper, the rule interpretation stage carried out in previous studies are reviewed. Common techniques used in interpreting the rules are discussed.

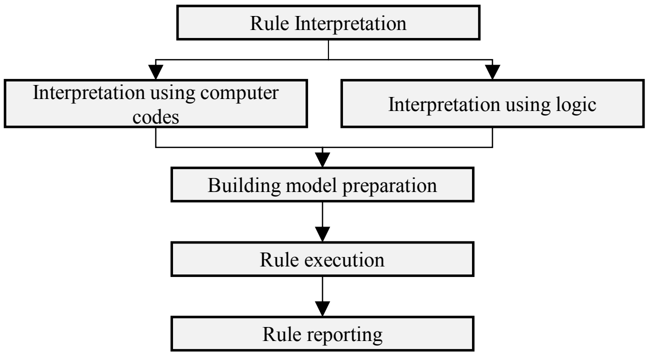

Fig. 1. General structure of rule checking process

II. RULE INTERPRETATION STAGE

In the rule interpretation stage, the rules which are originally produced in human language format must be interpreted into more conducive format for rule checking. There are currently two general methods in executing this, as presented in Fig. 1, in which rules are translated into computer codes by a programmer or rules are formally translated through the logic of human language [4]. The first method is comparatively more sophisticated than the second method as it need to be defined, translated, and maintained with high-level of expertise. On the other hand, the second method provides a

978-1-5090-6255-3/17/$31.00 ©2017 IEEE

mean for logical interpretation which has easier translation process as it can be executed by construction-skilled expert without any programming background.

III. TECHNIQUES USED IN DEVELOPING AUTOMATED CODE COMPLIANCE CHECKING

Exhaustive studies have been carried out by many researchers in exploring different techniques in interpreting the rules in their respective countries. These techniques vary from a straightforward techniques such as logical approach, object based approach and expert system to more complex techniques such as the ontological approach. In this section, selected techniques which are commonly used in interpreting the rules for the development of automated code compliance checking are explained.

A. Existing software or plug-in application

Solibri Model Checker (SMC) is a prominent BIM software application which assist designers in visualizing any issues or problems regarding the design model before and during construction [7]. Solibri Model Checker provides automated analysis and checking which includes indicating any potential design issues, detecting any design deficiencies and missing elements, and checking the building model against the accessibility rules. SMC contains built-in geometry-oriented rules related to space management and accessibility which are hard-coded [8]. Since SMC implements hard-coded functions, the information on rules is not displayed to the end-users. It is argued that adding new rules in SMC is a difficult task as it can only be custom made by Solibri software developers [9]. Nevertheless, this software has been applied in General Services Administration (GSA) project in United States and Norwegian State Administrative Agency project known as HITOS project [4]. Both projects applied SMC to check the compliance of accessibility rules with the building model.

In the GSA Project, checking method which involves validating the spaces in the building was developed for the preliminary and final concept design stage [10]. A set-based method was used in checking the model in preliminary concept design stage. It was assumed that spaces adjacent to each other are potentially connected which create adjacent spaces and checking was done based on these sets. For the final concept design stage, a topological connection graph-based method was applied to connect spaces through building elements such as doors, openings, elevators and stairs. Hence, if the spaces are not connected, this method cannot be used. As both of these methods are only restricted to preliminary and final concept design stage, modification by a third party is required to ensure the model is designed at the specific stage. SMC was applied to produce the area values and reports of the analysis which can then be exported to a reporting document [4]. Currently, all projects in United States are required to prepare the BIM model for validation by GSA [11].

In the HITOS project, SMC was employed to translate the accessibility rules into parametric table structure containing all related building objects such as spaces, doors, stairs, ramps and windows [4]. This parametric table structure also contains values related to the checking equations which can be edited by end-users. SMC then mapped the building objects from the model with the rules and values coded in the table. It is highlighted that SMC is very strict in the model structure requirements. For instance, the space layout must be designed without any overlaps and the attributes' names must be synced with the spaces.

