Practical Approaches for Computable Building Codes

Nawari O. Nawari nnawari@ufl.edu
University of Florida, College of Design, Construction & Planning, USA.

Adel Alsaffar, alsaffaradel@gmail.com
Kuwait University, Kuwait.

Abstract
Currently, the complexity of building regulatory and standards increases rapidly with the acquisition of new knowledge in the design and construction domain. The need for computable representation of the building codes and regulations for automating the code checking process is becoming ever more important. Within the framework of Building Information Modeling (BIM) workflow, model checking against building codes and standards is generally needed to be an automatic or semi-automatic process. These checking mechanisms generally do not modify a building design, but rather evaluates a design on the basis of the configuration of objects, their relations and attributes. This paper presents an overview of the main existing methods for computerizing building codes and regulations. It reviews main concepts for these methods including knowledge representation, reasoning procedures, and knowledge acquisition. Additionally, this research explores the capabilities of the cited approaches in terms of strengths and practical limitations for creating computable building regulatory.

Keywords: Building Information Modeling, BIM, Automated Code Checking, Building Code Computerization.

1 Introduction
Building rules and regulations are written by professionals to be read and applied by people. Since the reasoning and interpretation ability of the human brain is unlike anything implemented in computer systems, the computerization of this process poses a real challenge to the AEC industry (Nawari, 2012). Providentially, new advancements in the Artificial Intelligence (AI) research and Building Information Modeling (BIM) can offer effective solutions to resolve these problems.

AEC building standards and regulations commonly seek to organize, categorise, label, and define the rules, actions, and patterns of the build environment to attain safety against any kind of failure, efficiency and overall economy. However, their best-laid plans are overwhelmed by the inevitable change, growth, innovation, progress, evolution, diversity, and entropy (Nawari, 2012). Quiet often regulations can amend provisions and interpretive manuals which normally lead to massive volumes of semi-structured documents that amend, complement and potentially conflict with one another. These issues, which represent complications for both young engineers as well as experienced professionals, are also far more disorderly for the fragile traditional knowledge bases in computer systems. Even though precise definitions and specifications are essential for solving encoding building regulations, many code provisions aren't well defined and highly subjective. Furthermore, some code provisions are characterized by continuous progressions and open-ended range of exceptions that make it difficult to give complete, exact definitions for any concepts that are learned through experience.

The question of computerizing rules and provisions checking of building codes and standard has interested many researchers and practioners since the mid-sixties. More than 300 relevant research studies traversing over 40 years can been identified (Dimyadi et al, 2013). Some of the more significant primary studies identified and their influences have been presented in a timeline (Figure 1). A more comprehensive survey of previous developments for digital representation of design

codes and automated rule checking can be found in the work by Fenves et al. (1995); Eastman et al. (2009); Nawari (2012), and Dimyati et al.(2013).
Recent efforts on encoding building code provisions are focused more on the concept of marking-up regulatory texts to create a computable representation (Hjelseth and Nisbet, 2011; See, 2008). Other research investigation are centered primarily on the inquiries of automated or semi-automated extraction of information from regulatory texts into rules and other computable objects (Hjelseth, 2012; Kiyavitskaya et al, 2007; Zhang and El-Gohary, 2011, 2012, 2013). In the next sections, description of these methods will be given along with strengths and practical limitations.

