A FORMAL REPRESENTATION OF BUILDING CODES TO FACILITATE BIM-BASED AUTOMATED CODE-CHECKING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

YİĞİT SEMİH AKBAŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CIVIL ENGINEERING

SEPTEMBER 2019

Approval of the thesis:

A FORMAL REPRESENTATION OF BUILDING CODES TO FACILITATE BIM-BASED AUTOMATED CODE-CHECKING

submitted by **YİĞİT SEMİH AKBAŞ** in partial fulfillment of the requirements for the degree of **Master of Science in Civil Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _______________

Prof. Dr. Ahmet Türer
Head of Department, **Civil Engineering** _______________

Assist. Prof. Dr. Aslı Akçamete Güngör
Supervisor, **Civil Engineering, METU** _______________

Examining Committee Members:

Prof. Dr. İrem Dikmen Toker
Civil Engineering, METU _______________

Assist. Prof. Dr. Aslı Akçamete Güngör
Civil Engineering, METU _______________

Prof. Dr. Rıfat Sönmez
Civil Engineering, METU _______________

Assist. Prof. Dr. Güzide Atasoy Özcan
Civil Engineering, METU _______________

Assist. Prof. Dr. Saman Aminbakhsh
Civil Engineering, Atılım University _______________

Date: 11.09.2019

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Yiğit Semih Akbaş

Signature:

ABSTRACT

A FORMAL REPRESENTATION OF BUILDING CODES TO FACILITATE BIM-BASED AUTOMATED CODE-CHECKING

Akbaş, Yiğit Semih
Master of Science, Civil Engineering
Supervisor: Assist. Prof. Dr. Aslı Akçamete Güngör

September 2019, 155 pages

Building designs need to be checked against related building codes and regulations. This control is mainly a manual process done by government professionals. Therefore, it takes enormous amount of time and may lead to many errors. A more sophisticated system is required for checking building codes and regulations. Automated code-checking systems and tools can be used to aid this process with the help of Building Information Modeling (BIM) tools and these systems are started to be used for efficient design checks against building codes. However, building codes are textual documents that contain complex rule structures. Therefore, developing a formal and computer-interpretable building code representation is a challenging task.

This thesis describes a formal representation of building codes to help interpreting the rules in computer readable format. The proposed representation utilizes Solihin's rule classification as a baseline, and further modifies it for thoroughly representing the building codes. The representation decomposes the building codes into different classes and interpretability conditions in order to ease the BIM-based automated code-checking process.

Utilization of the proposed formal representation to decompose a building code is found as an effective initial process, to represent the related building code in a BIM-based automated code-checking system or tool. The verification of the proposed representation is done through an analysis of Ankara Municipality Housing and Zoning Code as well as implementation of the analyzed building code in a BIM-based automated code-checking tool. Although the analysis of the building code according to the proposed representation is done with success, the results of the implementation process indicates that there are certain limitations that may arise due to code rule complexities, BIM deficiencies, and/or code checking tool restrictions.

Keywords: Automated Code-Checking, Building Information Modeling, Building Code Representation, Code Compliance

ÖZ

YBM-TEMELLİ OTOMATİK YÖNETMELİK DENETLEMEYİ KOLAYLAŞTIRMAK İÇİN YAPI YÖNETMELİKLERİNİN BİÇİMSEL OLARAK MODELLENMESİ

Akbaş, Yiğit Semih
Yüksek Lisans, İnşaat Mühendisliği
Tez Danışmanı: Dr. Öğr. Üyesi Aslı Akçamete Güngör

Eylül 2019, 155 sayfa

Yapı projelerinin ilgili standart ve yönetmeliklere uygunluğunun kontrol edilmesi gerekmektedir. Bu kontrol süreci çoğunlukla resmi merciler tarafından manuel olarak yapılmaktadır. Sonuç olarak, bu süreç büyük miktarda zaman alıp, birçok hataya sebebiyet verebilmektedir. Bu nedenle yönetmelik denetimi için daha gelişmiş bir sistem gerekmektedir. Otomatik yönetmelik denetleme sistemleri ve araçları bu süreci iyileştirmek için Yapı Bilgi Modellemesi (YBM) araçlarının yardımıyla kullanılabilmektedir ve son dönemde bina yönetmeliklerini etkili bir şekilde denetlemek için bu sistemlerden faydalanılmaya başlanmıştır. Ancak, bina yönetmelikleri karmaşık kurallar içeren metinsel dokümanlardır. Sonuç olarak, biçimsel ve bilgisayara aktarılabilir bir yapı yönetmeliği modellemesinin geliştirilmesi zor bir görevdir.

Bu tez, kuralları bilgisayar tarafından okunabilir bir formata aktarmaya yardımcı olmak amacıyla biçimsel bir yapı yönetmeliği modellemesi tanımlamaktadır. Önerilen modelleme, Solihin'in kural sınıflandırmasını esas alarak ve bu kural sınıflandırmasını yeniden düzenleyerek yapı yönetmeliğini tam olarak modelleyebilecek hale getirmektedir. Bu modelleme YBM-temelli otomatik yönetmelik denetleme sürecini kolaylaştırmak için yapı yönetmeliklerini farklı sınıflara ve anlamlandırma koşullarına ayrıştırmaktadır.

Önerilen biçimsel modellemeyi yapı yönetmeliğini ayrıştırmak için kullanmak, ilgili yapı yönetmeliğini YBM-temelli otomatik yönetmelik denetleme sistem veya araçlarında temsil etmede etkili bir başlangıç işlemi olarak bulunmuştur. Önerilen modellemenin doğrulaması Ankara İmar Yönetmeliği'nin analizi ve analiz edilen yapı yönetmeliğinin YBM-temelli bir otomatik yönetmelik denetleme aracında uygulanması ile tamamlanmıştır. Yapı yönetmeliğinin önerilen modellemeye göre analizi başarıyla tamamlanmış olsa da uygulama sürecinin sonuçları; yönetmelik kurallarındaki karmaşıklıklar, YBM eksiklikleri, ve/ya denetleme aracı yetersizlikleri nedeniyle bazı kısıtlamalar olduğuna dikkat çekmektedir.

Anahtar Kelimeler: Otomatik Yönetmelik Denetleme, Yapı Bilgi Modellemesi, Yapı Yönetmelik Temsili, Yönetmeliğe Uygunluk

I dedicate this thesis to my beloved family.

ACKNOWLEDGEMENTS

I am very grateful to my supervisor Asst. Prof. Dr. Aslı Akçamete Güngör for her mentorship, guidance, comments and continuous support throughout this study.

I am grateful to my father Nurettin Akbaş, my mother Hasibe Akbaş and my brother Yavuz Selim Akbaş for their continuous support and encouragement under all circumstances. You always lifted me up when I fell.

I would like to gratefully thank and express my deepest respects to Fahri Levent Arslan for his mentorship and guidance.

Lastly, I am grateful to my girlfriend Ece Dimici for her presence, continuous support, everlasting patience and bringing joy into my life.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1. Introduction

In Architecture, Engineering and Construction (AEC) industry, building designs are checked against the required building codes and regulations. Building codes and regulations are determined and written by government professionals and professional bodies. They could be in various forms such as written texts, tables, equations, etc. Also, they are legally binding so that any building shall comply with the related building codes and regulations. The building codes and regulations are complex documents that include large amount of information. Different types of building codes and regulations exist such as for health and safety, acoustics, fire safety, housing and zoning, etc. In order to get approval for a building project, a building design shall satisfy all of the related building codes and regulations' rules.

Checking building models against building codes and regulations is mainly a manual process done by government professionals. Therefore, this process takes enormous time and may lead to many errors. Moreover, different codes contain information which usually tend to conflict or complement with each other. Therefore, the manual rule checking process is inconsistent (Han et al., 1998). The building codes and regulations checking process is subject to change thanks to the introduction of new technologies in the AEC industry. By utilizing these technologies for code-checking, it is aimed to overcome the subjectivity in building codes and regulations due to different interpretations. In short, a more sophisticated system is required for checking building codes and regulations more efficiently.

In order to accomplish this goal, two technologies shall be used for overcoming the difficulties and inefficiencies pertinent to traditional methods. These technologies are Building Information Modeling (BIM) and automated code-checking.

1.2. Problem Statement

Building codes and regulations are essential for the building designs. They guide building designs towards optimization and standardization. They determine the minimum requirements of building designs to offer sustainable and standardized buildings to the public. The checking process of building codes and regulations are manual and people-centered (Eastman et al., 2009). However, manual checking of building designs and regulations may cause different problems and challenges to both designers and controllers.

The building codes and regulations have a complex nature. They get amended by time and have ambiguities and inconsistencies. Also, they might conflict with each other. Therefore, the basic human nature may lead to different interpretations on the same provision by different users. This results in different applications and acceptance criteria for building designs which is undesirable and inefficient.

The building codes and regulations usually complement each other. Moreover, they include vast amount of information. To completely check a building design, the controller shall have the knowledge of different sets of building codes and regulations. Also, most of the building codes and regulations are written in legal and outdated language which makes them difficult to process. Therefore, checking process of building designs require expertise within different building codes and regulations for the controller and the designer.

The building designs generally contain excessive amount of objects to be checked from different set of building codes and regulations which makes the checking process time consuming and error-prone. Errors resulted from checking of building designs leads to negative building performance which are expensive to fix at later stages. On the other hand, utilization of BIM-based automated code-checking reduces human related errors since the process is automated and this results in lesser changes at construction stage. Therefore, these challenges imply that automated code-checking systems are needed in order to attain efficiency by saving time, preventing errors, and assuring consistency.

Automated code-checking systems require representation of building codes and regulations and also building designs in computer readable format. Its main process includes extraction of necessary components from the design and applying set of rules from building codes and regulations. By utilizing the automated code-checking process, it is aimed to improve the manual code-checking of building designs (Nawari, 2018). Correspondingly, Building Information Modeling (BIM) technologies are widely used to facilitate automated code-checking. This is because the BIM tools allow users to extract relevant information from the building design in computer interpretable format to be used in automated code-checking. Moreover, BIM tools allow to include extended information within each component such as the dimensions, type, material, area, volume, thermal conductivity, opposed to the traditional 2D tools. Moreover, there is a standard neutral representation of building models (Industry Foundation Classes, IFC) to exchange information, although it is not accepted by the few. However, there is not any standard representation of building codes and regulations.

To address these gaps and challenges, in this thesis, a building code representation is developed in order to help digitalizing the building codes and regulations with a formal methodology. The developed building code representation is implemented to one of the main building codes in Turkey which is Ankara Municipality Housing and Zoning Code (AMHZC). Then, a case study is conducted to verify the developed building code representation in a BIM-based automated code-checking tool.

1.3. Motivation

Motivation of this thesis takes foundation of the need for digitalizing building codes in a formal way to take one step further towards standardization. Since building code representations are the base of any automated code-checking system, representing building codes and regulations in a formal way may boost overall performance of automated code-checking systems in many ways as follows:

- Offering a base for the automated code-checking systems to reduce the challenges occurring from ambiguities, conflicts, and different interpretation of building codes
- Classifying rules in order to provide an understanding of the building code structure to ease the building code representation
- Defining complexity structure of building code provisions by analyzing them in a formal manner to ease the code-checking process

Representation of building codes and regulations has been a research area for more than five decades. The initial work has been undertaken by Fenves (1966). Fenves used decision tables to represent the American Institute of Steel Construction's (AISC) standard specifications. He offered "decision tables, an if-then novel programming and program documentation technique" to represent building code provisions in a precise and unambiguous form. This concept has found use in 1969 AISC Specification (AISC 1969) (Fenves et al., 1969). Then, the work by Fenves et al. (1969) further developed with the investigation of different foundations for restructuring the AISC Specification (Nyman et al., 1973). The AISC Specification has been examined at four levels that are "the organizational network for the requirements of provisions, the information network for connecting provisions, the detailed level for representing individual provisions in decision tables, the lowest level for representing basic data in provisions". Although different possibilities for restructuring the specification were presented, the study is summed up with reorganization (Nyman and Fenves, 1975).

Further studies of the topic evolved around assisting standards writers in authoring design standards with four-level representation (Fenves and Wright, 1977). Then, research activities around the subject has been led up to a software system called SASE (Standards, Analysis, Synthesis and Expression). SASE is used for generating and checking decision tables, information networks, classification systems and organizational provisions of building codes. SASE required two sets of information from the user which are classifying the entities of provision applied and classifying attributes required by the provision of building codes (Fenves et al., 1987).

Different approaches has been taken to represent building codes such as object-oriented model. Object-oriented model allows organization of the model according to the building design objects within the building code and gives the opportunity to the designer to select varying classes of data items required for the related building code's pieces. Object-oriented model of a building code is divided into four main groups of objects: the design object heterarchy for describing the building design objects contained by the scope of the building code with more abstract features of related systems and components such as shape, material, function, etc.; the performance-limitation hierarchy for representing the limitations of building design objects such as behavior, spatial clearance, etc. and the data and item hierarchy for representing the properties and features of different information in building codes such as requirements, rules, functions, graphs, etc., and the data item instance network for representing the logic of a building code (Hakim and Garrett, 1993).

Another approach to represent building codes is object-logic model. Object-logic model uses a unified object-oriented logic programming to combine two modules to represent and process building codes (Yabuki and Law, 1993). Also, it is possible to develop formal procedures for formulating to check entirety, eccentricity and conflicts of the building code provisions (Yabuki, 1992). Object-logic model consists of five modules: a standards base for representing the organization of a standard and individual provisions, an object data model for linking CAD model and extracting attributes, a conformance checking module for checking the compliance of a component against provision requirements, a component design module for trial design sections, and a standard analysis module for checking potential problems related to completeness, redundancy and contradictions in the rule sets (Yabuki, 1992; Yabuki and Law, 1993).

The context-oriented model is developed by expanding the object-oriented, object-logic and description logic models. This model attaches the classes to provisions and classes have a number of subclassification hierarchies. Provisions with related classes are also tagged with the subclasses of the class. The collection of subclasses is called "context" (Fenves et al., 1995). Associating the context with provisions eliminated the high number of subclasses (Kiliccote et al., 1994). Further study by Kiliccote and Garrett (1998) used isomorphic modeling which is the representation of a provision by only one computable entity supported by the Standard Modeling Language (SML) to eliminate the loss of locality.

Proposed by Kerrigan and Law (2003), the REGNET project aimed to develop an infrastructure for regulatory information management and compliance assistance. The REGNET project used a logic-based regulation assistance system built on XML-based framework to represent regulations in a document repository which contains a searchable user interface.

Development of rule-based systems has started in the late 1980s (Garrett and Fenves, 1987). With the development of Industry Foundation Classes (IFC) in 1990s, the early research for the building code-checking made possible. Han et al. (1998) and Vassileva (2000) illustrated general occasions and a client-server approach. Han et al. (1999) identified the need for different object views for different types of rule checking. Then, Han et al. (1999; 2002) developed a simulation approach of ADA wheelchair accessibility checking.

One of the earliest automated code-checking systems is Singapore's COnstruction and Real Estate NETwork (CORENET). CORENET was first introduced in 1995 and used on electronic drawings until 1998 when it has started operating on IFC building model data. Then, CORENET is provided with the FORNAX programming language to capture extended information from the building model objects (Solihin, 2004). Another automated code-checking system is Stattbygg's HITOS project. HITOS used Express Data Manager (EDM) model server (EDM website) for storing and accessing data, Solibri Model Checker (SMC) for accessibility checks and dRofus for spatial programming and programming requirements ("dRofus," n.d.). Moreover, Australia's DesignCheck automated code-checking system used object-based rule interpretation and EDM for rule bases. DesignCheck developed an internal model for enhancing IFC model data (Ding et al., 2006). Another automated code-checking system is The International Code Council's SMARTcodes project. The SMARTcodes project uses an International Energy Conservation Code (IECC) dictionary to facilitate definitions of objects and properties needed for automated code-checking. Also, SMARTcodes project offered SMARTcodes builder, a web based application, for rule interpretation. General Services Administration (GSA) has developed Design Assessment Tool (DAT) for circulation and security checks of courthouses. A more recent automated code-checking system is LicA system developed by the University of Porto, Faculty of Engineering. LicA system uses a relational database which includes set of tables to represent objects in water distribution network and calculation properties with hydraulic analysis tool and code-checking feature (Martins and Montiero, 2013). LicA is the only automated code-checking system mentioned in this thesis that does not utilize IFC data model.

Extensive research steps has been taken in the areas of representing building codes in a formal matter and developing automated code-checking systems. However, these research results are still not commonly accepted and used. There are some reasons and challenges behind this situation.

Many provisions in building codes are complex and competent. They require to be represented with more comprehensive building code representations. For instance, according to Fenves et al. (1995), almost 40% of the rules in the 1993 BOCA National Building Code use high order logic. Therefore, current building code representation initiatives are inadequate to express all or most of the provisions in building codes.

In most of the efforts, building code provisions are hard-coded into computer operable rules. They use different programming languages such as C++, java, etc. Therefore, these systems require high level of expertise and are not suitable for the most of the users. Also, building codes and regulations are updated and revised frequently. According to Malsane et al. (2015), The England and Wales Regulations Part L, Conservation of Fuel and Power, has been updated each year for the last four years. Changes in the building codes require changes in the code-checking systems. The hard-coding of provisions into systems require complex interlinks between provisions. Therefore, it is inefficient, time consuming and error-prone to update building code provisions in such systems.

Lastly, there is not any correlation between building codes and code-checking systems. Since, the building codes are competent and have complex nature, they lack the generic vision of building codes. Therefore, the completeness of the building code provisions in a whole matter is flawed.

This thesis mainly focuses on the previously mentioned gaps as its motivation. For this purpose, a formal methodology to represent building codes is leveraged to achieve interlinks between building codes and code-checking systems, for reducing the time-consuming processes and maintaining a generic view of building codes.

1.4. Objectives

The overall objective of this thesis is to develop a formal representation of the building codes based on the steps below:

- Discussing the previous research efforts on the topics of representing building codes and automated code-checking systems to define the limitations and challenges encountered in those efforts.
- Defining a formal methodology to digitalize building codes in order to ease the overall process of automated code-checking.
- Defining a framework to bridge Building Information Modeling (BIM) and automated code-checking with proposed methodology in order to consistently represent building codes.

- Developing frameworks that does not require coding knowledge to allow non-programmer users to update the proposed building code representation in their own way.

1.5. Methodology

The methodology of this thesis consists of seven steps which are explained as follows:

- Previous research efforts will be discussed and evaluated.
- Based on the literature review, formal approach steps will be defined for developing a building code representation to completely cover a building code in automated code-checking process.
- A formal representation of building codes will be developed.
- Ankara Municipality Housing and Zoning Code (AMHZC) will be analyzed according to the formal representation.
- Implementation of the formal representation with the AMHZC using the Solibri Model Checker (SMC) application will be performed in order to determine the limitations and benefits of the proposed approach.
- A case study will be performed to verify the proposed approach.

1.6. Outline

This thesis consists of six chapters which are: introduction, automated code-checking process, building code representation methodology, building code representation, building code implementation and case study and conclusion.

Chapter 1 gives brief information about manual checking process and its challenges, introduces problems and challenges encountered in the building code representations and automated code-checking systems, presents the motivation of this study, objective of the thesis and methodology steps to be taken.

Chapter 2 presents background information about the automated code-checking process, current automated code-checking platforms and review on the automated code-checking systems.

Chapter 3 explains the proposed formal representation methodology of building codes in detail.

Chapter 4 includes the analysis of a building code according to the proposed formal building code representation methodology.

Chapter 5 presents digitalization of the analyzed building code and a case study using a sample building design to validate the proposed building code representation.

Chapter 6 concludes with the study findings, discusses the limitations and challenges of the approach and explains future research about the study.

CHAPTER 2

AUTOMATED CODE-CHECKING PROCESS

2.1. Introduction

There are different terms related to the analyzing and confirming building designs against building codes. Different authors describe this process with different terms such as code-checking, rule-based checking, compliance checking or constraint checking. However, there are minor differences between them and they can be accepted as identical to each other.

All of the terms mentioned above describe a process that does not modify a building design, but rather reviews and validates it. Automation of code-checking is defined by Martins and Montiero (2013) as “the usage of a software to run a code, constraint or rule-based checking routine”. Moreover, according to the Eastman et al. (2009), an automated code-checking system can be; an application tied to a design tool, such as a plug-in, a stand-alone application running parallel with design generating tools, or a web-based application that can accept design from different sources. In general, automated code-checking systems can be identified as systems which apply rules or conditions to the design and report the result of the process with “pass”, “fail” or a warning about the errors.

In this thesis, automated code-checking process is structured into six stages (Figure 2.1). First stage is the analysis of the building codes. In this stage, building code is analyzed and building code provisions are divided into classes using a formal methodology. Then, in the second stage the provisions are converted into digitalized rules by implementing them according to the rule interpretation conditions. Third stage is the digital representation of the building design with BIM tools. In this stage, a building model is prepared for automated code-checking. The building model preparation depends on the analysis of the building code since required Level of Development (LOD) to do code-checking changes from code to code. Fourth stage is the extraction of the necessary information from the building model that will be used for code-checking in Industry Foundation Classes (IFC) data model format. The IFC data model represents a set of internationally standardized object definitions (Nawari, 2018). Usage of IFC is encouraged since it is aneutral model representation for describing a building and used and accepted by most of the researchers and commercial users (Eastman et al., 2009). Fifth stage is where the automated code-checking happens. In this stage, digitalized rules are applied to the extracted data from the building design to execute automated code-checking. The last stage is the reporting stage where the checking results are displayed in different formats such as text, graphical, etc. However, environment for checking the rules and reporting stages will be discussed together since most of the time automated code-checking platforms and systems offer built-in reporting features.

2.2. Analysis of the Building Codes

The building codes are the legalized documents which are controlled by the authorities. The main objective of the building codes is to specify the minimum requirements to efficiently safeguard the health, safety and welfare of the community. There are building codes for different purposes such as seismic code, fire code, housing and zoning code, etc. The building codes differ from country to country and even from region to region since the best practices and the material accesses are different. Therefore, it will be correct to say that the structure of the building codes are not constant.

The building code structure is compiled of provisions. These provisions are used to define the limitations of the building design. Also, provisions occasionally refer to the other provisions in the same or other building codes. The provisions of a building code can be used to define the terms, describe the limitations of the building design or even give reference to the laws. For the purpose of this thesis, the provisions which describe the limitations of the building design will be emphasized. Therefore, from this point on, the provision term will be used to describe the limitations of the building design.

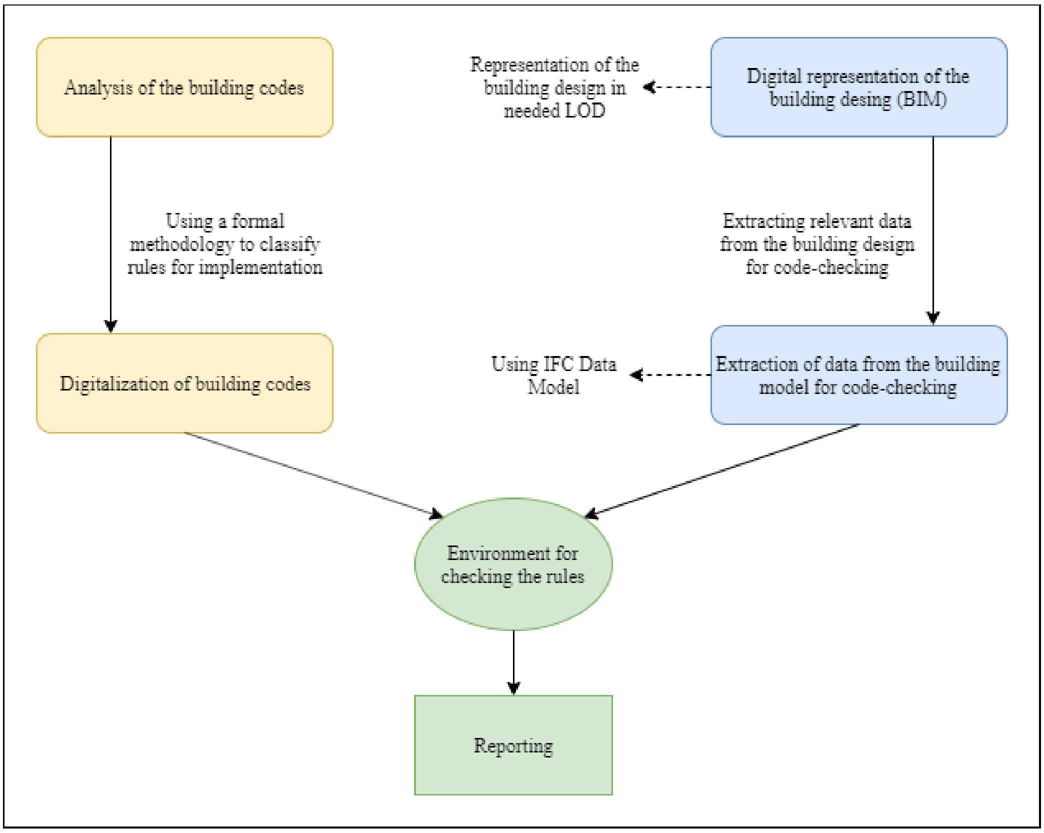

Figure 2.1. Visual Representation of Automated Code-Checking Process

To apply automated code-checking on a building code, the provisions of the building code should be carefully analyzed. Provisions in a building code may require basic data, straightforward derived data or expanded data structure. Also, provisions should be checked if they are computer interpretable or not. The analysis of the building codes is rather ignored by most of the authors working on this topic. However, to apply automated code-checking for a building code, it is necessary to develop a deep understanding of the structure of a building code.

