48

A Generalized Adaptive Framework for Automating Design Review Process: Technical Principles

Nawari O. Nawari

Abstract

Design review is the process of evaluating a design against its requirements to verify the performance of the design and identify issues before construction takes place. The cited methods for automating the design review process are either based on proprietary, domain-specific or hard-coded rule-based representations, which may be successful in their particular implementations, but they have the drawbacks of being costly to sustain, inflexible to change, lack generalized framework of rules and regulations modeling that can adapt to various engineering design realms, and thus don't support an open neutral standard. They are often referred to as 'Black Box' approaches. This study proposes a new comprehensive framework that minimizes the shortcomings of the cited methods. Building regulations for example, are legal documents written and authorized by people to be interpreted and applied by professionals. They are barely precise as formal logic. Engineers can read those documents and translate them into scientific notations and software applications. They can extract any information they need, reason about it, and apply it at various levels of precision. How these extraction and application are carried out is a critical component of automating design review process. The primary project goal is to address this issue by focusing on the development of a Generalized Adaptive Framework (GAF) for an open standard [based on Industry Foundation Classes (IFC)] that enables automating the design review processes to attain design efficiency and cost-effectiveness. The objectives of this paper include: (a) the conceptual and theoretical development of a framework that is adaptive to the target domain and supports an open standard for transforming the written design regulations and rules into a computable model, and (b) defining the different modules needed for the automation of the design review process.

Keywords

Design review • BIM • IFC • Generalized adaptive framework • GAF • Automation of design review

48.1 Introduction

Design review is the process of evaluating a design against its requirements to verify the performance of the design and identify issues before construction takes place. Design requirements typically take the form of written texts, tables, and equations. These rules, in general, have lawful status. However, the reasoning and analytic ability of the human brain is unlike algorithms implemented in computer systems. Thus, the automation of this process poses a real challenge to the engineering domains. For example, how can the interpretation of these rules in a computer interpretable format be performed, in a manner that the implementation can be validated as consistent with the written design requirements?

One of the first effort to automate design compliance is demonstrated by the work of Fenv [4], when he investigated the application of decision tables to represent the AISC (American Institute of Steel Construction) standard specifications. He made the remark that decision tables, If-Then-novel programming, and program documentation technique, could be

N. O. Nawari (✉)
University of Florida, College of Design, Construction and Planning, School of Architecture, Gainesville, FL 32611-5702, USA
e-mail: nnawari@ufl.edu

I. Mutis and T. Hartmann (eds.), *Advances in Informatics and Computing in Civil and Construction Engineering*,
https://doi.org/10.1007/978-3-030-00220-6_48

utilized to represent design standard provisions in a precise and unambiguous form. The concept was set to practical application in the [2] AISC Specification. It was described as a set of interrelated decision tables. Later, other researchers tried to build on Fenv work such as Lopez et al., who implemented the SICAD (Standards Interface for Computer Aided Design system [7, 8]. The SICAD system was a software prototype developed to demonstrate the checking of designed components as described in application program databases for conformance with design standards. The SICAD concepts were in production use in the AASHTO Bridge Design System [1].

Other recent research efforts have proposed to manually extract and translate written regulatory rules directly into computer code [10, 11]. In such approach, formalized regulatory information in the form of codified rules is then accessed internally by the software code of the compliance examination application. Further recent efforts on automating design review process are focused more on the concept of regulatory text mining and semantic web approach to creating a computable representation [5, 13, 16]. Other investigations are centered chiefly on the studies of automated or semi-automated extraction of information from regulatory texts into rules using Artificial Intelligence (AI) and Natural Language Processing (NLP) [15–18]. There are also some research efforts that target specific domains and are bounded by the particular area of computerized code compliance verification system [6, 9].

48.2 Statement of Contribution

The current methods for automated rules compliance auditing are either based on proprietary, domain-specific or hard-coded rule-based representations, which may be successful in their particular implementations, but they have numerous disadvantages. These methods are costly to maintain, inflexible to change, lack a generalized framework of rules and regulations modeling that can adapt to various domains, and thus don't support an open neutral standard. They are often referred to as 'Black Box' or 'Gray Box' approaches.

The proposed GAF for Automated Design Review (ADR) process involves the development of the computable representation of code regulations and the mechanisms for exchanging data between the different components of the framework and the Building Information Model (BIM) data. This paper focuses on the basic principles of the GAF. This includes the general concepts and the interpretation process where the semantic structure of the building codes regulations is manipulated and transformed into object rules or parametric models using Transformation Reasoning Algorithm (TRA) and the development of Model View Definition (MVD) that would lead to Industry Foundation Classes (IFC) data schema. This requires the development of taxonomy, knowledge conceptualization, modification, integration, and decomposition of the design regulations and rules.