While the GSA project and HITOS project relies on SMC for the checking process, certain studies utilized SMC to validate their framework. For instance, researchers tested their framework for checking of safety rules in SMC as it provides user-friendly visualization and virtual walk-through function [12]. The International Code Council in North America also developed a system known as SMARTcodes and use Solibri Model Checker as the platform to test the reliability of the code checking results [4].

Besides Solibri Model Checker, researchers explored the possibility of a plug-in, a software component that offers additional function to an existing software application for the purpose of automated code compliance checking [13]. In their research, they proposed a framework to be implemented into Autodesk Revit Architecture 2011 for the checking of openings in firewalls, fire resistance ratings, and horizontal continuity of the firewalls based on International Building Code (IBC) in United States. Additional feature in Revit was built to show a static graphical model which contains necessary information on the building properties in the graphical and non-graphical form. Certain information such as the area of opening in firewall and width of opening in firewall which are not readily available in the building model was extracted with the creation of new parameters as knowledge representation of the building codes. The parameters can be created easily as it is one of the features in Autodesk Revit. Hence, it is stated that Autodesk Revit provides suitable platform for developing an automated building design system which includes code compliance checking.

B. Object-based approach

Object-based or also known as object-oriented approach is a technique which organizes knowledge. This organization is done through the representation of object types as the knowledge. Defining the attributes, procedure, rules and machine learning are the examples of activities in organizing the knowledge [14].

In representing the building codes using this approach, three stages are executed as demonstrated in Fig. 2 – building codes classification and abstraction, rule representation modelling, and knowledge base establishment [15]. The building codes are initially classified and interpreted while identifying all related objects to the building codes. Then, relationship between the classes of building codes is established, followed by storing and maintaining the facts and values related to building codes in a tabular form as a knowledge base.

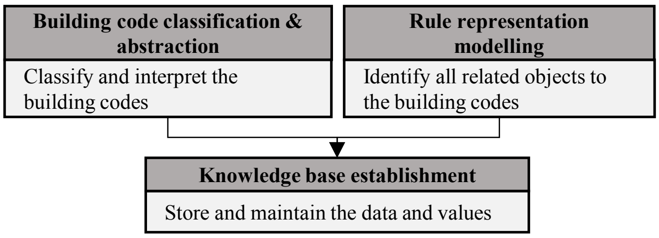

Fig. 2. Stages involved in object based approach for rule checking

While other studies applied similar concept with the procedure carried out in [15], they provided a more sophisticated object-based approach to facilitate the complexity of building codes. CORENET e-PlanCheck is a prominent example in the application of object-based approach as well as in the automated code checking system as it is officially implemented in September 2000 in Singapore, which is earlier compared to other countries [16]. A C++ object library which is an independent platform known as FORNAX was created by novaCITYNETS Pte. Ltd. to establish higher level knowledge base for code compliance checking [4][16]. By creating FORNAX objects through retrieving and extending the IFC model, rules written in natural language can be interpreted without the need to develop any algorithms [4].

Besides CORENET e-PlanCheck, an automated checking system called DesignCheck was developed in Australia [17][18]. This system uses Express Data Manager (EDM) which is equipped with object-based rule engines. This system gives leverage to the users in such way where it accommodates checking at different stages, namely, sketch design, detailed design and documentation as explained in Table I [18]. However, DesignCheck is not implemented commercially and no further development is currently being made [19]. In DesignCheck, each rule was analyzed and the sentence was categorized according to the description, performance requirements, objects, properties and relationships to ease the checking process [18]. This procedure was adopted in another research which they developed a system named Fire Codes Checker (FCC) [20]. Since certain rules are initially complex to be interpreted for automated checking process, a study had taken an approach to filter the rules before interpreting them [21]. The rules or clauses are first categorized into declarative clause category for clauses which are computer interpretable, informative clause category for clauses which need human interpretation, and other category which consists of clauses that are not suitable for automated code compliance checking.