According to Macit (2014), the provisions of a building code is classified into two types of rules which are “self-contained and linked explanatory rules”. The self-contained rules are the exact descriptions of the provisions such as width of a stair, height of a door, etc. The linked explanatory rules are defined as the clarifications, exceptions, exemptions, or modifications of other rules.

According to the Malsane et al. (2015), a filter is applied to the building code rules to reduce the complexity. The building code rules are categorized into declarative and informative clauses. Declarative clauses represent computer interpretable rules while informative clauses represent rules that require human interpretation in order to convert them to computer readable rules and building code rules that are not suitable to categorize are deemed not suitable for automated code-checking.

According to the Solihin (2016), the rules of a building code is classified into four different classes. Class 1 rules are defined as rules that require a single or small number of explicit data. Class 2 rules are defined as rules that need simple derived attribute values. Class 3 rules are defined as rules that need extended data structure. Class 4 rules are defined as rules that require a "proof or solution". However, this rule classification method is inadequate to completely cover a building code and Class 4 rule classification is not informative and not used in most of the building codes.

In this thesis, Solihin's (2016) rule classification will be used as a baseline and further modified in order to completely cover building codes.

2.3. Digitalization of Rules

In the digitalization of rules stage, the rules that are written in human language format are represented in computer readable format or codes. Extensive research efforts have been conducted by different researchers using various approaches for representing the building code rules. Three of the most common approaches are reviewed here which are object-based approach, logical approach, and ontological approach.

2.3.1. Object-Based Approach

Object-based or object-oriented approach is a method that arranges the information. This organization is reached through by representing the object types as the knowledge (Ismail et al., 2017).

In this approach, three stages are defined as building codes classification and abstraction, rule representation modelling and knowledge base establishment (Yang and Li, 2001). As stated in Figure 2.2, the building codes are classified and interpreted while defining all corresponding objects. After that, the connections between different classifications or building codes are established and lastly the facts and values are maintained and stored in a tabular form as an information base (Ismail et al., 2017).

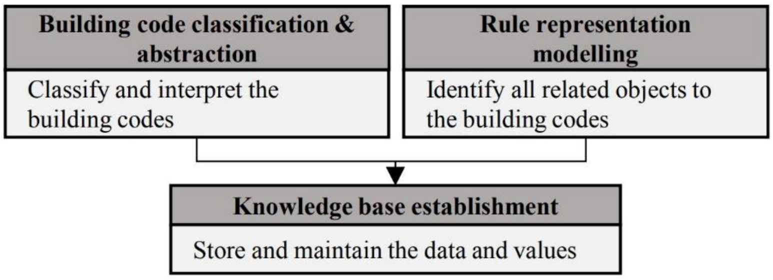

Figure 2.2. Stages of Object-Based Approach (Ismail et al., 2017)

Other studies around this topic also used similar approach as procedure such as Singapore's CORENET system. By utilizing a C++ object library named FORNAX, they made it possible to constitute higher level of information basis for automated code-checking. By using FORNAX objects, the IFC data model is easily extended for representing written rules in computer without the assistance of extensive algorithms (Eastman et al., 2009).

Another example of object-based approach is the Australia's DesignCheck system. DesignCheck system utilizes Express Data Manager which uses an object-based rule engine. This system is similar to the Singapore's CORENET system. However, DesignCheck aims to aid the users in different phases of building design stages such as early stage, detailed stage and specification stage. Each rule in DesignCheck is analyzed and categorized according to the different parameters and object properties such as description, objects, properties, relationships, etc. to ease the automated code-checking process (Ding et al., 2006). A similar approach to the DesignCheck system is taken as an approach known as Fire Codes Checker (Balaban et al., 2012).

In another approach, object-based approach and logical approach have been combined with a rule based engine framework to interpret fall protection and safety for the construction stage (Zhang et al., 2013). The rules identified with three stages. First, the components, properties and connections needed to represent a safety condition are described. Then, the logic required for carrying out the assessment determined. Lastly, method to resolve the safety issues is determined. To extend the automated code-checking process and to enhance the decision-making, rule-based algorithms are utilized. Similar to the Australia's DesignCheck system, this approach also uses rule categorization to further simplify the rules. Each rule is categorized according to the different building objects (Ismail et al., 2017).

2.3.2. Logical Approach

The building codes are fabricated by humans according to a logic. Therefore, mapping written rules to the computer codes is possible through logic. One of the most common approach to represent rules in computer readable codes is first order predicate logic.

The predicate logic has been utilized to supply validation of checking logically with different capabilities such as representation of the checking procedure, computation of rules chosen, different situations of rules and definition of building elements as representations of predicate logic (Lee, 2010). In this effort, the rules of United States Court Design Guide (USCDG) have been applied to the building elements in order to validate and conduct checking. In USCDG, rules are consisted and connected by many logical conditions like 'and', 'or', 'all' and 'if-then'. As a result, checking is done through validating logical conditions to give results either as 'true' or 'false'. In a similar study, KBimCode which contains a collection of building permit needs, used predicate logic to interpret rules (Lee et al., 2016).

Another approach used conceptual graph to represent rules in basic logic structure (Solihin, 2016). An advantage of conceptual graph is that it is possible to use conceptual graph without any knowledge of a programming language. Conceptual graph is used to take the rules, objects, relationships within building design components with constraints. Conceptual graph uses rectangles to represent concept nodes, ovals to represent conceptual relations and diamonds to represent functions. The arcs are used to connect nodes and arrowheads of arcs pointing to the ellipse is the first argument of relation. The last node is described with arrowhead of arc pointing from the ellipse (Figure 2.3).

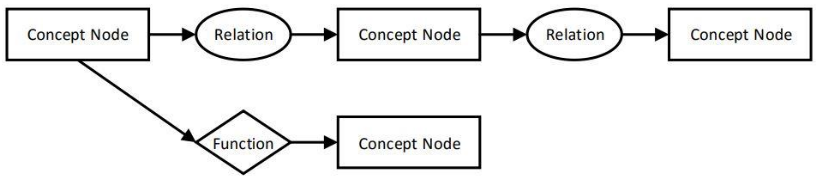

Figure 2.3. Basic Definitions of Conceptual Graph (Solihin, 2016)

Conceptual graph approach consists of four stages to translate rules. Firstly, main concepts of the rule is identified such as space notion. Secondly, atomic sub rules are defined since building code rules often have more than one sub-rule. Thirdly, atomic requirements and limits are determined. Lastly, conceptual graph of the rule which is connecting the concepts using relations and functions is defined (Figure 2.4).

BIM Rule Language (BIMRL) is developed under related study using Structured Query Language (SQL) to ease the automated code-checking process. In this framework, SQL was used to manage the data (Ismail et al., 2017). In Portugal's LicA system, SQL was also utilized (Martins and Montiero, 2013). The data have been controlled with a set of tables consisting of code checking framework, components and outcomes.

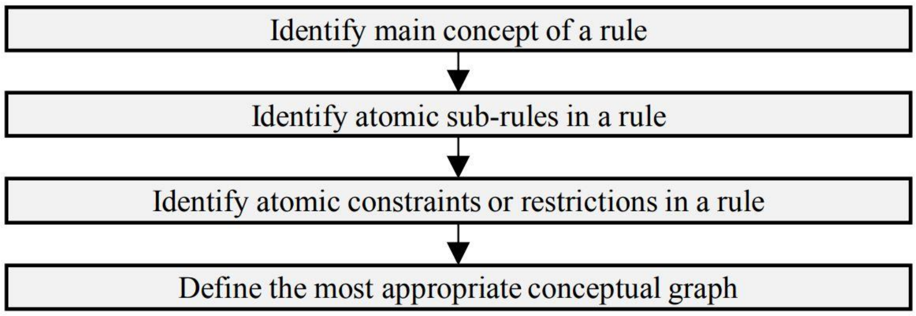

Figure 2.4. Conceptual Graph Four Stages of Rule Translation (Ismail et al., 2017)

In another approach, deontic logic which is a part of modal logic is used to deal with obligations, permissions, etc. (Salama and El-Gohary, 2011). Deontic logic approach uses reasoning to decide whether building model is permitted or forbidden according to the building code. Deontic logic effort has a more advanced capacity to deal with advanced information representation and reasoning than if, then and else logic (Ismail et al., 2017). The reasoning system consists of two layers which are deontology layer and ontology layer. Deontology layer represents laws and codes and reason about the checking of them. Ontology layer represents data about projects and its construction operations. This approach uses deontic conceptualization, deontic logic and Natural Language Processing (NLP) techniques (Salama and El-Gohary, 2011).

Decision table method was first introduced by Fenves (1966) which uses a then-novel programming and programming documentation technique and it is very similar to the logic approach. It has found use in 1969 AISC Specification (AISC 1969). Decision table method is a method of listing logical rules in parameter tables. It can be used to represent higher level of logic in a precise and straightforward format since it contains requirements and relations arranged in a matrix (Ismail et al., 2017). This approach was expanded in order to extend building codes (Tan et al., 2010). In that study, hyperlinks of the building codes have been included.

2.3.3. Ontological Approach

Semantic web technologies are introduced as an alternative approach to the IFC data model for automated code-checking (Pauwels et al., 2011). The semantic approach utilizes World Wide Web (WWW) to represent its semantic network of information. This semantic network describes concepts with a directed, labelled graph (Figure 2.5). Each node in this graph represents a concept or an object while arcs represent logical relation between nodes.

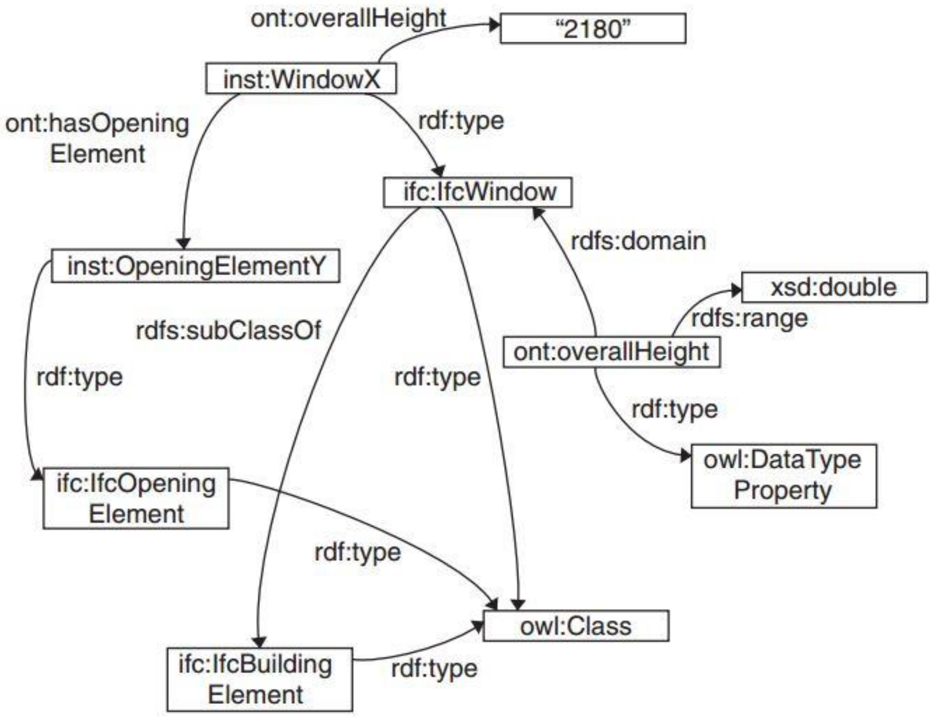

Figure 2.5. Graph Description of Window (Pauwels et al., 2011)

Semantic web approach utilizes the Resource Description Framework (RDF) to represent its graph structure (RDF Primer, 2014). An RDF graph is constructed using AND operator for a series of logical statements consisting of concepts or objects and these statements are also known as RDF triples (Figure 2.6). RDF graphs can utilize an improved semantic structure by using RDF vocabularies or ontologies. Required elements to create such ontologies are described and available in the RDF schema (RDFS) (RDF Schema, 2014). To describe ontologies with more expressive elements, the Web Ontology Language (OWL) is used which utilizes RDFS concepts as a subset (OWL Features, 2009) (Pauwels et al., 2010).

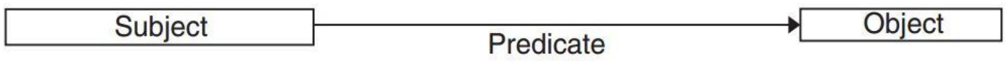

Figure 2.6. An RDF Triple Statement: Subject-Predicate-Object (Pauwels et al., 2010)

A formal ontological approach which is based on the semantic web has been developed by Yurchyshyna et al. (2008). In this approach, four stages are identified for modelling the conformity checking process. First, a knowledge acquisition method is utilized to represent data of checking process. Second, a reasoning model based on graph projection is developed. Third, a method for capitalizing of tacit expert knowledge on the checking process using expert rules is utilized. Lastly, these stages are integrated into the C3R system for validation.

Another ontological approach has been developed by International Code Council. The overall platform is named as SMARTcodes ("Digital Alchemy," n.d.). An International Energy Conservation Code dictionary was developed using SMARTcodes platform which utilizes similar knowledge acquisition method in C3R system. This dictionary is used as a platform of communication between rules and building model which is also similar to the C3R system's reasoning model (Ismail et al., 2017).

A different approach used semantic web technologies with Semantics of Business Vocabulary and Business Rules (SBVR) and SPARQL Protocol and RDF Query Language (SPARQL) to redevelop building codes (Bouzidi et al., 2012). Another approach developed CQIEOntology for the construction quality code-checking against building codes (Zhong et al., 2012).

In a different ontological approach, a concept named RASE was developed. The RASE concept is based on the four operators which are: Requirements (R), Applicabilities (A), Selections (S), and Exceptions (E) (Hjelseth and Nisbet, 2011). These operators can be attributed to contain a topic, a property, a comparator and a target value (Hjelseth and Nisbet, 2010). Requirements are defined with the imperatives ‘shall’ or ‘shall not’ and every check must contain at least one requirement. Applicabilities are defined as the phrases which do not directly related to the building code but can be used to compound building model components. Selections are defined as alternative concepts contained in building code rules. Lastly, exceptions are defined as opposite functions of applicabilities. RASE concept was tested in Norwegian accessibility standard, Dubai building regulation and United States Court design guidance document (Hjelseth and Nisbet, 2011).

The RASE concept is conducted in a method which primarily focused on building codes names as Code for Sustainable Homes (CSH) and Building Research Establishment Environmental Assessment Method (BREEAM) (Beach et al., 2013). In this study, RASE concept has been extended since BREEAM and CSH building codes do not produce pass or fail results. Instead, they give numeric scores from 1 to 5 to check the building model. In a different approach, RASE concept is combined with decision logic to interpret building codes (Kasim, 2015).

2.4. Digital Representation of Building Design and Data

The popularity of Building Information Modeling (BIM) tools is significantly increased over the last decade. The technology of BIM allowed users to visualize and examine in a virtual environment what to be constructed or fabricated (Nawari, 2018). Also, BIM allows detection of any design defects or problems in the building model. According to the National BIM Standard - United States® Version 3 (NBIMS-US, 2015), BIM is defined as “a digital representation of physical and functional characteristics of a facility. A building information model is a shared knowledge resource for information about a facility forming a reliable basis for decisions during its life-cycle; defined as existing from earliest conception to demolition”.

With the increased knowledge and use towards BIM tools, BIM is not only used for visualization and storing properties of building model components, but also for the semantically rich information which is embedded to these components that allows more sophisticated analysis to be achieved such as automated code-checking.

However, to utilize BIM in an automated code-checking system or tool, model requirements of the building design must be determined. In order to determine the model requirements, the aim of the related building code need to be precisely determined. In the next section, the model requirements of the BIM tools will be explained in detail.

2.4.1. Model Requirements

Automated code-checking systems include many data, functions and calculations embedded to them. Therefore, an appropriate building model shall provide automated code-checking systems with related information needed. For example, walls, windows, doors, etc. shall contain related information about structural properties, energy performance, noise reduction, etc. To specify the level of detail that the building model has, Level of Development (LOD) specification can be used. The primary purpose of the usage of LOD specification is to provide minimum necessary information to allow automated code-checking.

In BIM tools, all of the designing phases are model oriented. Different users are using BIM tools for different tasks and purposes. Therefore, to achieve the most interoperability and to reduce data loss by changing data model, different users shall operate on the same data model. The steps have been taken for standardization of data models with the emergence of interoperability and other problems. One of the most recognized and used data model by many of the practitioners and users is the Industry Foundation Classes (IFC) data model.

In the next section, the model requirements for the automated code-checking systems are explained in detail which are LOD specification and IFC data model.

2.4.1.1. Level of Development (LOD)

Building designs contain large sets of data embedded into them. Therefore, the minimum degree of information needed which the data possesses for the automated code-checking is a crucial task.

The Level of Development (LOD) is the degree of information that the building model components have. In short, LOD is how much detail the model has.

LOD has been developed by the American Institute of Architects (AIA) in 2008 as part of E202TM-2008, Building Information Modeling Protocol. Due to the continuously changing nature of BIM, LOD has been updated and reconfigured yearly. According to the AIA Document E203TM-2013 (American Institute of Architechts (AIA), 2013), LOD is described as the minimum dimensional, spatial, qualitative and other data in a building design component to support required uses. Aim of the LOD specification is to assist practitioners in AEC industry to define and specify the building information models with clarity at different stages of design and to standardize the application of LOD in BIM environments (Nawari, 2018).

According to AIA Document E203TM-2013, there are five definitions of different LOD that are LOD 100, 200, 300, 400 and 500. At 2015, BIMForum added the LOD 350 definition to the LOD specification. The definitions of different LODs have been given below (AIA, 2013; "Level of Development Specification," n.d.).

LOD 100

"The Model Element may be graphically represented in the Model with a symbol or other generic representation, but does not satisfy the requirements for LOD 200. Information related to the Model Element (i.e. cost per square foot, tonnage of HVAC, etc.) can be derived from other Model Elements."

LOD 200

"The Model Element is graphically represented within the Model as a generic system, object, or assembly with approximate quantities, size, shape, location, and orientation. Non-graphic information may also be attached to the Model Element."

LOD 300

“The Model Element is graphically represented within the Model as a specific system, object or assembly in terms of quantity, size, shape, location, and orientation. Non-graphic information may also be attached to the Model Element.”

LOD 350

“The Model Element is graphically represented within the Model as a specific system, object, or assembly in terms of quantity, size, shape, location, orientation, and interfaces with other building systems. Non-graphic information may also be attached to the Model Element.”

LOD 400

“The Model Element is graphically represented within the Model as a specific system, object or assembly in terms of size, shape, location, quantity, and orientation with detailing, fabrication, assembly, and installation information. Non-graphic information may also be attached to the Model Element.”

LOD 500

“The Model Element is a field verified representation in terms of size, shape, location, quantity, and orientation. Non-graphic information may also be attached to the Model Elements.”

According to the Solihin and Eastman (2015), for building models at the design development phase, usage of LOD 300 or LOD 350 is generally sufficient for automated code-checking. However, they also mentioned that LOD should be used at minimum level satisfactory for automated code-checking so as to minimize the efforts for modeling.

2.4.1.2. Industry Foundation Classes (IFC) Data Model

Industry Foundation Classes (IFC) data model provides detailed specifications as a set of internationally standardized object definitions (Nawari, 2018). IFC data model is registered by the International Organizations for Standardization (ISO) as ISO 16739 (ISO, 2018). According to the International Organization for Standardization (ISO, 2018), IFC data model is an open international standard for BIM data which allows exchange and sharing of data among different software applications used by different participants. The IFC data model is developed by the International Alliance for Interoperability (IAI) in 1996 (Liebich and Wix, 1999). On 2008, IAI changed its name to buildingSMART.

The IFC data model uses EXPRESS data description language. EXPRESS language is a part of the ISO Standard Exchange of Product (STEP) model (ISO 2016). EXPRESS language is a schema language that provides the specification of classes belonging to a defined domain, the attributes or information of those classes and the constraints on those classes. Also, EXPRESS language allows describing the relations between classes and the constraints applied (Nawari, 2018). EXPRESS language describes every component and different types of terminologies, connections and cardinality of building designs (Kasim, 2015).

An alternative specification is defined as Extensible Markup Language (XML) as ifcXML. ifcXML was developed and released in 2001. XML is readable by both humans and computers and self-descriptive. XML has a simple function of describing different rules to code documents with XML tags and results with a format that can be transferred between both sender and receiver. In summary, XML can be defined as a global format that is used to depict and transform a building design using mapping engine (Kasim, 2015).

The IFC data model is structured into four layers which are: Domain/Application layer, Interoperability layer, Core layer and Resource layer (Figure 2.7). The resource layer is used or referred by classes in the other layer. All information concerning a domain is collected together within the domain's schema and any classes within the Core, Interoperability and Domain/Application layers will reference to this domain. The Core layer is used to describe basic structure of the IFC object model and abstract concepts used by the higher levels of the IFC object model. The core layer has two levels: The Kernel level and Core Extensions level. The Kernel level determines the model level structure and decomposition and provides basic concepts of IFC models. Core Extensions level provides specialization of concepts from the Kernel level. The Interoperability layer is used to provide modules defining concepts and objects to different domain/application models. The Domain/Application layer is used for further model detailing within the AEC domain or different types of applications (International Alliance for Interoperability [IAI], 1999).

IFC data model is considered as a successful schema for providing knowledge interchange and interoperability (Nawari, 2018). Since the 1996 which is the development year of IFC data model, different versions of IFC data model has been developed to further enhance its capabilities (Figure 2.8).

In this thesis, the IFC data model is utilized to transfer information between BIM software and the automated code-checking system. The reason behind this can be described with the need of standardization of the overall process of automated code-checking and IFC data model is one of the most standardized building information data format available in the literature. According to the Yang and Li (2001), for representing the 3D CAD components of building design to conduct automated code-checking features, IFC and its accommodating knowledge representations could be useful due to their standardization, certainty, consistence and entirety of determination of building models.

