



Domain Knowledge Representation Languages and Methods for Building Regulations

Murat Aydın^(✉) and Hakan Yaman

Department of Architecture, Istanbul Technical University, Istanbul, Turkey
{aydinmurat12,yamanhak}@itu.edu.tr

Abstract. The development of computable building regulations is an important factor for shortening the communication of building code provisions and automated code compliance checking. The representation of building regulations plays an important role in a computer-readable format which recognizes and understands certain aspects of the domain knowledge in compliance checking of building regulations. It allows compliance checking of a building model according to building regulations, codes and standards, and it evaluates the building model with its building elements. The studies have continued to the present to obtain data from legal sources and to create an appropriate computable representation of building regulations. In this research, the studies on domain knowledge representation of computable building regulation compliance checking are reviewed in detail based on the literature in the last 50 years. It also discusses the languages and methods of the studies under common titles such as Human Languages, Formal Languages, Artificial Intelligence Methods, Markup Language Methods and Semantic Web Methods and also reviews the languages and methods which are used in the representation of building regulations.

Keywords: Domain knowledge representation · Automated Code Compliance Checking · Building Information Modeling

1 Introduction

1.1 Domain Knowledge Representation

In the recent years, many researchers have concentrated on engineering information systems for providing effective methods and tools for domain knowledge representations of the construction industry. Such problems are in focus like storing, retrieving, transferring, indexing, sharing and using the data in order to improve and accelerate the design and construction process. The domain knowledge has many facets regarding the building lifecycle tasks, but code checking and agency review processes deserve a special focus due to their content and information complexity. The current research in this domain includes Architecture, Engineering and Construction (AEC) Codes and Regulations and Automated Code Compliance Checking (ACCC).

AEC Codes and Regulations are legal documents written and authorized to be understood and implemented by legal experts and construction professionals. These are not definitive as formal logic. The flexibility of expression is important for an information retrieval system. Only experts can interpret these documents and translate them

into formal notations and software applications. They can select and use all kinds of information which they need and use them at various levels of accuracy. These are the practical tasks in which these extractions and applications are carried out, and where researchers and professionals are trying to develop automated or semi-automated methods for many years.

Most of the early studies in this specific research field focus on the conversion of the domain knowledge from natural language into a formal language such as First-Order Logic (FOL). In FOL, a predicate is a well-defined term (or a function) that can be evaluated as True, False or Unknown (if the terms are not defined). In addition, the quantity of predicate logic is related to the emerged logic statements or the application of expressions to all cases. There are general techniques developed for converting logic claims into executable statements, including Prolog Computer Language. The implementation of these methods in AEC regulations is laborious, and it has many limitations. For example, the interpretation of the rules for buildings and how many models should be applied to the rules are important issues. Furthermore, many subjective provisions cannot be translated into FLO.

One of the first implementations of FLO is the decision tables representing AISC specifications which was introduced in 1969. The decision logic tables approach has contributed to a technical standard with many objective data such as AISC specifications. Various applications based on this approach were used as a design tool for steel structures for at least fifteen years [1]. Other studies in this area include one of the most important early standard SASE (Standards, Analysis, Synthesis, and Expression) representation system developed by the NIST in 1984 (National Institute of Standards and Technology formerly known as the US National Bureau of Standards). SASE was used to establish and maintain the structure of decision tables and standards [2].

Other studies focus on using expert systems or AI (Artificial Intelligence) methods to encode regulatory data for use in building design [3–6]. These systems are only useful when basic information is kept up to date with regulatory provisions. Despite the intrinsic ineffectiveness and dependence on manual updates, the use of AI to transfer information from other computable objects and regulatory texts as automated or semi-automated has been utilized until today [7–10].

Other approaches to computerized building code checking include markup document modeling and the use of hypertext to represent regulatory provisions [11, 12]. The concept of marking regulatory texts in order to create a regulatory text are reconsidered in numerous studies [13–15]. Extensible Markup Language (XML) was proposed to represent regulations such as legal documents due to XML's ability to process semi-structured data in 2004 [14]. The studies have continued to the present for obtaining data from legal sources and to create an appropriate computable representation of the building regulations, codes and standards. The search for solutions continues in the AEC industry for more practical computable representation of the regulations, codes and standards.

1.2 Background of Automated Code Compliance Checking

Automated Code Compliance Checking (ACCC) is used for the computable representations of the building regulations. ACCC is a computational procedure for handling

the manual regulation verification problem in a limited number of reasoning steps. It includes deduction, induction, abstraction, generalization with a structured logic. It is a systematic extraction of logical rules from written documents and the development of a general implementation plan. ACCC also encompasses algorithmic strategies to search for repetitive patterns, universal principles, modifiable modules, and inductive connections. The rational strength of such systems lies in their ability to allow machine interpretation for building regulations, to obtain results that have missing data, and to expand certain limits of human intellect. Thus, ACCC shows rationality, consistency, coherence, organization and systemization.

The first step of the ACCC process is the creation of a computable representation of the content and context of building regulations. This involves an interpretation process in which the semantic structure of each arrangement is translated into rules or parametric models using certain formal languages. Following this procedure, the information including parametric data structures or rules are queried and activated using specific software tools. The next step includes the connection between these representations and the Building Information Modeling (BIM) data for interpretation. The ACCC process aims to improve the actions of manual design review by rationalizing the knowledge of a predictable result. It includes the actions to extract the necessary information automatically and verify particular details. This task is usually very time-consuming when applied manually to the initial data set of properties and conditions. Here, the computable models serve as a vessel to carry out automated code compliance checking processes.

In the second half of the twentieth century, various formal language models were developed to represent building codes and regulations. These models have been useful to handle various aspects of the knowledge domain. However, none of them is sufficient enough to address all the issues regarding ACCC. The development of computable building regulations is an important asset for the development of existing provisions by shortening the communication of code provisions and ACCC. The representation of building regulations and standards in a machine-readable format that recognizes and reads certain aspects of the knowledge domain plays an important role in the automated validation process of building regulations. The computable numerical scheme of building rules and specifications allows automated code compliance checking without changing building design. The scheme evaluates the compliance design principles of parametric objects, their relationships or their attributes. It contains a code-compliant building design framework for rule-based systems, and results are produced in the form of Success, Failure, Warning, Unidentified.