TABLE I. RULE CHECKING AT DIFFERENT STAGES

Design stage	Elements to be checked
Early stage	Accessible paths to or within the building, circulation space at doorway, disabled toilet, etc.
Detailed stage	Door widths, handrail, height, etc.
Specification stage	Floor surfaces, handrail materials, signs, etc.

In another study, a rule based engine framework were presented which adopted a combination of object-oriented approach and logical approach in interpreting fall protection and safety rules [22]. The rules were identified in three stages. Firstly, the objects, attributes, and their relationship were determined, followed by the establishment of logic in assessing the rules. Lastly, the method to resolve the safety issues was analyzed. Rule-based algorithms were then developed for supporting decision-making process. In the rule interpretation process, rules were simplified into categories which is quite similar to the DesignCheck system. The sentences in each rules were categorized based on building objects related in the rule such as floor and roof, object attributes such as holes and skylights, and lastly, the prevention systems suggested in the rule such as covers and guardrail systems.

C. Logical approach

Since rules are prepared by human, the rule interpretation can be done through the formal interpretation and translation of the logic in human language statements [4]. The most common language in interpreting the natural language is the first order predicate logic.

The adoption of predicate logic was explored in [10] to provide validation of checking logically as it is capable to represent the checking method, calculation in the selected rules, various conditions of the rules and able to regard building elements in a model as instances of predicate logic. In this approach, validation or checking process is the process of applying a rule to a building element. The rule, in this case is the occupant circulation rules in United States Courts Design Guide, consists of many logical conditions which are explicit and implicit, and connected through logical connections such as ‘and’, ‘or’ and ‘if-then’. Therefore, the checking can be done by validating the logical combinations of the conditions. Subsequently, this will give validation results comprising of either ‘true’ or ‘false’. In another research, KBimCode which is a set of building permit requirements readily translated for code compliance checking also implemented predicate logic in the interpretation process [23].

Besides predicate logic, conceptual graph were also adopted in interpreting rules into basic logic structure [24]. Conceptual graph is a tool which can be utilized by rule experts without any programming knowledge. It is able to extract the rules, building objects, the relationship between these two components including the constraints. The conceptual graph comprises of rectangles as concept nodes representation, ovals as conceptual relations representation, and diamond shapes as the function representation. In translating the rules into the conceptual graph, four stages must be followed as presented in Fig. 3. Firstly, the main concept of a rule must be identified, for instance, the space concept. Secondly, the atomic sub-rules must be identified which means that each rule contains many sub-rules which are independent from each other. Thirdly, atomic constraints or restrictions were identified. Lastly, the most appropriate conceptual graph was defined through the connections of all elements. BIM Rule Language (BIMRL) was then developed through Structured Query Language (SQL) to support the rule checking process in the BIM environment. SQL, an established query and manipulation language, was used to manage the data. SQL was also adopted in an automated code checking system called LicA [25]. Data was managed through a set of tables containing the code checking routines, related objects, and results of code checking analysis.

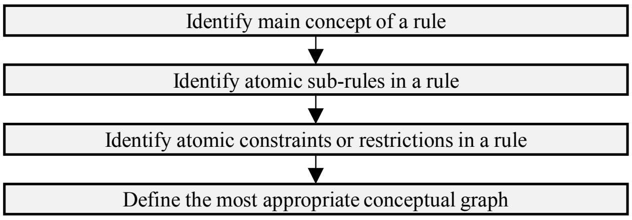

Fig. 3. Stages involved in developing a conceptual graph for logical interpretation

The logical approach was further explored through the introduction of deontology approach, also referred to as deontic logic which is a branch of modal logic [26]. This approach facilitates in reasoning whether a specific state, in this case, the building model, is permitted or forbidden according to relevant regulations. It is highlighted that this approach has higher capability to provide complex knowledge representation and reasoning compared to if-then-else code compliance checking. The framework is unique as it contains two layers, namely, a deontology layer, for representing the regulations and reasoning, and ontology layer, for representing project information as well as a branch of communication model and reasoning, while applying Natural Language Processing (NLP) techniques.