48.3 Goals and Objectives

The project primary goal is to address the issue of ADR process by concentrating on the development of a Generalized Adaptive Framework (GAF) for an open standard [based on Industry Foundation Classes (IFC)] that establishes the foundation for automating building design review processes.

The objectives of this research include the theoretical development of a framework that is adaptive to the target domain and supports an open standard for transmuting the written regulations and rules into a computable model, defining the different components needed for the automation of the design review process, and creating an algorithm for the data exchanges between the components of the framework to execute the virtual review process of a building design.

48.4 Methodology

48.4.1 Theoretical Framework

The GAF for ADR process involves the development of a computable representation of building regulations and the mechanisms for exchanging information between regulations and the Building Information Model (BIM) data. Figure 48.1 delineates an overview of the components and phases of the proposed framework. An overview of these main phases is given below:

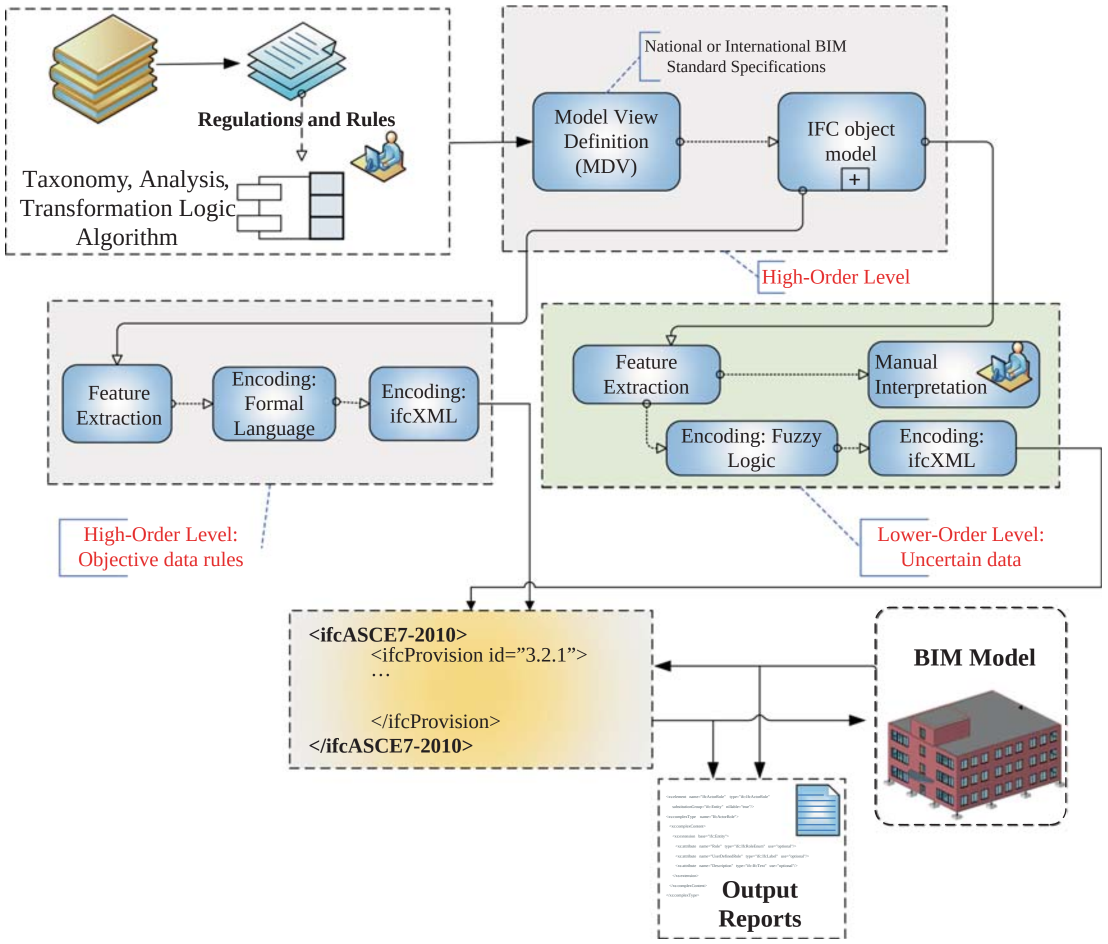
Regulations and Rules
Taxonomy, Analysis, Transformation Logic Algorithm
National or International BIM Standard Specifications
Model View Definition (MDV)
IFC object model
High-Order Level
Feature Extraction
Encoding: Formal Language
Encoding: ifcXML
High-Order Level: Objective data rules
Feature Extraction
Manual Interpretation
Encoding: Fuzzy Logic
Encoding: ifcXML
Lower-Order Level: Uncertain data
<ifcASCE7-2010>
<ifcProvision id="3.2.1">
…
</ifcProvision>
</ifcASCE7-2010>
BIM Model
Output Reports

