Zhang *Visualization in Engineering* (2017) 5:2
DOI 10.1186/s40327-017-0043-4

Visualization in Engineering

RESEARCH Open Access

A logic-based representation and tree-based visualization method for building regulatory requirements

Jiansong Zhang

Abstract

Background: Many research and development efforts have been made for automated compliance checking of building designs with regulatory requirements, but there is a lack of a non-proprietary and user-understandable representation of building regulations to support automated compliance checking in the construction domain that is experimentally tested for understandability and reading speed.

Methods: This research investigates a logic-based representation and tree-based visualization method for building regulatory requirements. The logic-based representation is based on classic logic programming language and can directly support automated compliance checking. The tree-based visualization is expected to improve the understandability and reading speed of the logic-based representation. Therefore, this method attempts to add to the limited research in non-proprietary and user-understandable representation of building regulations that is experimentally tested for understandability and reading speed. To test the understandability and reading speed of regulatory requirements using this representation and visualization method, a survey was conducted to compare different representations, namely, text, logic-based, and tree-based.

Results: Statistical analysis of the survey results shows that the proposed tree-based visualization method can significantly improve the understandability and reading speed of the logic-based regulatory requirement representation and this visualization method is at a comparable status with the original text representation of regulatory requirements in terms of understandability and reading speed.

Conclusions: (1) The investigated logic-based representation and tree-based visualization method for regulatory requirements serves as one potential non-proprietary and user-understandable representation of building regulations; (2) this research shows that computable representations of regulatory requirements can achieve understandability and reading speed that are comparable to the original text representation through tree-based visualization; and (3) this research reveals that the tree-based visualization of regulatory requirements improves the understandability and reading speed of regulatory requirements when such use is compared to the computable logic-based representation.

Keywords: Automated compliance checking, Logic-based representation, Tree-based visualization, Understandability, Reading speed, Building design

Correspondence:
jiansongzh@gmail.com; http://homepages.wmich.edu/~jyb5534/
Department of Civil and Construction Engineering, Western Michigan University, G-238 Parkview Campus, 4601 Campus Dr, Kalamazoo, MI 49008, USA

Background

The automation in building code compliance checking has a long history dating back to the 1960s when Fenves et al. (1969) digitized the American Institute of Steel Construction's (AISC) specifications into decision tables. Since then, a number of efforts have been made to explore the automated compliance checking of building designs with various types of code requirements using various types of computable representations. For example, Garrett and Fenves (1987) presented the automated compliance checking of structural design with structural codes using decision tables and information network to represent structural requirements from design standards; Delis and Delis (1995) examined the automated compliance checking of building architectural features (e.g., space configuration) with fire codes using IF/THEN rules to represent fire code requirements; Han et al. (1997) discussed the automated compliance checking of building components with accessibility requirements using simple simulations and rules to represent accessibility requirements; and Tan et al. (2007, 2010) proposed automated compliance checking of building envelope design using decisions tables to represent building codes and design regulations.

Some commercial automated compliance checking systems were developed. As surveyed by Eastman et al. (2009), several automated compliance checking projects have utilized rule-based checking platforms such as the Solibri Model Checker (SMC), the Express Data Manager (EDM) Checker, and FORNAX. Research efforts have also been designated for the representation of regulatory requirements for automated compliance checking purposes. For example, Lau and Law (2004) described a representation of regulations, standards, and codes using eXtensible Markup Language (XML) as well as ontology; Yurchyshyna et al. (2008) used a representation of regulatory requirements in the form of SPARQL Protocol and RDF Query Language (SPARQL) queries; Pauwels et al. (2011) proposed a representation of regulatory requirements using Notation 3 (N3) Logic; Hjelseth and Nisbet (2011) developed a Requirement, Applies, Select, and Exception (RASE) method for marking up regulatory requirements to support further processing into computable representations; Beach et al. (2013; 2015) extended the RASE method to a regulatory information representation at both the paragraph level and the word group level that can be converted to Semantic Web Rule Language (SWRL) rules; Dimyadi et al. (2014) adopted a representation of regulatory requirements using the Drools Rule Language (DRL); and Zhang and El-Gohary (2016a) developed a representation of regulatory requirements using first order logic. Compared to the methods used in commercial systems, these research efforts strive for a more flexible representation that potentially can be used to represent a variety of code requirements.

The need of a non-proprietary and user-understandable rule representation in automated compliance checking

The need of a non-proprietary and user-understandable representation of building regulations to support automated compliance checking has been pointed out by Garrett et al. (2014). Without such a representation, "many of these computable versions of regulations might go unverified, assumptions will not be understood or managed, and ambiguities may lead to misinterpretations" (Garrett et al. 2014). The aforementioned research efforts facilitated the transparency and openness of such representations and therefore helped with addressing this need. Many of these research efforts used well-established standard language representations such as N3 logic and LegalRuleML and thereby alleviated the problem of proprietary representation in most commercial systems. Some of these methods could use graphical representations to facilitate readers' understanding. For example, in the method proposed by Dimyadi et al. (2014, 2016), resource description framework (RDF) was used to represent regulatory information which can be visualized as diagrams consisting of nodes and directed-arcs, and Business Process Model and Notation (BPMN) diagrams were used at the compliant design procedure level. Both RDF and BPMN diagrams potentially facilitated reader's understanding. However, in spite of the seemingly spontaneous choice towards computable representations that can be easily visualized, little has been found by the author in previous research that focused on experimentally testing or improving the user-understandability of regulatory requirements in the building and construction domain for the various computable rule representations.

To address this research gap, in this paper, I propose here the use of a Prolog-based representation of regulatory requirement which uses a tree structure for visualization, and an experimental testing of the effect of this visualization on the understandability and reading speed of the regulatory requirements. This representation was implemented as part of a larger automated compliance checking system, the Semantic Natural Language Processing-based Automated Compliance Checking (SNACC) system. The regulatory information representation and visualization method is built upon the classic logic programming language – Prolog, which is an approximate realization of the logic programming computational model on a sequential machine (Sterling and Shapiro 1986). Prolog is the most widely used logic programming language; it is well developed with multiple open and freely available implementations. The tree-based visualization method was further developed to

provide a way to facilitate understandability of this Prolog-based rule representation. While there are other visualization methods such as RDF and BPMN diagrams, it is out of the scope of this paper to compare with them. The paper is organized as follows: the Method Overview section briefly introduces the SNACC system, and describes the logic-based representation and tree-based visualization method in detail; the Experimental Testing and Results section describes an experimental testing of the representation and visualization method in comparison to the normal text representation; the Conclusions section highlights the analysis results from the experiment and makes conclusions about such representations and visualizations based on the analysis.