A method known as decision table which was first introduced in year 1969 is very much related to the logical approach [27]. This decision table was formulated to interpret the steel design specifications in a building. As decision table was not computer-programmed, decision table is an approach of documenting logical rules in a form of parameter table. This method was further applied in [6] to extend the building codes. The decision table contains all criteria and decisions arranged in a matrix. Decision table is capable in representing complex logic in a very concise and simple manner. In the decision table created in [6], hyperlinks of the relevant regulations were included.

D. Ontological approach

The semantic web technologies have been an alternative to overcome the limitations faced in the application of Industry Foundation Classes (IFC) for the automation of code compliance checking [28]. The semantic web is constructed in a semantic network for the information representation on the World Wide Web (WWW). A graph is designed to describe concept in the semantic network. The graph displays objects and the logical relations between these objects. Resource Description Framework (RDF) is used as a mean to represent the graph structure as RDF provides statements and expressions to describe relationship between resources [29]. This statement is also known as RDF triples which contains a subject and an object representing the resources and predicate to represent to relationship between resources. RDF is applied in Web Ontology Language (OWL) to describe more complex ontologies [30]. Ontology can be defined as "a document or file that formally defines the relations among terms" [31].

A framework of a formal ontological approach based on semantic web technologies has been developed in [32]. In this framework, four aspects were covered to form a comprehensive conformity-checking model. First, a knowledge acquisition method was developed by applying RDF to represent the model and construction regulations. Second, reasoning model was created to match the model with the construction regulations. Third, capitalization of the use-based knowledge is carried out by formalizing and enriching tacit knowledge contained in the regulations. Lastly, these aspects were integrated into a prototype known as C3R system to validate the framework.

Ontological approach has also been adopted by International Code Council (ICC) in North America through the development of SMARTcodes in 2006 [4]. An International Energy Conservation Code (IECC) dictionary was established which has similar function as the knowledge acquisition method in C3R system. This dictionary also serves as a platform of communication between regulations and building model similar to the reasoning model developed in C3R system. SMARTcodes was employed in Solibri Model Checker to support rule-based code compliance checking. Besides that, a system based on semantic web technologies using Semantics of Business Vocabulary and Business Rules (SBVR) and SPARQL Protocol and RDF Query Language (SPARQL) to formalize the regulations was developed [33]. Besides regulations compliance checking on architectural and structural design domains, ontological approach was also adopted in [34] for construction quality inspection and evaluation known as CQIEOntology.

Semantic based concept was also introduced in automated code compliance checking through a concept named RASE. RASE concept is an abbreviation for Requirement, Applicabilities, Selection and Exceptions [35]. These four elements were used as operators in model checking and each rules are broken down into categories of these four elements. In their previous work, they explained the appropriate usage of these elements as summarized in Table II [36]. 'Requirements' are the most identifiable element in a rule in which the imperative 'shall' or 'shall not' is identified to categorise the text into this element. 'Applicabilities' is for identifying different phrases which refer to the same meaning such as 'external window' and 'external envelope' carry the same concept of 'window'. Meanwhile, 'Selection' is identified when the rules contain alternative concept, for instance, 'doors, windows, and other openings'. On the other hand, 'Exceptions' brings opposite function than 'Applicabilities'.

RASE concept was tested in selected texts in Norwegian accessibility standard, Dubai building regulation, and United States Court design guidance document and concluded that this concept provides reliable results [35]. RASE concept was also implemented in a framework developed in [37] which focused on checking of performance based regulations named

Code for Sustainable Homes (CSH) and Building Research Establishment Environmental Assessment Method (BREEAM). As RASE concept produces pass or fail results, they extended the concept further to support the establishment of numeric form results for the framework. This has allowed the model to be checked through the evaluation rating from 1 to 5 performance credits. While RASE concept was extended in this framework, a study had combined the RASE concept with Dialogue Language adapted from [38] to interpret Life Safety Codes (LSC) for Australia focusing on fire safety requirements [39]. With this combination, the RASE concept can be used to extract the rules according to the four elements while Dialogue Language is adopted to organize the hierarchical dialogue of codes. In another research, a combination of RASE concept and decision logic was executed to interpret the regulations [40].