It is clear that computable building regulations focus on data preparation and rule development. Each of these factors has its own characteristics and limitations. The difficulties are sourced from the nature of building regulations and standards. For example, building regulations are not private documents and they are often referred to as other sources. This means that all professionals must be familiar with most of the regulations in a building code or standard. However, such data is not always represented in a formal form. Furthermore, understanding a design standard requires the knowledge of the relevant design area. In AEC disciplines, basic professional knowledge (basic knowledge acquired by engineers and architects) is expected to benefit from architectural

engineering design standards [16]. In addition, both knowledge and tacit experience are required to decide whether a standard is audited and proceeded.

1.3 Building Regulations

People naturally live and carry out various activities together. Laws are required to regulate these activities socially and personally in order to ensure mutual living. Laws are also written rules that always show the necessary importance to maintain unity among people, to ensure harmony in the society and to preserve shared values. Specifically, codes, regulations and standards ensure the health, safety and welfare in the built environment and also contribute to the quality of life. Today, most of the information needed can be accessed through codes, regulations or standards. Codes, regulations and standards are created and updated by authorities, taking all necessary measures to prevent unwanted, unpredictable or controversial issues. They define the minimum requirements for the design and construction features of buildings and other structures that create the built environment. However, it is clear that the requirements of rules, regulations and standards are sometimes insufficient due to the unexpected situations that may cause loss of life and property. The concept of having minimum requirements for buildings and other structures is important for determining the acceptable level of living quality for users [17].

The construction laws and regulations are implemented in federal, county, city, state and local government levels to protect the health, safety and welfare of the society and to ensure the integrity of the completed construction projects. One of the most important challenges is the fact that the codes, regulations and standards are effective locally and different from each other in every specific authority. In general, each country has its own legal practice and local rules and regulations may not be applicable for other countries or domains. In order to address this issue, International Organization for Standardization (ISO) develops universally applicable codes, regulations and standards around the world. The sub-headings of codes, regulations and standards according to the CSI (The Construction Specifications Institute) are given below [17]:

- Codes
 - Design and Construction Requirements
 - Fire and Life Safety Requirements
 - AHJ Plan Reviews and Permits for Construction
 - Evaluation Reports
- Regulations
 - Zoning, Deed Covenants, and Regulations
 - Site Use and Environmental Requirements
 - Occupational Safety and Health Requirements
 - Accessibility Requirements
 - Health and Sanitary Requirements

- Standards
 - Standards Development Process
 - Standards Development Organizations
 - Governmental Standards Development Agencies

Clauses of Building Regulations. Regulations, sometimes known as rules, are developed and written by Authorized Having Jurisdictions (AHJs) for specific purposes. For example, a widely known Zoning Regulation is designed to regulate and determine the use of land within the city boundaries. Likewise, building regulations provide the necessary conditions for safe, healthy and optimum performance of buildings. These include specific issues such as fire protection, accessibility, energy performance, acoustic performance, elevator safety, electrical and gas safety, etc. Regulations also include sanctions or fines in case of non-compliance with the required conditions.

The content of building regulations is of great importance in the practical methods applied to the computerization of building regulations. The main objective of the existing practical methods is to digitize the building regulations through a computable model with clear syntax and semantics. This can be used to represent and justify building rules and clauses. Here, the provided model must comply with the general requirements of digital content providers. For example, an object-based schema of building codes should be represented with the minimum of data as possible in order to check the suitability of the building codes automatically. This amount may vary depending on the content of the specific building regulation. According to the content, some clauses may include short, concise, and quantitative statements, and some other clauses may contain expressive statements. These clauses can also be associated with other clauses by referring to each other. The clauses of a building regulation are classified into four main categories as shown in Fig. 1. These are:

- **Conditional Clauses:** Conditional clauses are applied to interpret a set of formal rules directly from textual documents. Examples of these are very common, and these clauses often contain rules with numerical values.
- **Content Clauses:** Contents clauses cannot be converted into TRUE or FALSE statements. These clauses usually include expression with definitions such as a definitions of a firewall, a fire stair, fire speed, smoke evacuation, or a high-rise building etc.
- **Ambiguous Clauses:** Ambiguous clauses are subjective. These clauses contain unstable or unclear words such as normally, approximately, mostly, nearly, maybe, etc.
- **Dependent Clauses:** Dependent clauses indicate that a section of a building regulation is related to one or more clauses. If some provisions fully meet the other provisions, it means that it complies with a certain condition. In general, it is somewhat difficult to convert these clauses into a set of formal rules. Therefore, it may be necessary to check compliance of dependent clauses manually.

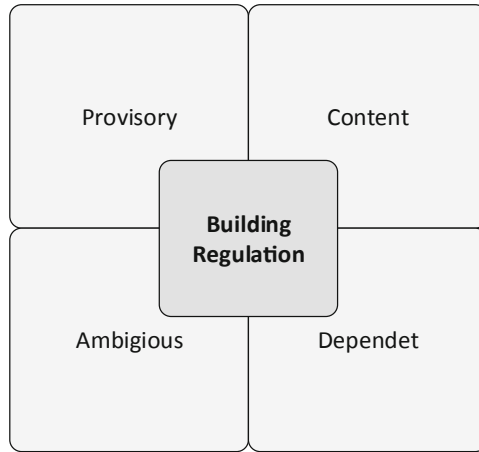


Fig. 1. Clause categories of a building regulation.

2 Domain Knowledge Representation Languages and Methods

The majority of the previous studies on modeling formal language representations of building regulations have focused only on syntax and grammar of rules. Understanding the meaning of a rule considered to be the most important task. This requires experts to have the knowledge and experience to interpret the meaning of the rules of a regulation. For example, CORENET uses a logic-based interpretation approach to translate provisions from natural language into a formal language. During the interpretation process, there are implied assumptions and expectations that provide an understanding of what needs to be examined.