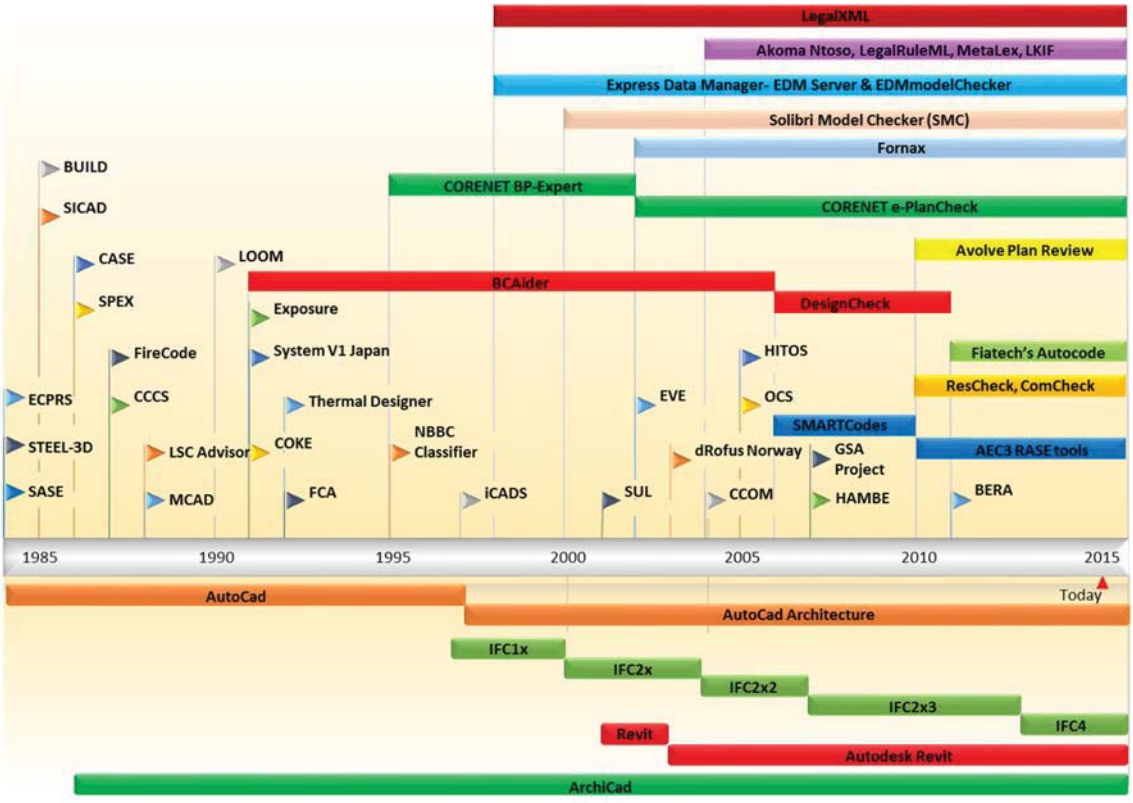

Figure 1 Timeline of related research to Automated Code Checking (modified from Dimyati et al, 2013).

2 General Domain Knowledge Digital Representations

Building codes and regulations are documents written by people to be interpreted and applied by people. They are seldom as precise as formal logic. That flexibility is essential for a system of knowledge acquisition. Yet professionals can read those documents and convert them into formal scientific notations and software applications. They can extract any kind of information they need, reason about it, and apply it at various levels of precision. The methods by which these extraction and application are conducted is an area that researchers have tried to automated or semi-automated for many decades.

The main approach to achieving computerized execution of regulations and guidelines in building informatics is to use modeling languages that can create computer-interpretable regulation and guideline provisions, i.e. provisions that a computer can run automatically. Recent researchers have investigated different modeling methods for the creation of computer-interpretable regulations and guidelines. The following is a summary of the two major modeling approaches for building codes and regulations:
Page layout

2.1 Artificial Intelligence Methods

In order to Methods based on AI (Artificial Intelligence) utilize different Natural Language Processing (NLP) algorithms. They are focused on the area of human–computer interaction. Many challenges in NLP approaches involve natural language understanding and extracting rules from the building codes text. They aim to enable computers to derive meaning from textual format of building regulations and to generate logical rules for further processing.

Most of the NLP methods extract data and metadata creating an array of technical critical tasks. These include for instance: (a) Text analytics; (b) Content monetization; (c) automatic classification of content for regulatory compliance checking; and (d) text mining. Content extraction is an important area of NLP. These methods can be based on the syntactic and/or semantic characteristics of the main content. They generally utilize the concept of "Entity". An Entity can be defined as the result of the extraction of proper nouns from text, such as people, places, building elements such as beams, columns, walls, windows, doors, … etc. Each extracted named entity is classified, tagged and assigned a sentiment score, which gives meaning and context to each entity.

The general approach using NLP algorithms starts with developing ontology to capture domain knowledge to enhance interpretability and understanding of domain particular text content (Zhang et al. 2013). The NLP methods can be rule-based approach or machine learning-based approach. Rule-based NLP uses manually-coded rules for meaning extraction and processing. These rules are then iteratively built and refined to enhance the accuracy of semantics processing. The machine learning-based NLP utilizes algorithms for training text processing models based on the content of a given training text. Rule-based NLP methods tend to yield better text processing performance (in terms of precision and recall), but requires more human intervention.

Recent research efforts used intermediate processing step before the final extraction of rules from the source text (e.g. Zhang et al, (2013)). This step generally results in 3-tuple semantic element in the form of: <Subject, Attribute, Value>. Such sematic element is characterized by: (i) an ontology concept; (ii) an ontology relation; and (iii) a deontic operator indicator.