Tree representation in natural language processing

Computational linguistics or natural language processing (NLP) emerged during the last century to enable computers to understand and process natural language (i.e., text and speech) in a human-like manner. Since the early phase of NLP research, visual representation methods have been utilized to help people analyze and understand the various natural language phenomena. A tree representation is one commonly used representation to fulfil such purpose; it was widely used in NLP contents especially for representing the structure of a sentence using phrase structure grammar. Phrase structure grammar (PSG), introduced by Noam Chomsky (Chomsky 1956), leveraged constituency relations to represent the structure of a sentence (Chomsky 1956). According to Chomsky (1956), "a phrase-structure grammar is defined by a finite vocabulary (alphabet) Vp, a finite set Σ of initial strings in Vp, and a finite set F of rules of the form: $X \rightarrow Y$, where X and Y are strings in Vp." In a PSG rule, a complex sequence of features on the right hand side of the rule can be represented by a smaller number of symbols or even just one symbol on the left hand side of the rule. Using a tree representation, the structure of a sentence can be clearly shown by applying PSG rules at each level of the tree to map sentence elements at the current level to the sentence elements at a higher level. For example, Fig. 1 shows the structure of the sentence "Bathrooms, toilet rooms, kitchens, storage rooms and laundry rooms shall be permitted to have a ceiling height of not less than 7 feet." The tree representation provides a rich set of visual cues by leveraging both the horizontal dimension and the vertical dimension of a page in a well-structured way. For example, Fig. 2 shows the parenthesis representation of the same sentence structure as that in Fig. 1, generated using Stanford Parser (Klein and Manning 2003). The tree representation is clearly more understandable than the parenthesis representation. Because of the similarity between parenthesis representation and Prolog syntax, the use of a tree-based visualization may also make the Prolog-based regulatory requirement more understandable.

Methods

Semantic natural language processing-based automated compliance checking (SNACC) system

The author developed a Semantic Natural Language Processing-based Automated Compliance Checking

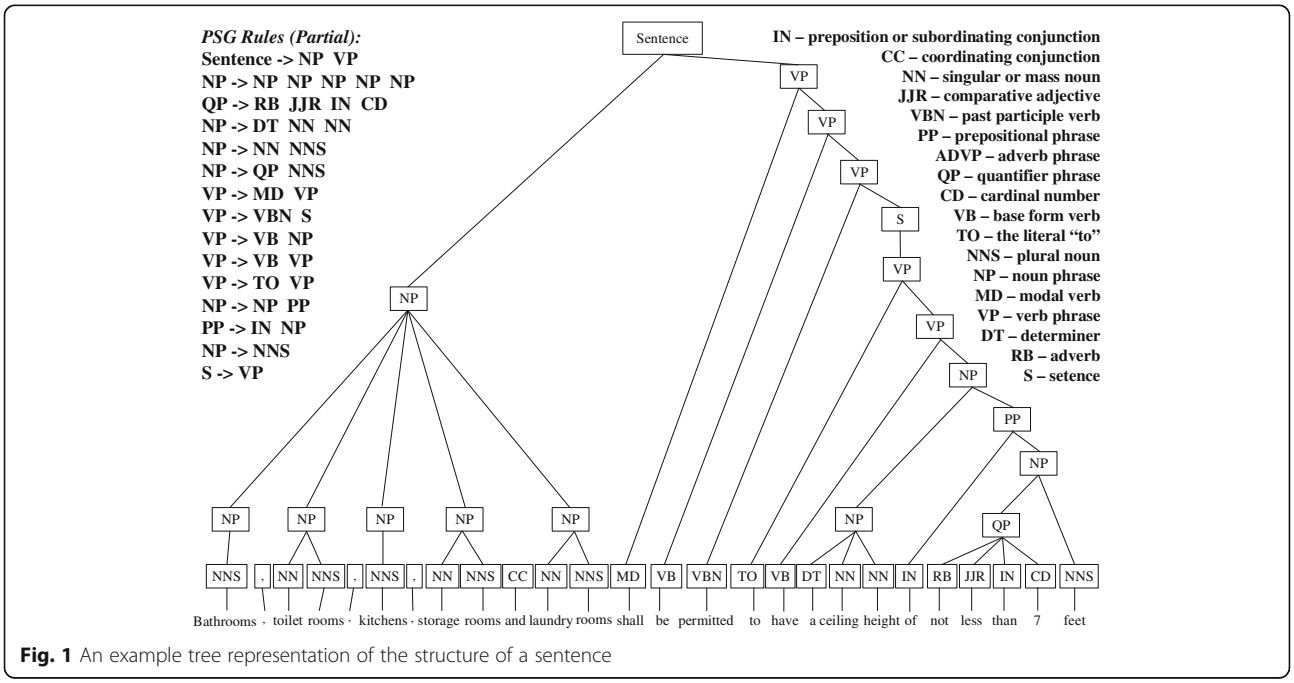

Fig. 1 An example tree representation of the structure of a sentence

```
(ROOT
  (S
    (NP
      (NP (NNS Bathrooms))
      (, ,)
      (NP (NN toilet) (NNS rooms))
      (, ,)
      (NP (NNS kitchens))
      (, ,)
      (NP (NN storage) (NNS rooms))
      (CC and)
      (NP (NN laundry) (NNS rooms)))
    (VP (MD shall)
      (VP (VB be)
        (VP (VBN permitted)
          (S
            (VP (TO to)
              (VP (VB have)
                (NP
                  (NP (DT a) (NN ceiling) (NN height))
                  (PP (IN of)
                    (NP
                      (QP (RB not) (JJR less) (IN than) (CD 7))
                      (NNS feet)))))))))
    (. .)))
```

Fig. 2 An example parenthesis representation of the structure of a sentence

(SNACC) prototype system that allowed the checking of quantitative requirements from building codes; that prototype system involves three main processing steps (Fig. 3): (1) a regulatory information extraction and transformation step that automatically extracts regulatory requirements from building codes and other types of construction regulatory documents and transforms the requirements into logic clauses (i.e., logic rules); (2) a design information extraction and transformation step that automatically extracts building design information from an industry foundation classes (IFC) model [i.e., the ISO standard for building information modeling (BIM)] and transforms the information into logic clauses (i.e., logic facts); and (3) an automated reasoning step that automatically checks the logic facts with the logic rules and generates compliance checking reports. Despite the commonality of these three steps in almost all ACC systems, the major advantages of the SNACC system are: (1) the extraction and transformation of regulatory information is fully automated, by virtue of the underpinning semantic modeling and natural language processing technologies; (2) the design information is automatically aligned with the regulatory information, by virtue of semantic transformation rules encoded in logic clauses; (3) the representation of regulatory information and design information based on logic programming languages were designed to reflect the original meaning of each concept and relation thus making the concepts and relations easier to read and understand by humans. All these advantages were achieved without affecting the computability of the regulatory requirement representations in supporting automated compliance checking. For a detailed description of each component of the SNACC system, the reader is referred to Zhang and El-Gohary (2016b). The next subsection introduces the logic-based representation and tree-based visualization method in detail.

Regulatory information representation and visualization method

The regulatory information representation and visualization method is built upon the classic logic programming language – Prolog: more specifically, the syntax of B-Prolog (i.e., an extended Prolog platform for programming concurrency, constraints, and interactive graphics) (Zhou 2014) is used.