Fig. 48.1 The generalized adaptive framework (GAF) for automated buildings design review

(A) High-Order Level I: Taxonomy formation, knowledge conceptualization, modification, integration, and decomposition of the design regulations and rules. This encompasses data analysis, partitioning and classification of regulatory text into broad categories. This is shown in the first step of Fig. 48.1. This phase is referred to as Transformation Reasoning Algorithm (TRA).

(B) High-Order Level II: Requires the Development Model View Definition (MVD), leading to Industry Foundation Classes (IFC) schema of the information obtained from phase A. The final data format is ifcXML representation. IfcXML is defined as the XML equivalent to the EXPRESS based specification of the IFC data model.

(C) Higher-order level III: The development feature extraction algorithm for all objective data (unambiguous data) leading to full translation into the object-based model. This extraction and transformation will lead to ifcXML data object model (see Fig. 48.1).

(D) Lower-order level: Necessitates feature extraction of uncertain data, then employing an algorithm for partial translation using fuzzy logic and approximate reasoning methods. Fuzzy logic provides a means of expressing linguistic rules in such a form that they can be combined into a coherent reasoning model. Such a model consists of three main parts: (i) fuzzification, (ii) inference engine (fuzzy rule base), and (iii) defuzzification (the process of transforming the aggregation result into a crisp output). The resulting data model from this phase is projected to be in ifcXML.

(E) The execution phase, which carries out the communications between the different layers of development. This encompasses the design of an algorithm linking the data from (B), (C), (D) and the BIM model. This phase will result in generating reports of the ADR outputs.

This paper centers primarily on the TRA. Subsequent publications will address the other phases of the GAF along with the criteria used to evaluate this part of the framework comprise interpretation accuracy, dealing with complexity, and the degree of completeness.

48.4.2 Transformation Reasoning Algorithm (TRA)

The TRA introduces the taxonomy for the building regulations knowledge followed by conceptualization process. Subsequently, knowledge created will be transformed into a new formalized form to deduce various facts to carry out automated reasoning. For example, building regulation or provision X_i will be transformed into a concept Y_i using TRA principles. TRA taxonomy defines the following major concepts:

Content (C_i)
Provisory (P_i)
Dependent (D_i)
Ambiguous (A_i)

Contents (C_i … C_n) are the sections of the building codes and regulations that cannot be transformed into object rules. These clauses are usually devoted for definitions, such as the definition of types of loads, firewall, fire rate, smoke evacuation, high-rise building etc. For instance, the live load is defined by the ASCE7-10 standard as: "*A load produced by the use and occupancy of the building or other structure that does not include construction or environmental loads, such as wind load, snow load, rain load, earthquake load, flood load, or dead load*".

Provisory (P_i … P_n) are clauses of the regulations that can be transformed from the textual format into a set of object rules. Examples of such clauses are prevalent and typical structures include rules with specific values such as those given in tables or equations in the building codes and standards.

Dependent (D_i … D_n) clauses specify that one clause is reliant on one or more other provisions. This means that some requirements are only appropriate for a specific condition when other clauses are satisfied. These clauses generally contain provisory clauses (P_i … P_n) and are often challenging to transform into sets of immediate object rules. These sections quite often may require manual checking for compliance. For instance, in Florida Building Code [3], section 503.1 regarding building height and area, states that "*The building height and area shall not exceed the limits specified in Table 503 based on the type of construction as determined by Section 602 and the occupancies as determined by Section 302 except as modified hereafter. Each portion of a build separated by one or more firewalls complying with Section 706 shall be considered to be a separate building.*"

Ambiguous (A_i … A_n) clauses are the vague or inexact provisions that would need an expert judgment to be evaluated. They usually include words such as: approximately, about, relatively, close to, far from, maybe, etc. An example of such provision is the footnote of the design lateral soil pressure for the clause given in ASCE 7–10: "*For* ***relatively*** *rigid walls, as when braced by floors, the design lateral soil load shall be increased for sand and gravel type soils to 60 psf* (9.43 kPa) *per foot (meter) of depth. Basement walls extending not more than* 8 ft (2.44 m) *below grade and supporting* ***light*** *floor systems are not considered as being* ***relatively*** *rigid walls*." This concept covers all regulations that are not capable of being computerized and some of them may have to be rewritten to enable implementation in automated code compliance auditing environment. Interpretation and revising both must adhere to understanding terms from both the legal and construction perspectives.