Due to conventional methods in the review process, many inconsistencies, mismatches, human errors, and abuses more likely to occur in manual building regulation compliance checking. As a result of checking, uncertainties arise based on experience. Setting standards for building code ontology and BIM data can be listed as solutions to these particular types of problems. In order to create consistent, accurate and measurable conditions and constraints, the suitability of each rule must be checked by machine-oriented automated processes. The current research studies largely include different modeling techniques for creating a formal language (computer-readable rules of regulations written in human language). Below is a summary of domain knowledge representation languages and methods which are used to automate building regulation compliance checking. As seen in Fig. 2, the most effective way is to apply modeling languages that are capable of generating rules that can be interpreted by computers. According to Fig. 2 workflow, transformation of a building regulation into a building code that is parsed and controlled by a computer is carried out in 3 stages:

- **Decomposition (Raw Data/Building Regulation)**
 At the initial stage, the relevant building regulation is analyzed and decomposed into the smallest base unit in the decomposition phase. Data tree structure related to the Regulation, Part, Clause, Statement and Textual Expression of Statement is formed until the stage of decomposition into statements. The second stage includes the examination of statement structures. These statements are generally grouped into two types of classification as a Clarification or a Rule. As a result of this classification, a statement is determined which can be a rule. As seen in Fig. 3, the regulation statement is detailed as the following:
 - Building Regulation Name: Planned Area Zoning Regulation (PAZR)
 - Part No: Part 05 (P.05.)
 - Part Heading: Provisions for Building
 - Clause Heading: Doors and Windows
 - Clause No: Clause 39 (C.39.)
 - Statement ID: PAZR.P.05.C.39.ST.(01).a).
 - Textual Expression of Statement: All door heights cannot be less than 210 centimeters.
- **RASE Method (Translation/Formal and Logic Languages)**
 The decision tables proposed by Nyman and Fenves for steel structures are handled for the representation of the relevant building regulation in the digital environment. According to Nyman and Fenves, a regulation statement is defined as a rule in 4 basic features. These are Requirement, Applicability, Selection and Exception. They suggested this model as a RASE Method. As seen in Fig. 3, the regulation statement “Statement ID: PAZR.P.05.C.39.ST.(01).a.” according to RASE Method is defined as a rule:
 - Building Code Name: Planned Areas Zoning Regulation Code (PAZRCode)
 - Part No: Part Five (RSG.05.)
 - Part Heading: Provisions for Building
 - Clause Heading: Doors and Windows
 - Clause No: Clause 39 (RS.39.)
 - Rule ID: PAZRCode.RSG.05.RS.39.R.(01).a).
 - RASE Method Expression of Rule:
 - Requirement: Door Height $\geq 2,10$ m
 - Applicability: <IfcDoor> <DOO - ...> <Door> <Door>
 - Selection: <IfcDoor> <OverallHeight> ≥ 2100 mm
 - Exception: Null
- **C# Language (Encoding/Computer-Readable Statement)**
 C# Object-Oriented Programming Language is used to encode the selected rule in the relevant building code. C# is a powerful, modern, object-oriented and type-safe programming language. At the same time, C# provides both the strength of the C++ language and the convenience of Visual Basic. The C# language is used in the Microsoft Visual Studio environment. C# language is a popular programming

language mainly preferred by the AEC industry. It also provides easy operations for standard data formats. IFC is among the standard data format that can be further processed with C#. IFC.XML format is selected as standard data. It is checked in the following coding whether it is selected from IFC.XML data for the related feature to be controlled by the related rule and provides the numerical value specified in the regulation statement. As seen in Fig. 4, the representation of “Rule ID: PAZRCode.RSG.05.RS.39.R.(01).a.” in C# language is as follows:

```

- #region PAZRCode.RSG.05.RS.39.R.(01).a.
  if (Convert.ToDouble(element.height) >= 2100)
    {text = " compliance with door statement. " }
  else
    {text = " not compliance with door statement. " }

```

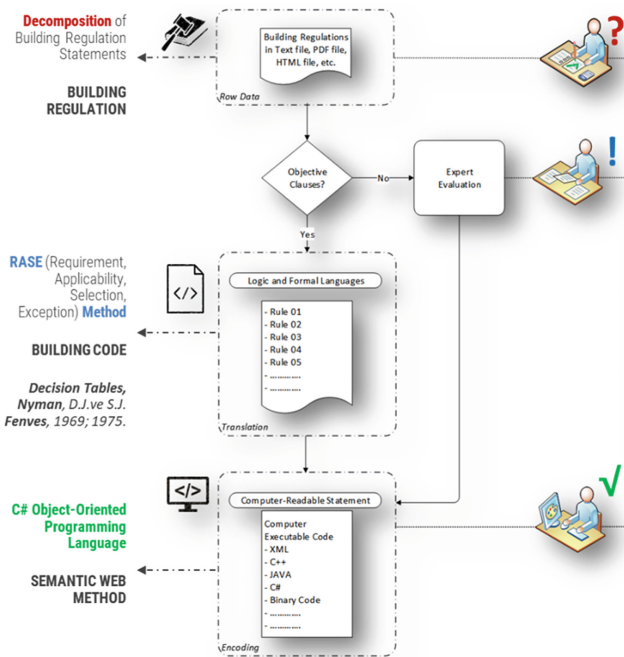


Fig. 2. A framework of developing a building regulation into a building code

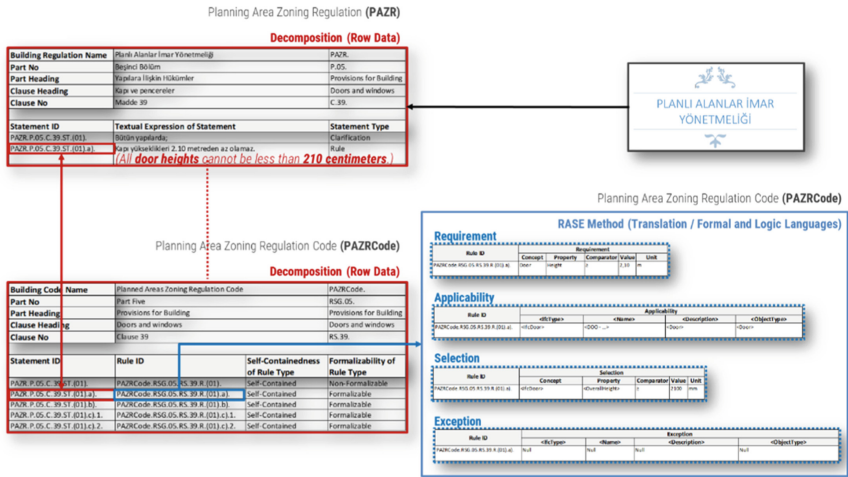


Fig. 3. Decomposition (raw data/building regulation) phase and RASE method (translation/ formal and logic languages) phase