TABLE II. ELEMENTS IN THE RASE CONCEPT

Elements	Description
Requirements	• Most identifiable element • Associated with the phrase 'shall' or 'shall not'
Applicabilities	• Where different phrases carry the same meaning • Example : 'external envelope' and 'external windows' are 'windows'
Selection	• Where a rule contains alternative subjects • Example : 'doors, windows and other openings'
Exceptions	• Opposite of 'applicabilities'

IV. CONCLUSION

This paper highlights the variety of techniques used in interpreting the rules to develop an automated code compliance checking. Based on the review, it can be concluded that the basis of these techniques is the employment of rule-based concept. As the checking relies on the rules to be complied and the building model to be checked, these two elements are matched through an 'if-then-else' relationship. Hence, the rule-based concept is to be embedded in every technique in the rule interpretation process.

The variety of different techniques will always continue to develop in this field. As stated in [14], "different tasks can have different characteristics that require different kinds of approaches". Therefore, the challenge is not on how new approaches can be developed, but on how to select and integrate these approaches, in other words, the techniques. In a nutshell, the importance of an automated code compliance checking can be observed through the development of many techniques as summarized in Table III.

It is argued that the term automated code compliance checking has been heavily used that it becomes too liberal and vague [41]. Hence, the next step in this field is to create collaborations between different research and the various techniques to have a direct goal in the automation of code compliance checking as well as converting the research ideas into implementations.

ACKNOWLEDGMENT

The authors would like to thank Universiti Teknologi Malaysia for supporting this research.

REFERENCES

[1] Miettinen, R., & Paavola, S. (2014). Beyond the BIM utopia: Approaches to the development and implementation of building information modeling. *Automation in Construction*, *43*, 84–91.

[2] buildingSMART (2016). *IFC Description.* Retrieved March 28 from http://buildingsmart.org/ifc/

[3] Eastman, C., Teicholz, P., Sacks, R., & Liston, K. (2011). BIM Handbook A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors (2nd Editio). New Jersey: John Wiley & Sons.

[4] Eastman, C., Lee, J. min, Jeong, Y. suk, & Lee, J. kook. (2009). Automatic rule-based checking of building designs. *Automation in Construction*, *18*(8), 1011–1033.

[5] Solihin, W., & Eastman, C. (2015). Classification of rules for automated BIM rule checking development. *Automation in Construction*, *53*, 69–

TABLE III. SUMMARY OF TECHNIQUES USED FOR RULE INTERPRETATION

Technique/s	Countries	Regulations involved	References
SMC	U.S.	Occupant circulation rules	[4][10][11]
SMC	Norway	Building accessibility rules	[4]
Autodesk Revit (Plug-in)	U.S.	Fire safety requirements	[13]
Object-based approach	Singapore	Clear area in household shelters	[15]
Object-based approach	Singapore	Rules in building plans and services	[4][16]
Object-based approach	Australia	Disabled access code	[17][18][19]
Object-based approach	Turkey	Fire safety requirements	[20]
Object-based approach	U.K.	Fire safety requirements for dwelling houses	[21]
Object-based & logical approach	U.S.	Fall protection and safety rules	[22]
Logical approach	U.S.	Occupant circulation rules	[10]
Logical approach	Korea	Building permit requirements	[23]
Logical approach	U.S. & Singapore	Hospital visibility, environment and safety, fire safety & accessibility rules	[24]
Structured Query Language	Portugal	Domestic water system regulations	[25]
Decision table	U.S.	Steel design specifications	[27]
Decision table	Canada	Hygrothermal performance of an exterior wall	[6]
Ontological approach	France	Construction regulations	[32]
Ontological approach & SMC	U.S.	Energy conservation codes	[4]
Ontological approach	France	Technical guides of tile roofs	[33]
Ontological approach	China	Construction quality inspection and evaluation	[34]
RASE	Norway, UAE, U.S.	Accessibility, building habitable spaces, and occupant circulation rules	[35][36]
RASE	U.S.	Sustainability and environmental requirements	[37]
RASE & Dialogue Language	Australia	Fire safety requirements	[39]
RASE & decision logic	U.K.	Sustainability requirements	[40]

82.