These concepts can then be modified, integrated or decomposed to enable computable representation of building regulations and standards. Knowledge concepts X_i can be transformed or combined with another concept into Y_i and then Y_i can be transmuted into Z_i to enable efficient computable representation. Thus, the TRA is defined as the conceptualization of knowledge representation by mapping building code and regulations into sets of object rules. Figure 48.2 is a pictorial description of the TRA for building regulations.

Building regulations will be classified using the taxonomy defined earlier and can also be translated into conceptual representations that closely approximate the meaning of the building code provision. These figurative structures can then be transformed and manipulated to deduce various facts and rules to carry out automatic compliance validation.

The TRA is based partially on first-order logic calculus. For example, the building code provision that says, "only Professional Engineer(PE) must approve structural design" can be stated as following using TLA:

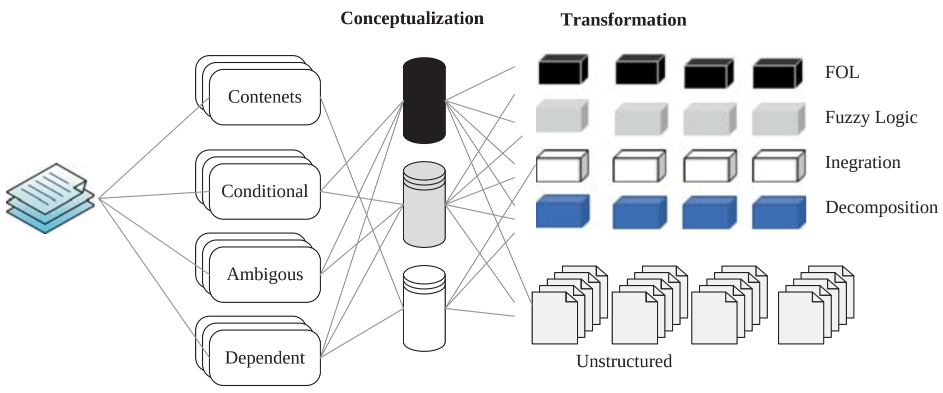

Fig. 48.2 The transformation reasoning algorithm (TRA)

Fig. 48.3 **a** Section 48.304 of Florida building code—residential [3]. **b** Florida building code—residential [3]

(a)

SECTION R304
MINIMUM ROOM AREAS

R304.1 Minimum area.
Habitable rooms shall have a floor area of not less than 70 square feet (6.5 m^2).
Exception: Kitchens.
R304.2 Minimum dimensions.
Habitable rooms shall be not less than 7 feet (2134 mm) in any horizontal dimension.
Exception: Kitchens.
R304.3 Height effect on room area.
Portions of a room with a sloping ceiling measuring less than 5 feet (1524 mm) or a furred ceiling measuring less than 7 feet (2134 mm) from the finished floor to the finished ceiling shall not be considered as contributing to the minimum required habitable area for that room.

(b)

SECTION R305
CEILING HEIGHT

R305.1 Minimum height.

Habitable space, hallways and portions of *basements* containing these spaces shall have a ceiling height of not less than 7 feet (2134 mm). Bathrooms, toilet rooms and laundry rooms shall have a ceiling height of not less than 6 feet 8 inches (2032 mm).

Exceptions:

1. For rooms with sloped ceilings, the required floor area of the room shall have a ceiling height of not less than 5 feet (1524 mm) and not less than 50 percent of the required floor area shall have a ceiling height of not less than 7 feet (2134 mm).
2. The ceiling height above bathroom and toilet room fixtures shall be such that the fixture is capable of being used for its intended purpose. A shower or tub equipped with a showerhead shall have a ceiling height of not less than 6 feet 8 inches (2032 mm) above an area of not less than 30 inches (762 mm) by 30 inches (762 mm) at the showerhead.
3. Beams, girders, ducts or other obstructions in *basements* containing *habitable space* shall be permitted to project to within 6 feet 4 inches (1931 mm) of the finished floor.

R305.1.1 Basements.

Portions of *basements* that do not contain *habitable space* or hallways shall have a ceiling height of not less than 6 feet 8 inches (2032 mm).