B-prolog syntax

In the syntax of B-Prolog, the logical conjunction operator is represented using a comma and the logical disjunction operator is represented using a semicolon. This explicit representation of logical disjunction in addition to logical conjunction is important to ensure a clear and concise representation of the disjunctive relation between concepts, namely, the relation between multiple concepts that share the same property or requirement. For example, in the requirement from Chapter 12 of the International Building Code (IBC) 2006 (International Code Council 2006) shown in Listing 1, “bathrooms,” “toilet rooms,” “kitchens,”

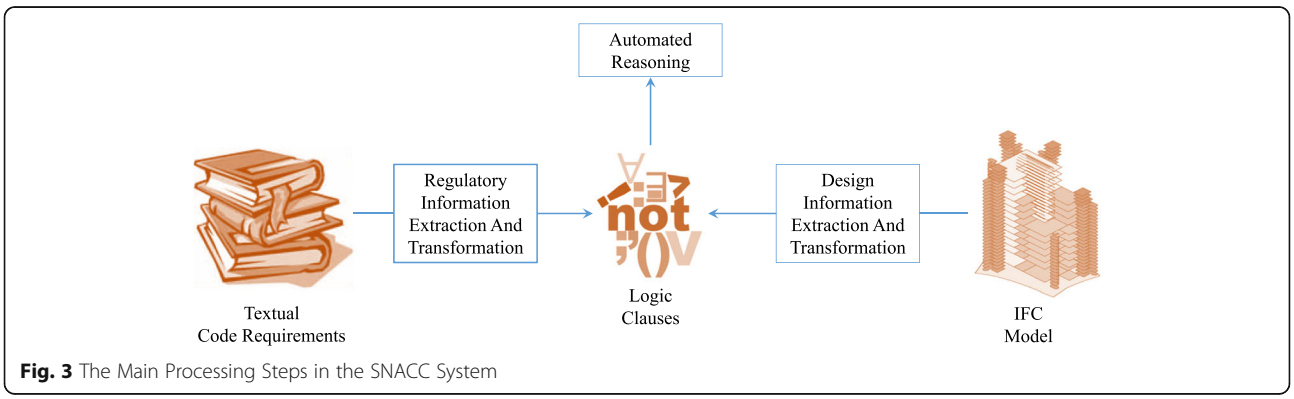

Fig. 3 The Main Processing Steps in the SNACC System

"storage rooms," and "laundry rooms" share the same requirement of "have a ceiling height of not less than 7 feet." Therefore, they have a disjunctive relation.

"Bathrooms, toilet rooms, kitchens, storage rooms and laundry rooms shall be permitted to have a ceiling height of not less than 7 feet (2134 mm)." (Provision 1208.2 of the IBC 2006)

Listing 1: Example regulatory requirement

The implication operator is represented using the combination of a colon and a dash (i.e., ":-"). The implication operator is used to connect the conditions and conclusion of a rule, where the conditions appear to the right of the operator and the conclusion appear to the left of the operator. The negation operator is simply the lower-cased word "not." In the B-Prolog environment and in any Prolog environment in general, every logic statement is universally quantified by default; therefore, there are no operators for universal quantification or existential quantification. In addition to the logical operators, constant, variable, predicate, and function are the four major building blocks of logic clauses in B-Prolog. A constant is represented as an alphanumeric string with the first letter lowercased. A variable is represented as an alphanumeric string with the first letter uppercased. A predicate is represented as a function that has arguments in parenthesis, with each argument being a constant, a variable, or a predicate. There are three main types of logic clauses in B-Prolog: rules, facts, and directives. A rule is a logic clause using the implication relationship to connect conditions and a conclusion; multiple conditions may exist, but there can be only one conclusion in a B-Prolog rule. A fact is a rule without conditions. A directive is a rule without conclusion. The basic syntax pattern for these three types of logic clauses are shown in the bullets below.

- Rule: *p0 :- p1,p2,p3...,pn.*
- Fact: *p0.*
- Directive: *:- p1,p2,p3...,pn.*

Regulatory information representation method

In the proposed regulatory information representation method, regulatory requirements are represented using the rule type of logic clause in B-Prolog. Each regulatory requirement is represented using one B-Prolog rule. For example, the logic clause in Listing 2 is a B-Prolog rule representing the regulatory requirement in Listing 1. In each B-Prolog rule, the condition part represents the compliance conditions as defined in the corresponding regulatory requirement, and the conclusion part specifies the compliant result. In the condition part in the B-Prolog rule in Listing 2, the first five predicates are conjoined using the disjunction operator ";" to represent their disjunctive relation, meaning the requirement applies to all these concepts. Moreover, the same argument (i.e., the variable "*Rooms*") is used for all the five predicates. This arrangement attempts to ensure the computability of the B-Prolog rule, as the variable "*Rooms*" will be instantiated by the instances of all bathrooms, toilet_rooms, kitchens, storage_rooms, and laundry_rooms in the actual execution of the rule in supporting compliance reasoning. A predicate can represent a concept or a relation. The starting five predicates in disjunctive relation in the conditions of the rule in Listing 2 [i.e., *(bathrooms(Rooms);toilet_rooms(Rooms);kitchens(Rooms);-storage_rooms(Rooms);laundry_rooms(Rooms))*] represent concepts for bathroom, toilet, kitchen, storage, and laundry rooms, whereas the next predicate "*has(Rooms,Ceiling_height)*" represents a relation between those rooms and the ceiling height. In other words, the "*Ceiling_height*" variable will be instantiated by a ceiling height instance that belongs to one of those types of rooms. As shown in the last predicate in the rule in Listing 2, a predicate can embed other predicates to show more complicated relations: the predicate "*greater_than_or_equal(Ceiling_height,quantity(7,feet))*" represents the quantitative relation between ceiling height and the quantity 7 feet, where the predicate "*quantity(7,feet))*" is embedded as an argument. The conclusion of a rule appears to the left of the implication operator ":-". The conclusion in the rule in Listing 2 is "*compliant_ceiling_height(Rooms),*" which indicates the ceiling height for a bathroom, toilet room, kitchen, storage room, or laundry room is in compliance with the requirement defined in this regulatory requirement.

```
compliant_ceiling_height(Rooms) :-
(bathrooms(Rooms);toilet_rooms(Rooms);kitchens(Rooms);storage_rooms(Rooms);
laundry_rooms(Rooms)),has(Rooms,Ceiling_height),ceiling_height
(Ceiling_height),greater_than_or_equal(Ceiling_height,quantity(7,feet)).
```

Listing 2: Example B-Prolog rule representing a regulatory requirement

Regulatory information visualization method

To facilitate the understandability of the regulatory requirements represented in B-Prolog rules, a tree structure was adopted to develop a visualization method for the conditions of the rules. This tree-shaped visualization is expected to enhance the representation of information in a regulatory requirement from one dimension (i.e., horizontal) to two dimensions (i.e., horizontal and vertical), and therefore provide more visual cues to the reader. In the developed regulatory information visualization method, the root node is used to represent a "seed concept," which is typically the subject of the requirement. For example, in the regulatory requirement in Listing 3, the subject "courts" is the seed concept. The tree-shaped visualization of a requirement

unfolds from the seed concept through relationships that connect the seed concept with other concepts. Each relation or related concept in the relation is represented at a different level of the tree, except for the disjunctive relation between concepts. The concepts that have disjunctive relation are represented at the same level of the tree using the disjunction operator ";". The tree grows in this way until all the concepts and relations in the requirement have been represented. For example, Fig. 4 shows the tree-shaped visualization of the regulatory requirement in Listing 3. The visualization starts from the root node "courts" and connects to the concept "width" through a possessing relationship represented by "has;" then the concept "width" is connected to the quantity "3 feet" through a quantity comparative relation "greater_than_or_equal." The quantity "3 feet" is broken into its value "3" and the unit "feet" and is represented as two leaf nodes of the tree because neither is further connected to other concepts in the requirement. Note that leaf nodes are represented in green color whereas other types of nodes (i.e., root node and branch node) are represented in blue color. This color coding can help user quickly identify the quantitative requirements because the quantities usually reside in leaf nodes.