The process often has many steps before the final encoding results of the regulatory text. These include, preprocessing phase, feature generation, analysis of the extracted information, extraction rules, and the final execution of the encoding instances. In the preprocessing phase of the original content underrun tokenization, sentence splitting, de-hyphenation, and morphological analysis. In the second step a set of semantic and syntactic features that describe the content are derived based on domain specific ontology (Zhang et al, 2013). An example illustrating these steps is depicted in table 1.

Table 1 Encoding of part of the International Building Code (IBC 2006) using NLP method.

Information Tuple	**Extracted Semantic Information Element**	**Formal Logic Representation**
Subject	Airspace	∀ (a,i, r, s) ((airspace(a) ^ insulation(i) ^ roof_sheathing(r) ^between(a, i, r) ^ has(a, s)) → O (greater_than_or_equal(s, quantity(1, inch)))
Subject Restriction	Relation(between, insulation, roof_sheathing)	
Compliance Checking Attribute	N/A	
Deontic Operator Indicator	Obligation	
Quantitative Relation	Provide	
Comparative Relation	greater_than_or-equal	
Quantity Value	1	
Quantity Unit/Reference	Inch	
Quantity Restriction	N/A	

Some of the weaknesses of these modeling representations include the needs for human interpretation, in order to make them: (i) accessible electronically; (ii) structured and understandable by machines; (iii) represented in a standard format; (iv) interoperable; (v) explicit; and (vi) transparent. Furthermore, considering the fact that code regulations are complex and often highly subjective in nature, thus these modeling approaches do require building regulation officials to be involved in the process to ensure the correct interpretations of the encoding.

Nawari (2012a) proposed First-Order Logic (FOL) as a modeling language for encoding regulatory text. The method is dedicated to the knowledge extraction approach that aims to directly transform text regulations into formal languages. The approach has the advantage of simplicity but has the limitation of manual work as well as the technical knowledge to convert the regulatory texts into a set of rules.

First-order logic is a formal logical system which includes domains of discourses over which quantifiers range. Like natural language, it assumes that the world contains: (i) Objects: foundations, beams, columns, trusses, colors, etc.; (ii) Relations: deep, round, square, bigger than, part of, comes between, …etc; (iii) Functions: maximum deflection, maximum bending moments, …etc. There are two key parts of FOL: The syntax determines which collections of symbols are legal expressions, while the semantics determine the meanings behind these expressions (Nawari, 2012a). Unlike natural languages, such as English, the language of first-order logic is completely formal, so that it can be automatically determined whether a given expression is legal. There are two key types of legal expressions: terms, which represent objects, and formulas, which express predicates that can be true or false. The terms and formulas of first-order logic are strings of symbols which together form the alphabet of the language. These symbols of the alphabet are divided into logical symbols, which always have the same meaning, and non-logical symbols, whose meaning varies by interpretation. An example of the application of these approach is depicted in table 2.

Table 2 Example of using FOL method (Nawari, 2012a)

Building Code	Provision	Encoding
ASCE 7-10 Standard for minimum design loads for buildings and other structure has the following provision (3.2.1) for computing lateral pressure.	*"In the design of structures below grade, provision shall be made for the lateral pressure of adjacent soil. If soil loads are not given in a soil investigation report approved by the authority having jurisdiction, then the soil loads specified in Table 3.2-1 shall be used as the minimum design lateral loads. Due allowance shall be made for possible surcharge from fixed or moving loads. When a portion or the whole of the adjacent soil is below a free-water surface, computations shall be based upon the weight of the soil diminished by buoyancy, plus full hydrostatic pressure. The lateral pressure shall be increased if soils with expansion potential are present at the site as determined by a geotechnical investigation."*	$\forall$x *Structure*(x) ^ *BelowGrade*(x) → *Required*(x, lateral pressure) $\forall$x ¬ *SoilReport*(x) → *Required*(x, Table 3.2.1) $\forall$x *Structure*(x) ^ *BelowGrade*(x) ^ *BlelowWaterSurface*(x) → (*Required*(x, lateral pressure submerged) ^ *Required*(x, hydrostatic pressure)) $\forall$x *Structure*(x) ^ *BelowGrade*(x) ^ *ExpansiveSoil*(x) → *Required*(x, increase lateral pressure)

2.2 Mark-Up Language Methods

Currently, building regulatory documents are available in Hypertext Markup Language (HTML), Portable Document Format (PDF) or hardcopy. To ease the knowledge representation for further processing, a number of researchers have recommended different variations of Mark-up languages to formalize the building regulatory rules and guidelines.