Exception: At beams, girders, ducts or other obstructions, the ceiling height shall be not less than 6 feet 4 inches (1931 mm) from the finished floor.

Table 48.1 Syntax of transformation reasoning algorithm (TRA)

Symbol	Definition
:: =	Is defined as
$\wedge$	Conjunction
$\vee$	Disjunction
$\subset$	Subset of
$\neg$	Negation
$\forall$	Universal Quantifier
$\exists$	Existential Quantifier
$\in$	Belongs to
$\rightarrow$	Implication
$\leftrightarrow$	Biconditional
$\Rightarrow$	Transform into
Constant	String starting with an uppercase letter
Variable	String starting with a lowercase letter
Pred (arg1, arg2, …)	Predicate
Fun (arg1, arg2, …)	Function
Pred1 (arg1, arg2, …) $\wedge$ Pred2 (arg1, arg2, …) $\vee$ …	Rule

$$Prov(PE) \in Conditional; \forall \mathrm{x}(PE(\mathrm{x}) \rightarrow Permitted(\mathrm{x}, \text{approve design}))$$

$$\forall \mathrm{x}\,(\neg PE(\mathrm{x})\,\forall \mathrm{x}\,\neg\, Permitted(\mathrm{x},\ \text{approve design}))$$

This TRA algorithm can be illustrated further by considering Florida Building Code—Residential 2017 [3]. Figure 48.3 depicts various section from the [3]-Residential (Table 48.1).

The provision shown in Fig. 48.3a can be transformed into computable representation using the TRA as follows:

Let REG_i = "Section R304"; Where i varies from 1 to n number of code provisions. Then we have

$$REG_i \in \mathrm{P_i} \Rightarrow \mathrm{Y_i} \Rightarrow \mathrm{X_i} \tag{48.1}$$

where,

The subscript i stands for the counts of the code sections being processed and varies from 1 to n sections.

$\mathrm{P_i}$ designates that this is a provisory clause, and describes the minimum room area ($\mathrm{Y_i}$) which is given by $\mathrm{X_i}$ that expresses the various Rules describing $\mathrm{Y_i}$:

$$\mathrm{X_i} = \{\mathrm{R_1}, \mathrm{R_2}, \ldots, \mathrm{R_m}\} \tag{48.2}$$

where, $\mathrm{R_1}$, $\mathrm{R_2}$, … $\mathrm{R_m}$ are the rules defining $\mathrm{X_i}$.

$$\text{Let}\, \mathrm{Z_{1j}} = \left\{\mathrm{z_{11}} \ldots \mathrm{z_{1q}}\right\} \tag{48.3}$$

$$\mathrm{z} = \text{IfcSpace}; \mathrm{z_{11}} = \text{``R304.1''}; \text{and}; \mathrm{z_{12}} ::= \text{Floor area} > \ = 70\,\mathrm{ft^2}\left(6.5\mathrm{m^2}\right) \tag{48.4}$$

$$R_2{:}\forall\, \mathrm{z}\,(REG_i(\mathrm{z})) \rightarrow Minimum\ Area\left(\mathrm{z}, \mathrm{Z_{1j}}\right) \wedge \neg\, Space\ Name(z, KITCHEN) \tag{48.5}$$

$$\mathrm{Z_{2j}} = \left\{\mathrm{z_{21}} \ldots \mathrm{z_{2q}}\right\} \tag{48.6}$$

$$\mathrm{z_{21}} = \text{``R304.2''};\ \text{and}\, \mathrm{z_{22}} ::= \text{least horizontal dimension of any habitable room} > \ = 7\,\mathrm{ft}(2.134\,\mathrm{m})$$

$$R_3{:}\forall\, \mathrm{z}\,(REG_i(\mathrm{z}) \rightarrow Minimum\ Dimension\left(\mathrm{z}, \mathrm{Z_{2j}}\right) \wedge \neg\, Space\ Name(z, KITCHEN) \tag{48.7}$$

$$\mathrm{Z_{3j}} = \left\{\mathrm{z_{31}} \ldots \mathrm{z_{3q}}\right\} \tag{48.8}$$

$Z_{31} = \text{``R304.3''};\ z_{32} ::= \text{Ceiling height} > 5\,\text{ft for sloped ceiling; and}\ z_{33} ::= \text{Ceiling height} > 7\,\text{ft for furred ceiling}$

$$R_4{:}\forall z(REG_i(z) \rightarrow \mathit{Ceiling\ Height\ Limitation}(z, Z_{3j})) \tag{48.9}$$

$$X_i = \{R_1 \wedge R_2 \wedge R_3 \wedge R_4\} \tag{48.10}$$

Equation 48.10 represents the knowledge transformation process to generate a computable model for the code specifications expressed in FBC 2017 [3] Residential, section R304. This knowledge representation can then be mapped into IFC schema as given by [12].