[6] Tan, X., Hammad, A., & Fazio, P. (2010). Automated Code Compliance Checking for Building Envelope Design. *Journal of Computing in Civil Engineering*, *24*(April), 203–211.

[7] Solibri Inc. (2017). *Solibri Model Checker*. Retrieved March 23 2017, from https://www.solibri.com/products/solibri-model-checker/

[8] Preidel, C., & Borrmann, A. (2015). Automated Code Compliance Checking Based on a Visual Language and Building Information Modeling. In *Proceedings of the 32nd International Symposium of Automation and Robotics in Construction, 15-18 June* (pp. 1–8). Oulu, Finland.

[9] Greenwood, D., Lockley, S., Malsane, S., & Matthews, J. (2010). Automated compliance checking using building information models. In *Proceedings of the Construction, Building and Real Estate Research Conference, 2-3 September* (pp. 363–371).

[10] Lee, J. M. (2010). *Automated Checking of Building Requirements on Circulation Over a Range of Design Phases*. *Georgia Institute of Technology*. Georgia Institute of Technology. Retrieved from http://smartech.gatech.edu/bitstream/handle/1853/34802/lee_jae_m_201008_phd.pdf;jsessionid=10FA65D2354A62739D238AB3B3A15577.smart2?sequence=1

[11] U.S. General Services Administration (GSA). (2015). *GSA Building Information Modeling Guide Series 02 - Spatial Program Validation*. *National 3D-4D-BIM Program*. Washington, U.S.

[12] Zhang, S., Lee, J., Venugopal, M., Teizer, J., & Eastman, C. (2011). Integrating BIM and Safety: An Automated Rule-Based Checking System for Safety Planning and Simulation. In *Proceedings of CIB W99 Conference* (pp. 1–13). Washington.

[13] Nguyen, T. H., & Kim, J. L. (2011). Building code compliance checking using BIM technology. In *Proceedings - Winter Simulation Conference* (pp. 3395–3400). United States.

[14] Martin, J. and Odell, J. (1998). *Object-oriented methods A Foundation.* (2nd ed.) Upper Saddle River, N. J.: Prentice Hall.

[15] Yang, Q. Z., & Li, X. (2001). Representation and Execution of Building Codes for Automated Code Checking. *Proceedings of the Ninth International Conference on Computer Aided Architectural Design Futures*, 315–329.

[16] Khemlani, L. (2005). CORENET e-PlanCheck : Singapore's automated code checking system. Retrieved from http://www.aecbytes.com/buildingthefuture/CORENETePlanCheck.htm

[17] Ding, L., Drogemuller, R._, Jupp, J., Rosenman, M. A., & Gero, J. S. (2004). Automated Code Checking. In *CRC for Construction Innovation, Clients Driving Innovation International Conference* (pp. 1–17). Australia.

[18] Ding, L., Drogemuller, R., Rosenman, M., Marchant, D., & Gero, J. (2006). Automating code checking for building designs - DesignCheck. In *CRC for Construction Innovation, Clients Driving Innovation International Conference* (pp. 1–16). Sydney, Australia.

[19] Dimyadi, J., & Amor, R. (2013). Automated Building Code Compliance Checking – Where is it at? In *Proceedings of the 19th World Building Congress: Construction and Society, 5-9 May* (pp. 172–185).

[20] Balaban, O., Sezen, E., Kilimci, Y., & Cagdas, G. (2012). Automated Code Compliance Checking Model for Fire Egress Codes. *Digital Applications in Construction - eCAADe*, *2*, 1–10.

[21] Malsane, S., Matthews, J., Lockley, S., Love, P. E. D., & Greenwood, D. (2015). Development of an object model for automated compliance checking. *Automation in Construction*, *49*, 51–58.