For example, Lau et al. (2004) proposed the eXtensible Markup Language (XML) as a unified format to represent regulations because of XML's capability to handle semi-structured data such as legal documents. To semi-automate the translational process, they did develop a shallow parser as the first phase to consolidate different formats of regulations into XML. The structures of

regulations, namely its hierarchical and referential structures, are reconstructed in XML schema. For non-structural characteristics of regulations, a feature extraction mechanism had been proposed.

They further developed parse trees using a context-free grammar and a semantic modeling system that is capable of tagging regulation provisions with the list of references they contain. An example of an XML reference tag is shown in Listing 1, where Section 4.7.4 of ADA regulations cites Section 4.5 once. When appropriately rendered and linked, references provide users with additional but crucial information to better understand the regulations (Lau et al.,2004).

Listing 1 Showing an example of XML format for building regulations (Lau et al., 2004).

```
<regulation id="adaag" name="ADA Accessibility Guidelines" type="Federal"> ...
 <regElement id="adaag.4" name="Accessible Elements and Spaces..."> ...
  <regElement id="adaag.4.7" name="Curb Ramps"> ...
   <regElement id="adaag.4.7.4" name="Surface">
    <regText> Surfaces of curb ramps shall comply with 4.5. </regText>
    <reference id="adaag.4.5" num="1" />
   </regElement> ...
  </regElement> ...
 </regElement> ...
</regulation>
```

Another example of research on the transformation of regulation texts into formal languages is the automated mark-up of Italian legislative texts in XML (Bolioli et al, 2002). The automated classification of legal documents is achieved by a content-based clustering of regulations and labelling of Italian law texts. The importance of different terms that can identify a regulation text is highlighted such as terms appearing in head notes, heading section, case name, etc. These terms are then taken into consideration when dealing with information extraction techniques for legal case retrieval and are formalized as mark-up values in the XML representation of a legal text.

Hjelseth et al (2010) suggested a method that can model building regulations rules from direct semantic interpretation of text by use of four semantic mark-up operators: Requirement, Applies, Select and Exception. The semantic based approach is able to handle a wide range of purposes as validating, guiding systems, adaptive and content based model checking (Hjelseth et al, 2010).

This approach still requires a manual identification of the computable rules by a person with AEC domain skills and implementation of applicable software code based on common predefined measures. The proposed mark-up language model is based on few operators, namely: Select (S), Applies (A), Requirement (R) and Exception (E). Applied to provisions text, the user may highlight any phrase that means: more scope as a 'Select (S)' less scope as an 'Applies (A)'; 'shall'/'must' etc. as a 'Requirement (R)', (including alternative requirements); 'unless' etc. as an 'Exception (E)', (including composite exceptions). (R), (A), (S) and (E) constructs can be generally attributed to have a topic, a property, a comparator and a target value. The topic and property will ideally be drawn from a restricted dictionary composed of terms defined within the code provisions and normal practice. The value (with any unit) may be numeric, whereupon the comparators will include 'greater', 'lesser', 'equal' and their converses. If the value is descriptive, then only the 'equal' or 'not equal' comparators are relevant. If the value represents a set of objects, then the comparator may be any of the set comparison operators such as 'includes', 'excludes' (Hjelseth et al, 2010).

The approached is further suggested to be linked to IFC constraint sub-schema. This constraint can be instantiated as an objective and qualified as a specification constraint. An example illustrating the application of this approach to the International Building Code ICC IECC 2006 moisture control specification is shown in table 3.

This approach can be applied successfully in the case of fully normative documents that are expected to result in a pass or fail results. However, the approach has limitations to handle other type of code regulations text specifically the deep hierarchies and heavy cross referencing among provisions in code regulations.

Table 3 Mark-up specification ICC IECC 2006 502.5 Moisture control (Hjelseth et al., 2010).