As a further example, consider section R305 in Fig. 48.3b. In this example, the variables used above will have the following values:

REG_2 = "Section R305"; then we have

$$REG_2 \in (P_2) \Rightarrow Y_2 \Rightarrow X_2 \tag{48.11}$$

where,

P_2 designates that this is a provisory clause, and describes the minimum ceiling height (Y_2) which is given by X_2 that expresses the various Rules describing Y_2:

$$X_2 = \{R_1, R_2, \ldots, R_m\} \tag{48.12}$$

where, R_1, R_2, …, R_m are the rules defining X_i.

$$\text{Let}\ Z_{1j} = \{z_{21}, \ldots, z_{2q}\} \tag{48.13}$$

$$z = \text{IfcSpace};\ z_{11} = \text{``R305.1''} \tag{48.14}$$

$$z_{12} ::= \geq 7\,\text{ft};\ z_{13} ::= \geq 6.667\,\text{ft};\ z_{14} ::= \geq 6.333\,\text{ft};\ z_{15} \geq 5\,\text{ft} \tag{48.15}$$

$$\begin{gathered} R_2{:}\,\forall z\,(REG_2(z) \rightarrow \mathit{Ceiling\ Height}(z, z_{12}) \wedge \mathit{Habitable\ Space}(z) \wedge \neg\, \mathit{Space\ Name}(z, \mathit{BATHROOM}) \wedge \neg \\ \mathit{Space\ Name}(z, \mathit{TOILETROOM}) \wedge \neg\, \mathit{Space\ Name}(z, \mathit{LAUNDRYROOM}) \wedge \neg\, \mathit{Sloped\ Ceiling}(z) \end{gathered} \tag{48.16}$$

$$\begin{gathered} R_3{:}\,\forall z\,(REG_2(z) \rightarrow \mathit{Ceiling\ Height}(z, z_{13}) \wedge \mathit{Habitable\ Space}(z) \wedge (\mathit{Space\ Name}(z, \mathit{BATHROOM}) \vee \\ \mathit{Space\ Name}(z, \mathit{TOILETROOM}) \vee \mathit{Space\ Name}(z, \mathit{LAUNDRYROOM})) \wedge \neg\, \mathit{Sloped\ Ceiling}(z) \end{gathered} \tag{48.17}$$

$$A = \mathit{Floor\ Area}(z);\ z_{16} ::= \geq 0.5 * A \tag{48.18}$$

$$R_4{:}\,\forall z\,(REG_2(z) \rightarrow \mathit{Ceiling\ Height}(z, z_{15}) \wedge \mathit{Sloped\ Ceiling}(z, z_{16}) \tag{48.19}$$

$$\text{Let}\ Z_{2j} = \{z_{21}, \ldots, z_{2q}\} \tag{48.20}$$

$$z = \text{IfcSpace};\ z_{21} = \text{``R305.1.1''} \tag{48.21}$$

$$z_{23} ::= \geq 6.667\,\text{ft};\ z_{24} ::= \geq 6.333\,\text{ft};\ z_{25} = \text{IfcBeam};\ z_{26} = \text{IfcDuct}; \tag{48.22}$$

$$\begin{gathered} R_5{:}\forall z\,(REG_2(z) \rightarrow \mathit{Ceiling\ Height}(z,\ z_{23}) \wedge \neg\,(\mathit{At\ Beam\ Elevation}(z, z_{25}) \\ \vee\ \mathit{At\ Duct\ Elevations}(z, z_{26})) \wedge \neg \mathit{Habitable\ Space}(z) \end{gathered} \tag{48.23}$$

$$R_6{:}\forall z\,(REG_2(z) \rightarrow \mathit{Ceiling\ Height}(z,\ z_{24}) \wedge (\mathit{At\ Beam\ Elevation}(z, z_{25}) \vee \mathit{At\ Duct\ Elevations}(z, z_{26}) \tag{48.24}$$

$$X_2 = \{R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5 \wedge R_6\} \tag{48.25}$$