[22] Zhang, S., Teizer, J., Lee, J. K., Eastman, C. M., & Venugopal, M. (2013). Building Information Modeling (BIM) and Safety: Automatic Safety Checking of Construction Models and Schedules. *Automation in Construction*, *29*, 183–195.

[23] Lee, H., Lee, J.-K., Park, S., & Kim, I. (2016). Translating building legislation into a computer-executable format for evaluating building permit requirements. *Automation in Construction*, *71*, 49–61.

[24] 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. Georgia Institute of Technology.

[25] Martins, J. P., & Monteiro, A. (2013). LicA: A BIM based automated code-checking application for water distribution systems. *Automation in Construction*, *29*(23), 12–23.

[26] Salama, D. M., & El-Gohary, N. M. (2011). Semantic Modeling for Automated Compliance Checking. *Computing in Civil Engineering*, 641–648.

[27] Fenves, S. J., Gaylord, E. H., & Goel, S. K. (1969). *Decision table formulation of the 1969 AISC specification*. Urbana, Illinois.

[28] Pauwels, P., Van Deursen, D., Verstraeten, R., De Roo, J., De Meyer, R., Van De Walle, R., & Van Campenhout, J. (2011). A semantic rule checking environment for building performance checking. *Automation in Construction*, *20*(5), 506–518.

[29] W3C (2014). *RDF 1.1 Primer*. Retrieved March 22 2017, from https://www.w3.org/TR/rdf11-primer/

[30] W3C (2012). *OWL 2 Web Ontology Language Document Overview (Second Edition)*. Retrieved March 22 2017, from https://www.w3.org/TR/owl2-overview/

[31] Berners-Lee, T., Hendler, J., & Lassila, O. (2001, May). The Semantic Web. *Scientific American Magazine*, *21*, 1–36.

[32] Yurchyshyna, A., Faron-Zucker, C., Thanh, N. Le, & Zarli, A. (2008). Towards an Ontology-based Approach for Formalisation of Expert Knowledge in Conformity Checking Model in Construction. In *Proceedings of the 7th European Conference on Product and Process Modelling (ECPPM)* (pp. 447–456). France.

[33] Bouzidi, K. R., Fies, B., Faron-Zucker, C., Zarli, A., & Thanh, N. Le. (2012). Semantic Web Approach to Ease Regulation Compliance Checking in Construction Industry. *Future Internet*, *4*(3), 830–851.

[34] Zhong, B. T., Ding, L. Y., Luo, H. B., Zhou, Y., Hu, Y. Z., & Hu, H. M. (2012). Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking. *Automation in Construction*, *28*, 58–70.

[35] Hjelseth, E., & Nisbet, N. (2011). Capturing Normative Constraints by Use of the Semantic Mark-up RASE Methodology. In *Proceedings of the 28th International Conference of CIB W78* (pp. 1–10). Sophia Antipolis, France.

[36] Hjelseth, E., & Nisbet, N. (2010). Exploring semantic based model checking. In *Proceedings of CIB W78 27th International Conference* (Vol. 27, pp. 1–11). Cairo, Egypt.

[37] Beach, T. H., Kasim, T., Li, H., Nisbet, N., & Rezgui, Y. (2013). Towards automated compliance checking in the construction industry. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8055 LNCS*(PART 1), 366–380.

[38] Omari, A. & Roy, G.G. (1993). A representational scheme for design code information in an expert systems approach to building design. *Computing Systems in Engineering*, *4*(2), 253–269

[39] Shih, S.-Y., Sher, W., & Giggins, H. (2013). Assessment of the building code of Australia to inform the development of BIM-enabled code checking system. In *Proceedings of the 19th World Building Congress: Construction and Society, 5-9 May* (pp. 1–12). Brisbane, Australia.

[40] Kasim, T. (2015). BIM-Based Smart Compliance Checking to Enhance Environmental Sustainability. Cardiff University.

[41] Dimyadi, J., & Solihin, W. (2016). CIB W78 2015 Special track on Compliance Checking. *Journal of Information Technology in Construction*, *21*, 315–316.