Code Clause source	ICC IECC 2006 502.5 Moisture control
Rule Description	All {*A/*} framed {*\A*}{*S/*} walls, floors{*\S*} and {S/}ceilings{\S} {E/}not ventilated{*\E*} to allow moisture to escape shall be provided with an {*R/*}approved vapor retarder{*\R*} having {*R/*}a permanence rating of 1 perm{*\R*} (5.7 × 10 −11 kg/Pa s m^2) or less, when tested in accordance with the desiccant method using Procedure A of ASTM E 96. The vapor retarder shall be {*R/*} installed on the warm-in-winter side {*\R*} of the insulation. Exceptions: {*E/*} Buildings located in Climate Zones 1 through 3{*\E*} as indicated in Figure 301.1 and Table 301.1. In construction where {*E/*} moisture {*\E*} or its {*E/*} freezing {*\E*} will not damage the materials. Where other approved means to avoid {*E/*} condensation {*\E*} in unventilated framed wall, floor, roof and ceiling cavities.

3 Practical Approaches

One of the main goals of practical approaches of encoding building regulations is to provide a computable model with clear syntax and semantics that can be used to represent and reason about building codes regulations and provisions. In addition, the model must be well-suited to the needs of digital content providers. It is crucial that computable building regulation model is achieved before one starts the development of automatic code checking systems. An object-based representation of building regulations should define the minimum amount of data required to enable automatic checking of compliance and thus help achieving integrated practice goals when utilizing BIM workflow in practice.

The paper recommends that code representations can be done by accomplishing a number of development levels:

- High-Order Level : → Development of Model View Definition (MVD) → IFC schema
- Higher-order level: → feature extraction → all certain data objective concepts → goal is full encoding.
- Lower-order level: → feature extraction →all subjective information and uncertain data → partial encoding using fuzzy logic or full manual checking.

Figure 2 illustrates the practical approach model for building regulations and standard provisions digital encoding. The approach relies primarily on the XML standard as it offers many benefits (Nawari, 2012b):

(i) In-memory XML programming interface that facilitates communicating with XML from within the various software programing languages.

(ii) The powerful extensibility of the query architecture can provide implementations that work over both XML and traditional SQL data stores.

(iii) The query operators over XML use an efficient, easy-to-use, in-memory XML facility to provide XPath/XQuery functionality in the host programming language. This integration provides strong typing over relational data objects while retaining the expressive power of the relational database model and the performance of query evaluation directly in the underlying data store.

(iv) All standard query operators can be replaced with user-defined implementations that provide additional services such as remote evaluation, query translation, and optimization.

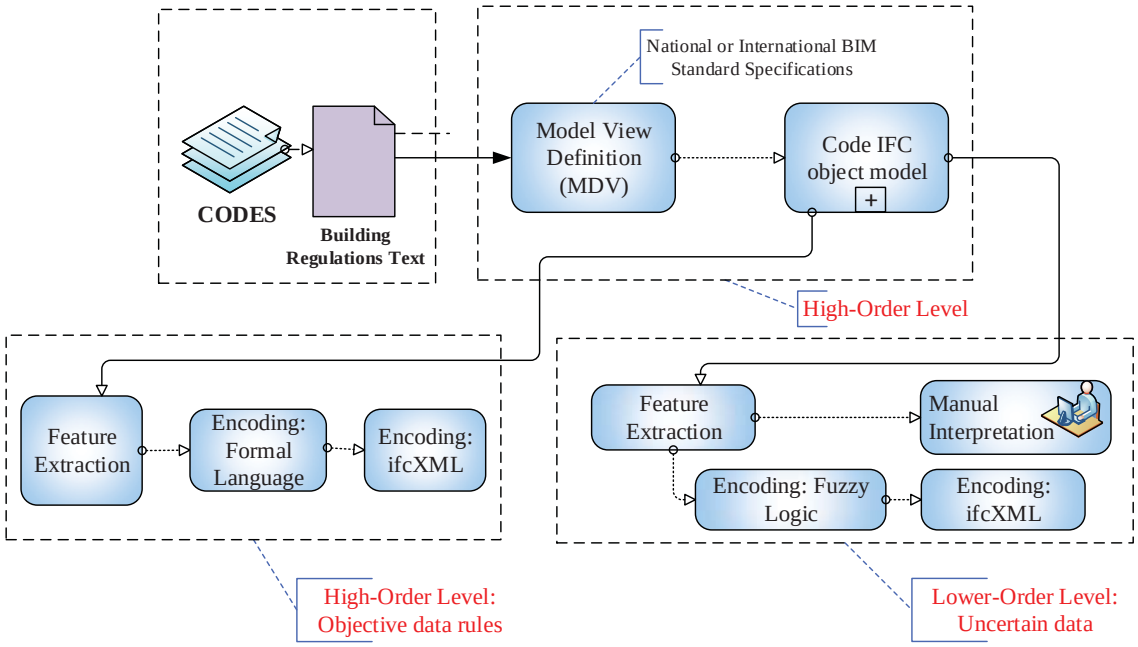

Figure 2 Practical approach for computable building regulation model.