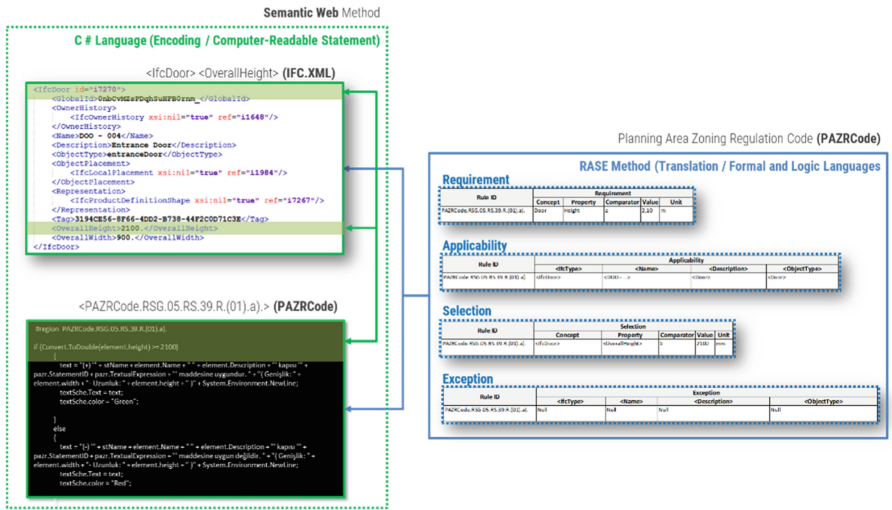


Fig. 4. C# language (encoding/computer-readable statement) phase

2.1 Human Languages

Human is a social entity as all individuals communicate with each other. A language is method of communication between people in written and spoken form. Various arguments were made about the emergence of this tool which provides unity, order and mutual agreement between people. Although these arguments cannot be proved, but, it is widely accepted that the body language emerged and then the speech-language was formed as a result of the human need for communication.

Human languages are easy to learn by children. The learning occurs during the one ended transmission of verbal expressions from the adults. In this expression process, the adult speaks the human language according to the breathing rate and the limitations of short-term memory [18]. This implies that human language is the first language that an infant can comprehend. In the process of cognitive development, a child quickly acquires the speaking and understanding abilities close to his/her parents. This illustrates the numerous extensibility of expressions with a limited vocabulary. Many words have an open-ended number of senses and uncertainty in a human language. As seen in Fig. 5, some of the basic characteristics of human languages are as follows:

- Phonology
- Morphology
- Syntax
- Semantics
- Pragmatics
- Discourse Information
- Realm Knowledge

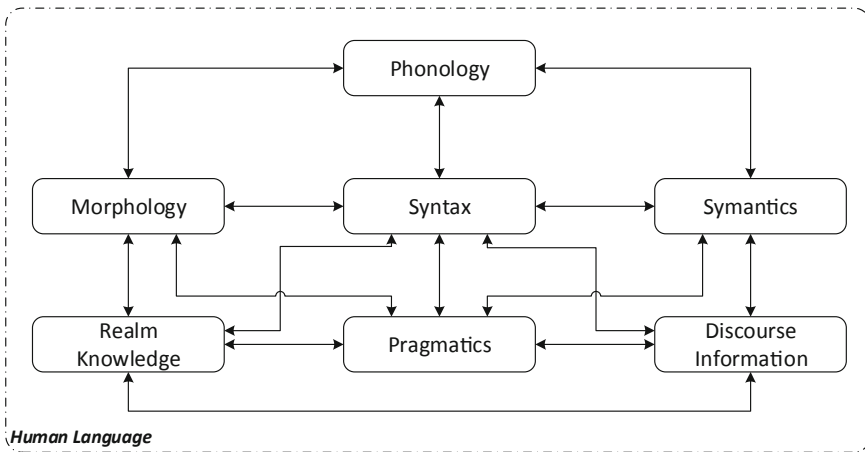


Fig. 5. Human language characteristics.

Rules, which combine sounds and words together, create new sentences of a language grammar. A language grammar is equally complex and logical. It can produce an infinite set of sentences to express any thought. Therefore, it is very difficult to talk about a single meaning for a sentence in human language. Instead, there are numerous possibilities of different meanings within a single sentence. In addition, human languages are constantly changing over time and it is impossible for a person to understand every text or dialogue in his/her language. Taking this complexity into consideration, it is unrealistic to expect a computer to overcome this problem with acceptable levels of effort and computation power.

2.2 Formal Languages

A formal language is a set of symbols' strings that can be limited to specific rules in mathematics, computer science and linguistics. The alphabet of an official language is a set of symbols, letters, or markers. The strings of a particular alphabet are called words. The words of a formal language are sometimes referred to as well-formed words or well-formed formulas. A formal language is often referred to as a formation rule through a formal grammar or context-free grammar.

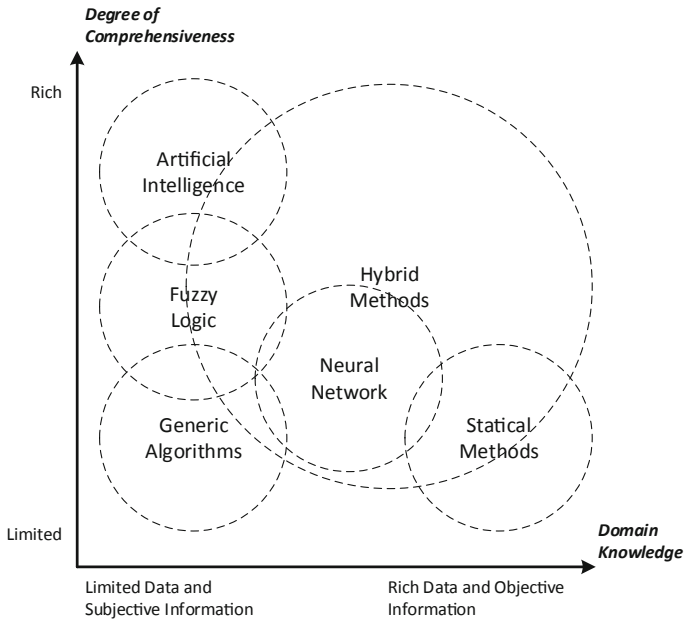


Fig. 6. Application areas of formal languages.