"Courts shall not be less than 3 feet (914 mm) in width." (Provision 1206.3 from Chapter 12 of the IBC 2006)

Listing 3: Example regulatory requirement

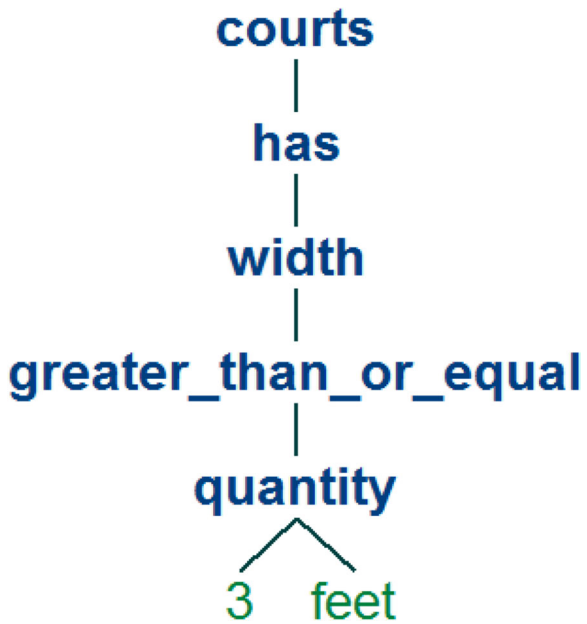

Fig. 4 The tree-shaped visualization of a simple requirement from the IBC 2006

The tree-shaped visualization for the regulatory requirement in Listing 1 is shown in Fig. 5. For processing, the first concept in the subject "bathrooms" was selected as the seed concept. But this was only an arbitrary choice, because all the concepts in the subject are in disjunctive relation which means any of them can be selected as the seed concept with no difference in the final visualization. As shown in Fig. 5, the disjunctive relation between these concepts in the subject are represented using the semicolon. This usage provides consistency with the B-Prolog rule representation for which the visualization was created. The root node is then connected with the concept "ceiling_height" through a possessing relationship represented by "has;" then the concept "ceiling_height" is connected to the quantity "7 feet" through a quantity comparative relation "greater_than_or_equal." The quantity "7 feet" is broken into its value "7" and unit "feet" and is represented as the leaf nodes of the tree.

The examples in Figs. 4 and 5 show trees with one branch because the requirements that the figures were representing had only one path of relationships from the subject to the quantitative requirement. If there are multiple paths from the subject to leaf nodes, the corresponding tree-shaped visualization will have multiple branches. The multiple paths could be used to represent different conditions or different requirements enforced on the subject. Fig. 6 shows the tree-shaped visualization of the regulatory requirement in Listing 4, where the subject of the regulatory requirement has multiple conditions and a quantitative requirement. In the subject of the regulatory requirement in Listing 4, the concepts have the disjunctive relation as represented by the disjunction operator ";". Compared to the visualization of requirements in Figs. 4 and 5, the visualization of the requirement in Fig. 6 better demonstrates the power of leveraging the vertical dimension in the tree-shaped visualization. This structure inherent in the tree-shaped representation is expected to provide more visual cues as a way to help people understand the concepts and their relationships in a regulatory requirement.

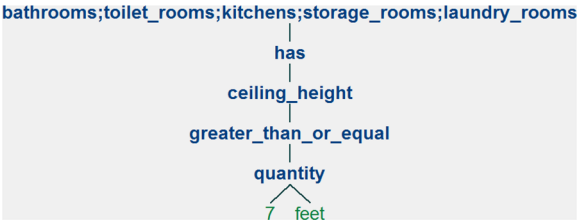

Fig. 5 The tree-shaped visualization of a requirement from the IBC 2006 with disjunctive relation

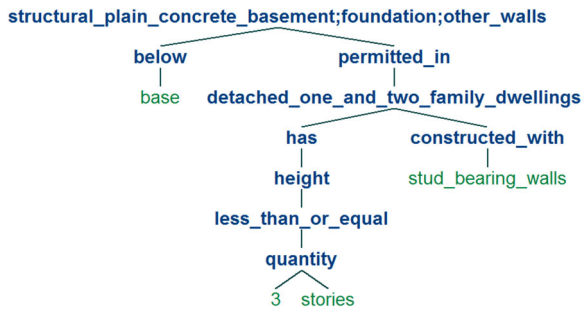

Fig. 6 The tree-shaped visualization of a requirement from the IBC 2009 with disjunctive relation and multiple branches

"Structural plain concrete basement, foundation or other walls below the base are permitted in detached one- and two-family dwellings three stories or less in height constructed with stud-bearing walls." (Provision 1908.1.8 of the IBC 2009)

Listing 4: Example regulatory requirment

Results and discussion

Experimental setup

To test the proposed tree-shaped visualization method in terms of the understandability of the regulatory requirements represented using it, a survey was sent to 300 people with various backgrounds (e.g., student, researcher, engineer, entrepreneur, administrator) randomly. A brief explanation of the B-Prolog syntax was included at the beginning of the survey. Four major pieces of information were collected about the survey participants: sex, age, highest degree earned, and knowledge on building codes. The survey asked participants to compare the text, logic, and visual representations of regulatory requirements in Listing 1 and Listing 4, in terms of understandability and reading speed. The text representation of the regulatory requirement in Listing 1 is the original text of the regulatory requirement coming from the IBC 2006. The logic representation of the regulatory requirement in Listing 1 is shown in Listing 2. The visual representation of the logic requirement in Listing 1 is shown in Fig. 5. The text representation of the regulatory requirement in Listing 4 is the original text of the regulatory requirement coming from the IBC 2009. The logic representation of the regulatory requirement in Listing 4 is shown in Listing 5. The visual representation of the regulatory requirement in Listing 4 is shown in Fig. 6.

```
compliant_story_height :-
((structural_plain_concrete_basement(Structural_plain_concrete_basement);
foundation(Structural_plain_concrete_basement);other_walls(Structural
_plain_concrete_basement)),below(Structural_plain_concrete_basement,
Base),base(Base),detached_one_and_two_family_dwellings(Detached_
one_and_two_family_dwellings),permitted_in(Structural_plain_concrete_
basement,Detached_one_and_two_family_dwellings),height(Height),
has(Detached_one_and_two_family_dwellings,Height),less_than_or_
equal(Height,quantity(3,stories)),constructed_with(Detached_one_and_
two_family_dwellings,Stud_bearing_walls),stud_bearing_walls
(Stud_bearing_walls).
```

Listing 5: Example logic representation

Experimental results and analysis

Among the 300 people contacted, 93 participated in the survey (i.e., 31% response rate). The distribution of the survey respondents in terms of sex, age, highest degree earned, and knowledge on building codes are shown in Figs. 7, 8, 9 and 10.