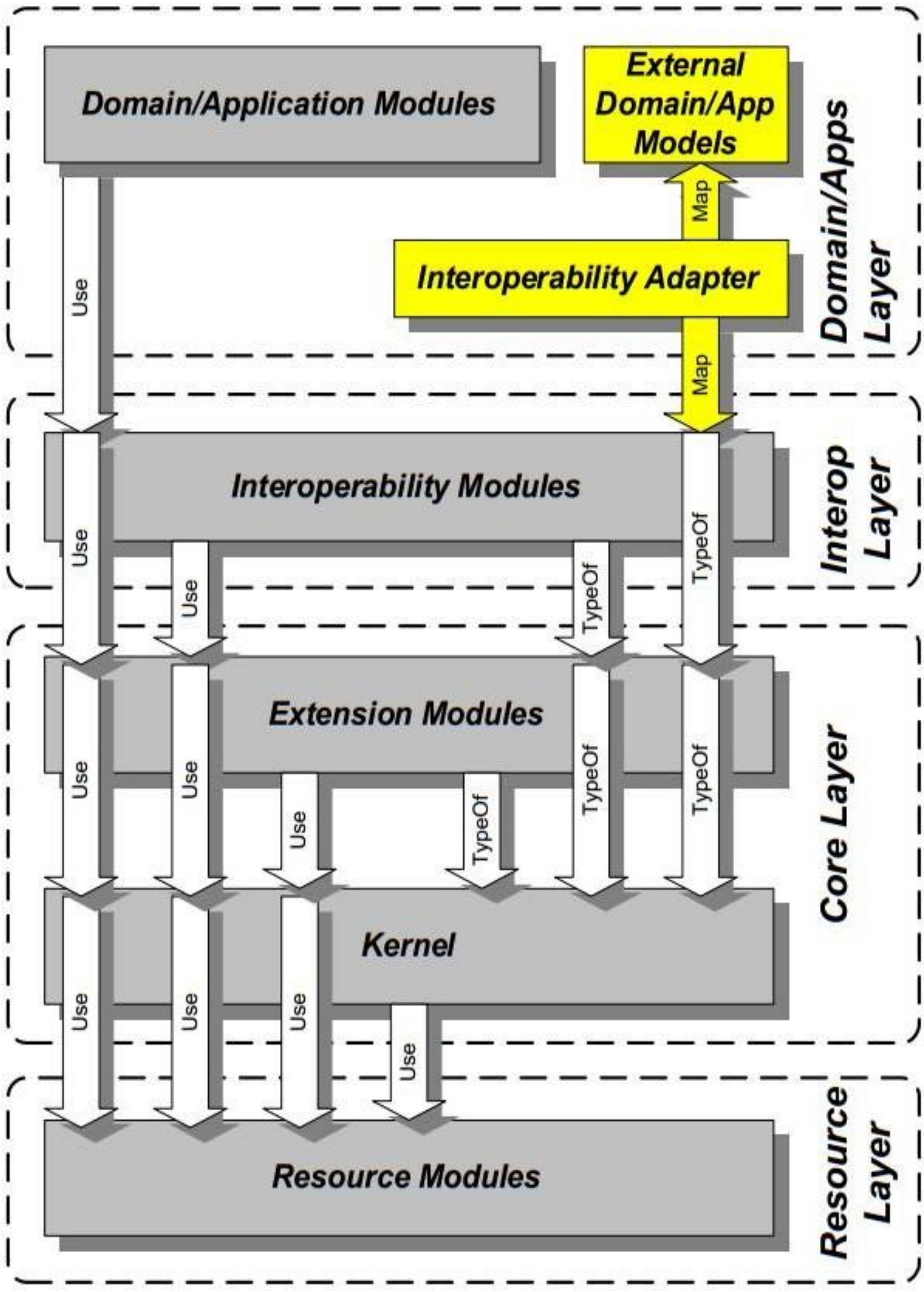

Figure 2.7. IFC Data Model Structure (International Alliance for Interoperability, 1999)

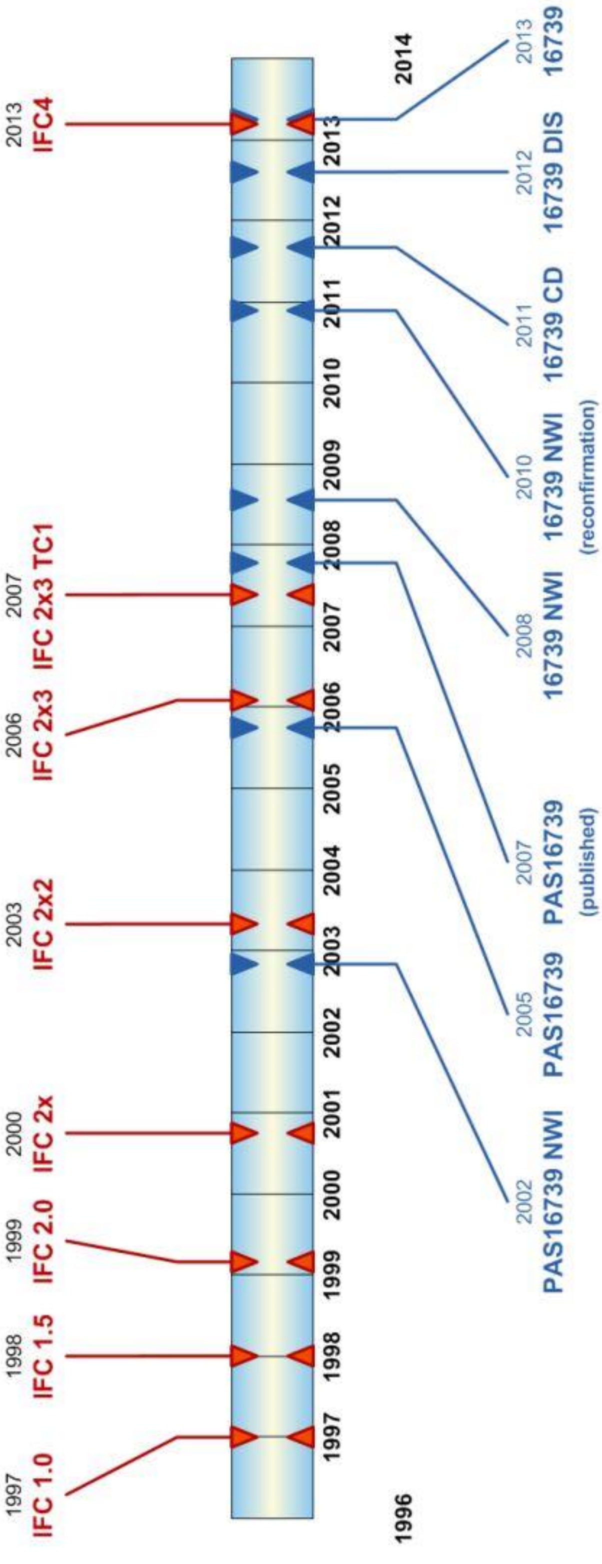

Figure 2.8. IFC Development Time Scale (Liebich, 2014)

2.5. Environment for Checking the Rules and Reporting

In this section, the information from previous sections are gathered in an environment to check their compatibility. After checking, the results are reported. As stated before, since automated code-checking tools mostly offer built-in reporting features, environment for checking the rules and reporting stages will be discussed under the same headline.

There are extensive research efforts conducted by different researchers and governments by utilizing various approaches for checking the rules and reporting the code-compliance. These approaches will be viewed and explained under automated code-checking platforms and automated code-checking systems. The difference between automated code-checking platforms and systems is that, automated code-checking systems are composed of automated code-checking platforms and additional supplementary softwares.

2.5.1. Automated Code-Checking Platforms

Research evolution of automated code-checking platforms for building designs began more than three decades ago (Garrett and Fenves, 1987). However, the technologies in this topic are continuing to evolve towards an actual automated code-checking platform. Still, there is not any automated code-checking platform that covers full functionality to check a building code since most of the efforts are region specific or covering specific rules.

There are a number of automated code-checking platforms developed to support automated code-checking features. They tend to differ from each other in their capability of automated code-checking processes, flexibility in modeling, encoding of building codes, reporting features and integration with other software. In terms of information sharing features, their methods are different. Some of them are considered to be using black box methods which users have limited access to the rule creation engine. Other methods utilize gray box and white box methods which users have varying customization and creation of rules (Figure 2.9) (Nawari, 2018). In this thesis, four different automated code-checking platforms are reviewed which are Solibri Model Checker, FORNAX, SMARTcodes and JOTNE EDModel Checker. These platforms utilize object-based rule engines to apply automated code-checking feautes. All of the platforms discussed here apply rules on the IFC model data as a neutral file format to represent building information.

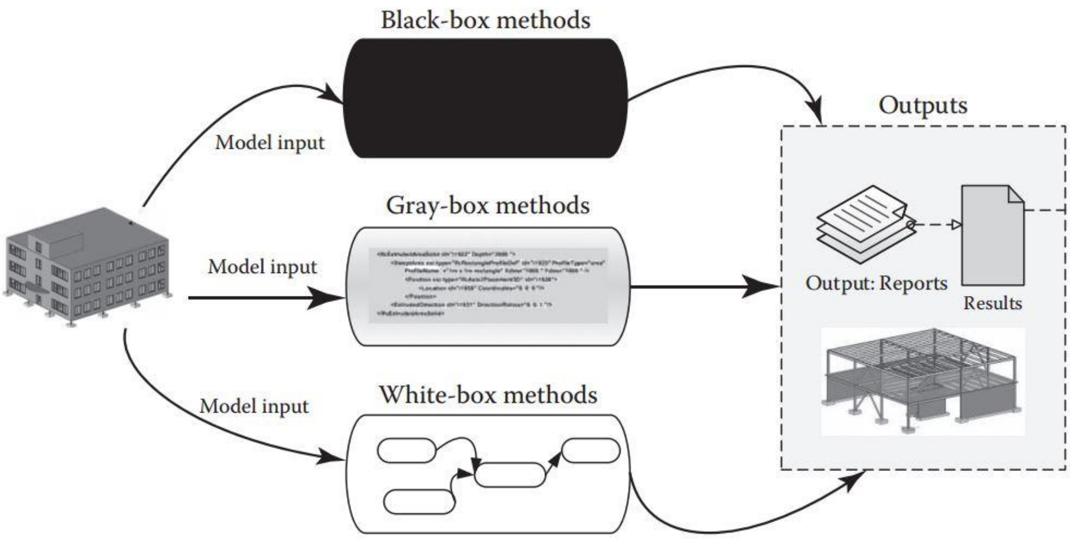

Figure 2.9. Different Platforms of Automated Code-Checking

2.5.1.1. Solibri Model Checker

Solibri Model Checker (SMC) is a java-based desktop platform application. SMC can be used to read IFC data files from different applications and combine them to apply checks on them. It has the ability to check for model clashes, geometric attributes, fire code exits, path distances, space requiements, etc. (Figure 2.10). The main purpose of the SMC to continuously improve the quality of the BIM model during its life cycle.

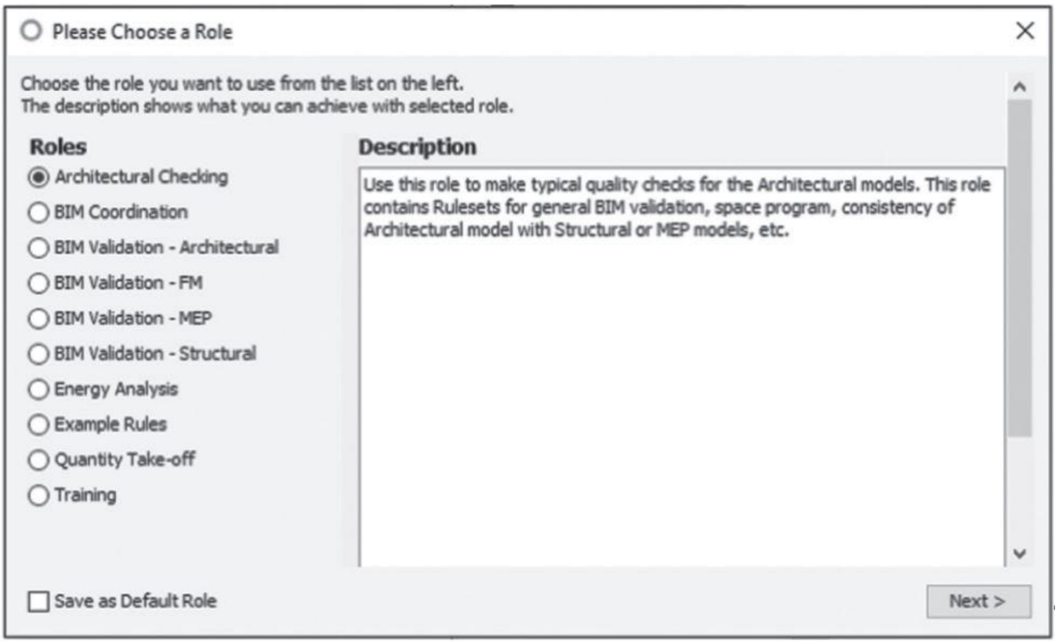

Figure 2.10. SMC Rule Set Selection

Rules can be parametrically customized through table-set control parameters. In short, the users have the ability to change the parameters of constraints in a desired way. However, creation of new rulesets is only possible through Java programming language by utilization of the SMC application program interface (API). But, the SMC API does not have open access and does require expertise of Java programming language and SMC data structure. However, SMC allows users to request new rules throughout their subscription using the Ruleset Manager in order to satisfy the needs of the customer (Figure 2.11) (Nawari, 2018).

SMC require files to be in IFC data model format. Also, it can combine models from different disciplines to perform checks on them. The combined models can be saved in a SMC file format which is more compressed than IFC data model format.

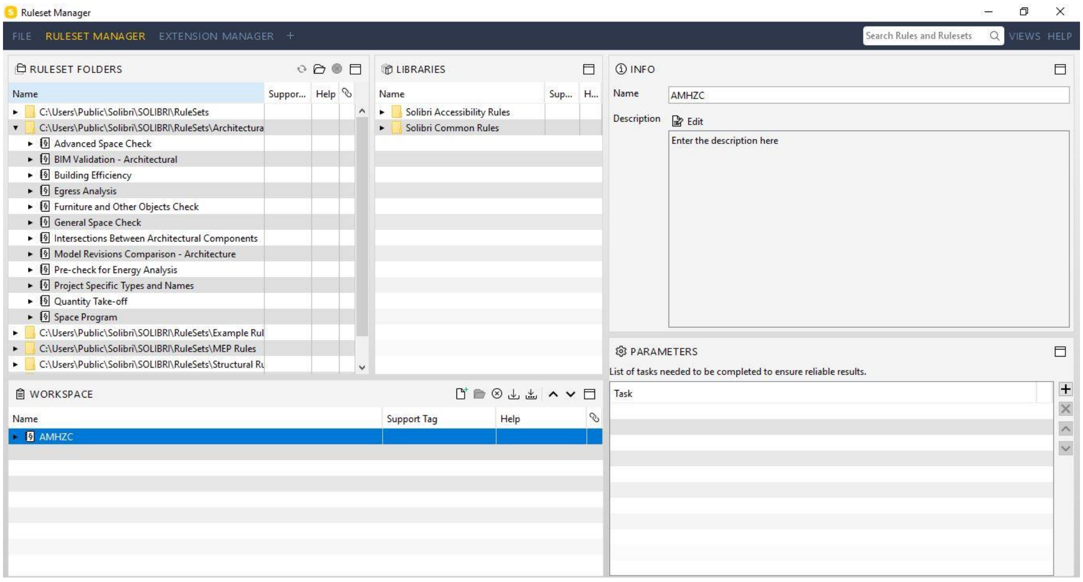

Figure 2.11. SMC Ruleset Manager

SMC is a sophisticated application with vast amount of features to improve the user experience. The building model can be browsed through the SMC user interface (UI). Also, it has the standard model manipulation tools such as zoom, pan, rotation, walkthroughs, etc. SMC allows clash examination of different building elements by toggling on and off related features from UI (Figure 2.12). Any object in the building model can be grouped, highlighted or hided from the view. In SMC, since the model provides a basis of underlying data, any object's information can be viewed such as attributes, location, quantity, area, etc. (Nawari, 2018). Also, SMC can report its results in a visual manner such as pdf, xls or xml file formats (Salama and El-Gohary, 2011).

The issues found in the SMC are usually fixed in the original BIM software tool. Since SMC does not have a direct link between BIM software tools, the process of locating issues in the building model done manually. To address this issue, SMC suggests exporting results in the BIM Collaboration Format (BCF) which is based on XML schema of buildingSMART. BCF allows object or object views in BIM software tools to identify and locate the results automatically. However, BFC is independent from the IFC data model schema. SMC offers tools to compare two versions of the building model to further enhance the interoperability that shows the differences between models (Nawari, 2018). Lastly, SMC is an adequate application to conduct automated code-checking since it does not require specific expertise of knowledge.

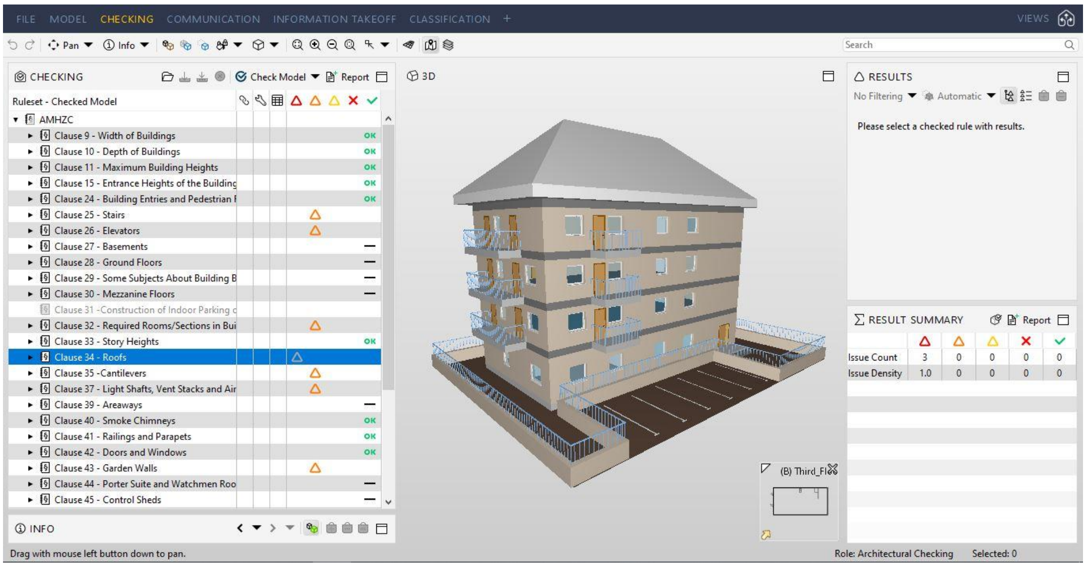

Figure 2.12. SMC UI Code-Checking Results

2.5.1.2. FORNAX

FORNAX is an independent platform that was developed for the Singaporean CORENET project. FORNAX is developed by novaCITYNETS (novaCITYNETS, 2000). FORNAX is a C++ object library which is able to create new information and expanded views of IFC data model. FORNAX is used to augment IFC data model with higher-level of semantics to conform automated code-checking features (Nawari, 2018). FORNAX objects carry rules themselves to asses them which provides good object-based modularity (Eastman et al., 2009). These objects are designed for the customization which allows different codes and conditions to be checked. FORNAX can be used for variety of checks such as building codes, access, fire codes, parking, ventilation, etc. (Kasim, 2015). CORENET performs rule checking in three phase which are; checking with current IFC information, property set extensions to IFC and derived information from IFC (Eastman et al., 2009). Basically, FORNAX objects are used to obtain new information and produce expanded views of IFC to process automated code-checking.

2.5.1.3. SMARTCodes

An automated code-checking platform has been developed by International Code Council which is called SMARTcodes. SMARTcodes allows users to interpret written language rules to the computer codes with the help of a dictionary of field particular terms and semi-formal methods (as cited in Eastman et al., 2009). More descriptive information about the SMARTcodes is given under the International Code Council automated code-checking system.

2.5.1.4. Jotne EDModel Checker

The Express Data Manager (EDM) is an application that was developed by the Jotne EDM Technology in Norway in 1998 (Nawari, 2018). EDM provides users with an object database and support open-access of rule checking by utilizing EXPRES modeling language which is the part of ISO (2016) (Eastman et al., 2009). By using EXPRESS modeling language, users can interpret written rules to the computer codes. EDM also supplies users with modules such as EDMmodelserver and EDMmodelchecker. By using EDMmodelchecker, users can validate a dataset and ensure its conformity to the rules and constraints defined in EXPRESS schema (Nawari, 2018). EDMmodelserver which is an object-based database server allows users to deal with large building model data (Eastman et al., 2009). Also, EDM provides users with textual reporting and server services to enhance the user experience.

In EDM, new model views are possible to develop by utilizing EXPRESS and EXPRESS-X languages which are used for connecting data from different EXPRESS schemas (Nawari, 2018). Also, EDM allows users to utilize Java, C, C# and C++ programming languages to enable creation of application programming interfaces based on the EXPRESS schema (“Express Data Manager,” n.d.).

2.5.2. Automated Code-Checking Systems

In this section, six of the automated code-checking systems are reviewed. Most of these systems are still under development and utilize black box methods for information interoperability. The systems are summarized in Table 2.1.

Table 2.1. *Automated Code-Checking Systems Summary*

Developer	System	Year	Target Rules	Rule Platform
Singapore	CORENET	1998	Building Code	FORNAX
Norway	Stattsbygg	2005	Accessibility	SMC
Australia	DesignCheck	2006	Accessibility	EDM
USA	ICC	2006	Building Code	SMARTcodes, SMC, Xabio
USA	GSA	2008	Circulation and Security	SMC
Portugal	LicA	2013	Water Systems	LICAD

2.5.2.1. CORENET System

CORENET (Construction Real Estate NETwork) is the first code checking system that has been used in practice. CORENET was first developed by the Singapore's Ministry of National Development. CORENET offers an infrastructure to establish information exchange and interoperability between relevant government agencies and real estate industry on different phases of building life cycle such as design, construction and maintenance. CORENET contains three modules which are e-Submission, e-PlanCheck and e-info. Real estate industry professionals use e-Submission system to transfer building designs to system for approval of different type of building permits. e-Info system provides related information to the professionals. e-PlanCheck system allows professionals to check building model against related building codes. Therefore, our main interest among these three modules is the e-Plancheck.

The e-PlanCheck system has used electronic drawings for compliance checking until 1998, then the Industry Foundation Classes (IFC) was introduced into the system for importing building model data. The main objective of e-PlanCheck is to cover building plans and building services compliance checks against related building codes such as fire safety, building design, environmental health, vehicle parking, etc. However, IFC building model only offers a schema for objects in the design. As a result, CORENET is provided with FORNAX. FORNAX is developed by novaCITYNETS which is an e-government solution provider in Singapore. FORNAX is based on C++ programming language which extracts related information from the IFC data to accomplish compliance checking. Basically, FORNAX objects are used to derive new data and produce extended views of IFC to accomplish automated compliance checking (Figure 2.13). Also, by using FORNAX objects, users does not need to use additional algorithms to retrieve certain attributes from the design. CORENET performs rule checking in three phases which are; checking with current IFC information, property set extensions to IFC and derived information from IFC (Eastman et al., 2009).

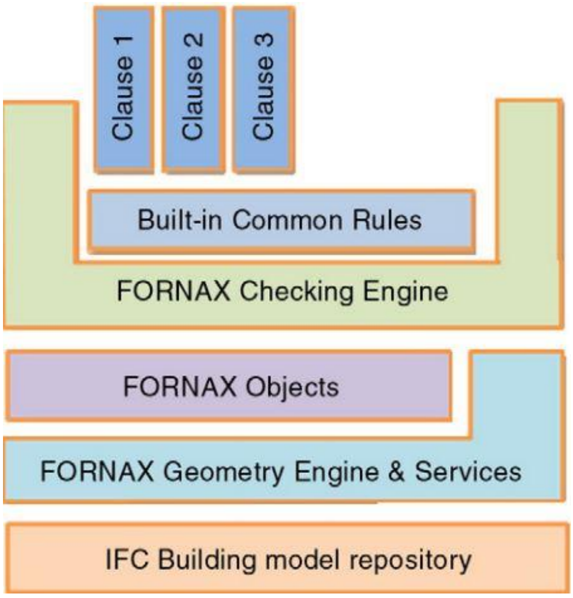

Figure 2.13. CORENET Architecture (novaCITYNETS, 2000)

2.5.2.2. Norwegian Statsbygg's Design Rule Checking

The Singapore's CORENET system was modified and adopted in Norway by the Norwegian Agency of Public Construction and Property (Statsbygg). The modified system was tested in specific projects such as Selvaag Group's "Munkerud" housing project and Akershus University Hospital (AHUS) project for different set of rules (Sjøgren, 2007; Sjøgren, 2005). After these early efforts, with the purpose of extending the use of IFC over overall project life cycle, different platforms was experimentally adopted such as SMC, dRofus, EDM model server and checker, e-PlanCheck, etc. The HITOS project was named after by Tromsø University College and managed by Statsbygg and Tromsø University College since 2005.

HITOS project used EDM model server to store and access building model data using IFC data format. The accessibility rules are parameterized, mapped to objects and executed using Solibri Model Checker (SMC) Ruleset Manager. SMC is used to extract related objects' attributes from the building model. For spatial programming and programming requirements, HITOS utilized dRofus (Figure 2.14) ("dRofus," n.d.).