The aim of formal languages is to represent existing problem areas as much as possible to predict objective functions. Some of the major formal languages, which have been developed over the years, show the efforts of researchers to develop a formal language to understand and comprehend the complexity of real-world systems. These include statistics, syntax, binary and fuzzy logic, lexical semantics, neural networks and genetic algorithms. In the 1950s, Claude E. Shannon's knowledge theory and other statistical methods were popular in both linguistics and psychology. But, the speed and storage capacity of the first computers were not sufficient to handle the required data volumes. Towards the end of the century, the increase in computer power made it possible to processes large volumes of data using different methods. Many of these formal languages try to represent the problem area knowledge as adequately as possible to better predict the target function. These languages attempt to develop processes by using heuristic rules and data.

Figure 6 shows the main application areas of these formal languages. Each of these modeling languages is based on a specific technology. These are mathematical statistics, grammar rules, dictionary formats, fuzzy logic and networks of neurons. These formal languages are given below:

- Hybrid Methods
- Statistical Methods
- Generic Algorithms
- Artificial Intelligence Methods
- Neural Networks
- Fuzzy Logic Approaches

2.3 Artificial Intelligence Methods

The aim of Artificial Intelligence (AI) is always to simulate human intelligence, knowledge and perception. This simulation focuses on two main domains:

- The first domain is associated with learning relationships
- The second domain focuses on encapsulating and reusing information.

From the ACCC perspective, the AI methods aim to fully automate a building code compliance checking process by extracting and coding legal requirements to ensure computer processing. These methods are generally based on Natural Language Processing (NLP) models, which predict the probability distribution of language expressions. It includes two main types:

- A rule-based approach and
- A machine learning (ML)-based approach

A rule-based approach uses manually developed rules to process documents. A machine learning (ML)-based approach refers to a system learned from existing data or previous experience and it uses ML algorithms to process text. An ML-based approach may be one of the following types:

- Supervised,
- Unsupervised and
- Semi-supervised.

Supervised ML-based algorithms require intensive manual effort for preparing a data set. However, their sensitivity and performance are relatively higher than other methods. In general, a rule-based approach provides better text processing performance than an ML-based approach [19].

NLP methods can be classified into shallow and deep approaches. These are distinguished by their different emphasis on text processing. If an emphasis is placed on the analysis of missing sentences or specific topics, the NLP method is considered shallow. On the other hand, if an emphasis requires full sentences, the NLP method is

considered deep [20]. An NLP method that achieves the most reasonable performance results is shallow. Utilization of a deep NLP method is relatively difficult due to the requirement of detailed information and efficient reasoning about a domain in AI [21].

There have been many research studies focusing on NLP techniques in the AEC. Caldas and Soibelman conducted an ML-based text classification study of construction documents [22]. As seen in Fig. 7, Zhang and El-Gohary proposed several approaches to automate building regulation compliance checking using NLP methods. These approaches include semantic modeling and semantic NLP techniques to facilitate the automatic processing of building regulation documents to extract regulatory rules in computable formats [23–25]. Typically, they involve a set of algorithms on a computable platform. These are Text Classification (TC), Information Extraction (IE) and Information Transformation (IT).

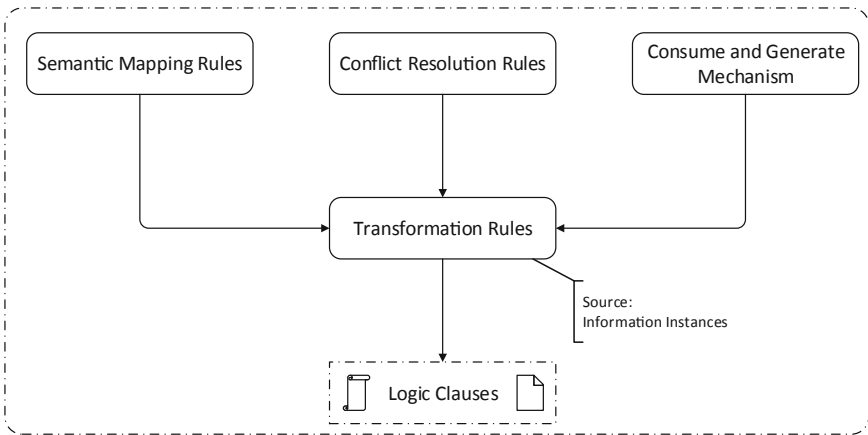


Fig. 7. Transformation rules study of Zhang and El-Gohary [25].

2.4 Markup Language Methods

A markup language is a system used to note attributes of a document. Historically, the term “marking” has been used to refer to the process of marking a string. It usually contains fonts, dimensions, spaces, letters, and other formatting features. Marking specifies the order of characters or other symbols that are placed in a text or a word to define the logical structure of the document or to specify how the document should appear when viewed or printed. Markup languages are static, unlike programming languages. Programming languages process data through various operations in a dynamic fashion. Basically, a markup language defines similar units of information in a document. It enables them to read and to process applications more effectively by introducing a form of into a document.

Hypertext Markup Language (HTML). Hypertext Markup Language (HTML) is a standard text markup language used to create web pages. HTML cannot be defined as a programming language. Because HTML contains no programming logic and it doesn't have common conditional and flow control statements. Basically, HTML is necessary to simply connect different data such as text, image, video and a page together for the proper viewing by a web browser software. As shown in Fig. 8, HTML commands are written between <and>. They are often used to indicate the beginning and the end of the highlighted text (<FamilyName>AYDIN</FamilyName>). However, they can also be used individually if a sign is placed in the text (<ThePerson>). Most of the building regulations and standard documents are obtained in addition to a printed copy with an HTML.

```

<IfcPersonAndOrganization id="i1643">
  <ThePerson>
    <IfcPerson xsi:nil="true" ref="i1637"/>
  </ThePerson>
  <TheOrganization>
    <IfcOrganization xsi:nil="true" ref="i1639"/>
  </TheOrganization>
</IfcPersonAndOrganization>
<IfcPerson id="i1637">
  <FamilyName>AYDIN</FamilyName>
</IfcPerson>
<IfcOrganization id="i1639">
  <Name>Murat</Name>
</IfcOrganization>

```

Fig. 8. An example of HTML text in IFC.XML of a BIM model.