To help with the statistical analysis on understandability and reading speed results, a scoring method was used: for each vote of a representation over another representation, the winning representation gets a score of 1.5 and the losing representation gets a score of 0.5; for each tie between two representations, both representations get a score of 1. The experimental results are shown in Table 1. Equivalent percentages for each score and their confidence intervals (CIs) at 99, 95 and 90% confidence levels were calculated. Equivalent percentages were defined by dividing each score by 279 (i.e., the score of a representation if it wins over both other representations from all the 93 respondents). The Wilson score method without continuity correction (Wilson 1927) was used to calculate the CIs. Table 1 shows that for both the simple regulatory requirement in Listing 1 and the complex regulatory requirement (i.e., including more concepts and relationships and therefore having more than one branches in the tree-based representation) in Listing 4: (1) the text representation was significantly more understandable than the logic representation at all the three confidence levels; (2) the visual representation was significantly more understandable than the logic representation at all the three confidence levels; (3) while the text representation had higher scores than the visual representation in terms of understandability, the differences were not necessarily significant. For the simple regulatory requirement, the difference was significant at the 90 and 95% confidence levels but not significant at the 99% confidence level. For the complex regulatory requirement, the difference was not significant at any of the three confidence levels; (4) the text representation was significantly faster to read than the logic representation at all the three confidence levels; (5) the visual representation was significantly faster to read

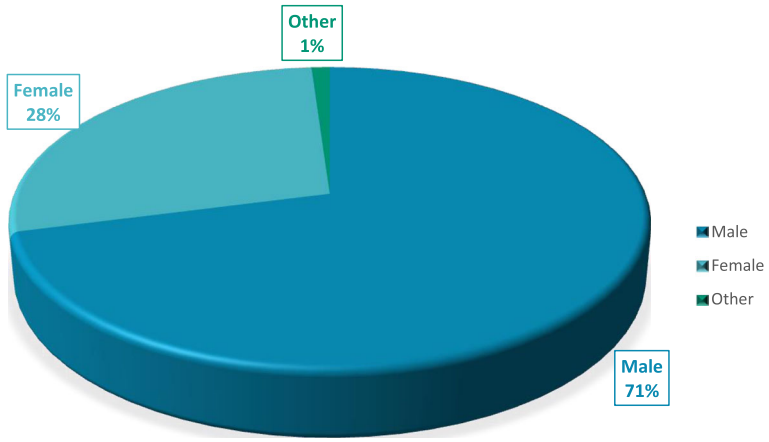

Fig. 7 Sex distribution of the survey respondents

than the logic representation at all the three confidence levels; and (6) while the text representation had higher scores than the visual representation in terms of reading speed, the difference was only significant at the 90% confidence level, but not significant at either the 95% or the 99% confidence level.

To explore if the level of knowledge on building codes affects the above results. A similar analysis was conducted on the four parts of collected data with four levels of knowledge on building codes. The results are shown in Tables 2, 3, 4 and 5. As shown in Tables 2, 3, 4 and 5 the score ranking in terms of understandability and reading speed among the three representations is always in the following order: text representation > visual representation > logic representation, whereas the differences in the scores may or may not be statistically significant. Table 6 summarizes the status of all differences in terms of their statistical significance. It shows that: (1) the difference between text and logic representations is almost always significant with the exception of three small cases (out of 48 cases) on complex requirement when the respondents' level of knowledge on building codes were low; this can be explained by the fact that low level of knowledge on building codes could have led to more difficulty in reading and understanding the text representation of complex building code requirements; (2) the difference between visual and logic representations is almost always not significant at the 99% confidence level but always significant at the 95% and 90% confidence levels, except for four small cases (out of 48 cases) on the understandability of simple requirement when the respondents' level of knowledge on building codes were high, or on the reading speed of complex requirement when the respondents' level of knowledge on building codes were relatively high; this can be explained by the fact that high level of knowledge on building codes may have led to a better capability in understanding the logic representation of simple building code requirements and a faster reading speed of the logic representation of complex building code requirements; (3) the difference between text and visual representations is almost never significant except for one

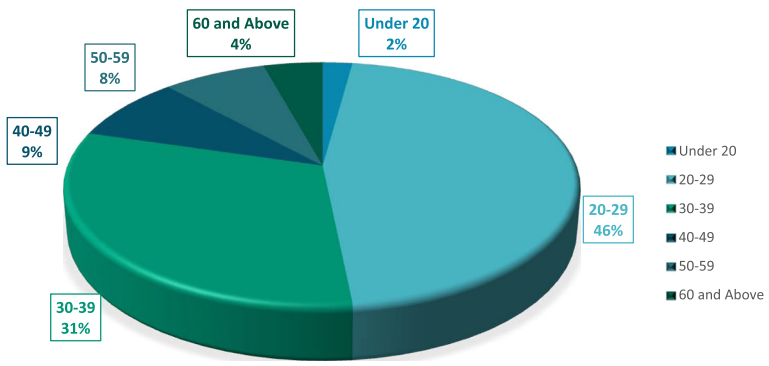

Fig. 8 Age distribution of the survey respondents

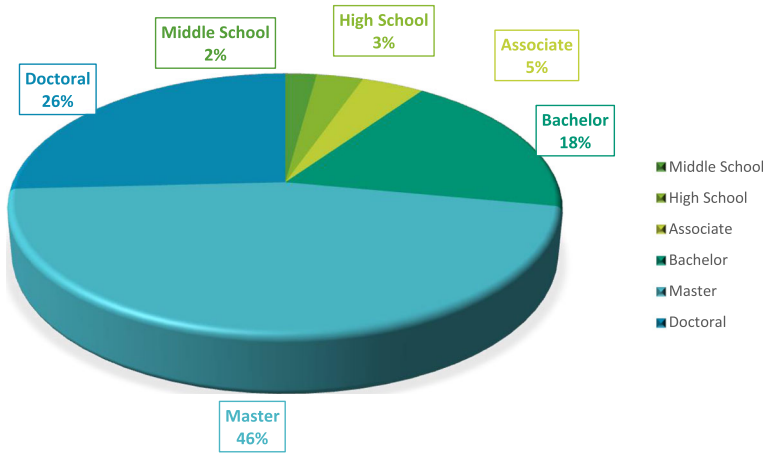

Fig. 9 Degree earned distribution of the survey respondents

small case (out of 48 cases) on the reading speed of simple requirement when the respondents' level of knowledge on building codes were relatively high; this can be explained by the fact that high level of knowledge on building codes may have enabled faster reading of the text representation.

Conclusions

Automated code compliance checking is an important application of modeling and computing technology in the architectural, engineering, and construction industry. Many efforts have been made in academia, government, and industry to develop automated compliance checking methods and systems. In spite of the many efforts in automated compliance checking research and development, there is still a lack of a non-proprietary and user-understandable representation of building regulations to support automated compliance checking. Little has been done toward testing or improving the user-understandability of regulatory requirements in the building and construction domain to support automated compliance checking. To address this research gap, this paper demonstrates a logic-based representation and tree-based visualization method for regulatory requirement. The method leveraged B-Prolog rules to represent regulatory requirements where the conditions of the B-Prolog rule represent the premises of the regulatory requirement, and the conclusion of the B-Prolog rule represents the compliance status with the regulatory requirement. Each concept or relation in a regulatory requirement is represented as a predicate in the B-Prolog rule, with the original names of the concepts and relationships used as the names and arguments of the predicates. This consistency in naming helps users understand the meanings of the concepts and relations simply by looking at the logic rule representation. In the tree-shaped visualization method, a tree structure is used to illustrate the concepts and relations. The tree representation used visual cues both in the horizontal dimension and the vertical dimension to provide a better visualization of the regulatory requirements compared to the pure logic-based representation.