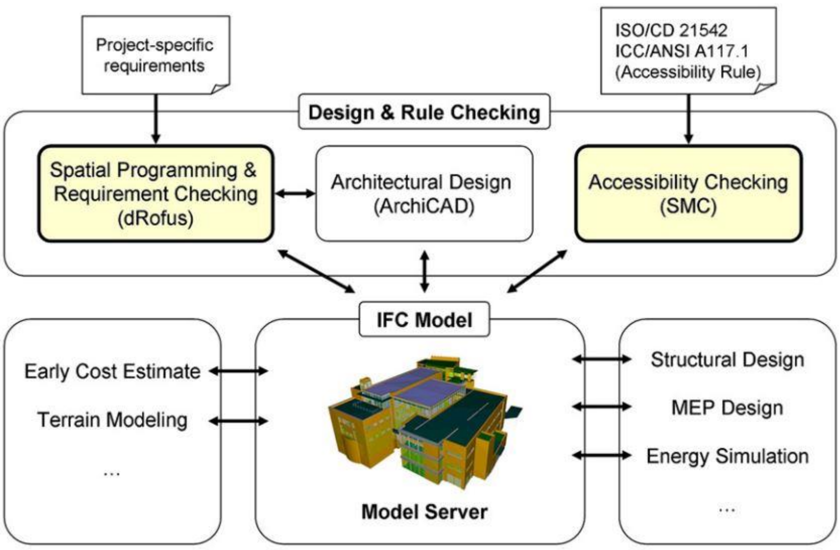

Figure 2.14. Overview of HITOS Project (Eastman et al., 2009)

2.5.2.3. Australia's DesignCheck System

Australia's DesignCheck system was created by researchers of CRC for Construction Innovation (Ding et al., 2006). The DesignCheck system utilizes object-based interpretation to define building codes and Express Data Manager (EDM) rule bases for encoding (Figure 2.15). The DesignCheck systems uses IFC model data for design. However, an internal model developed in order to enhance IFC models to cover enriched information necessary building codes. Also, developed internal model is further structured to cover new attributes.

The system architecture of DesignCheck composes of three tools which are: "main user interface, EDM database and the report system" (Ding et al., 2006) (Figure 2.16). EDM checker utilizes the EXPRESS language to describe rule diagram.

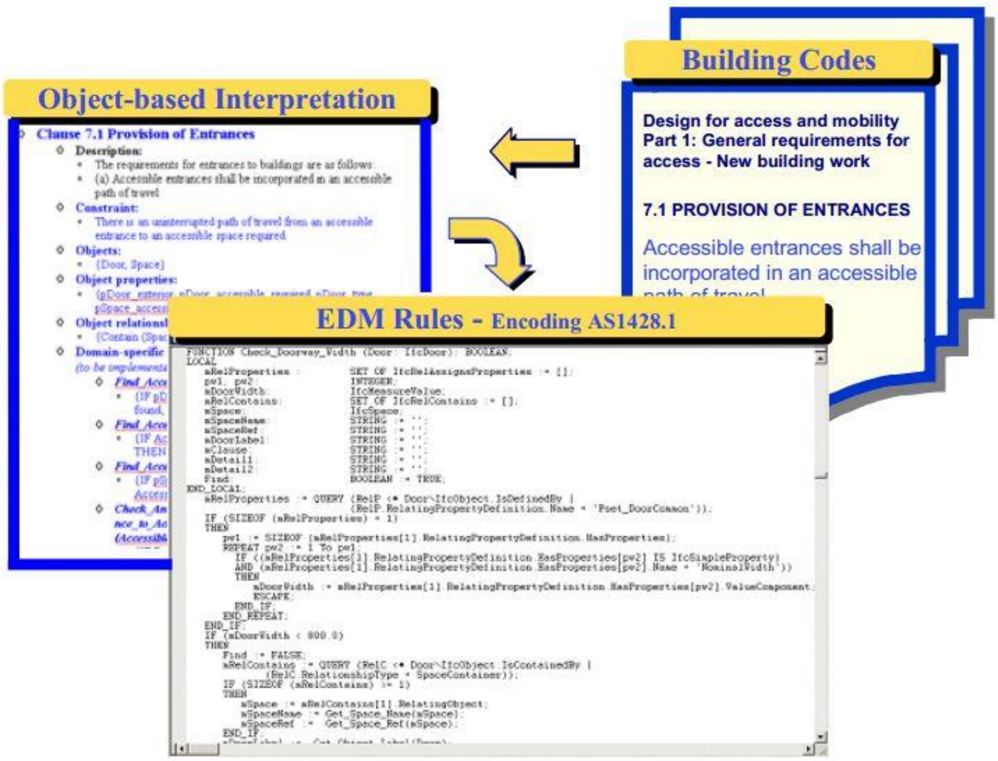

Figure 2.15. Design Check System (Ding et al.,2006)

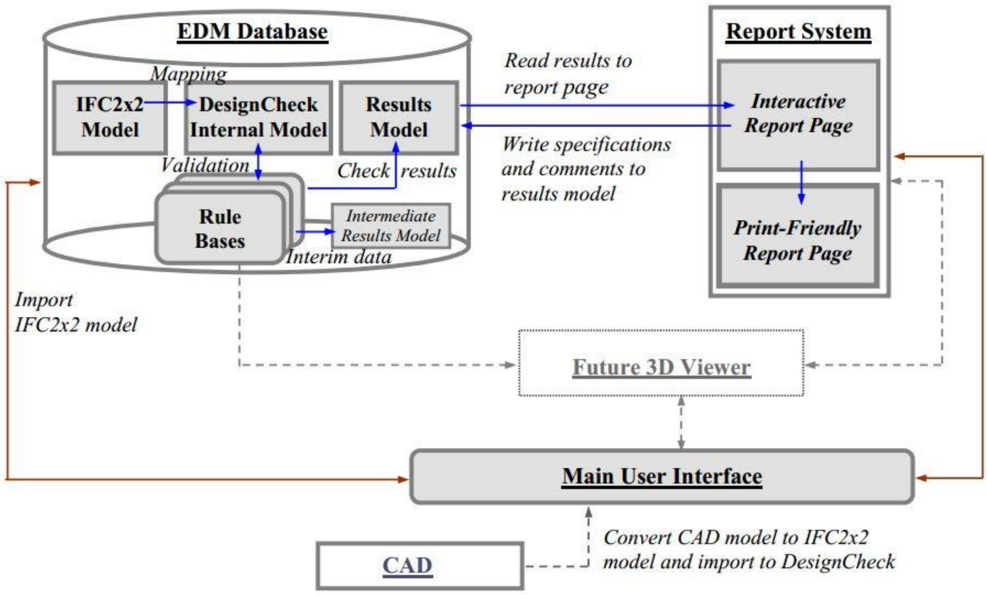

Figure 2.16. DesignCheck System Architecture (Ding et al.,2006)

It can be said that Australia's DesignCheck system is similar to the Singapore's CORENET system. However, DesignCheck system also offers rule check at different phases of building design.

2.5.2.4. International Code Council

The International Code Council (ICC), which is the organization responsible for master building codes in North America, began supporting SMARTcodes project in 2006. The SMARTcodes project was implemented by AEC3 and Digital Alchemy. One of the main objectives of SMARTcodes project to provide link between written codes to computer-interpretable code sets. Developers of SMARTcodes project focused mainly on automating and simplifying automated code checking of federal, local and ICC codes. However, the SMARTcodes project is ended in 2008 due to recession ("Digital Alchemy," n.d.).

The SMARTcodes project has been utilized on International Energy Conservation Code in 2006 ("Projects International Code Council," n.d.). The SMARTcodes project uses an International Energy Conservation Code (IECC) dictionary to facilitate definitions for objects and properties needed for automated code checking.

ICC offered a web-based software, SMARTcodes Builder, to create SMARTcodes. The software allows users to utilize expressions and expressions' logical parts, then formalizing the expressions utilizing the terms from the database (Nawari, 2018). The IECC dictionary is not limited to be used for rule interpretation, also allows information exchange between SMARTcodes system and the IFC data model of the building design (Eastman et al., 2009). SMARTcodes software required users to utilize components like building model, building location, code to be checked and model checking system.

The initial effort, SMARTcodes for SMC, was created by Digital Alchemy and AEC3 ("Projects International Code Council," n.d.). It allowed users to conduct rule-based code checking from ICC 2006 on limited test models such as Lawrence Berkeley National Laboratory, the National Association of Realtor Headquarters building, and four building models provided by the U.S. Coast Guard (Eastman et al., 2009).

Figure 2.17 shows the framework of SMARTcodes concept. The reporting part is supported through the reporting engine of the model-checking software (MCS) (Nawari, 2018). The reports can be viewed in various file formats such as .html, .pdf, .rtf, .xls and .xml (Eastman et al., 2009).

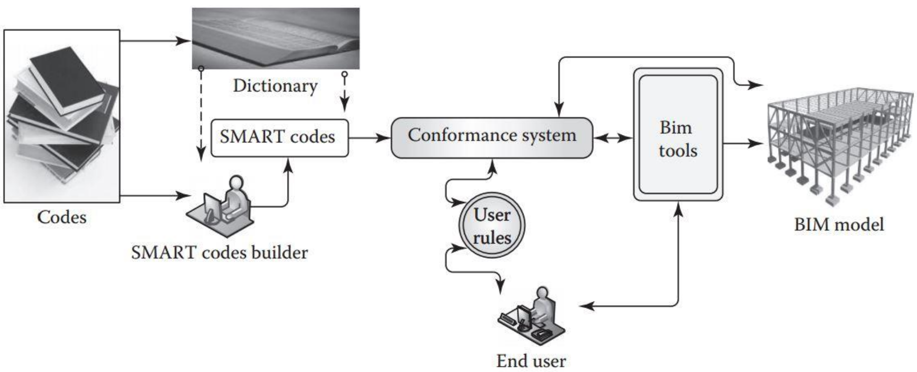

Figure 2.17. SMARTcodes Framework (Nawari, 2018)

2.5.2.5. General Services Administration

The US General Services Administration (GSA) is one of the pioneers of BIM utilization and usage in USA. The GSA started publishing BIM guidelines in 2007 (General Services Admisinistration [GSA], 2007a) and their latest publication was in 2016 (GSA, 2016).

The GSA has started to develop a rule-checking system named Design Assessment Tool (DAT) for circulation and security validation of courthouses. The rules of DAT were extracted from the U.S. Court Design Guide (CDG) (GSA, 2007b). The CDG rules were translated into DAT in 302 statements. The circulation rules of CDG were constructed into four high-level conditions: “start space, intermediate space, destination space and transition” (Eastman et al.,2009). All of the space requirements include a space name and a security level such as public, restricted and secure (Figure 2.18). The rules are specified in a SMC parametric table. Rules defined by the parameters shown in Figure 2.18 become computer-processible by utilizing a SMC plug-in developed by Georgia Tech (Eastman et al., 2009).

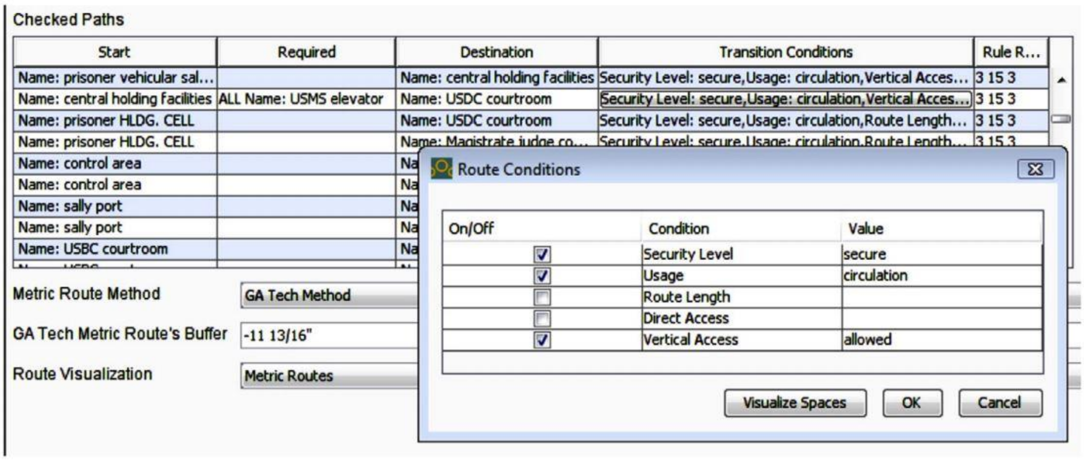

Figure 2.18. Rule Description of DAT System (Eastman et al., 2009)

2.5.2.6. LicA System

LicA is a system which is defined by Martins and Monteiro (2013) as “a system that performs automated code-checking of Portuguese domestic water systems”. LicA has been developed by the University of Porto Faculty of Engineering (FEUP). The framework of LicA system has been showed in Figure 2.19. The LicA system uses a relational database which includes set of tables to represent objects in water distribution network and calculation properties with hydraulic analysis tool and code-checking features. The LicA system database is developed in Transact Structured Query Language (T-SQL) (Martins and Montiero, 2013). The LicA system results are gathered in table for post-processing and the database is accessible through an Open Data Base Connectivity (ODBC).

A 3D graphical user interface (UI) named LICAD was created in order to enter and indicate results which then used to illustrate network model in 3D with color and text explanations. The LicA database and LICAD was developed separately to provide further development, usage freedom, preventing versioning and interoperability. The conceptual relations of LicA and LICAD are represented in Figure 2.20.

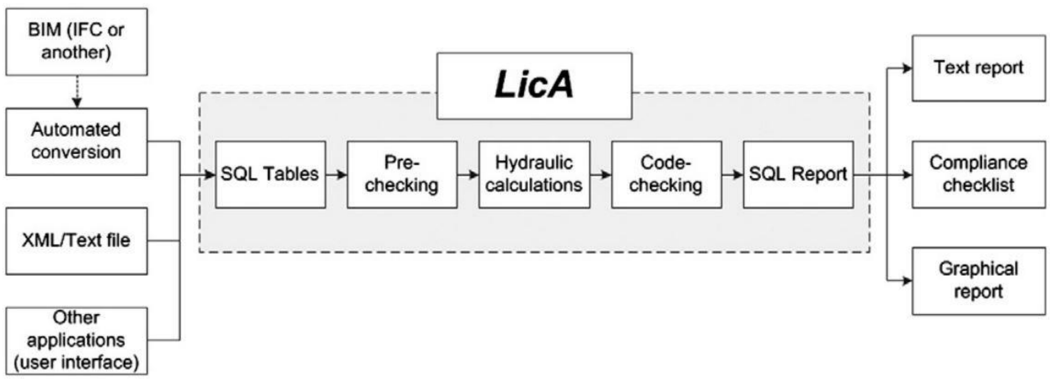

Figure 2.19. LicA System Framework (Martins and Montiero, 2013)

In its current version the LicA system does not support IFC data format. The system and database has been strictly developed for automated rule-checking of Portuguese building codes. Also, LicA system has solution for ambiguities in building codes with developing various sets of groups for code-checking results such as a class of checks done manually (Nawari, 2018).

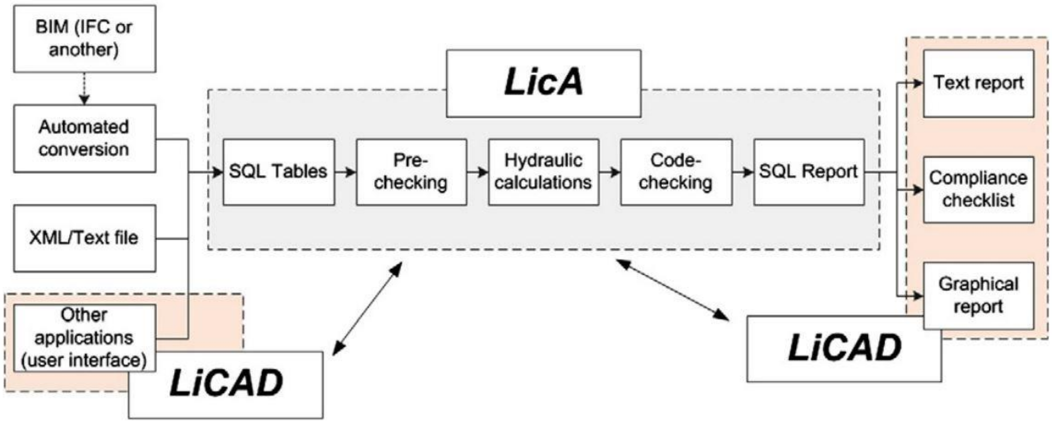

Figure 2.20. Conceptual Relations Between LicA and LICAD (Martins and Montiero, 2013)

The UI of the LicA system is divided to four components (Figure 2.21). In the figure, first tab from the left is the model definition tab which allows user to design the network and navigate the 3D model, the second tab from the left is the hydraulic analysis tab which allows user to run hydraulic calculations, the third tab from the left is the code-checking tab which allows user to check code provisions defined and the fourth tab from the left is the maps and views tab which allows user to generate text-based reports such as quantity takeoffs, hydraulic calculation and code-checking results (Martins and Monteiro, 2013).

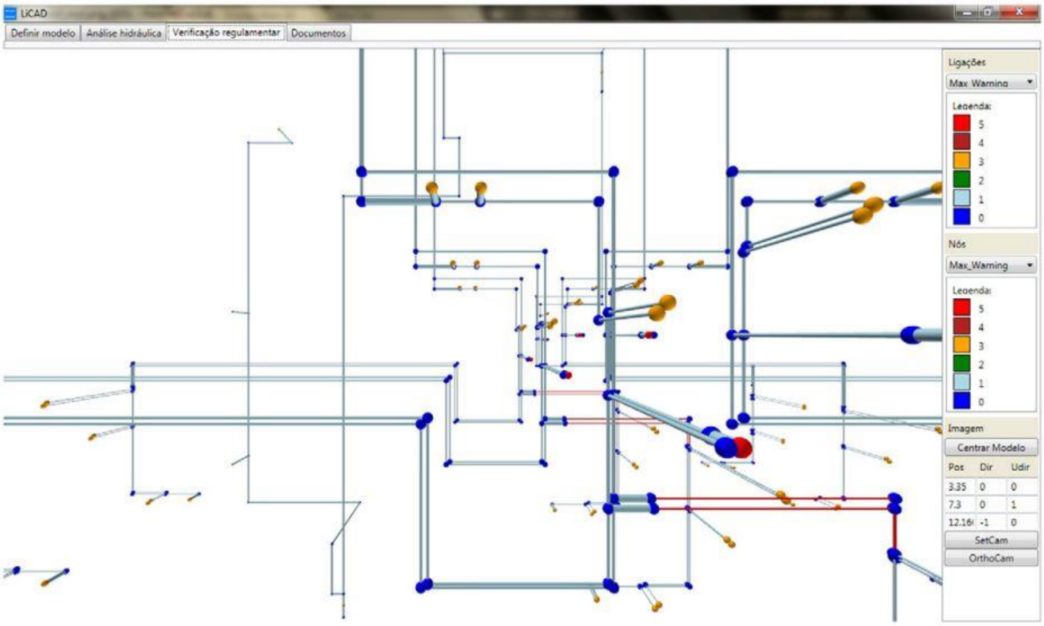

Figure 2.21. LicA User Interface and Code-Checking Results (Martins and Monteiro, 2013)

2.6. Limitations and Needs

In this chapter, the automated code-checking process and its components are explained in detail. Moreover, studies related to the automated code-checking process and its components are summarized. However, most of studies presented in this chapter require expert knowledge like programming languages such as C++, java, EXPRESS, etc. Also, most of the efforts are using black box methods which do not share the details of the code checking process with the users. Therefore, it is difficult for non-experts to conduct automated code-checking using these systems and platforms. Also, most of the automated code-checking systems and tools are focusing on the selected building codes which makes them not compatible with other building codes than the selected ones. Therefore, application of automated code-checking on different building codes is cumbersome and limited. Moreover, the tools or systems used for digitalizing and representing the rules mostly focus on specific/selected rules which are inadequate to completely cover building codes. Therefore, an integrative work is needed to represent a building code completely and to allow non-experts to utilize automated code-checking.

In the next chapter, a building code representation methodology is defined and explained in detail to represent a building code completely.

CHAPTER 3

BUILDING CODE REPRESENTATION METHODOLOGY

3.1. Introduction

This thesis presents a building code representation methodology for analyzing building codes in order to ease the automated code-checking process. The proposed methodology is used to classify the building code rules in a formal manner and to check their interpretability conditions to categorize whether the rules can be expressed in a computer readable format or not. The proposed building code representation methodology firstly analyses the building code and decomposes it. Then, it organizes the building code rules into different classes such as Class 1, Class 2, Class 3, and Reference rules and into different interpretability conditions such as Interpretable, Semi-Interpretable, and Non-Interpretable rules. Lastly, the developed building code representation is utilized to determine necessary variables in order to process code-checking in automated systems and platforms in a more efficient and precise way. For instance, determination of LOD, decision of which automated code-checking system or platform will be used for automated code-checking, or prediction of external software solutions required for completely checking a building code, are carried out.

3.2. Building Code Analysis

Building codes are written in human language and contain complex information. Therefore, to interpret a building code in computer readable format, first it is necessary to develop a deep understanding of it. Most of the automated code-checking studies focus on how to represent written rules in computer readable format. However, these studies ignore the natural characteristics of the building codes. Building codes are not independent documents and they occasionally refer to other building codes. Also, building codes do not always present information in a formal expression. Moreover, the application of the building codes rely heavily on the controller's experience and judgment. To accomplish automated code-checking, analyzing first the building code is a significant and needed task. In this section, the typical building code structure is investigated.

In most of the building codes, from general to the specific, there are chapters, clauses, statements, and rules (see Figure 3.1). Building codes' main headlines are chapters. Chapters are the most general concepts in a building code. They frame and divide the building code into main parts and refer to the subjects generally. These chapters contain many clauses. Clauses are the parts of chapters which define specific subjects in a general manner. Statements are the parts of clauses and they refer to the specific conditions and situations in a specific subject defined by the clauses. Rules may refer to the statements or they can be a part of the statements. The rules are the smallest parts of building codes. The detailed definitions and examples about the structural components of the building codes are explained in the following headlines.

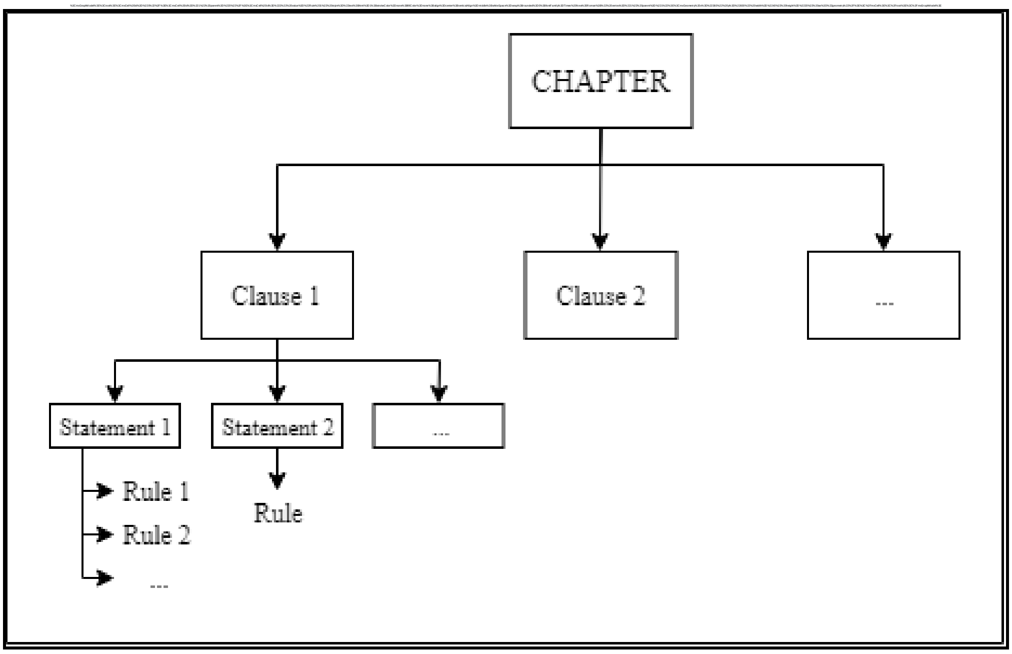

Figure 3.1. Building Code Sections From General to Specific

3.2.1. Chapters

Chapters constitute the main structure of building codes. They frame the main concepts in a building code such as general principles, land related principles, building related principles, projects, building permits, etc. They may contain different sets of clauses. For instance, AMHZC Chapter III Land Related Principles cover clauses from 6 to 8 while AMHZC Chapter V Building Related Principles cover clauses from 23 to 53. Chapters in a building code determine the scope of the subject inside.

3.2.2. Clauses

Clauses constitute the main structure of chapters. They refer to the specific subjects in a chapter. Also, they may belong to different related areas. For instance, AMHZC Chapter V Building Related Principles Clause 9 includes the width of the building information related to the building, while Clause 16 includes the levelling information related to the parcellation.

3.2.3. Statements

Statements constitute the main structure of clauses. They refer to more specific subjects in clauses. Statements in a building code are used to distinguish the clause subject into related smaller pieces which helps organizing the building code for easier understanding. For instance, in AMHZC Clause 32, statement 1 was used to refer to the residential buildings, statement 2 was used to refer to the commercial buildings and statement 3 was used to refer to the public buildings.