An example of inexact code provision can be seen in section R322.1 of [3]-Residential (Fig. 48.4). In this provision, the regulations states: "*Buildings and structures constructed in whole or in part in flood hazard areas, including A or V Zones and Coastal A Zones, as established in Table R301.2(1), and substantial improvement and restoration of substantial damage of buildings and structures in flood hazard areas, shall be designed and constructed in accordance with the*

Fig. 48.4 Part of Florida building code—residential [3]

SECTION R322
FLOOD-RESISTANT CONSTRUCTION

R322.1General.
Buildings and structures constructed in whole or in part in flood hazard areas, including A or V Zones and Coastal A Zones, as established in Table R301.2(1), and substantial improvement and restoration of substantial damage of buildings and structures in flood hazard areas, shall be designed and constructed in accordance with the provisions contained in this section. Buildings and structures that are located in more than one flood hazard area shall comply with the provisions associated with the most restrictive flood hazard area. Buildings and structures located in whole or in part in identified floodways shall be designed and constructed in accordance with ASCE 24.

provisions contained in this section." The word substantial is never defined precisely. Using the TRA, then we have, REG_3 = "Section R322";

$$REG_3 \in (C_1 \wedge A_1) \Rightarrow Y_3 \Rightarrow X_3 \tag{48.25}$$

$(C_1 \wedge A_1)$ designates that this is a content clause with ambiguous statements describing flood resistance construction (Y_3) which is given by X_3 that defines the various conditions unfolding Y_3.

$$X_3 = \{R_1, R_2, \ldots, R_m\} \tag{48.26}$$

where, R_1, R_2, …, R_m are the rules defining X_3.

$$\text{Let } Z_{3j} = \{z_{31}, \ldots, z_{3q}\} \tag{48.27}$$

z = IfcBuilding; z_{31} = "[3]—R322"; z_{32} = "ASCE 24"

$$R_1{:}\forall z\,(\mathit{In\ Flood\ Zone}(z) \rightarrow \mathit{Required\ Provision}(z, z_{31})) \tag{48.28}$$

$$R_2{:}\forall z\,(\mathit{In\ Flood\ Ways}(z) \rightarrow \mathit{Required\ Provision}(z, z_{32})) \tag{48.29}$$

As for the conceptualization of the term "substantial damage", a fuzzy logic and predicates will be utilized to transform concept into a rule representation. A fuzzy set is defined as [14]: ***A*** is a fuzzy subset of the universe of discourse ***U***, is characterized by a membership function $\boldsymbol{\mu_A}$: *U* ➔ *[0…1]* which associates with each element ***u*** of ***U*** a number $\boldsymbol{\mu_A}$ *(u)* in the interval [0,1]. This definition can be employed to define fuzzy predicate (first-order predicate.) The truth-value of any proposition can be evaluated as the degree of membership of the responding fuzzy relation. Thus, a fuzzy predicate can be considered as the membership function of a fuzzy relation over individual variables' universe of discourse. Each fuzzy predicate represents a concept in the TRA.

For example, the building damage stated in section R322 can be represented as a fuzzy variable having values depicted in Fig. 48.5. These include small damage, medium damage and substantial damage. These definitions need to be determined from experience or local guidelines.

Now, let z_{32} = a fuzzy variable defined by

$$\left.\begin{aligned} \boldsymbol{\mu_A}(u) &= 0 && 25\% < = u > \ = 0 \\ &= (1/15)u - 25/15 && 25\% < u > \ = 0 \\ &= 1 && u > 40 \end{aligned}\right\} \tag{48.30}$$

where, $0 \le \mu_A(u) \le 1$

Now, section R322 of [3]-Residential is transformed into the following rule:

$$R_3{:}\forall z\,(\mathit{In\ Flood\ Zone}(z) \wedge \mathit{Damage}(z, z_{32}) \rightarrow \mathit{Required\ Provision}(z, z_{31})) \tag{48.31}$$

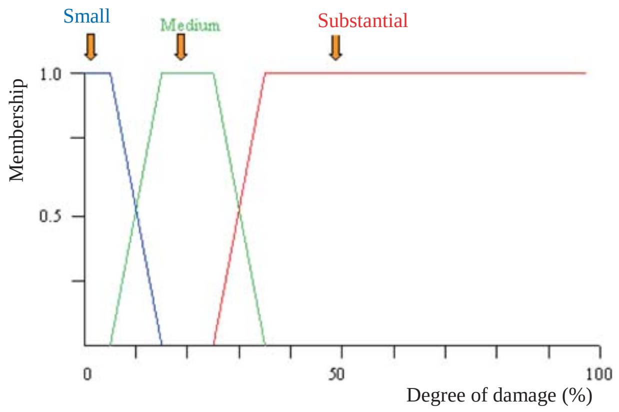

Fig. 48.5 The concept of fuzzy transformation of building damages

Building codes do have quite often such imprecise terms to describe specific conditions. Future research will study these terms and their transformation using fuzzy predicates.