The feature extraction is usually done with the help of ontological representation of all the knowledge taking part in the building regulation text along with the Model View Definition (MVD) and the resulting IFC schema.

The proposed approach aims to formalize encoding of building regulation in a way that is close to natural language and easily verifiable with several syntactic and semantic features that are capable of handling objective as well as subjective regulation provisions. The transformation into rules of the uncertain data can be handled partially by utilizing fuzzy logic.

Fuzzy logic has been applied in many areas successfully to assist in processing information that may not be completely defined. More details about this transformation is being currently investigated by the first author. Future publications will through more lights on the subject. Finally, it is important to acknowledge the fact that there will always be certain part of building regulations which still require manual verifications for compliance.

4 Conclusions

With the increasing amount of building regulations and expert knowledge, the need for computable representation of the building codes and regulations for automating the code checking process is becoming ever more important. Particularly, in the environment of Building Information Modeling (BIM) workflow, model checking against building codes and standards is critically needed to be an automatic or semi-automatic process.

The paper seeks to propose methods with practical flexibility of encoding building standards and domain knowledge and at the same time possess easily verifiable syntax and sematic features. The proposed approaches relies on clearly identifying objective and subjective data of the regulatory text before formalizing building codes. The suggested methods require development of MVD and IFC schema along with FOL, Fuzzy logic, and ifcXML for encoding building provisions and guidelines. These approaches realize the limitations of the formalization system by openly designating which part of the codes and standards can be encoded and which part can't be computerized and require manual compliance verification.

Acknowledgements

Authors are sincerely grateful to Fulbright U.S. Scholar Program for funding and supporting this research. Also, authors would like to express their gratitude to Collage of Design, Construction and Planning at the University of Florida as well as the School of Architecture at the University of Florida for promoting the international exchange activities and facilitating various supportive information and tools.

References

Bolioli, A., Dini, L., Mercatali, P., and Romano, F. (2002). 'For the Automated Mark-Up of Italian Legislative Texts in XML' in T.J.M. Bench-Capon, A. Daskalopulu and R.G.F. Winkels (eds.), *Legal Knowledge and Information Systems. The Fifteenth Annual Conference*. Amsterdam: IOS Press, 2002, pp. 21-30.

Dimyadi1, J., and Amor, R. (2013). "Automated Building Code Compliance Checking – Where is it at?" *Procceding of the 19th International CIB World Building Congress,* 5th-9th May 2013, Australia.

EASTMAN, C. M., JAE-MIN LEE, YEON-SUK JEONG, JIN-KOOK LEE, 2009. "Review Automatic rule-based checking of building designs ", *Journal of Automation in Construction (18), pp. 1011–1033,* Elsvier.

FENVES, S. J., GARRETT, J. H., KILICCOTE. H., LAW. K. H., AND REED, K. A., 1995. Computer representations of design standards and building codes: U.S. perspective, *The Int. J. of Constr. Information Technol., 3(1), pp. 13-34.*

Hjelseth, E. & Nisbet, N., 2010. Exploring semantic based model checking. In the *Proceedings of the 2010 27th CIB W78 International Conference.*

Hjelseth, E. and Nisbet, N. (2010). "Overview of concepts for model checking". In the *Proceedings of the 2010 27th CIB W78 International Conference, Cairo, Egypt, 2010.*

Lau, G.T., and Law, K. (2004). "An Information Infrastructure for Comparing Accessibility Regulations and Related Information from Multiple Sources". In the *Proceedings of 10th International Conference on Computing in Civil and Building Engineering, Weimar, Germany, 2-4 June, 2004.*

Nawari, O. N. (2012a). "Automated Code Checking in BIM Environment". In the *Proceedings of the 14th International Conference on Computing in Civil and Building Engineering, Moscow, Russia, 27-29 June, 2012.*

Nawari, N. (2012b). "Automating Codes Conformance." *Journal of Architectural Engineering, ASCE, 18(4), 315–323.*

Zhang, J. and El-Gohary, N. (2013). "Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking." *Journal of Computing in Civil Engineering, 10.1061/(ASCE)CP.1943-5487.0000427, B4015001.*