3.2.4. Rules

The rules are defined as the smallest parts of building codes. In other words, the rules are the foundation of the building codes. They refer to single or multiple information about a single specific subject. They may be a part of a statement or be the statement itself. They are used for the code-checking processes in automated code-checking systems. Different classes of rules and interpretability of rules are explained in the following sections.

3.3. Interpretability Conditions of Building Code Rules

Since the building codes are written in human language, the logic and connections within the building codes may be complex and even impossible to interpret in computer readable format. Some of the rules in a building code may be ambiguous and even contradictory to another rule. Also, some rules may only be definitions of terminology. Interpretability conditions check on the rules will improve the time efficiency of the building code analysis since non-interpretable rules and definitions will be excluded from the analysis. Therefore, the interpretability check of the building code rules are required and helpful. In this section, the interpretability conditions of building code rules will be explained. However, the interpretability conditions defined here reflect the present situation of the related technologies. Further studies and technologies may improve the interpretability conditions of the building code rules. The interpretability of building code rules are divided into 3 general conditions which are Interpretable, Semi-Interpretable, and Non-Interpretable rules. The definitions and the examples of the rule conditions are explained below.

3.3.1. Interpretable Rules

Interpretable rules are the simple rules that represent straightforward interpretation to computer format. These rules are unambiguous and most of the time modeled with a single step. They contain precise concepts and can be directly interpreted to the computer readable format. However, they can contain operators such as “if-then”, “or” and “and”. Some examples of Interpretable Rules from the Ankara Municipality Housing and Zoning Code can be found below:

> “Control cabinets’ areas cannot exceed 9 m^2.” (AMHZC, Clause 45-Statement 2)

> “The stair steps’ height and width are found using $2h + b = 61 – 65$ function. (h) is the step height as cm and (b) is the step width as cm.” (AMHZC, Clause 25-Statement 2)

> “Building entrance corridor width until it reaches main stairs and elevators is minimum 2.20 meters for public buildings and minimum 1.50 meters for other buildings.” (AMHZC, Clause 24- Statement 1)

3.3.2. Semi-Interpretable Rules

Semi-Interpretable rules contain ambiguous concepts that obstruct the interpretation process. They require human clarification in order to convert them into Interpretable rules. Most of the time they present fuzzy and subjective information that can be differently interpreted by different experts and as a result of this difference the application of the related rule may be subjected to change. Some examples of Semi-Interpretable rules from the Ankara Municipality Housing and Zoning Code can be found below:

> “Toilets must have enough number of urinals and sinks.” (AMHZC, Clause 50- Statement 3)

> “Doors cannot have sill. In compulsory situations that require construction of door sills, the precautions must be taken to allow disable person movement, fire exits and similar actions.” (AMHZC, Clause 42- Statement 2)

> “Elevators must have unblocked access from the building entrance according to the accessibility standards.” (AMHZC, Clause 26- Statement 10)

> “Elevators must have necessary equipment according to the accessibility standards.” (AMHZC, Clause 26- Statement 11)

3.3.3. Non-Interpretable Rules

Non-Interpretable rules are the rules that are not interpretable to computer format. These rules depend on aesthetics, definitions of terminology, and evaluations of authority. As a result, these rules are impossible to represent in computer readable format due to their open-ended structures. Also, Non-Interpretable rules cannot have any rule classification embedded to them. Some examples of Semi-Interpretable rules from the Ankara Municipality Housing and Zoning Code can be found below:

> “The roofs must be similar to the characteristics of the neighboring streets and designed according to the building to be constructed.” (AMHZC, Clause 34- Statement 1)

> “Related municipalities are authorized to determine the air conditioner locations and closing the installation areas after the last story according to the building architecture.” (AMHZC, Clause 35- Statement 2)

> “Shunt chimneys: Main chimneys rising from ground floor to the roof and every unit that are connected are called shunt chimney. Natural gas devices cannot connect to this type of chimneys.” (AMHZC, Clause 40- Statement 2)

3.4. Classification of Building Code Rules

In this section, the building code rules are classified. The rule classification process is based on the rule classification study of Solihin (2016). According to the Solihin’s (2016) rule classification study, the rules are divided into four general classes which are Class 1, Class 2, Class 3, and Class 4 rules. Class 1 rules are defined as rules that require a single or small number of explicit data, Class 2 rules are defined as rules that require simple derived data, Class 3 rules are defined as rules that require extended data structure, and Class 4 rules are defined as rules that require a “proof of solution”. However, this rule classification is inadequate for covering a building code completely, as building codes complement each other. There are some rules referencing different standards, codes, and regulations. By using this rule classification, some of the rules in a building code are left uncovered since it is not possible to ignore rules. Also, Class 4 rules defined in this study are limited to certain building codes and may require additional building modeling during different stages of construction. An example is the Occupational Safety and Health Administration’s (OSHA) protection from falling in construction rules (OSHA, 2015) which requires modeling of protective equipment during the construction of the project. However, most of the building codes focus on checking as-built building models which do not allow adding new features to the existing building model. Therefore, Class 4 rules have limited use in building codes.

In this thesis, a modified version of the rule classification of Solihin (2016) is used. The modified version of rule classification consists of four general classes which are Class 1, Class 2, Class 3, and Reference rules. By using the modified version, the rule classification of Solihin (2016) is aimed to be improved to wholly analyze building codes. The modified version of rule classification takes out Class 4 rules and adds Reference rules to the rule classification process. As a result, the modified version of the rule classification will be able to cover a building code completely. The definitions and the examples of the rule classes are explained below.

3.4.1. Class 1 Rules

Class 1 rules are the most basic rules that are classified. This type of rules mostly require explicit attributes and entity references that can be found inside the BIM model. The information of Class 1 rules is explicitly available from the building model entities or properties of the building model components. Some examples of Class 1 rules from the Ankara Municipality Housing and Zoning Code can be found below:

> "According to the implementary development plan, buildings with 3 stories shall leave space for the elevators and buildings with 4 or more stories shall utilize elevators." (AMHZC, Clause 26-Statement 1)
>
> "Door heights cannot be shorter than 2.10 meters." (AMHZC, Clause 42-Statement 1.a)
>
> "On road fronts, consoles cannot be wider than 1.5 meters" (AMHZC, Clause 35-Statement 1.a.1)
>
> "Slope of the roofs cannot exceed 40 %." (AMHZC, Clause 34- Statement 1.a)
>
> "Effective chimney height is minimum 4 meters" (AMHZC, Clause 40- Statement 3.g)

3.4.2. Class 2 Rules

Class 2 rules cover the basic data or small set of derived values from the building model attributes or properties. This type of rules require simple derivations which does not generate new data from the building model. In other words, Class 2 rules operates between different properties or attributes to reach the required state. The information of Class 2 rules are not explicitly presented by BIM tools but implicitly gathered from the building model data and relationships. Some examples of Class 2 rules from the Ankara Municipality Housing and Zoning Code can be found below:

> "According to the given story number, eave elevation (H= building height, n= number of stories except ground floor)
>
> a. On the regional floor defined area residential and commercial parcels; H=3.50 + n x 3.00
> b. Other floor defined residential parcels H= 5.00 + n x 3.60
> c. Other floor defined non-residential parcels H= 5.50 + n x 4.00
>
> can be calculated according to the given formulas." (AMHZC, Clause 11-Statement 1.a-b-c)

> "The stair steps' height and width are found using 2h + b = 61 – 65 function. (h) is the step height as cm and (b) is the step width as cm." (AMHZC, Clause 25-Statement 2.a)

3.4.3. Class 3 Rules

Class 3 rules are the most complex rules that are classified in this thesis. This class of rules require an extension to the building model data for providing improved conditions. Class 3 rules may use more than one data structure strategies. Also, they may involve complex geometric, topological and different algorithms. To solve Class 3 rules, extensive software tools are generally used. Some examples of Class 3 rules from the Ankara Municipality Housing and Zoning Code can be found below:

"Parcels which do not have proper geometric shape will be converted to the basic and proper shape by using area balancing, then after this process the depth of this shape will be accepted as the parcel's average depth." (AMHZC, Clause 10- Statement 1.a.2)

"In case of Rooms'/Sections' areas are not geometrically arranged, a square must be able to be fitted according to the given minimum side length in this clause's statement 1.a." (AMHZC, Clause 32- Statement 1.c)

3.4.4. Reference Rules

As mentioned before, the building codes complement each other. Therefore, building codes often reference to other codes. Reference Rules are used to cover the references given in the building codes. Reference Rules are not the focus of this thesis and they will not be further investigated. Reason behind this is that most of the time reference rules completely cover the building codes that it refers. Some examples of Reference Rules from the Ankara Municipality Housing and Zoning Code can be found below:

"At buildings and their entrances, the access of disabled persons must be configured according to the TSE standards." (Clause 24- Statement 7)

"At public buildings, all structure, facility and public area arrangements are done for the access and usage of disabled persons according to the TSE standards." (Clause 32- Statement 3.a)

"Garbage shutes' design and arrangement are done according to the TS 2166." (Clause 38- Statement 2)

3.5. Building Code Representation Methodology Framework

Representation of the building codes are not a straightforward process due to the complex nature of building codes. To accomplish this task, there are several steps that must be taken in order to represent a well-defined process. Each step defined below explains the pathway to the desired milestone in the framework.

- Determination of scope: Definition of what to be represented from a building code.
- Building Code Analysis: Extraction of related clauses, statement,s and rules according to the scope of the automated code-checking.
- Interpretability of rules: Determination of interpretability conditions of rules obtained through the building code analysis step.
- Classification of rules: Determination of rule classes of the Interpretable and Semi-Interpretable rules.
- Determination of necessary variables according to the results: Determination of the building model LOD and decision of which automated code-checking system to use according to the results from previous steps.

In the next sections, the components of this framework will be explained in detail. Figure 3.2 illustrates the process of the proposed building code representation methodology framework.

3.5.1. Determination of Scope

The first step of the building code representation methodology is the determination of scope. As mentioned before, building codes are written by humans considering human interpretation capabilities and they contain complex sets of information. They also consist of different types of information which may be out of scope for the automated code-checking process. For instance, clauses in a building code may contain information about the building, construction issues, land adjustment as well as general issues, terminology, definitions, scope, aim, legal basis, permit issues. Therefore, the building code must be separated from out of scope information and rules. By applying this step, the building code will be reduced to the relevant information about the automated code-checking focus. The information gathered from this step will be then used in the automated code-checking process. After applying this step to a building code, the information is reduced to the relevant chapters and clauses within the determined scope.

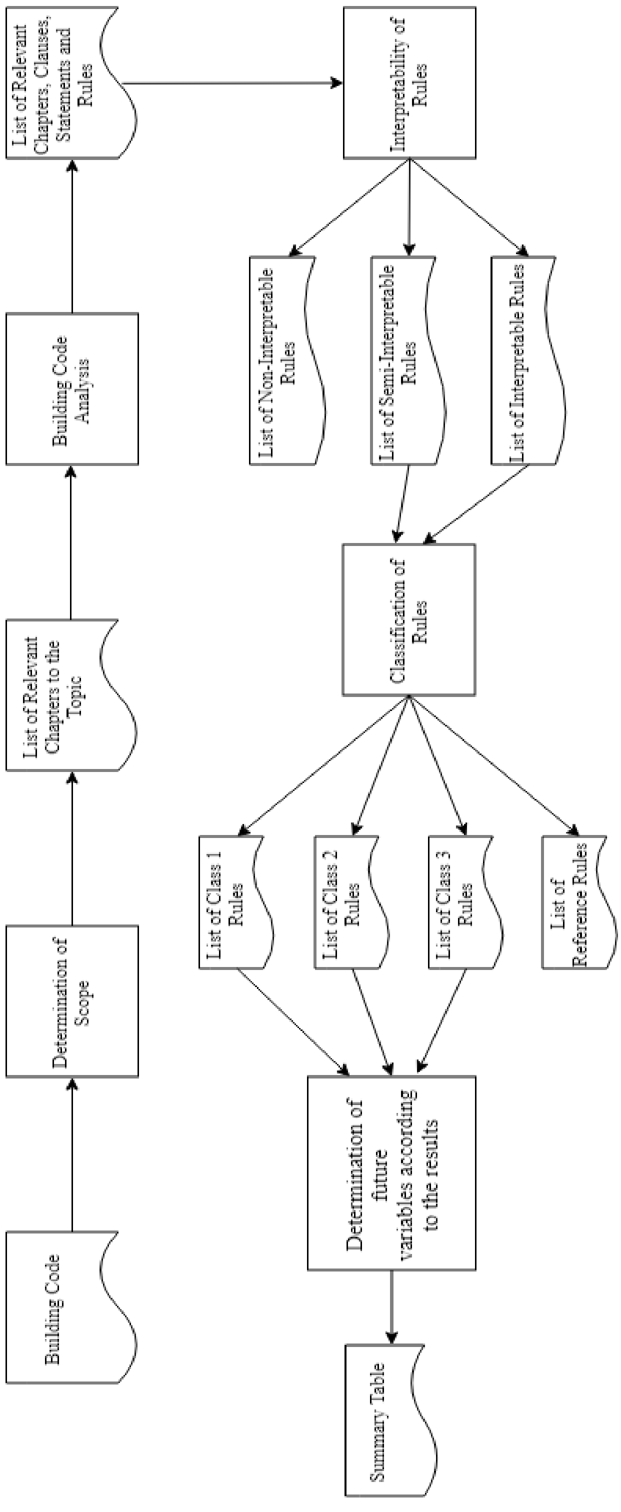

Figure 3.2. Building Code Representation Framework

3.5.2. Building Code Analysis

The second step of the building code representation methodology is the analysis of the building code. In this step, the building code is decomposed from chapters to rules as explained in Section 3.2. All of the rules related to the automated code-checking are listed under their clauses and statements, since automated code-checking systems and platforms apply rules to the building models. By applying this step, the building code's structure is extracted. From the step 1, the related chapters and clauses are determined. Then, in this step, the clauses are analyzed and separated into statements and rules. Therefore, the output of this step will be the list of relevant clauses, statements, and rules for the automated code-checking process.

3.5.3. Interpretability of Rules

The third step of the building code representation methodology is the definition interpretability conditions of the building code rules. In this step, every rule inside the scope of the automated code-checking are evaluated as interpretable to computer readable format or not. The reason for this step is that building codes consists of rules which may be interpretable, be ambiguous, be contradictory to the other rules, be impossible for modeling or just explain some terminology that does not require any checks in the model. Therefore, this step helps understanding how much of the building code can be represented within the automated code-checking systems or platforms. By applying this step, every rule's interpretability to computer readable format is decided. The decision process of the interpretability conditions of the rules are given in Section 3.3. The output of this step will be the list of rules with defined interpretability conditions.

3.5.4. Classification of Rules

The third step of the building code representation methodology is the classification of the building code rules. In this step, Interpretable and Semi-Interpretable rules are classified into different classes according to their complexity and requirements. The building codes are complex documents which includes different rules with different requirements. Rules may require simple values such as height, width, area, etc., derived values from the properties and values of the building model or even an external functions in order to interpret them in the computer readable format. The classification of rules section is especially important for developing an understanding of the rule structure of the building code. By applying this step, the rules are classified according to the predefined constraints. The classification process is explained in Section 3.4. The output of this step will be the list of classified rules which can be represented in computer readable format.

3.5.5. Determination of Necessary Variables According to the Results

The last step of the building code representation methodology is the determination of necessary variables and decisions needed for automated code-checking systems and platforms. In this step, the results from the previous steps are gathered into a summary table. In this table, classifications such as Class 1, Class 2, Class 3 and Reference rules and their interpretability conditions such as Interpretable, Semi-Interpretable and Non-Interpretable rules are summed into a cross table where classification is listed on the columns and interpretability is shown on the rows. After that, the percentage of related rules to the total number of rules are written into the corresponding cell. By constructing this table, the rule structure of the building code can be examined such that how many rules present in the building code, how many of the rules are interpretable to computer readable format, how many rules are of Class 1, etc. By using this information, it is possible to predict the required automated code-checking system or platform to be used for the automated code-checking according to the percentages from different cells in the table. In Table 3.1, the automated code-checking systems and platforms that were explained in Chapter 2 of this thesis are listed according to their possible usage for different rule classes. Although different automated code-checking systems and platforms presented are in Table 3.1, it is possible to develop new automated code-checking systems and platforms for the needs. Therefore, the table illustrated here acts merely as a suggestion for representing different classes of rules in a building code according to present research work and with commercialy available applications.

Using the results from the summary table, it is possible to develop an understanding of Level of Development (LOD) requirement of the building model for enabling code check. In the literature, LOD 300 or LOD 350 is considered generally sufficient for building model to be used for automated code-checking (Solihin and Eastman, 2015). However, lesser or greater LODs can be required for checking different building codes. Although there is not a direct link between the classifications of rules represented here with the LODs, still it can be inferred that basic rules such as Class 1 rules are most likely to be modeled with lower LODs than LOD 350. For example, a building code composed of 80% of rules as Class 1, is expected to be modeled with lower LODs than LOD 350 since Class 1 rules require simple explicit attributes and entity references that can be found inside the BIM model unlike other rule classes. As a result, utilizing lower LODs on specific building model objects will reduce the object detail for modeling the building compared to using LOD 350, when only code checking requirements are considered.. Therefore, by utilizing lower LODs for specific building model objects, the time spent for modelling the building may also be reduced. However, the LOD suggestion according to the classes is an inference from the classification requirements from Section 3.4. As stated in Section 3.4, from Class 1 to Class 3 rules, the complexity of the rule structure increases and therefore, the increase in the LOD requirement is naturally expected.

Table 3.1. *List of Research Work and Commercial Applications Used for Different Classes of Rules (Modified from Solihin, 2016)*

Class 1	Class 2	Class 3
Checks based on explicit data	**Checks based on simple derived attribute values**	**Checks based on extended data structure**
Solibri Model Checker	Solibri Model Checker	General Services Administration
FORNAX	FORNAX	FORNAX
CORENET System	CORENET System	CORENET System
General Services Administration	General Services Administration	
LicA System	LicA System	
Australia's DesignCheck System		
Norwagian Stattsbygg System		
SMARTCodes		
International Code Council		
JOTNE EDModel Checker		

CHAPTER 4

BUILDING CODE REPRESENTATION CASE STUDY

4.1. Introduction

In this chapter, Ankara Municipality Housing and Zoning Code (AMHZC) is represented according to the methodology defined in Chapter 3. The reason for choosing AMHZC is that Ankara is the capital city of Turkey and it is also the second most populated city in Turkey with 5.270.000 inhabitants in 2015, according to data of the Governorship of Ankara (“Nüfüs ve İdari Durum,” n.d.). Therefore, it will be true to say AMHZC is one of the major building codes applied in Turkey since there is continuous need of construction in this big city.

The representation of AMHZC follows the sections of the methodology defined in Chapter 3. The application steps in this chapter are explained in detail and results are illustrated in the corresonding steps below according to the defined framework.

4.2. Determination of Scope

The proposed building code representation method is applicable to most of the building codes and rules. However, a complex building code is needed to represent every concept explained in the previous chapter. Therefore, AMHZC has been chosen to be represented. In Turkey, each building project is checked against the related housing and zoning code of the area. The Municipalities define housing and zoning codes according to the standards of different Ministries. Therefore, it can be said that housing and zoning codes contain information from different standards, codes, regulations, and specifications from different building codes, regulations and standards.

In this step, the main structure of Ankara Municipality Housing and Zoning Code (AMHZC) is analyzed. AMHZC is divided into nine parts which are illustrated in Table 4.1. The chapters of AMHZC are divided according to their different uses and definitions.

Table 4.1 *Main Structure of AMHZC*

Chapter	Chapter Definition	Clauses
I	Purpose, Scope, Reference and Definitions	1-4
II	General Principles	5
III	Land Related Principles	6-8
IV	Settlement Related Principles	9-23
V	Building Related Principles	23-53
VI	Projects and Building Permits	54-66
VII	Audit Related Principles	67-68
VIII	Principles Related to Application of Regulation	69
IX	Various Principles, Last Principles and Entry into Force	70-71

For the purpose of automated code-checking process, the unrelated chapters must be excluded from the representation. In this study, only Chapter IV settlement related principles and Chapter V building related principles of the AMHZC are related to the building specifications. Other chapters are used to define purpose, scope, reference, definitions, different principles and permits which are out of the scope of this study as these cannot be checked from a building model. Therefore, in the next stages of this building code representation, Chapter IV and Chapter V of the AMHZC will be further analyzed in detail.

4.3. Building Code Analysis

In Chapter IV and Chapter V of AMHZC, there are clauses related to different areas such as building, parcellation, building adjustment, installation, etc. In this section, the clauses of Chapter IV and Chapter V of AMHZC that are related to the buildings are extracted since only building related clauses are subject of this thesis. Therefore, among these chapters, 27 clauses are found relevant to buildings. The clauses found in Chapter IV and Chapter V of AMHZC are listed below in Table 4.2.

Table 4.2. *Clauses of Chapter IV and Chapter V of AMHZC*

Clause	Clause Headline	Related Area
9	Building Road Façade Lengths	Building
10	Depth of Buildings	Building
11	Maximum Height of Buildings	Building
12	Construction of Different Height Buildings	Building
13	Multiple Construction in a Parcel	Parcellation
14	Road Profiles and Height Principles	Parcellation
15	Entrance Heights of the Buildings	Building
16	Levelling	Land Readjustment
17	Measures Required for Public Welfare	Parcellation
18	Municipality Owned Parcel Applications	Parcellation
19	Construction According to the Parcellation	Parcellation
20	Building Base Area	Building Adjustment
21	Building Total Interior Area	Building Adjustment
22	Exceptions of Building Total Interior Area	Building Adjustment
23	Municipality Authority on Building Aesthetics	Authority Principles
24	Building Entries and Pedestrian Ramps	Building
25	Stairs	Building
26	Elevators	Building
27	Basements	Building
28	Ground Floors	Building
29	Some Subjects About Building Basements and Ground Floors	Building
30	Mezzanine Floors	Building
31	Construction of Indoor Parking on the Front Yard of Ground Level Elevated Parcels	Building
32	Required Rooms/Sections in Buildings and Their Minimum Dimensions	Building

Table 4.2. (cont'd)

33	Floor Heights	Building
34	Roofs	Building
35	Cantilevers	Building
36	Eaves	Building
37	Sun Shadings, Vent Stacks and Air Shafts	Building
38	Garbage Chute	Building
39	Areaways	Building
40	Smoke Chimneys	Building
41	Railings and Parapets	Building
42	Doors and Windows	Building
43	Garden Walls	Building
44	Porter Suite and Watchmen Room	Building
45	Control Sheds	Building
46	Other Additions and Common Areas	Installation
47	Gas Service Stations	Special Types
48	Bread, Cake and Pita Bakeries	Special Types
49	Passages and Shopping Stores	Special Types
50	Public Restrooms	Special Types
51	Tea Houses	Special Types
52	Parking Lots	Special Types
53	Lightning Rods, Warning Lights and Antennas	Installation

As a result of the building code analysis, 305 rules are found from these 27 clauses that are related to the buildings. The analysis of all 27 clauses are given in Appendix A in detail. An example of the analysis of this section is given in Table 4.3. The example clause illustrated in Table 4.3 is about the stairs. This example clause contains 18 rules. However, as explained in Section 3.2 the rules may be a part of the statements or the whole statements. In the example below, statements 6, 7, and 8 each contain only 1 rule and therefore the statements act as rules. After this section, the interpretability conditions of these rules will be analyzed.