48.5 Conclusions

The current design review process lacks a generalized framework of rules and regulations modeling that can adjust to several engineering design domains and support an open neutral standard for automating the review process. This paper proposes a framework that is adaptive to the target domain and endorses a neutral standard for transforming the written design regulations and rules into a computable model to enable ADR process. The paper outlines and defines the different components of the proposed framework and their relationships. This paper centers on the higher order level I of the framework which is denoted by the Transformation Reasoning Algorithm (TRA). The TRA introduces the taxonomy for the building regulations knowledge along with the conceptualization and transformation processes. Subsequently, knowledge created is a new formalized object representation that model objective and ambiguous building regulations and can deduce various facts to carry out automated reasoning. This approach minimizes the shortcomings of the cited methods by transforming objective and vague data of building code into a concise formal representation that can be mapped into IFC data schema. Future studies will investigate the other levels of the GAF and how they would influence the broader impacts of this research project. These impacts include the enormous benefits to the AEC industry depicted in the consistency of the interpretation of regulatory provisions, the ability to self-check required aspects before bidding, saving time and resources during design review, optimum design, the quicker turnaround in feedback, and faster approvals for construction permits by building authorities.

References

1. AASHTO: AASHTO Guide for Design of Pavement Structures, 4th edn. American Association of State Highway and Transportation Officials, Washington, DC (1998)
2. AISC (American Institute of Steel Construction): Specification for the Design, Fabrication and Erection of Structural Steel for Building (1969)
3. FBC: Florida Building Code, 6th edn. International Code Council (ICC), Washington, DC (2017)
4. Fenves, S.J.: Tabular decision logic for structural design. J. Struct. Eng, ASCE **9–92**, 473–490 (1966)
5. Kiyavitskaya, N., Zeni, N., Mich, L., Cordy, J.R., Mylopoulos, J.: Text mining through semi automatic semantic annotation. In: Proc. of PAKM'06, vol 4333 LNCS, pp 143–154. Springer-Verlag (2006)
6. Lee, J.: Building environment rule and analysis (BERA) language. PhD Thesis, Georgia Institute of Technology (2011)
7. Lopez, L.A., Elam, S., Reed, K.: Software concept for checking engineering designs for conformance with codes and standards. Eng. Comput. **5**, 63–78 (1989)
8. Lopez, L.A., Wright, R.N.: Mapping Principles for the Standards interface for Computer Aided Design, NBSIR 85-3115. National Bureau of Standards, Gaithersburg, MD (1985)
9. Mazairac, W., Beetz, J.: BIMQL—An open query language for building information models. Adv. Eng. Inform. **27**(2013), 444–456 (2013)
10. Nawari, N.: Automating codes conformance. J. Archit. Eng. ASCE **18**(4), 315–323, (2012a)
11. Nawari, N.O.: Automated code checking in BIM environment. In: Proceedings of the 14th International Conference on Computing in Civil and Building Engineering, Moscow, Russia, 27–29 June, (2012b)

12. Nawari, O.N.: Building information modeling: automated code checking and compliance processes, CRC Press, ISBN-10:1498785336. (2018)
13. Pauwels, P., Van Deursen, D., Verstraeten, R., De Roo, J., De Meyer, R., Van de Walle, R., Van Campenhout, J.: A semantic rule checking environment for building performance checking. Autom. Constr. **20**(5), 506–518 (2011)
14. Zadeh, L.A.: Fuzzy sets. Information and control, vol. 8, pp. 338–353. (1965)
15. Zhang, J., El-Gohary, N.: Extraction of construction regulatory requirements from textual documents using natural language processing techniques. J. Comput. Civil Eng. ASCE **2012**, 453–460 (2012)
16. Zhang, J., El-Gohary, N.: Semantic NLP-Based information extraction from construction regulatory documents for automated compliance checking. J. Comput. Civil Eng. ASCE **30**, 04015014 (2013)
17. Zhang, J., El-Gohary, N.: Automated information transformation for automated regulatory compliance checking in construction. J. Comput. Civil Eng. ASCE **29**, B4015001 (2015)
18. Zhang, J., El-Gohary, N.: Semantic-based logic representation and reasoning for automated regulatory compliance checking. J. Comput. Civil Eng. ASCE **31**, 04016037 (2016)