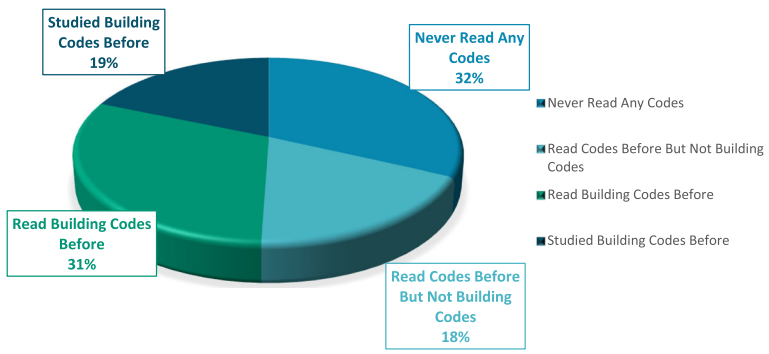

Fig. 10 Knowledge on building codes distribution of the survey respondents

Table 1 Total score for the three types of representations

Representation	Total Score			
	Understandability (simple regulatory requirement)	Reading Speed (simple regulatory requirement)	Understandability (complex regulatory requirement)	Reading Speed (complex regulatory requirement)
Text	203	191	180	184.5
Equivalent Percentage	72.8%	68.5%	64.5%	66.1%
99% CI	(65.4%, 79.0%)	(60.9%, 75.1%)	(56.9%, 71.5%)	(58.5%, 73.0%)
95% CI	(67.3%, 77.6%)	(62.8%, 73.6%)	(58.7%, 69.9%)	(60.4%, 71.4%)
90% CI	(68.2%, 76.9%)	(63.7%, 72.8%)	(59.7%, 69.1%)	(61.3%, 70.6%)
Logic	97	81	91	87.5
Equivalent Percentage	34.8%	29.0%	32.6%	31.4%
99% CI	(27.9%, 42.4%)	(22.6%, 36.5%)	(25.9%, 40.2%)	(24.7%, 38.9%)
95% CI	(29.4%, 40.5%)	(24.0%, 34.6%)	(27.4%, 38.3%)	(26.2%, 37.0%)
90% CI	(30.2%, 39.6%)	(24.8%, 33.7%)	(28.2%, 37.4%)	(27.0%, 36.1%)
Visual	166	158	169	156
Equivalent Percentage	59.5%	56.6%	60.6%	55.9%
99% CI	(51.8%, 66.8%)	(48.9%, 64.0%)	(52.9%, 67.8%)	(48.2%, 63.3%)
95% CI	(53.6%, 65.1%)	(50.8%, 62.3%)	(54.7%, 66.1%)	(50.0%, 61.6%)
90% CI	(54.6%, 64.2%)	(51.7%, 61.4%)	(55.7%, 65.3%)	(51.0%, 60.7%)

To test the understandability and reading speed of the proposed visualization method and how it compares to the original text representation and the logic representation, a survey was conducted during which 300 people with various backgrounds were contacted and 93 of them participated. The survey collected background information of the respondents' sex, age, education, and knowledge level on building codes, and tested the respondents' understanding and reading speed of the different representations of two regulatory requirements. Between the two regulatory requirements one is relatively simple and the other is relatively complex. The testing results were statistically analyzed. Results showed that: (1) the visual representation of regulatory requirements using the tree-based visualization method was significantly better than the logic-based representation in

Table 2 Results for respondents that "never read any codes"

Representation	Total Score			
	Understandability (simple regulatory requirement)	Reading Speed (simple regulatory requirement)	Understandability (complex regulatory requirement)	Reading Speed (complex regulatory requirement)
Text	53.5	51.5	49	49.5
Equivalent Percentage	61.5%	59.2%	56.3%	56.9%
99% CI	(47.7%, 73.7%)	(45.4%, 71.6%)	(42.7%, 69.1%)	(43.2%, 69.6%)
95% CI	(51.0%, 71.0%)	(48.7%, 68.9%)	(45.9%, 66.3%)	(46.4%, 66.8%)
90% CI	(52.7%, 69.4%)	(50.4%, 67.5%)	(47.5%, 64.7%)	(48.1%, 65.3%)
Logic	28.5	26.5	26	26
Equivalent Percentage	32.8%	30.5%	29.9%	29.9%
99% CI	(21.4%, 46.5%)	(19.5%, 44.2%)	(19.0%, 43.6%)	(19.0%, 43.6%)
95% CI	(23.8%, 43.2%)	(21.8%, 40.8%)	(21.3%, 40.2%)	(21.3%, 40.2%)
90% CI	(25.1%, 41.4%)	(23.0%, 39.1%)	(22.5%, 38.5%)	(22.5%, 38.5%)
Visual	50	50	49	48.5
Equivalent Percentage	57.5%	57.5%	56.3%	55.7%
99% CI	(43.8%, 70.1%)	(43.8%, 70.1%)	(42.7%, 69.1%)	(42.1%, 68.6%)
95% CI	(47.0%, 67.3%)	(47.0%, 67.3%)	(45.9%, 66.3%)	(45.3%, 65.7%)
90% CI	(48.7%, 65.8%)	(48.7%, 65.8%)	(47.5%, 64.7%)	(46.9%, 64.2%)

Table 3 Results for respondents that "read codes before but not building codes"

Representation	Total Score			
	Understandability (simple regulatory requirement)	Reading Speed (simple regulatory requirement)	Understandability (complex regulatory requirement)	Reading Speed (complex regulatory requirement)
Text	42.5	39.5	37	36
Equivalent Percentage	83.3%	77.5%	72.5%	70.6%
99% CI	(66.3%, 92.7%)	(59.8%, 88.8%)	(54.6%, 85.3%)	(52.6%, 83.9%)
95% CI	(70.9%, 91.1%)	(64.3%, 86.8%)	(59.1%, 82.9%)	(57.0%, 81.3%)
90% CI	(73.1%, 90.2%)	(66.6%, 85.5%)	(61.3%, 81.5%)	(59.3%, 79.8%)
Logic	16.5	14.5	16	15
Equivalent Percentage	32.4%	28.4%	31.4%	29.4%
99% CI	(18.4%, 50.4%)	(15.4%, 46.4%)	(17.6%, 49.4%)	(16.1%, 47.4%)
95% CI	(21.1%, 46.0%)	(17.9%, 42.0%)	(20.3%, 45.0%)	(18.7%, 43.0%)
90% CI	(22.7%, 43.8%)	(19.3%, 39.7%)	(21.9%, 42.8%)	(20.2%, 40.7%)
Visual	33	30	33	29
Equivalent Percentage	64.7%	58.8%	64.7%	56.9%
99% CI	(46.7%, 79.3%)	(41.1%, 74.5%)	(46.7%, 79.3%)	(39.3%, 72.9%)
95% CI	(51.0%, 76.4%)	(45.2%, 71.2%)	(51.0%, 76.4%)	(43.3%, 69.5%)
90% CI	(53.2%, 74.7%)	(47.3%, 69.4%)	(53.2%, 74.7%)	(45.4%, 67.6%)

terms of understandability and reading speed for both simple and complex regulatory requirements; (2) the original text representation of regulatory requirements was slightly better than the visual representation in understandability when the requirement was simple (i.e., significant at the 90% and 95% confidence levels but not significant at the 99% confidence level). the text representation was slightly better than the visual representation in understandability when the requirement was complex, and the text representation was slightly better than the visual representation in reading speed both for simple and complex requirements (i.e., only significant at the 90% confidence level, or at the 95% confidence level for simple regulatory requirement). But none of these differences were significant unless otherwise specified; (3) the original text representation of regulatory requirements was

Table 4 Results for respondents that "read building codes before"