Table 4.3. *Textual Expression of the Clause 25 from the AMHZC*

ID	Textual Expression of Rule	Type
Clause 25- Stairs		
1	Minimum stair treads widths are;	
1.a	Internal stair tread width cannot be lesser than 1 meter for single independent sectioned residental buildings	Rule
1.b	Common stair tread width cannot be lesser than 1.20 meters for more than one indenpendent sectioned residental buildings	Rule
1.c	Stair tread width cannot be lesser than 1.50 meters for non-residental buildings	Rule
2	The stair step's heights and widths are;	
2.a	The stair step's height and width; are found using 2h + b = 61 - 65 function. (h) is the step height as cm and (b) is the step width as cm.	Rule
2.b	The stair step's height, cannot be higher than 0.16 meters for buildings without elevators, 0.175 meters for building with elevators, 0.15 meters for exterior stairs.	Rule
2.c	The stair step's width cannot be lesser than 0.28 meters for residental buildings, 0.30 meters for buildings other than residental and 0.35 meters for exterior stairs.	Rule
2.ç	The stair step's width cannot be lesser than 0.10 meters 0.15 meters from the the narrowest edge, 0.28 meters at the step midst for spiral staircases.	Rule
3	Stair arm, floor and intermediate stairheads and their dimensions;	
3.a	In case of risers are more than 17 in a stair, it is compulsory to construct an intermediate stairhead between floors.	Rule
3.b	Intermediate and floor stairheads' width cannot be lesser than stair width for double-return stairs.	Rule

Table 4.3. (cont'd)

3.c	The intermediate stairheads dimension at stairs and spiral staircases, is found using (n x 63 + b) function. For instance; 1 x 63 + 29 = 92 cm 2 x 63 + 29 = 155 cm	Rule
4	For, cases of usage change or public-private buildings stairs are;	
4.a	In case of changing the usage of existing buildings to the public buildings, existing stair dimensions must satisfy the new building's stair dimension requirements.	Rule
4.b	At public-private buildings, for every usage, seperate stair enclosure must be arranged.	Rule
5	About service stairs and their dimensions;	
5.a	Even if it is not inhabited, roof and basement stairs shall comply with the minimum dimension of stairs.	Rule
5.b	Basement public areas shall be connected with the main stairs for all buildings.	Rule
5.c	For single sectioned buildings, in case of basements are not inhabited, they can be accessed through outside.	Rule
6	Regulation on Fire Protection statements effective for fire measures and escape stairs.	Rule
7	Independent section stairs serving more than one story with public access for non-residental buildings also must comply with this statement.	Rule
8	On each side of the stairs, gunwale and railings must be constructed according to the related TSE standards for disable and also tread, floor and intermediate stairheads' coverings shall comply with the standards.	Rule

4.4. Interpretability of Rules

As stated before Ankara Municipality Housing and Zoning Code (AMHZC) has a complex structure which contains different rules with different properties. These rules can be interpretable, or contain subjective information, only define terminology, or even contain contradictory information which make them impossible to interpret in computer readable format. The interpretability conditions of rules are explained in Section 3.3. After applying the interpretability conditions to the selected 305 rules, 185 of the rules are found Interpretable, 74 of the rules are found Semi-Interpretable, and 29 of the rules are found Non-Interpretable. In other words, 61% of the rules are Interpretable, 24% of the rules are Semi-Interpretable and 10% of the rules are Non-Interpretable out of the total 305 rules defined. As can be seen from the sum of the interpretability condition rule results, rules that contain interpretability conditions are less than the total number of rules. The reason for this is that some of the rules from the AMHZC are excluded from the interpretability condition check since they refer to the other standards, specifications, etc. The interpretability results of Clause 40 - smoke chimneys are given in Table 4.4 as example for the interpretability conditions check.

Table 4.4. *Interpretability of Conditions of AMHZC Clause 40 – Smoke Chimneys*

Clause	**Statement**	**Rule**	**Rule Properties**
			Interpretability
C40	ST1	R40.1	
		R40.1.a	Interpretable
		R40.1.a.1	Interpretable
		R40.1.a.2	Interpretable
		R40.1.b	Interpretable
		R40.1.c	Interpretable
	ST2	R40.2	
		R40.2.a	Non-Interpretable
		R40.2.b	Non-Interpretable
		R40.2.c	Non-Interpretable
		R40.2.ç	Interpretable

Table 4.4. (cont'd)

	R40.2.d	Interpretable
	R40.2.e	Semi-Interpretable
	R40.2.f	Non-Interpretable
	R40.2.g	Semi-Interpretable
	R40.2.ğ	Interpretable
	R40.2.h	
	R40.2.ı	Semi-Interpretable
	R40.2.i	Interpretable
	R40.2.j	Interpretable
ST3	R40.3	
	R40.3.a	Non-Interpretable
	R40.3.b	Interpretable
	R40.3.c	Interpretable
	R40.3.ç	Interpretable
	R40.3.d	Interpretable
	R40.3.e	Semi-Interpretable
	R40.3.f	Interpretable
	R40.3.g	Interpretable
	R40.3.ğ	Interpretable
	R40.3.h	Semi-Interpretable
	R40.3.ı	Semi-Interpretable
	R40.3.i	Semi-Interpretable
	R40.3.j	Interpretable

As can be seen in Table 4.4, R40.2.h has been intentionally left blank because it does not meet the interpretability conditions. Reason behind this is that the related rule is out of the scope of this study since it refers to the different regulations, specifications, building codes and laws. However, it is not possible to label this rule with interpretability conditions because it may or may not be interpretable. Therefore, this rule will be explained and labeled in the classification of rules section. The rule is illustrated below:

> "For dimensioning, design and construction of chimneys, TSE standards, 5/12/2008 dated and published in the 27075 numbered Official Gazette of Turkey Energy Performance Regulation for Buildings, Turkey's Regulation on Fire Protection, 18/9/2002 dated and published in the 24880 numbered Official Gazette of Turkey Natural Gas Market Interior Installation Regulation and related public institutions' specification statements must be followed." (AMHZC, Clause 40- Statement 2.h)

4.5. Classification of Rules

The AMHZC rules are divided into classes as Class 1, Class 2, Class 3 and Reference rules. The classification process of the rules are explained in Section 3.5. After classifying the 305 rules that are defined in the Section 4.3, 184 of the rules are found as Class 1, 70 of the rules are found as Class 2, 5 of the rules are found as Class 3 and 17 of the rules are found as Reference rules. In other words, 60% of the rules are Class 1, 23% of the rules are Class 2, 2% of the rules are Class 3 and 6% of the rules are Reference rules out of the total 305 rules defined. The number and percentage of the Class 3 rules are less than other rule classes since they require more complex checks and explicit tools and applications which are difficult to interpret in computer readable format that require unique properties on rules. Moreover, fewer Class 3 rules makes it easier to represent building code in computer format. As can be seen from the classification of rules results, the classified rules are less than the total number of rules. The reason for this is that some of the rules from the AMHZC are excluded from the classification process since they are Non-Interpretable rules. Non-Interpretable rules cannot be represented in computer readable format and therefore they are not suitable for classification. The classification of Clause 32 - required rooms/sections in buildings and their minimum dimensions is given in Table 4.5 as an example of the classification.

Table 4.5. *Classification of Clause 32 Required Rooms/Sections in Buildings and Their Minimum Dimensions*

Clause	Statement	Rule	Rule Properties	
			Rule Class	Interpretability
C32	ST1	R32.1	Class 1	Interpretable
		R32.1.a	Class 2	Interpretable
		R32.1.b	Class 1	Interpretable
		R32.1.c	Class 3	Interpretable
	ST2	R32.2		
		R32.2a	Class 1	Interpretable
		R32.2b	Class 2	Interpretable
		R32.2c	Class 1	Interpretable
		R32.2ç	Class 1	Semi-Interpretable
	ST3	R32.3		
		R32.3a	Reference	
		R32.3b	Class 1	Interpretable
		R32.3c	Class 2	Semi-Interpretable
		R32.3ç	Class 2	Semi-Interpretable

The classification of rules and interpretability conditions of 305 rule that are found in the AMHZC are provided in Appendix B.

4.6. Determination of Necessary Variables According to the Results

In this section, the data from the previous sections are gathered to create a summary table to see the complete structure of the Ankara Municipality Housing and Zoning Code (AMHZC). Also, since it is a cross table it is possible to see classification and interpretability of rules altogether. In this table, Class 1- Interpretable rules are 40%, Class 2-Interpretable rules are 20 %, Class 3- Interpretable rules are 1%, Class 1- Semi-Interpretable rules are 21%, Class 2- Semi-Interpretable rules are 3%, Class 3- Semi-Interpretable rules are 1%, Reference rules are 6% and Non-Interpretable rules are 10% of the total rules. Also in brackets, the number of rules for each cell are illustrated. Total number of rules can be found in two ways which are the sum of the Class 1, Class 2, Class 3, Reference and Non-Interpretable rules or the sum of the Interpretable, Semi-Interpretable, Non-Interpretable and Reference rules. The reason for this is that the Non-Interpretable rules do not have classifications and Reference rules do not have interpretability conditions.

Table 4.6. *Summary Table of the AMHZC Rules*

	Interpretable	**Semi-Interpretable**	**Non-Interpretable**	**Total #**
Class 1	40% (121)	21% (63)		60% (184)
Class 2	20% (62)	3% (8)		23% (70)
Class 3	1% (3)	1% (2)		2% (5)
Reference				6% (17)
Total #	61% (201)	24% (74)	10% (29)	

As stated in Section 3.5.5, the summary table can be used to define necessary variables for the automated code-checking systems and tools. In Table 4.6, Class 1 rules constitutes 60% of the total rules in the AMHZC while Class 2 rules are 23% and Class 3 rules are 2%. An automated code-checking system or tool can be chosen to fulfill the requirements of this building code. As can be seen from Table 3.1 in Section 3.5.5, there are different automated code-checking systems and tools that are able to check Class 1 and Class 2 rules. However, scope of the automated code-checking systems and tools shall be taken into consideration before utilizing them. Because scope of the automated code-checking system or tool may not be related to the building code at hand.

Also, the Level of Development (LOD) decision can be made according to the classification of rules since rules are classified from simple to complex. As can be seen from Table 4.6, Class 1 rules constitutes 60% of the total rules in the AMHZC. Therefore, it can be said and suggested that the simple rules are easier to be modeled with lower LODs.

In the next chapter, implementation of the AMHZC rules within an automated code-checking platform will be explained, discussed and verified with the use of the proposed building code implementation framework.

CHAPTER 5

BUILDING CODE IMPLEMENTATION CASE STUDY

5.1. Introduction

In this chapter, the rules of Ankara Municipality Housing and Zoning Code (AMHZC) are represented in an automated code-checking system by using the developed building code representation methodology. However, in the AMHZC, the Semi-Interpretable rules contain ambiguous concepts that impede the interpretation process. Moreover, Semi-Interpretable rules require human clarification because they contain fuzzy and subjective information that lead to different interpretation by different experts. Therefore, it is not possible to represent Semi-Interpretable rules in computer readable format without an initial study to develop a proof of concept system that encapsulates expert knowledge. Also, Non-Interpretable rules cannot be represented in the computer readable format and they will be excluded from the automated code-checking. Lastly, Reference rules' interpretability conditions are unknown and the referred regulations, specifications, laws and codes must be examined in detail in order to represent them in computer readable format. In this study, Semi-Interpretable, Non-Interpretable and Reference rules will not be represented in the code-checking tool since the requirements of these rules are beyond the scope of this study. In this thesis, only Interpretable rules will be covered and investigated thoroughly.

Solibri Model Checker (SMC) will be used as the automated code-checking tool for variety of reasons. Firstly, most of the automated code-checking systems and tools are using black box methods. They do not share information about their systems and tools with the public. On the other hand, SMC allows users to create their own rules according to the predefined rulesets. Secondly, SMC is a commercial application used widely around the world. Also, IFC data model can be imported and the tool offers variety of interoperability options between different applications and software. Therefore, conducting this study on SMC allows an easier BIM-based application than some other automated code-checking systems and tools. Lastly, SMC has a user-friendly interface which can be used and understood without any expert knowledge. Most of the automated code-checking systems and tools require expertise about different programming languages such as C, C++, java, EXPRESS, etc. However, representing rules in computer readable format using SMC does not require special expertise in coding knowledge domains. Moreover, SMC uses parametric tables to represent rules which eases the process of transferring rules into the tool. For this reasons, SMC has been chosen as the environment to conduct automated code-checking in this study.

In the next section, building code implementation framework will be defined, explained and illustrated.

5.2. Building Code Implementation Framework

In this section, building code implementation will be discussed in detail. In the previous section, it is explained why Solibri Model Checker (SMC) was chosen to implement Ankara Municipality Housing and Zoning Code (AMHZC) rules and to conduct automated rule-checking. Therefore, building code implementation framework is prepared according to the checking process of SMC. In Figure 5.1, the building code implementation framework is illustrated. For different automated code-checking systems and tools, different frameworks can be formed. In other words, this framework is prepared solely for SMC.

In this framework, the processes and data outside the dotted line are already explained in previous sections. Within the dotted line, the processes that are performed using the SMC is defined. The inputs to the SMC are the list of relevant rules to the automated code-checking and BIM. The list of relevant rules are the rules defined in Section 5.1 which are the Interpretable rules from the AMHZC. The BIM in the framework is the model of the building that will be checked.

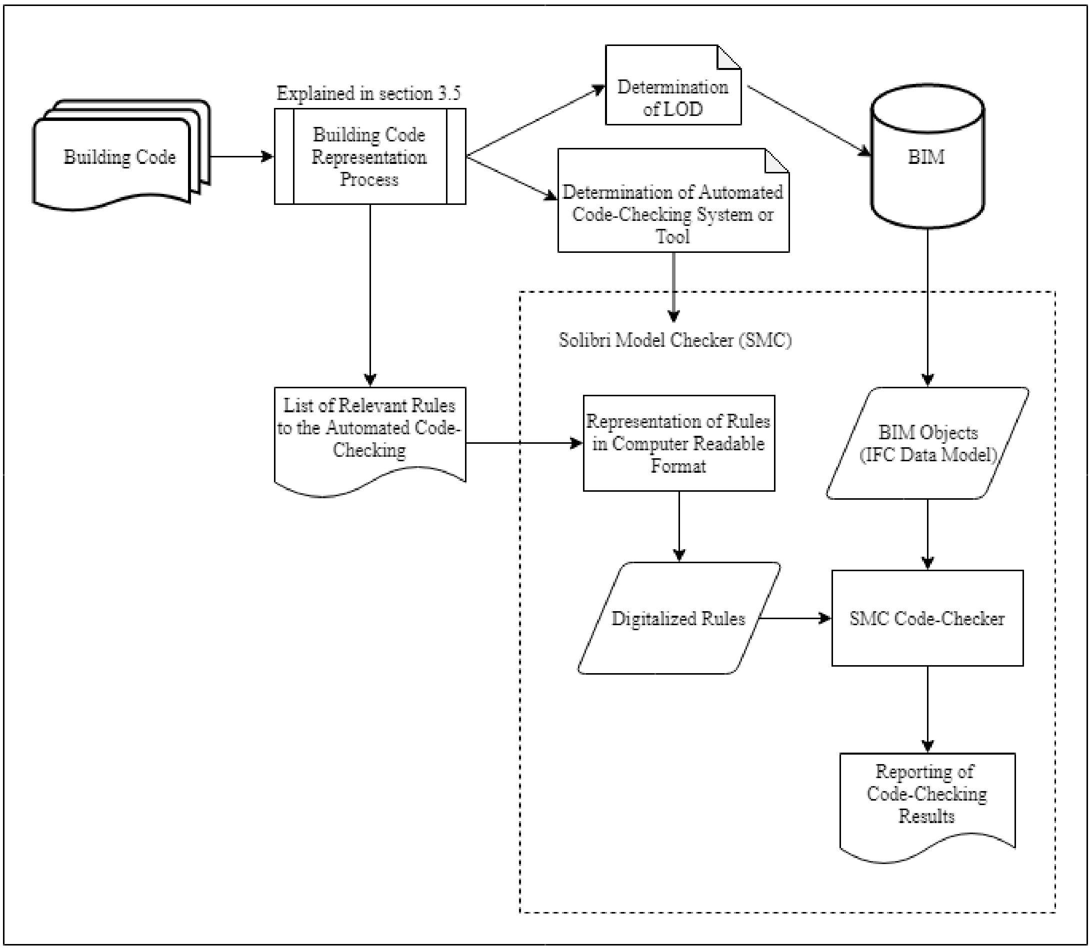

Figure 5.1. Building Code Implementation Framework

In the representation of rules in computer readable format process, firstly the related rules will be arranged using the SMC Ruleset Manager. Ruleset Manager in SMC offers different sets of rules in different domains such as architectural, structural, MEP rulesets. These rulesets contain different rules to check various properties of the building model components. These rules can be modified to reach the desired rule requirements and parameters. The product of this process is the digitalized rules in the computer readable format. Secondly, the model formed using the BIM tools such as Autodesk Revit, Archicad, Tekla, etc., is exported into IFC data model and uploaded to the SMC. As a result, the model components' properties can be viewed in SMC. The outcome of this process is the BIM objects. Another feature of the building code implementation framework is the SMC Code-Checker. SMC has a built-in code-checking feature. In this study, it has been named as "SMC Code-Checker" and from now on will be used to refer to SMC's built-in code-checking feature. To use SMC Code-Checker, BIM objects and digitalized rules are needed. It has the ability to check the building model according to the defined rulesets. To be able to perform this check, building model shall be designed parallel to the needs of defined rulesets. Also, SMC Code-Checker can be used to visualize the problems and errors of building design components. In short, issues can be individually examined. For reporting results, SMC also offers a built-in reporting feature which can be used to export results in .xls, .pdf, etc. formats. In the next sections, representation of rules in computer readable format, SMC Code-Checker, and reporting of results will be explained in detail.

5.2.1. Representation of Rules in Computer Readable Format

To represent written rules in computer readable format, SMC Ruleset Manager is used. SMC Ruleset Manager is a built-in application of SMC which allows creating rules according to the given rulesets. There are different rulesets provided in SMC to create and manage the rules such as architectural, structural, MEP rulesets. These rulesets contain sub-rulesets for specific checks (Figure 5.2). For instance, architectural ruleset contains 12 sub rulesets for different rules such as advanced space check, BIM validation – architectural, pre-check for energy analysis, building efficiency, etc. SMC rules are coded in parametric tables (Eastman et al., 2009). Therefore, SMC does not require any expert knowledge for creating and modifying rules.

SMC Ruleset Manager has a user-friendly UI that allows views to be modified and contains all the necessary tabs for creating, modifying, and managing rules. Views of the UI can be modified by using the Views tab on top right. Also, it allows searching rules and rulesets at the top right of the UI. In the Workspace tab, it is possible to create user-defined rules and rulesets for the desired use. In the Info tab, written information related to the rules are illustrated. Moreover, Libraries tab is where the simple common rules are presented.

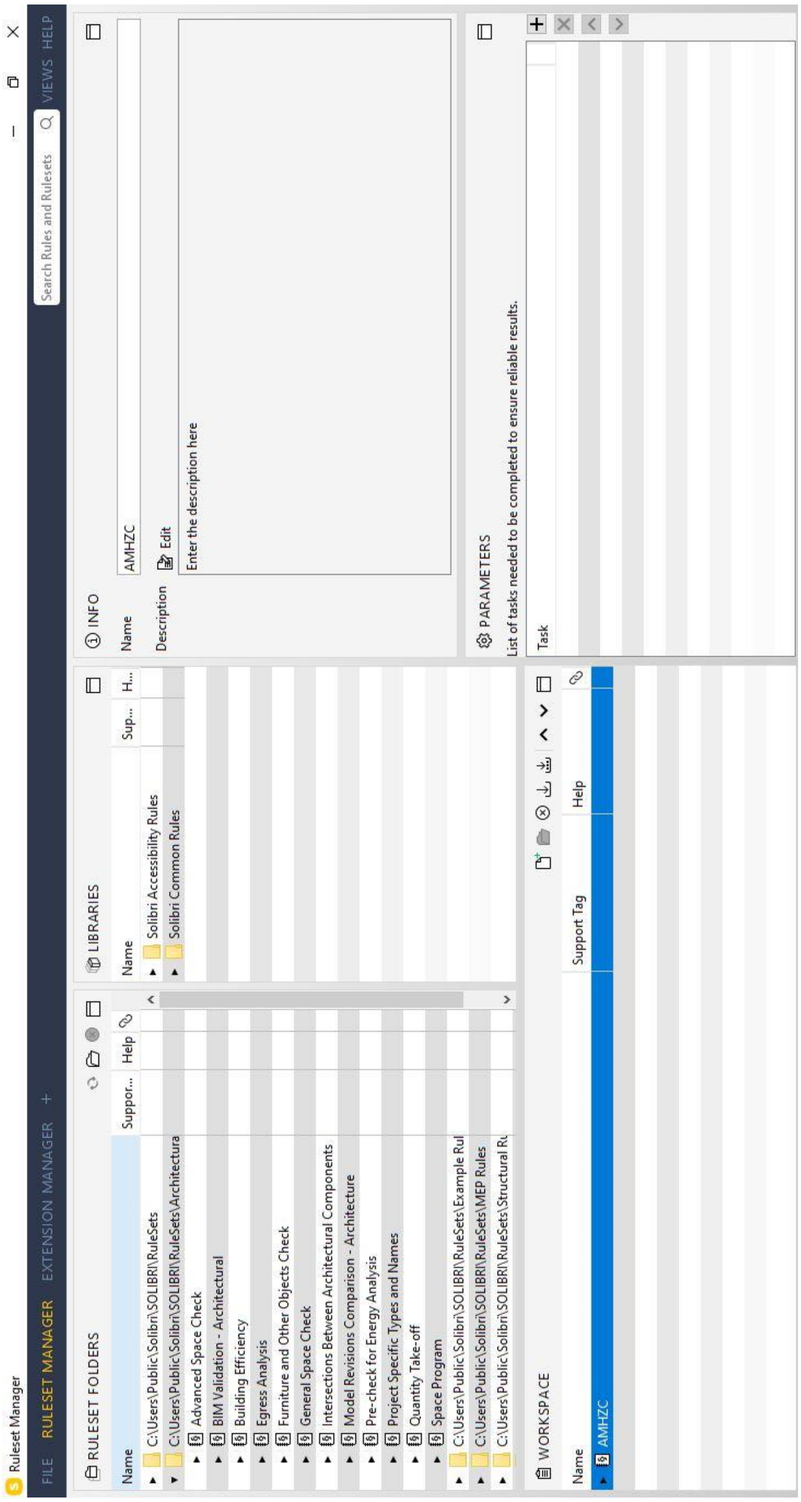

Figure 5.2. SMC Ruleset Manager UI

In the Parameters tab, rule structures are defined and modified. Most of the rules' parameters consists of components to check, requirements, and categorization of results sub-tabs (Figure 5.3). Components to check tab is used to define which components are included, excluded and ignored for checking the related rule. Requirements tab is used to define specific requirements of the components to pass rule by defining state, component, property, operator, and value similar to the components to check tab. Lastly, categorization of results tab is used for categorizing issues' properties.

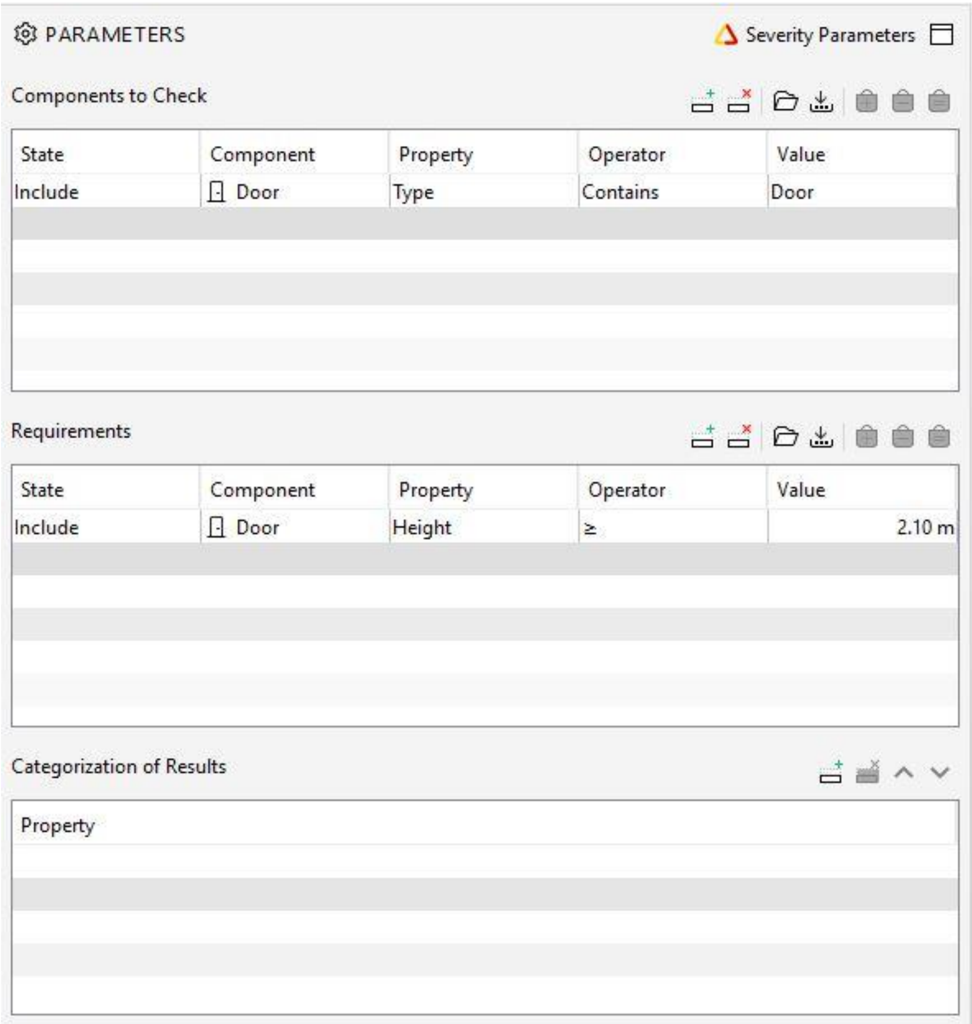

Figure 5.3. SMC Rule Parameters of AMHZC Clause 42 – Statement 1.a

In Figure 5.3, rule parameters of the door heights are illustrated. In AMHZC R42.1.a, it was stated that the door heights cannot be shorter than 2.10 meters. Therefore, to configure this rule, first the doors of the building must be defined in a precise manner. In this case, all of the door objects containing "door" statement in their type which covers all the doors in the building is used. After defining which component to check, the requirements are determined. For this case, the height of the door objects must be equal or higher than 2.10 meters.