Extensible Markup Language (XML). Extensible Markup Language (XML) is a markup language for creating documents that can be easily read by both humans and computer systems. The XML standard is defined by W3C. Figure 9, shows the “xmlns:xlink” referenced to this standard. In addition to data retention, it also serves as an intermediate format for exchanging data between different systems. For example, Lau and Law proposed an integrated format to represent Extensible Markup Language (XML) regulations. Because XML is capable of modeling semi-structured data such as legal documents [26]. In fact, XML has a binary property as a markup language and a Web standard.

```

<?xml version="1.0" encoding="UTF-8"?>
  <ex:iso_10303_28 xmlns:xsi="http://www.w3.org/2001/XMLSchema-in-
stance" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:ex="urn:iso.org:standard:10303:part(28):version(2):xm
lschema:common" xsi:schemaLocation="urn:iso.org:standard:10303:part(28):ver-
sion(2):xm
lschema:common ex.xsd" version="2.0">
  <ex:express id="exp_1" external="" schema_name="IFC2X3">
  <!--external: When the EXPRESS schema is represented 'by-reference', the
external XML attribute shall be present and its value shall identify a resource that
contains the EXPRESS schema (text).-->
  <!--schema_identifier: For an EXPRESS schema that is defined by a Part of
ISO 10303 schema_identifier shall contain the ASN.1 (official identifier for a part
of 10303 ) identifier value associated with that schema by that Part of ISO 10303-
->
  <ex:uos id="uos_1" description="" schema="exp_1" configuration="i-
ifc2x3" edo="" xmlns="http://www.iai-tech.org/ifcXML/IFC2x3/FINAL" xsi:sche-
maLocation="http://www.iai-tech.org/ifcXML/IFC2x3/FINAL ifc2x3.xsd">

```

Fig. 9. A short example of an IFC.XML format.

LegalRuleML and LegalRuleML Modeling Languages. In the last few years, several legal XML standards have been proposed in order to identify and present legal text information with XML-based rules [27–30]. Other studies have focused on Legal Ontology Research, which is combined with Semantic Web to model law concepts and provisions [31–33]. Many of these studies have used the Expressive XML Annotation, combined with Semantic Web technology, to meet the unique features of legal rules and norms. Examples of these efforts have resulted in RuleML and LegalRuleML Modeling Languages [30, 34–36]. The purpose of LegalRuleML is to determine the characteristics (comprehensive, articulated, and meaningful markup) of legal rules and norms.

2.5 Semantic Web Methods

The Semantic Web is an internet add-on that aims to enable web content to be understood, interpreted and used not only by natural languages but also the related software. This software can easily find, share, and integrate data. The semantic web essentially consists of a philosophy, a set of design principles, collaborative working groups and assistive technologies. Some components of the semantic web are not yet developed or implemented but these are more likely to be achieved in the near future. The other parts are expressed in official descriptions. All of these parts are expected to formally describe the concepts, terms and connections in a particular problem space. These parts are:

- Resource description framework (RDF)
- Various data conversion formats such as XML, N3, Turtle, N-Triples.

- Notations like resource description framework Diagram (RDFs)
- Web ontology language (OWL)

The Internet (World Wide Web) was originally designed as a content service for documents shown by web browsers. It is a content service that is meaningful to users rather than machines [37]. The data and meanings expressed in web pages are difficult for a computer to extract, understand and process automated information [38]. It is recommended to solve the problem by adding contextual information to the existing information in order to gain meaning on the internet with the Semantic Web approach. This semantic approach is the key concept underlying web marking including:

- Web coding
- Information processing
- Universal usability
- Search engine visibility
- Maximum display flexibility

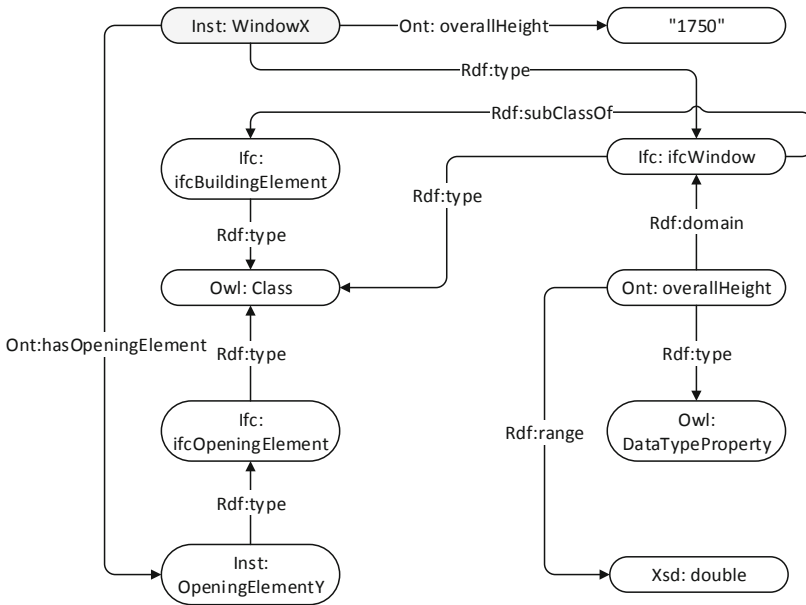


Fig. 10. Semantic network study of Pauwels et al. [39].

The semantic web method is focused on the development of an IFC model based on logic theory and rules by using a descriptive language. One of the earliest studies belongs to Pauwels [39]. The study concentrated on semantic web technologies for building code compliance checking. The main purpose of this study was to provide building acoustic regulation compliance checking using detailed BIM models. The concept of the study was based on a semantic network. This semantic network defines

concepts through a directed, labeled graph. Each node represents a concept or an object, and each arc represents the logical relationship between these two concepts or objects [39]. The graph shown in Fig. 10, is an instance of a combination of logic-based declarative sentences, each of which consists of two nodes and a relational arc. The semantic of a particular concept is defined by a graph associated with this concept.