Representation	Total Score			
	Understandability (simple regulatory requirement)	Reading Speed (simple regulatory requirement)	Understandability (complex regulatory requirement)	Reading Speed (complex regulatory requirement)
Text	63	60.5	55.5	58
Equivalent Percentage	75.0%	72.0%	66.1%	69.0%
99% CI	(61.3%, 85.0%)	(58.2%, 82.7%)	(52.0%, 77.8%)	(55.1%, 80.2%)
95% CI	(64.8%, 83.0%)	(61.6%, 80.5%)	(55.4%, 75.3%)	(58.5%, 77.9%)
90% CI	(66.5%, 81.9%)	(63.4%, 79.3%)	(57.2%, 73.9%)	(60.3%, 76.6%)
Logic	32	25.5	32	30
Equivalent Percentage	38.1%	30.4%	38.1%	35.7%
99% CI	(25.8%, 52.1%)	(19.3%, 44.3%)	(25.8%, 52.1%)	(23.8%, 49.8%)
95% CI	(28.4%, 48.8%)	(21.6%, 40.9%)	(28.4%, 48.8%)	(26.3%, 46.4%)
90% CI	(29.9%, 47.1%)	(22.8%, 39.1%)	(29.9%, 47.1%)	(27.7%, 44.6%)
Visual	49	44	50.5	44
Equivalent Percentage	58.3%	52.4%	60.1%	52.4%
99% CI	(44.4%, 71.1%)	(38.7%, 65.7%)	(46.1%, 72.6%)	(38.7%, 65.7%)
95% CI	(47.7%, 68.3%)	(41.8%, 62.7%)	(49.4%, 69.9%)	(41.8%, 62.7%)
90% CI	(49.4%, 66.8%)	(43.5%, 61.1%)	(51.1%, 68.5%)	(43.5%, 61.1%)

Table 5 Results for respondents that "studied building codes before"

Representation	Total Score			
	Understandability (simple regulatory requirement)	Reading Speed (simple regulatory requirement)	Understandability (complex regulatory requirement)	Reading Speed (complex regulatory requirement)
Text	40.5	36	35	37.5
Equivalent Percentage	79.4%	70.6%	68.6%	73.5%
99% CI	(61.9%, 90.2%)	(52.6%, 83.9%)	(50.6%, 82.4%)	(55.6%, 86.0%)
95% CI	(66.5%, 88.2%)	(57.0%, 81.3%)	(55.0%, 79.7%)	(60.1%, 83.7%)
90% CI	(68.7%, 87.1%)	(59.3%, 79.8%)	(57.2%, 78.1%)	(62.4%, 82.3%)
Logic	18.5	13.5	15.5	15.5
Equivalent Percentage	36.3%	26.5%	30.4%	30.4%
99% CI	(21.5%, 54.2%)	(14.0%, 44.4%)	(16.9%, 48.4%)	(16.9%, 48.4%)
95% CI	(24.5%, 50.0%)	(16.3%, 39.9%)	(19.5%, 44.0%)	(19.5%, 44.0%)
90% CI	(26.2%, 47.8%)	(17.7%, 37.6%)	(21.0%, 41.8%)	(21.0%, 41.8%)
Visual	31	30.5	33.5	31
Equivalent Percentage	60.8%	59.8%	65.7%	60.8%
99% CI	(42.9%, 76.2%)	(42.0%, 75.3%)	(47.7%, 80.1%)	(42.9%, 76.2%)
95% CI	(47.1%, 73.0%)	(46.1%, 72.1%)	(52.0%, 77.2%)	(47.1%, 73.0%)
90% CI	(49.3%, 71.2%)	(48.3%, 70.3%)	(54.2%, 75.6%)	(49.3%, 71.2%)

significantly better than the logic-based representation in terms of understandability; (4) the original text representation of regulatory requirements was significantly better than the logic-based representation in reading speed. This result shows that the proposed tree-based visualization method can significantly improve the understandability and reading speed of the logic-based regulatory requirement representation, and it is at a comparable status with the original text representation of regulatory requirements in terms of understandability and reading speed. Further analysis into the different levels of knowledge on building codes showed that high level of knowledge on building codes may help with the understanding (of simple regulatory requirement) and reading speed (of complex regulatory requirement) of the logic representation, but the difference between visual and logic representation is significant in most cases.

Contributions to the body of knowledge

This research contributes to the body of knowledge in three main ways: (1) before this research, little has been done for testing or improving the user-understandability of regulatory requirements in the building and construction domain to support automated compliance checking, as far as the author is aware. This research is among the first to experimentally compare the understandability and reading speed between different representations of building regulations in a quantitative manner; (2) there is a lack of a non-proprietary and user-understandable representation of building regulations to support automated compliance checking in the construction domain; the proposed logic-based representation and tree-based visualization method serves as one possible non-proprietary and user-understandable representation of building regulatory requirements; (3) this research shows that when given an effective visualization method, the computable representations of regulatory requirements can achieve understandability and reading speed that are comparable to the original text representation (i.e., with no significant difference); and (4) this research reveals that the tree-based representation of regulatory requirements improve the understandability and reading speed of regulatory requirements as compared to the computable logic-based representation.

Limitations and future work

Two main limitations of the current work are acknowledged. First, the work presented in this paper only serves as an initial investigation on the potential effects of visualization on computable regulatory representations in terms of understandability and reading speed, and this investigation focused on a specific type of computable regulatory information representation (i.e., Prolog-based representation) and a specific type of visualization technique (i.e., tree-based visualization). How the tree-based visualization performs for other types of computable regulatory information representations and how other types of visualization technics compare with the tree-based visualization can be conducted by follow-up research efforts. Second, there are regulatory requirements that are much more complex than the ones used in the experiment in this research; how the different types of

Table 6 Significant status of the difference between representations

Difference Between Representations	Confidence Level	Total Score			
		Understandability (simple regulatory requirement)	Reading Speed (simple regulatory requirement)	Understandability (complex regulatory requirement)	Reading Speed (complex regulatory requirement)
Text & Logic ①	99%	Significant	Significant	Not Significant	Not Significant
	95%	Significant	Significant	Significant	Significant
	90%	Significant	Significant	Significant	Significant
Text & Logic ②	99%	Significant	Significant	Significant	Significant
	95%	Significant	Significant	Significant	Significant
	90%	Significant	Significant	Significant	Significant
Text & Logic ③	99%	Significant	Significant	Not Significant	Significant
	95%	Significant	Significant	Significant	Significant
	90%	Significant	Significant	Significant	Significant
Text & Logic ④	99%	Significant	Significant	Significant	Significant
	95%	Significant	Significant	Significant	Significant
	90%	Significant	Significant	Significant	Significant
Visual & Logic ①	99%	Not Significant	Not Significant	Not Significant	Not Significant
	95%	Significant	Significant	Significant	Significant
	90%	Significant	Significant	Significant	Significant
Visual & Logic ②	99%	Not Significant	Not Significant	Not Significant	Not Significant
	95%	Significant	Significant	Significant	Significant
	90%	Significant	Significant	Significant	Significant
Visual & Logic ③	99%	Not Significant	Not Significant	Not Significant	Not Significant
	95%	Not Significant	Significant	Significant	Not Significant
	90%	Significant	Significant	Significant	Not Significant
Visual & Logic ④	99%	Not Significant	Not Significant	Not Significant	Not Significant
	95%	Not Significant	Significant	Significant	Significant
	90%	Significant	Significant	Significant	Significant
Text & Visual ①	99%	Not Significant	Not Significant	Not Significant	Not Significant
	95%	Not Significant	Not Significant	Not Significant	Not Significant
	90%	Not Significant	Not Significant	Not Significant	Not Significant
Text & Visual ②	99%	Not Significant	Not Significant	Not Significant	Not Significant
	95%	Not Significant	Not Significant	Not Significant	Not Significant
	90%	Not Significant	Not Significant	Not Significant	Not Significant
Text & Visual ③	99%	Not Significant	Not Significant	Not Significant	Not Significant
	95%	Not Significant	Not Significant	Not Significant	Not Significant
	90%	Not Significant	Significant	Not Significant	Not Significant
Text & Visual ④	99%	Not Significant	Not Significant	Not Significant	Not Significant
	95%	Not Significant	Not Significant	Not Significant	Not Significant
	90%	Not Significant	Not Significant	Not Significant	Not Significant

① never read any codes
② read codes before but not building codes
③ read building codes before
④ studied building codes before

representations perform on these significantly complex requirements still need to be investigated. In future work, the author plans to scale up the experiment to include other types of computable regulatory information representations and other types of visualization methods as well as incorporating the investigation of scalability of those methods to the much more complex regulatory requirements.