5.2.2. SMC Code-Checker

SMC Code-Checker is the built-in code-checking application of SMC. To process rule checking, SMC Code-Checker requires a building model and a set of rules defined in SMC. To import a building model into SMC, it must be in defined formats such as .ifc, .pdf, .dwg, etc. After importing the building model into SMC, the building model's each component can be visualized in Model layout's Model Tree tab. SMC's default layouts are Model, Checking, Communication, and Information Takeoff (Figure 5.4). However, the new layouts according to the different needs can be created by using the plus button next to the default layouts.

To add a ruleset to the SMC, Checking tab inside the Checking layout can be used. As stated before, there are predefined rulesets and rules that can be used to check a building according to predefined parameters. Also rules can be configured using the SMC Ruleset Manager according to the desired requirements. After adding rulesets and rules to the SMC Code-Checker, code-checking is simply done by clicking the "Check Model" button on the Checking tab (Figure 5.4). Then, SMC Code-Checker illustrates the issues found in the model in the Results and Result Summary tabs. By using the Results tab, it is possible to view each issue visually in tree view. Results tab also illustrates the issues with different notices. Moreover, SMC has the capability of classifying issues according to their severity such as critical, moderate, and low severities with the red, orange, and yellow triangles. The severity conditions can be changed from the rule parameters. Lastly, the parameters of a rule can be modified from the Parameters tab in Checking layout although adding new rules to the rulesets can only be done through SMC Ruleset Manager.

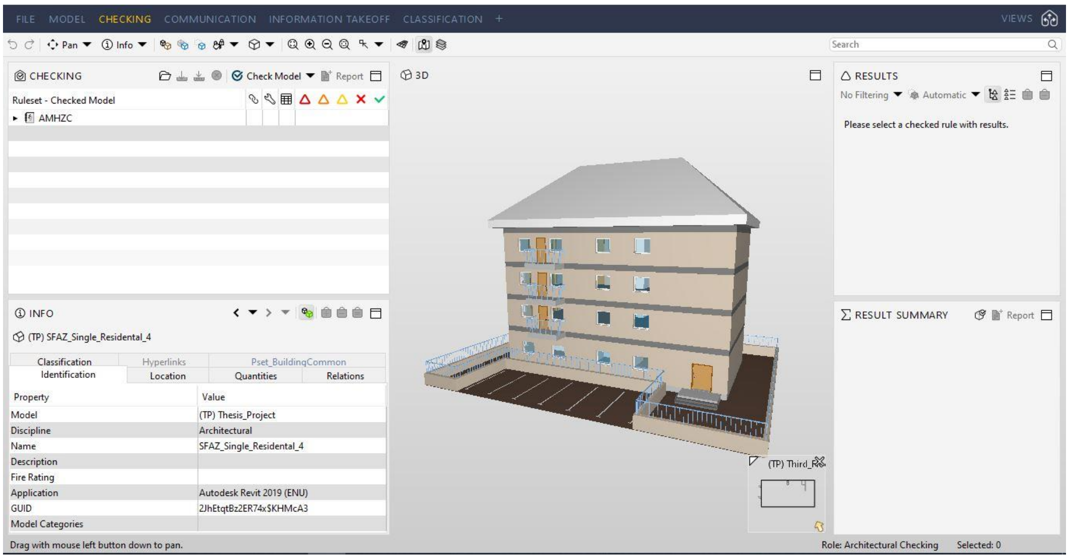

Figure 5.4. SMC Code-Checker UI

5.2.3. Reporting of Results

In this section, reporting of code-checking results of SMC will be explained and discussed. By using SMC's default reporting feature, issues found in the building model can be reported in various formats. Firstly, from the Results Summary tab, the issues found in the building model can be reported in .xls format as a list without visualization. Secondly, the issues found in the building model can be illustrated with slides which contains the visualization of the issue. Adding slides to an issue, captures the current view of the 3D model and embeds it to the issue. Therefore, for the practical uses, the 3D view of the building model should be adjusted properly to examine the issue. After adding slides to the issues, they are used in presentations using Presentation tab in Communication layout to give visual information about the issue. The presentations can be reported in different formats such as .xls, .pdf, .bcf and .rtf (Figure 5.5). It is possible to add more views to an issue by using the Issue Details tab. Therefore, the issue can be viewed from different perspectives. Also, different slides can be added to a single issue. By adding more views to an issue, it is aimed to improve the issue details. Also, the slide of the issue can be updated with different 3D views using the Issue Details tab.

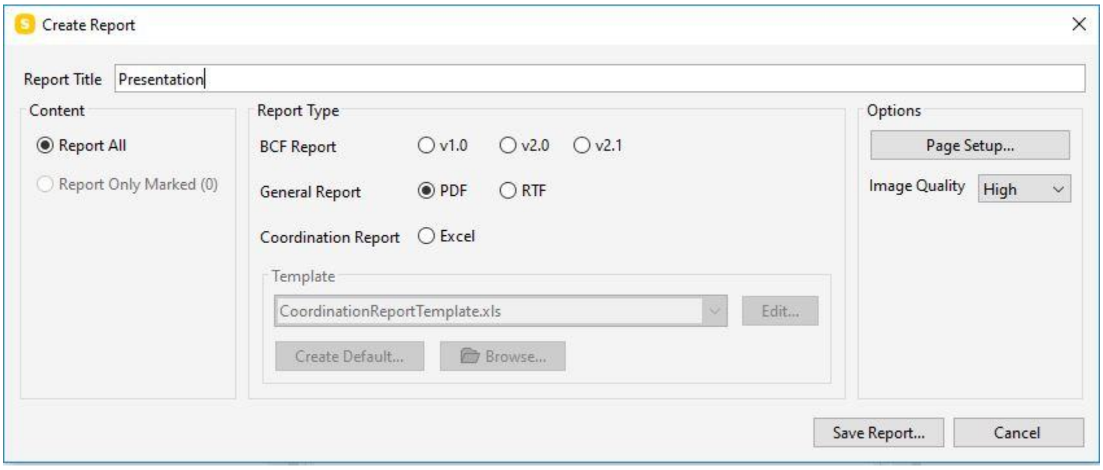

Figure 5.5. SMC Reporting Options

5.3. Implementation of Framework through a Case Study

In this section, the selected relevant rules from Section 5.1 of Ankara Municipality Housing and Zoning Code (AMHZC) are implemented as a case study through the building code implementation framework. To accomplish this task, first, the relevant rules from AMHZC are represented in computer readable format using SMC Ruleset Manager.

As stated in Section 4.4, AMHZC has 185 Interpretable rules with different classifications. Since each rule contains different parameters, components and requirements to check, different type of rule parameter tabs from SMC Ruleset Manager are used in order to represent them in the tool. For this purpose, SMC Ruleset Manager parameter tabs are divided into different types.

Type 1 parameter tab is used to check defined components' property requirements. In Figure 5.6, the parameter tab of the AMHZC R42.1.c which is used to limit the independent section's entrance door's width is illustrated. By using this type of parameter tab, it is only possible to do simple checks such as the quantity, identification, location, classification, property set, and relation checks according to their defined properties and values. Therefore, most of the rules are represented in computer readable format using this type of parameter tab.

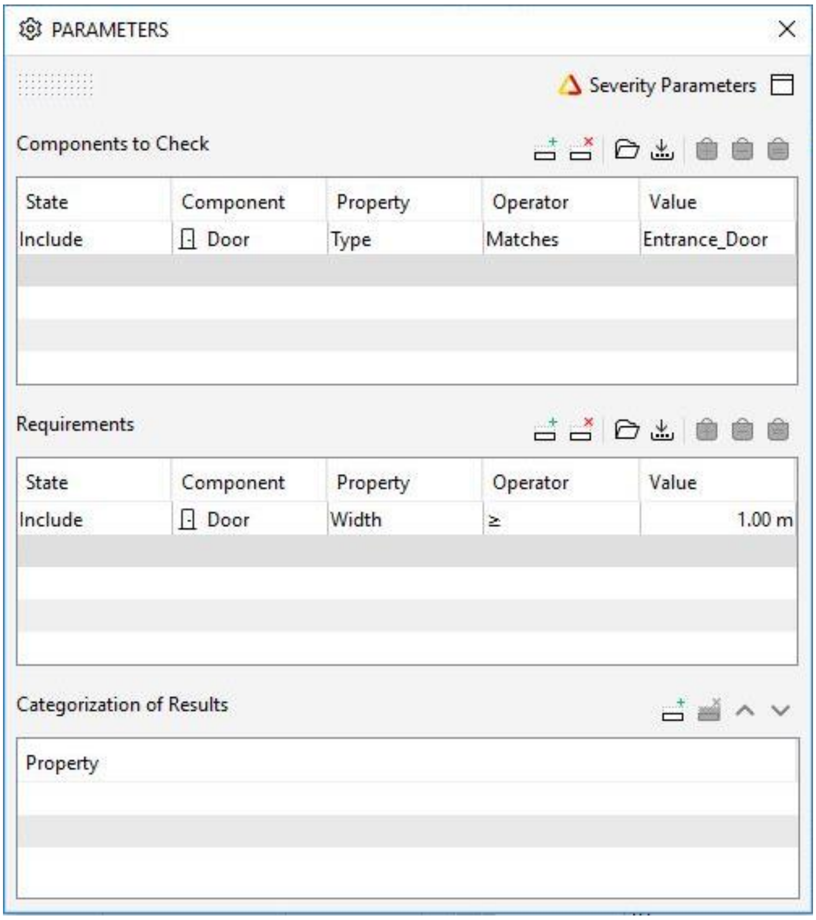

Figure 5.6. AMHZC R42.1.c SMC Ruleset Manager Parameter Tab

Type 2 parameter tab is used for distance calculation between defined components. By using this parameter tab, it is possible to define maximum or minimum distance between defined components. Distance check contains features such as shortest distance, horizontally alongside, directly below/above, etc (Figure 5.7). In Figure 5.7, parameter tab of the AMHZC R42.1.ç which requires revolving doors must be alongside the main entrance doors is illustrated.

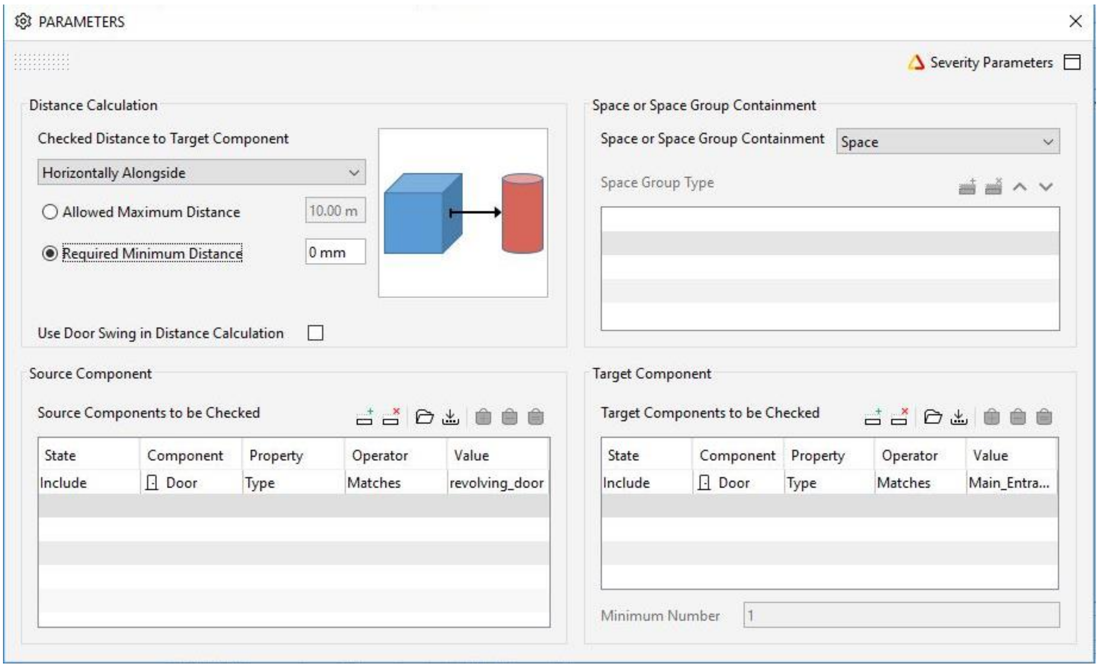

Figure 5.7. AMHZC R42.1.ç SMC Ruleset Manager Parameter Tab

Type 3 parameter tab is used for defining the free corridor width of classified spaces. To use this parameter tab, the spaces must be classified using the SMC's Classification tab. The Classification tab can be opened using the view button in SMC UI. In the Classification tab, new classifications according to the user's need can be created by using the new classification button. Then, classification settings of the newly created classification opens for setting the parameters of the classification. In Figure 5.8, the classification settings for accomplishing the AMHZC R24.1 is illustrated. AMHZC R24.1 states that the entrance corridor of the building must have a minimum free corridor width of 2.2 meters for public buildings and 1.5 meters for other buildings. Therefore, spaces that contain 'entrance' in their name are used for this classification. Then, related classifications are created and appointed to the relevant components.

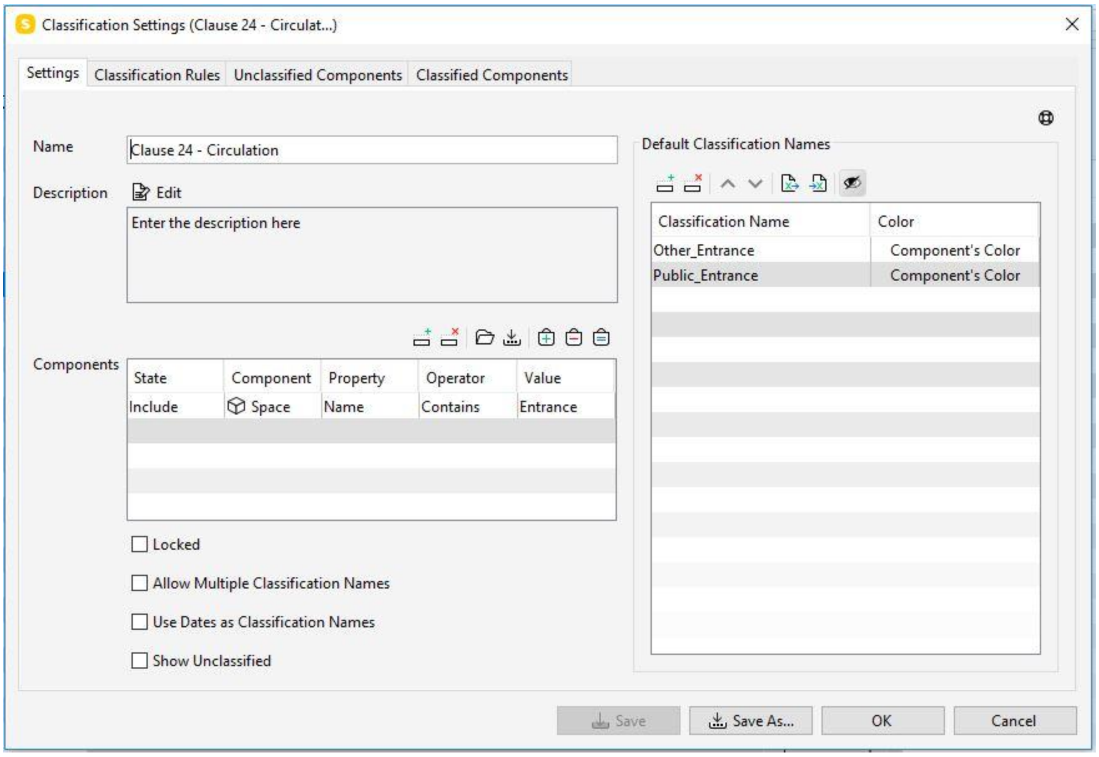

Figure 5.8. AMHZC R24.1 Classification Tab Settings

Type 4 parameter tab is used for limiting the ramp properties according to the needs. It contains various properties related to ramps such as slope requirements, length requirements, width requirements, handrail checks, connection to stairs, etc. Like type 3 parameter tabs, this type of parameter tab also requires classified components in order to process the related rules. In AMHZC R24.10, the slope requirements for the internal and external ramps are defined. For instance, ramps with 0 to 15 cm height are limited to have maximum slope of 8%.

Type 5 parameter tabs are used for limiting the stair properties according the requirements similar to the Type 4 parameter tab (Figure 5.9). This parameter tab contains vast amount of requirements in order to check stairs such as height of stair, minimum width of stair, minimum/maximum riser height, minimum/maximum tread length, landing check of stairs, handrail check of stairs, etc. This parameter tab also requires classification of stair in order to process them. In AMHZC R25.2.b, the tread length of stairs are defined. For instance, residential building stair treads must have minimum tread length of 0.28 meters.

Type 6 parameter tab is used for space component requirements such as spaces must contain specific number of defined components. For instance, in AMHZC R44.2.a, porter suites are required to have openings for air and light needs. Therefore, by using this type of parameter tab, spaces defined as porter suites can be arranged to have required openings.

Type 7 parameter tab is used to check connections between chosen spaces. This parameter tab can be arranged to consider doors and openings between the chosen spaces. Also, this parameter tab can be modified to cover direct exit to outside of the building. For example, in AMHZC R25.5.b, it was stated that the main stair of the building must be connected to the basement. Therefore, using this parameter tab, connection from the entrance corridor space to basement space is arranged.

Type 8 parameter tab is used to check that each building floor contains required number of spaces. By using this parameter tab, the number of spaces can be defined such as living room, bedroom, bathroom, etc. in the related building design. Although this type of parameter tab is explained in this section, it did not find any use in the AMHZC.

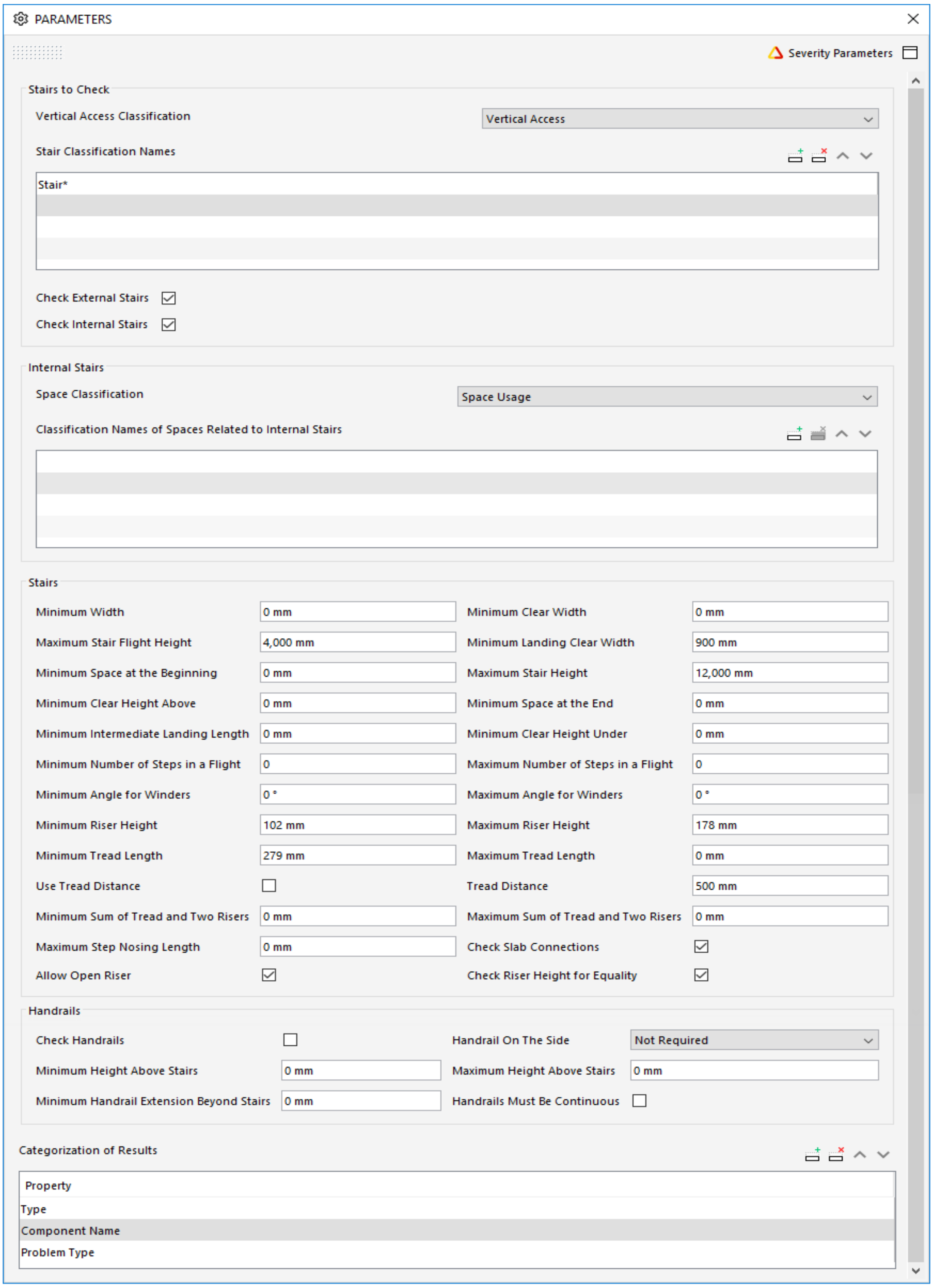

Figure 5.9. SMC Stair Check Parameter Tab (Retrieved from https://solution.solibri.com/help/smc/9.10/en/html_sol_210.htm)

In addition to the parameter tabs, to be able to represent AMHZC rules in SMC, an object definition system is utilized to define specific components. By doing this, it was possible to distinguish specific components from other same type components. For instance, building's name is coded in a specific way in order to specify if the building is in statuted floor zone area or other zone, how many stories does the building have, what is the type of building such as residential, commercial, high, industrial, detached, adobe, etc. and if the building is single or double. Since, SMC and BIM tools do not have the capabilities for specifying the components according to the AMHZC needs, it was required to implement building code in the SMC Ruleset Manager. However, for different building codes these specifications can be modified in order to satisfy the component requirements since these types and requirements vary from building code to building code. In Figure 5.10, the necessary specifications to satisfy the stair requirements are illustrated.

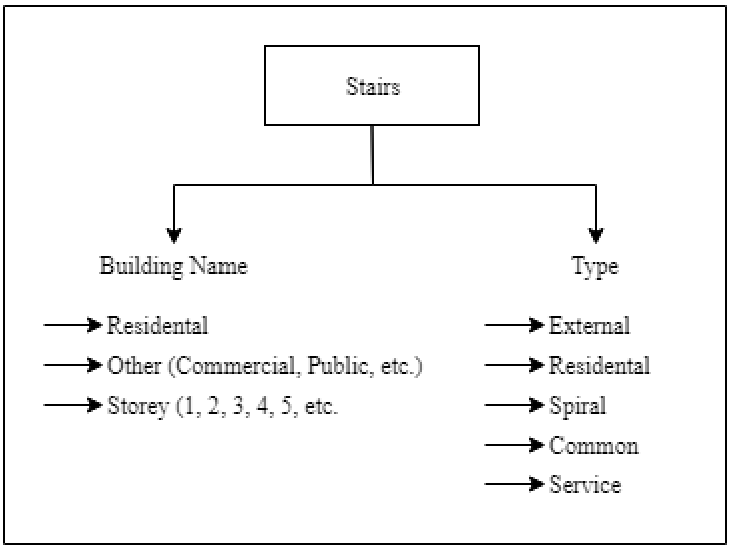

Figure 5.10. Stair Specification

By utilizing defined parameter tabs and object definition system in this section, the Interpretable rules of AMHZC are represented in SMC. However, although 185 Interpretable rules are found in the AMHZC, only 45% (83) of them are represented via SMC Ruleset Manager. Among those 185 rules, 121 of them are classified as Class 1, 61 of them classified as Class 2, and 3 of them classified as Class 3. 48% (58) of the Class 1 rules, 39% (24) of the Class 2 rules, and 33% (1) of the Class 3 rules are represented in computer readable format using SMC Ruleset Manager. Interpretability of the rules in SMC are listed in Appendix C.