3 Evaluation

The above-mentioned domain knowledge representation methods and languages are used during automated code compliance checking process. Figures 2, 3 and 4, show the three stages of checking and transforming an example regulation. The first stage is the decomposition of the building regulation written in human language. The second stage is the RASE method proposed by Nyman and Fenves, one of Formal Languages, and the last stage is the semantic web method with C# programming language. Preferred languages and methods have advantages and disadvantages according to time, cost, programming labor and their impacts. The most important implication of these approaches is the digital transformation of building regulation checking processes using available technological methods.

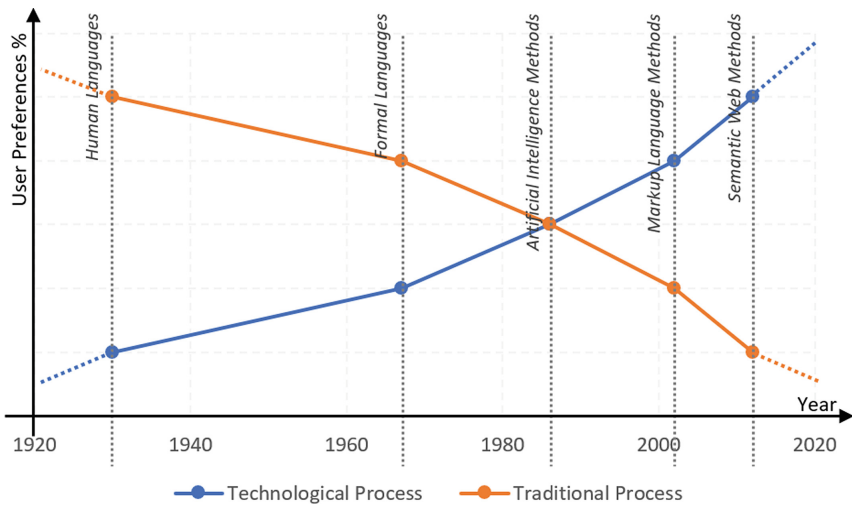


Fig. 11. Traditional and technological checking process of building regulation according to domain knowledge representation languages and methods by year.

As shown in Fig. 11, the traditional checking process of building regulations written in human language had been carried out exclusively by human experts using analog methods. In the late 60s, the development of formal languages provided the opportunity for checking building regulations in using technology through the conversion of regulations into logic building rules. The AI methods which came in the mid-80s initiated the transformation of building regulation checking from conventional

to computerized methods. With the increased technological capabilities at the beginning of the 2000s, Markup Languages Methods made it possible for computers to digitally parse and process logic representations of building regulations. As a continuation stage of these advancements, the semantic web methods enabled the specific software to understand and interpret building regulations and perform automatic checking. Currently, web-based systems are being developed for better accessibility and platform-free operations for ACCC. These developments clearly demonstrate the current trajectory of ACCC approaches as they are being rapidly transformed from analog to digital.

4 Conclusion

In this research, the studies on domain knowledge representation of computable building regulation compliance checking are reviewed in detail focusing on the last fifty-year research work. Existing languages and the representation methods are discussed under common categories such as Human Languages, Formal Languages, Artificial Intelligence Methods, Markup Language Methods and Semantic Web Methods. This extensive literature review helped to illustrate the potentials and limitations of the existing methods and the markup languages for Automatic Code Compliance Checking.

In summary, building regulations and standards are legal documents created by the legislation experts and used by AEC professionals. These texts are generally in the form of natural language. They usually include texts, mathematical formulas, tables and other legal provisions. These expressions are as precise as official languages. The flexibility of a text is very important for an information retrieval system. Engineers and architects can use these documents and translate them into formal scientific representations and software applications. This may reduce the complexity of code checking procedures where experts can extract any data type, find problems regarding code checking and apply them in various application stages. The existing methods allow automated or semi-automated code checking procedures using specific software tools. The current literature documents novel research studies on this domain in order to provide more effective methods and approaches that are capable of creating rules and semantic links through new modeling languages. Due to the increasing complexity of construction projects, it can be argued that both research and practical applications of ACCC will increase in the near future using data derived from advanced BIM models and databases. Here, the link between ACCC and BIM methods will be the key to more advanced and effective applications.

References

1. Fenves, S., Goel, S., Gaylord, E.: Decision Table Formulation of the 1969 AISC Specification. University of Illinois, Urbana (1969)
2. Fenves, S.J., Garrett, J.H., Kiliccote, H., Law, K.H., Reed, K.A.: Computer representations of design standards and building codes: U.S. perspective. *Int. J. Constr. Inf. Technol.* **3**(1), 13–34 (1995)