Acknowledgements
The author would like to acknowledge the contribution of Kim Ballard at Western Michigan University Writing Center for her writing suggestions on the first draft.

Funding
Not applicable.

Availability of data and materials
The dataset(s) supporting the conclusions of this article is(are) included within the article (and its additional file(s)).

Competing interests
The author declares that he has no competing interests.

Publisher's Note

Received: 3 November 2016 Accepted: 6 March 2017
Published online: 16 March 2017

References

Beach, T. H., Kasim, T., Li, H., Nisbet, N., & Rezgui, Y. (2013). Towards automated compliance checking in the construction industry. *LNCS, 8055*, 366–380. doi:10.1007/978-3-642-40285-2_32.

Beach, T. H., Rezgui, Y., Li, H., & Kasim, T. (2015). A rule-based semantic approach for automated regulatory compliance in the construction sector. *Expert Syst Appl, 42*(12), 5219–5231. doi:10.1016/j.eswa.2015.02.029.

Chomsky, N. (1956). Three models for the description of language. *IEEE Trans Inform Theor, 2*(3), 113–124. doi:10.1109/TIT.1956.1056813.

Delis, E. A., & Delis, A. (1995). Automatic fire-code checking using expert-system technology. *J Comput Civ Eng, 9*(2), 141–156. http://dx.doi.org/10.1061/(ASCE)0887-3801(1995)9:2(141)#sthash.VGCZZmk4.dpuf.

Dimyadi, J., Clifton, C., Spearpoint, M., Amor, R. (2014). Regulatory knowledge encoding guidelines for automated compliance audit of building engineering design. *Proc., Comput. Civ. Build. Eng.*, ASCE, Reston, VA, 536-543. http://dx.doi.org/10.1061/9780784413616.067#sthash.lTElCOvH.dpuf. Accessed 8 Feb 2017.

Dimyadi, J., Pauwels, P., & Amor, R. (2016). Modelling and accessing regulatory knowledge for computer-assisted compliance audit. *J Inf Technol Constr, 2016*, 317–336.

Eastman, C., Lee, J., Jeong, Y., & Lee, J. (2009). Automatic rule-based checking of building designs. *Autom Constr, 18*(8), 1011–1033. http://dx.doi.org/10.1016/j.autcon.2009.07.002.

Fenves, S.J., Gaylord, E.H., Goel, S.K. (1969). Decision table formulation of the 1969 AISC specification. *Civ. Eng. Studies: Structural Research Series, 347*. http://hdl.handle.net/2142/14275. Accessed 13 Oct 2016.

Garrett, J. H., & Fenves, S. J., Jr. (1987). A knowledge-based standard processor for structural component design. *Eng Comput, 2*(4), 219–238. doi:10.1007/BF01276414.

Garrett, J. H. J., Palmer, M. E., & Demir, S. (2014). Delivering the infrastructure for digital building regulations. *J Comput Civ Eng, 28*, 167–169. http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000369#sthash.g1Z7WQtn.dpuf.

Han, C. S., Kunz, J. C., Law, K. H. (1997). Making automated building code checking a reality. *Facility Manage. J., 1997*(September/October), 22-28. DOI: 10.1.1.60.8891

Hjelseth, E., & Nisbet, N. (2011). *Capturing normative constraints by use of the semantic mark-up RASE methodology* (Proc., CIB W78 2011). Rotterdam: Conseil International du Bâtiment (CIB). http://2011-cibw078-w102.cstb.fr/papers/Paper-45.pdf. Accessed 08 Feb 2017.

ICC (International Code Council). (2006). 2006 international codes. https://law.resource.org/pub/us/code/ibr/icc.ibc.2006.pdf. Accessed 31 Jan 2017.

Klein, D., & Manning, C. D. (2003). Accurate unlexicalized parsing. *Proc., 41st Meeting of the Association for Computational Linguistics*, Sapporo, Japan, 423-430. doi:10.3115/1075096.1075150.

Lau, G. T., & Law, K. (2004). An information infrastructure for comparing accessibility regulations and related information from multiple sources. *Proc., 10th Int. Conf. on Computational Civil and Building Engineering (ICCCBE)*, ISCCBE, Hong Kong, China. http://eil.stanford.edu/publications/gloria_lau/icccbe.pdf. Accessed 13 Oct 2016.

Pauwels, P., Van Deursenc, D., Verstraetena, R., De Rooc, J., De Meyera, R., Van de Wallec, R., & Van Campenhoutb, J. (2011). A semantic rule checking environment for building performance checking. *Autom Constr, 20*(5), 506–518. http://dx.doi.org/10.1016/j.autcon.2010.11.017.

Sterling, L., & Shapiro, E. (1986). *The art of Prolog: advanced programming techniques*. Cambridge: MIT Press. http://www.cuceinetwork.net/archivos/prolog/The_Art_of_Prolog.pdf.

Tan, X., Hammad, A., Fazio, P. (2007). Automated code compliance checking of building envelope performance. *Proc., 2007 ASCE Int. Workshop on Comput. Civ. Eng.*, ASCE, Reston, VA, 256-263. http://dx.doi.org/10.1061/40937(261)32. Accessed 13 Oct 2016.

Tan, X., Hammad, A., & Fazio, P. (2010). Automated code compliance checking for building envelope design. *J Comput Civ Eng, 24*(2), 203–211. http://dx.doi.org/10.1061/(ASCE)0887-3801(2010)24:2(203)#sthash.6DUU1389.dpuf.

Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference. *J Am Stat Assoc, 22*(158), 209–212. doi:10.2307/2276774.

Yurchyshyna, A., Faron-Zucker, C., Thanh, N.L., Zarli, A. (2008). Towards an ontology-enabled approach for modeling the process of conformity checking in construction. *Proc., CAiSE'08 Forum 20th Intl. Conf. Adv. Info. Sys. Eng.*, dblp Team, Germany, 21-24. http://ceur-ws.org/Vol-344/paper6.pdf. Accessed 13 Oct 2016.

Zhang, J., & El-Gohary, N. (2016a). Semantic-based logic representation and reasoning for automated regulatory compliance checking. *J Comput Civ Eng*. doi:10.1061/(ASCE)CP.1943-5487.0000583. http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000583.

Zhang, J., & El-Gohary, N. M. (2016b). Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking. *Autom Constr, 73*(2017), 45–57.

Zhou, N. (2014). B-Prolog user's manual (version 8.1): Prolog, agent, and constraint programming. Afany Software. http://www.picat-lang.org/bprolog/download/manual.pdf. Accessed 13 Oct 2016.