3. Eastman, C., Lee, J., Jeong, Y., Lee, J.-K.: Automatic rule-based checking of building designs. *Autom. Constr.* **18**(8), 1011–1033 (2009)
4. Frye, M., Olynick, D., Pinkney, R.: Development of an expert system for the fireprotection requirements of the national building code of Canada. In: *Proceedings of CIB W78 Conference*, Montreal, Canada, pp. 215–226 (1992)
5. Mugridge, W., Hosking, J., Amor, R.: *Adding a Code Conformance Tool to an Integrated Building Design Environment*. Auckland Uniservices, Auckland (1996)
6. Rosenman, M.A., Gero, J.S., Oxman, J.: An expert system for design codes and design rules. In: Sriram, D., Adey, R. (eds.) *Applications of Artificial Intelligence in Engineering Problems*, vol. II, pp. 745–758. Springer, Heidelberg (1986). https://doi.org/10.1007/978-3-662-21626-2_60
7. Hjelseth, E., Nisbet, N.: Overview of concepts for model checking. In: *27th International Conference—Applications of IT in the AEC Industry, CIB W78 2010*, Cairo, pp. 16–18 (2010)
8. Kiyavitskaya, N., Zeni, N., Mich, L., Cordy, J.R., Mylopoulos, J.: Text mining through semi automatic semantic annotation. In: Reimer, U., Karagiannis, D. (eds.) *PAKM 2006. LNCS (LNAI)*, vol. 4333, pp. 143–154. Springer, Heidelberg (2006). https://doi.org/10.1007/11944935_13
9. Zhang, J., El-Gohary, N.: Automated information extraction from construction related regulatory documents for automated compliance checking. In: *Proceedings of the 28th International Conference of CIB W78*, Sophia Antipolis, France, pp. 1–10 (2011)
10. Zhang, J., El-Gohary, N.: Extraction of construction regulatory requirements from textual documents using natural language processing techniques. In: Issa, R., Flood, I. (eds.) *International Conference on Computing in Civil Engineering*, pp. 453–460. ASCE American Society of Civil Engineers (2012)
11. Turk, Z., Vanier, D.: Tools and models for the electronic delivery of building codes and standards. In: *Proceedings of CIB W78 Conference*, Stanford, CA, pp. 20–30 (1995)
12. Vanier, D.: Computerized building regulations. In: *Proceedings of the International Conference on Municipal Code Administration Building Safety and the Computer*, pp. 43–62. National Research Council Canada, Winnipeg (1989)
13. Bolioli, A., Dini, L., Mercatali, P., Romano, F.: For the automated mark-up of Italian legislative texts in XML. In: Bench-Capon, T.J.M., Daskalopulu, A., Winkels, R. (eds.) *Legal Knowledge and Information Systems: Jurix 2002: The Fifteenth Annual Conference*, pp. 21–30. IOS Press, Amsterdam (2002)
14. Lau, G., Law, K.: 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, p. 11 (2004)
15. Hjelseth, E., Nisbet, N.: Capturing normative constraints by use of the semantic mark-up (RASE) methodology. In: *Proceedings of CIB W78-W102 International Conference*, Sophia Antipolis, France, pp. 1–10 (2011)
16. Nawari, N.O., Alsaffar, A.: Methods for computable building codes. *J. Civil Eng. Archit.* **3**, 163–171 (2015)
17. Construction Specifications Institute: *The CSI Project Delivery Practice Guide*, Chapter 4, Codes, Regulations, and Standards, pp. 51–63. Wiley, Hoboken (2011)
18. Sowa, J.F.: The challenge of knowledge soup. In: Ramadas, J., Chunawala, S. (eds.) *Research Trends in Science, Technology and Mathematics Education*. Homi Bhabha Centre, Mumbai (2006)

19. Crowston, K., Liu, X., Allen, E., Heckman, R.: Machine learning and rule based automated coding of qualitative data. In: Proceedings of 73rd ASIS&T Annual Meeting: Navigating Streams in an Information Ecosystem, pp. 1–2. Association for Information Science and Technology, Silver Spring (2010)
20. Zouaq, A.: An overview of shallow and deep natural language processing for ontology learning. In: Ontology Learning and Knowledge Discovery Using the Web: Challenges and Recent Advances, pp. 16–38. IGI Global, Hershey (2011)
21. Tierney, P.J.: A qualitative analysis framework using natural language processing and graph theory. *Int. Rev. Res. Open Distance Learn.* **13**(5), 173–189 (2012)
22. Caldas, C.H., Soibelman, L.: Automating hierarchical document classification for construction management information systems. *Autom. Constr.* **12**, 395–406 (2003)
23. Zhang, J., El-Gohary, N.: Automated regulatory information extraction from building codes leveraging syntactic and semantic information. In: Proceedings of 2012 ASCE Construction Research Congress (CRC), West Lafayette, IN (2012)
24. Zhang, J., El-Gohary, N.M.: Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking. *J. Comput. Civil Eng.* **30**, 04015014 (2013)
25. Zhang, J., El-Gohary, N.: Automated information transformation for automated regulatory compliance checking in construction. *J. Comput. Civil Eng.* **29**, B4015001 (2015)
26. Lau, G.T., Law, K.: An information infrastructure for comparing accessibility regulations and related information from multiple sources. In: Proceedings of 10th International Conference on Computing in Civil and Building Engineering, Weimar, Germany (2004)
27. Lupo, C., et al.: General XML format(s) for legal sources: Estrella European Project IST-2004-027655. Deliverable 3.1, Faculty of Law. University of Amsterdam, the Netherlands (2007)
28. Boer, A., Radboud, W., Vitali, F.: MetaLex XML and the legal knowledge interchange format. In: Casanovas, P., Sartor, G., Casellas, N., Rubino, R. (eds.) *Computable Models of the Law*. LNCS (LNAI), vol. 4884, pp. 21–41. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85569-9_2
29. Sartor, G., Palmirani, M., Francesconi, E., Biasiotti, M.: *Legislative XML on Semantic Web*. Springer, Heidelberg (2011). <https://doi.org/10.1007/978-94-007-1887-6>
30. Palmirani, M., Governatori, G., Rotolo, A., Tabet, S., Boley, H., Paschke, A.: LegalRuleML: XML-based rules and norms. In: Olken, F., Palmirani, M., Sottara, D. (eds.) *RuleML 2011*. LNCS, vol. 7018, pp. 289–312. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24908-2_30
31. Mazzei, A., Radicioni, D.P., Brighi, R.: NLP-Based Extraction of Modificatory Provisions Semantics, pp. 50–57. ACM Press, New York (2009)
32. Francesconi, E., Montemagni, S., Peters, W., Tiscornia, D.: *Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language*. Springer, Heidelberg (2010). <https://doi.org/10.1007/978-3-642-12837-0>
33. Palmirani, M., Brighi, R.: Model regularity of legal language in active modifications. In: Casanovas, P., Pagallo, U., Sartor, G., Ajani, G. (eds.) *AICOL 2009*. LNCS, vol. 6237, pp. 54–73. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16524-5_5
34. Boley, H., Tabet, S., Wagner, G.: Design rationale for RuleML: a markup language for semantic web rules. In: Cruz, I.F., Decker, S., Euzenat, J., McGuinness, D.L. (eds.) *Proceedings of SWWS 2001, the First Semantic Web Working Symposium*, pp. 381–401 (2001)
35. Wagner, G., Antoniou, G., Tabet, S., Boley, H.: The abstract syntax of RuleML: towards a general web rule language framework. In: *Proceedings of Web Intelligence 2004*, pp. 628–631. IEEE Computer Society (2004)

36. RuleML. The Rule Markup Initiative. <http://www.ruleml.org>. Accessed 01 Apr 2018
37. Cardoso, J.: Semantic Web Services: Theory, Tools and Applications. IGI Global, Hershey (2007)
38. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. *Sci. Am.* **284**(5), 34 (2001)
39. Pauwels, P., et al.: A semantic rule checking environment for building performance checking. *Autom. Constr.* **20**(5), 506–518 (2011)