There are two reasons behind this result. Firstly, SMC's capabilities to completely check a building code is limited. For instance, in AMHZC R34.1.a, it was stated that the slope of roof cannot exceed 40%. Although the slope of roofs can be specified in BIM tools, SMC does not have slope property for the roofs. Secondly, the complexity of the building code comes with complex requirements for its rules. Even a single rule in a building code may contain several requirements, components and relations with other rules to be checked. For instance, the AMHZC R37.9.a states that the buildings with 2 stories or eave height lower than 7.5 meters can contain air shafts with dimensions 0.45x0.45 m. and only 2 sections in same floor can benefit from these air shafts. Therefore, representing this rule using the SMC Ruleset Manager is not possible.

After representing the Interpretable rules in computer readable format using SMC Ruleset Manager, second stage is to import the building model to be checked into SMC. For this purpose, a sample building model was created using Autodesk Revit 2019 tool (Figure 5.11). The sample building model has 20 meters length, 10 meters depth and 12.5 meters height (excluding the roof). Each floor is designed to have one independent unit. Each independent section has one living room, two bedrooms, one kitchen, and one bathroom. The building site is surrounded by garden walls with railings on them. Also, a parking lot for six cars has been designed. The roof of the building is designed with 40% slope and the stairs of the building contain 22 risers for ground to first floor and 19 risers for other floor connections. Lastly, the building contains different cantilevers as balconies.

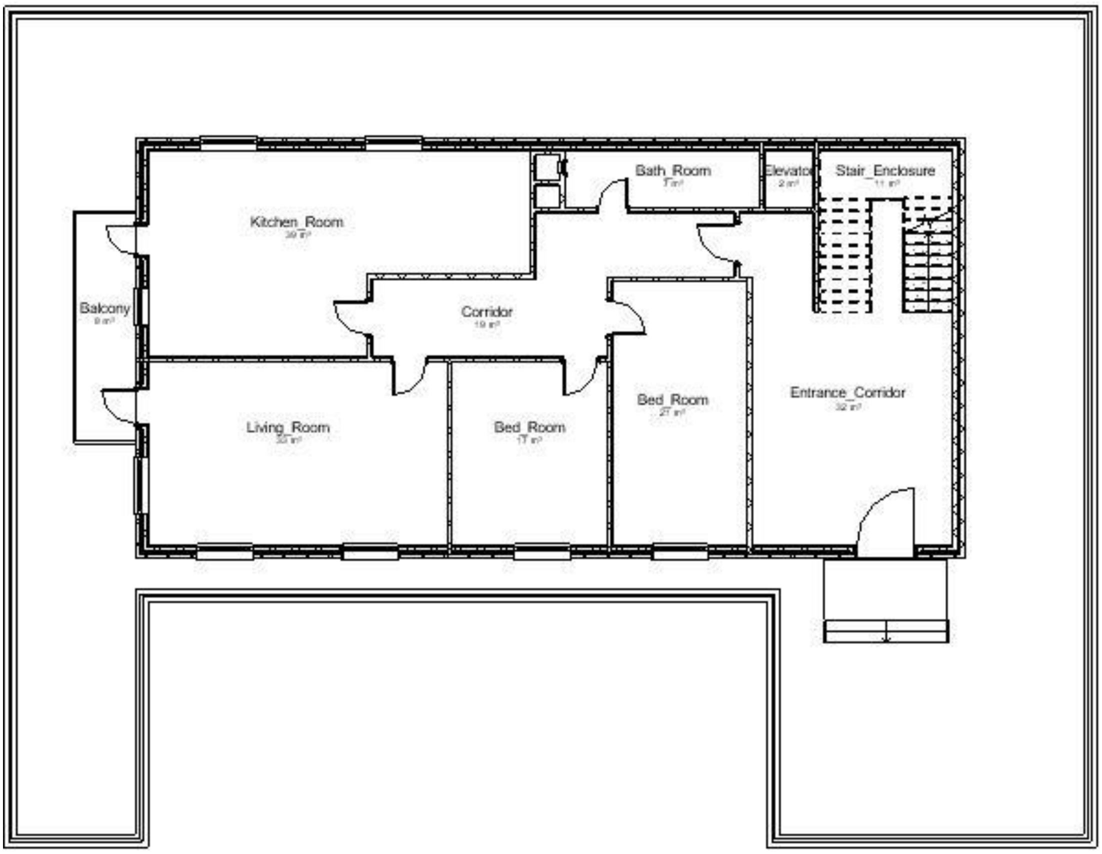

Figure 5.11. Sample Building Model Floor Plan

The building model has been exported as IFC data model from Autodesk Revit 2019 tool in order to be able to utilize it in SMC. Then, the building model is imported into SMC for checking it against the rules defined in SMC Ruleset Manager.

To verify that the automated code-checking can be performed, several properties in the building are purposely set against the AMHZC rules, such as external stair on the building entrance, balcony on the side of the building, bathroom in the independent section, etc. This aimed to enable assessing if the rules are defined according to the needs of the AMHZC and see whether SMC Code-Checker is able to process automated code-checking.

After applying SMC Code-Checker to the building model and represented rules, the results illustrated in Figure 5.12 are obtained.

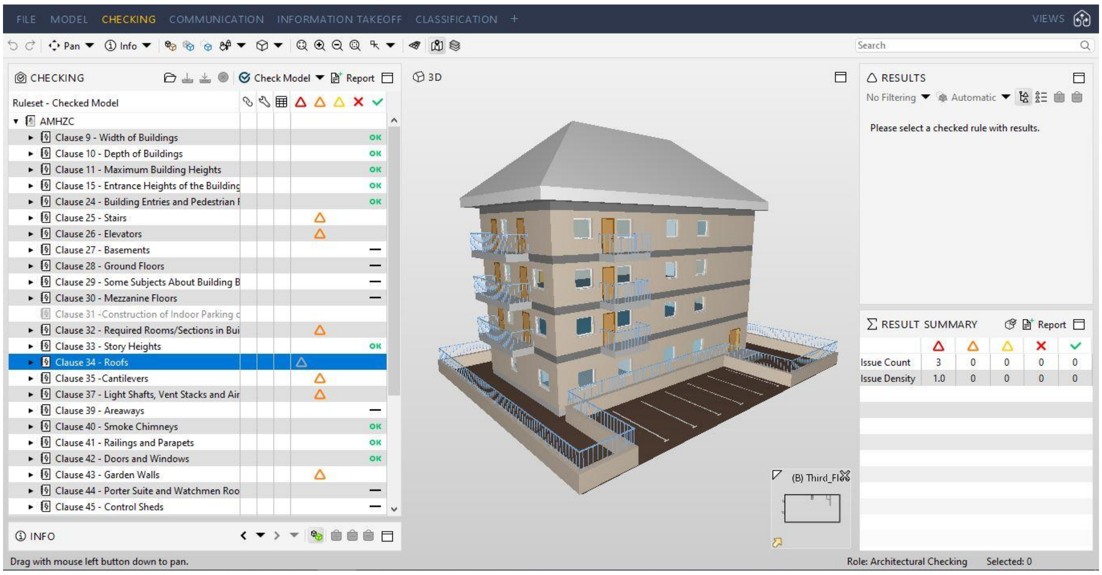

Figure 5.12. Checking Results of Sample Building Model

In Figure 5.12, the Checking tab shows the results of the code-checking with different features. Firstly, dash indicates that there are no available components to check in the model. For instance, there is no basement space in this building model and the results for the AMHZC Clause 27 show the result as dash. This means, there is not any basement to check in the building model. Secondly, “OK” indication is used if the corresponding rule is passed in the building model. For example, the model comply with Clause 9 rules of the AMHZC. Thirdly, there are indications for the issues found in the building that are represented with red, orange, and yellow triangles. The color code of these triangles refer to the severity of the issue such as red is used for critical severity, orange is used for moderate severity, and yellow is used for low severity. If needed, the severity properties of any rule can be modified from the Parameter tab. In our case, all rules severity is equal to each other since they are all evaluated as compliant or non-compliant to rule.

In Figure 5.13, the results of the bathroom dimension check is illustrated. In AMHZC Clause 32.1.a, it was stated that the minimum width available in bathrooms cannot be smaller than 1.5 meters. However, in the sample building model, the minimum width is designed as 1.41 meters. Therefore, after checking the rules, SMC illustrates the issue about this width parameter. In the Info tab, it says the width of the bathroom is smaller than the required (1.5 meters) dimension.

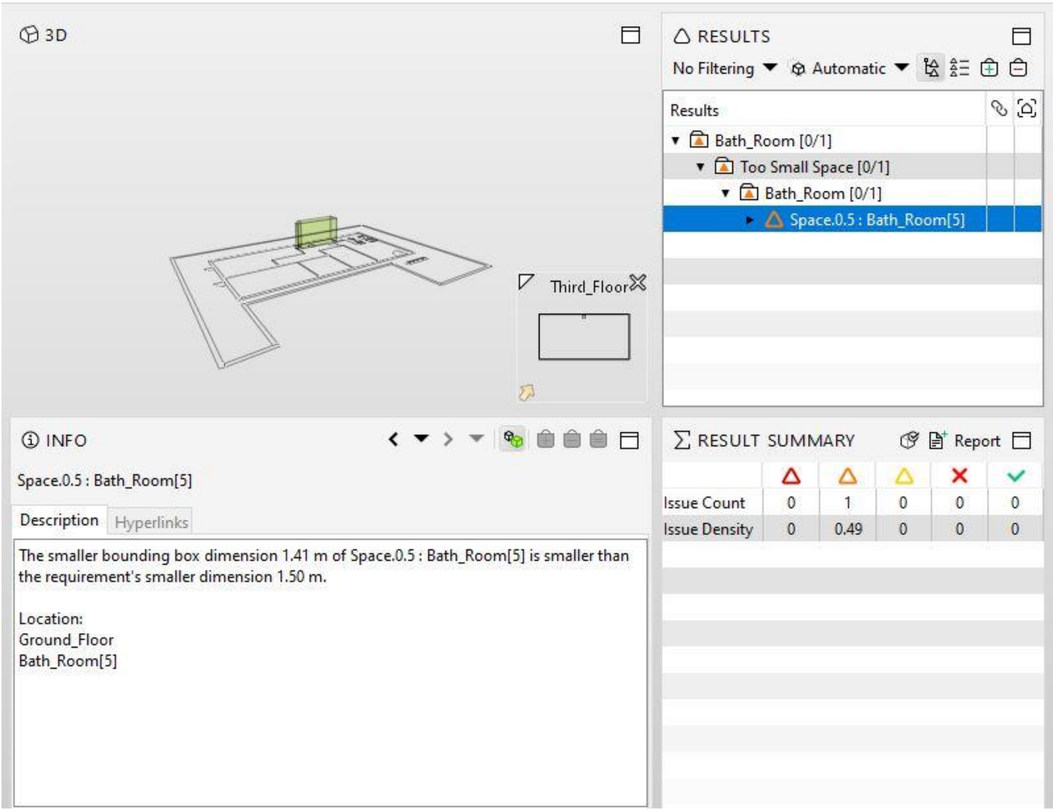

Figure 5.13. Checking Results of AMHZC R32.1.a Bathroom Minimum Width

After checking of the sample building model, it is possible to report these results in different formats and ways. One way to report issues is by using the Result Summary tab’s report button. By using this feature, the results can be viewed in .xls format. Also, the .xls format can be edited even before publishing the results. In Figure 5.14, Result Summary tab report settings are illustrated.

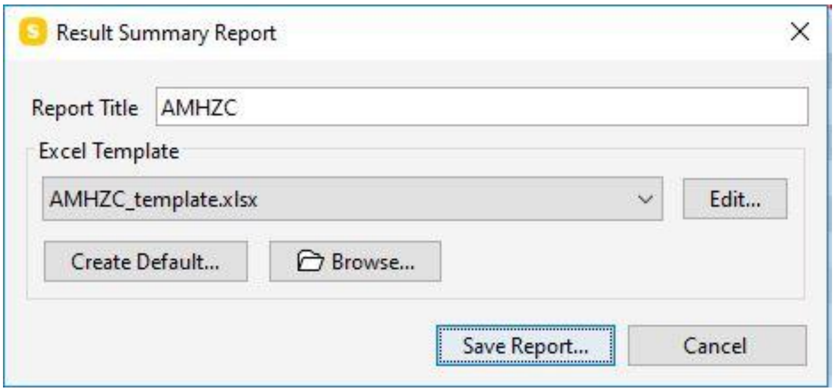

Figure 5.14. SMC Result Summary Tab Report Settings

Another way to report results using SMC is to create slides of the related issues and then using these slides to make visual presentations about the issue. To create slides of the related issues, first the issue is chosen from the Results tab and then the little house icon with rectangular line around it is double-clicked (Figure 5.15). After creating slides of the related issues, by using the Presentation tab's new presentation button, new presentation about the issues is created (Figure 5.16). Then, by using the report button on the Presentation tab, different formats of the related issues can be created and visualized.

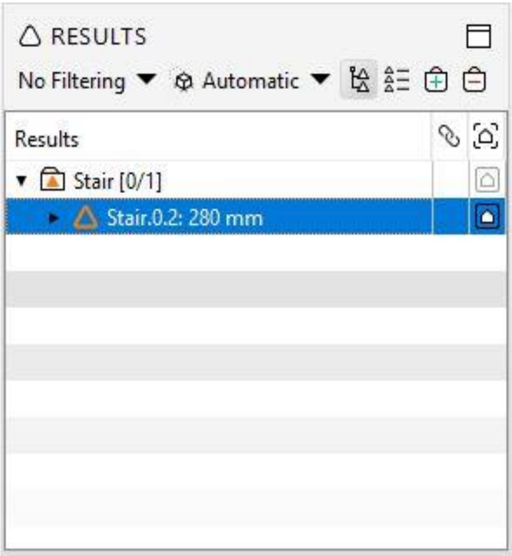

Figure 5.15. SMC Adding Slides for Presentation

Figure 5.16. SMC Presentation View of Issues

CHAPTER 6

CONCLUSIONS

6.1. Finding and Discussions

In this thesis, a formal representation of building codes is proposed and demonstrated with a methodology which is explained in detail in Section 3.5. This representation provides a basis to decompose a building code, a classification to group the building code rules, and the interpretability conditions of the building code rules to eliminate not-interpretable rules. Utilization of the proposed representation is intended to ease the automated code-checking by investigating the building code structure, determining rule properties, and representing the building code completely.

Automated code-checking has been a research field for academicians for a long time. However, most of the studies in this field focus on analysis of several rules rather than representing a building code completely. It can be said that these studies have limited application in practice. Also, the application of these studies require expert knowledge such as different programming languages and hence they are not widely adopted. Therefore, utilization of these studies are limited to specific people with specific knowledge. Moreover, the majority of the studies utilize black box methods which do not share open-access information about the implementation of codes in code-checking tools. The proposed building code representation in this thesis aims to improve the above limitations.

The main contribution of this thesis is the building code representation methodology as explained in Chapter 3 in detail. The proposed building code representation takes Solihin's (2016) rule classification study as a baseline and further modifies it in order to completely cover building codes. Although Solihin's study has a solid base for classifying the rules, it is found limitted in covering a building code completely when codes refer to each other. Therefore, it is modified to represent building codes completely and without ambiguities. This thesis demonstrates that utilizing the proposed building code representation eases the representation of building codes in computer readable format.

In the building code representation methodology, a framework is used to illustrate the process steps of building code representation methodology as explained in Section 3.5. The proposed building code representation methodology consists of five steps:

- Determination of scope: Definition of what to be represented from a building code.
- Building Code Analysis: Extraction of related clauses, statements and rules according to the scope of the automated code-checking.
- Interpretability of rules: Determination of interpretability of rules obtained from the building code analysis step.
- Classification of rules: Determination of rule classes of the Interpretable of Semi-Interpretable rules.
- Determination of necessary variables according to the results: Determination of the building model LOD and decision of which automated code-checking system to use according to the results from previous steps.

Second contribution of this thesis is the automated code-checking process analysis which is done in Chapter 2. It is crucial to understand automated code-checking process in order to conduct studies related to this topic. Therefore, in Chapter 2, the automated code-checking process and their components are defined. Moreover, the detailed information and previous studies about automated code-checking process is carefully explained. By using this information, the building code representation framework is proposed.

Third contribution of this thesis is the building code implementation framework defined and explained in detail in Section 5.2. This framework defines the steps to be taken in order to implement the proposed building code representation in an automated code-checking system or tool. However, this framework is developed to be used in Solibri Model Checker (SMC) only. On the other hand, with a few adjustments and modifications, this framework can be utilized in other automated code-checking systems and tools. The developed building code implementation framework consists of three process steps:

- Representation of rules in the computer readable format defined in the building code representation for the purposes of automated code-checking.
- Utilization of the computer readable rules and the building model in the automated code-checking process.
- Reporting of the results.

Fourth contribution of this thesis is the building code implementation case study which aimed to represent a building code completely in an automated code-checking environment. In similar studies, building codes are fully represented with their methodologies. However, it is not attempted to fully implement building codes within an automated code-checking environment.

In this thesis, Ankara Municipality Housing and Zoning Code (AMHZC) is used to verify the proposed building code representation. The verification process consists of analysis of AMHZC, implementation of the AMHZC within an automated code-checking tool and a case study to test the capabilities of the proposed building code representation, automated code-checking tool, and Building Information Modeling (BIM) tool. The proposed building code representation can be successfully used to analyze the AMHZC. Moreover, this study resulted in some lessons-learned for understanding which rules in Turkish codes are representable in computer interpretable format and what are the characteristics of rules that prevents the computer interpretability. However, the implementation of automated code-checking has showed several limitations. Firstly, in Solibri Model Checker (SMC) there are some building components and information that cannot be represented, such as roof slopes. Although these components and their relevant information are defined in BIM tools and AMHZC, SMC does not have the capabilities to represent them. Secondly, building codes have a complex structure which comes with complex requirements in their rules. Even a single rule in a building code may contain several requirements, components, and relations with other rules. Therefore, representing these rules in SMC is not possible. Lastly, the case study was successful to test capabilities of the proposed building code representation and automated code-checking tool, but the utilized BIM tool have several limitations. As stated before, Autodesk Revit 2019 is used for creating a sample building model. There are some information that cannot be defined in this BIM tool such as naming of building site, naming of zones, etc. which needed to be utilized in SMC through IFC data model.

6.2. Limitations and Future Research

In this thesis only Ankara Municipality Housing and Zoning Code (AMHZC) rules are represented using the proposed methodology. The reason for choosing the AMHZC is that Ankara is the capital of Turkey with more than 5 million population and AMHZC has higher level of complexity than most of the standards and regulations which makes representation of AMHZC in computer readable format a hard task. For future research, different building codes from different countries can be represented using the proposed building code representation methodology.

Solibri Model Checker (SMC) has been used to represent textual building codes in computer readable format. The reasons for choosing SMC are explained in Section 5.1. For future research, different automated code-checking systems and tools can be used to implement proposed building code representation methodology. Also, it is possible to develop new automated code-checking systems and tools to further improve the implementation of the proposed methodology.

Another limitation of this thesis is that only interpretable rules are chosen to be represented in computer readable format. Semi-Interpretable, Non-Interpretable and

Reference rules are not included in this study. For future research, efforts to auto-check these rules can be developed with modifying the building code representation methodology proposed in this study to cover a building code without any exceptions.

The study presented in this thesis can be improved further by adding different concepts to the automated code-checking process. There can be several approaches for future research to increase efficiency of converting building codes into digital formats. For instance:

- A system can be developed to automatically detect and interpret building codes in written format to represent them in automated code-checking systems and tools.
- An approach can be developed to store different building codes in a database, which allows modifications of rules and creates a building code database available in computer readable format for future applications.
- A plug-in can be developed for BIM tools to check component's compatibility with the selected building code while in designing phase.

REFERENCES

American Institute Of Steel Construction. (1969). *Specification for the Design, Fabrication & Erection of Structural Steel for Buildings.* Retrieved from https://www.aisc.org/globalassets/aisc/manual/15th-ed-ref-list/specification-for-the-design-fabrication-and-erection-of-structural-steel-for-buildings.pdf

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

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

Bouzidi, K. R., Fies, B., Faron-Zucker, C., Zarli, A., & Thanh, N. Le. (2012). Semantic Web Approach to Ease Regulation Compliance Checking in Construction Industry. Future Internet, 4(3), 830–851. DOI: 10.3390/fi4030830

Digital Alchemy. (n.d.). Retrieved from https://d-alchemy.com/services

Ding, L., Drogemuller, R., Rosenman, M., Marchant, D., & Gero, J. (2006). Automating code checking for building designs: DesignCheck. In K. Brown, K. Hampson, and P. Brandon (eds), Clients Driving Innovation: Moving Ideas into Practice (12–14 March 2006). Cooperative Research Centre (CRC) for Construction Innovation, Wiley-Backwell, UK.

dRofus. (n.d). Retrieved from https://www.drofus.no/en/product.html

Eastman, C. M., Lee, J.-m., Jeong, Y.-s., & Lee, J.-k. (2009). Automatic Rule-Based Checking of Building Designs. *Automation in Construction*, 18(8), 1011-1033. doi: 10.1016/j.autcon.2009.07.002.

Express Data Manager. (n.d.). Retrieved from http://www.jotneit.no/express-data-manager-edm

Fenves, S. J. (1966). Tabular Decision Logic for Structural Design. *Journal of Structural Division ASCE*, 92, 473-490.

Fenves, S.J., Garrett Jr., J. H., Kiliccote H., Law, K. H., & Reed, K.A. (1995). Computer representation of design standards and building codes. *The International Journal of Construction Information Technology*, 3(1), p. 13-34.

Fenves, S. J., Gaylord, E. H., & Goel, S. K. (1969). Decision Table Formulation of the 1969 AISC Specification. *Civil Engineering Studies*, SRS-347. Retrieved from https://www.ideals.illinois.edu/handle/2142/14275

Fenves, S. J., & Wrigth, R. N. (1977). The Representation and Use of Design Specifications, Technical Note 940, NBS, Washington, DC.

Fenves, S. J., Wright, R.N., Stahl, F, I., & Reed, K. A. (1987). Introduction to SASE: Standards Analysis, Synthesis and Expression, NBSIR 873513, NBS, Washington, D.C.

Garrett Jr., J.H., & Fenves, S.J. (1987). A knowledge-based standards processor for structural component design. *Engineering with Computers*, 2 (2) (1987) 219–238. Retrieved from https://link.springer.com/content/pdf/10.1007/BF01276414.pdf

General Services Administration. (2007a). *GSA Building Information Modeling Guide Series 01 – Overview* .Retrieved from https://www.gsa.gov/cdnstatic /GSA_BIM_Guide_v0_60_ Series01_Overview_05_14_07.pdf

General Services Administration. (2007b). *U.S. Courts Design Guide* .Retrieved from https://www.gsa.gov/cdnstatic/Courts_Design_Guide_07.pdf

General Services Administration. (2016). *GSA Building Information Modeling (BIM) Guide 07 - Building Elements* .Retrieved from https://www.gsa.gov/cdnstatic/BIM_Guide_07_v_1.pdf

Hakim, M. M., & Garrett Jr, J. H. (1993). A Description Logic Approach for Representing Engineering Design Standards, *Engineering with Computers*, 9, pp 108-124.

Han, C. S., Kunz, J. C., & Law, K. H. (1998). Client/Server Framework for on-Line Building Code Checking. *Journal of Computing in Civil Engineering*, 12(4), 181.

Han, C., Kunz, J., & Law, K. (1999). Building design services in a distributed architecture. *Journal of Computing in Civil Engineering*, ASCE 13 (1) 12–22. https://doi.org/10.1061/(ASCE)0887-3801(1999)13:1(12)

Han, C., Kunz, J., & Law, K. (2002). Compliance Analysis for Disabled Access, Advances in Digital Government Technology, Human Factors, and Policy, in: William J. McIver Jr., Ahmed K. Elmagarmid (Eds.), Kluwer, Boston, MA. 149–163.

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

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

International Alliance for Interoperability. (1999). *IFC Object Model Architecture*. Retrieved from https://iaiweb.lbl.gov/Resources/IFC_Releases/IFC_Release_2.0/FINAL_Documents/IFC_R2_ObjectModelArchitectureGuide_US .PDF

International Organization for Standardization. (2018). *Industry Foundation Classes (IFC) for Data Sharing in the Construction and Facility Management Industries -- Part 1: Data Schema* (ISO Standard No. 16739-1:2018). Retrieved from https://www.iso.org/standard/70303.html

International Organization for Standardization. (2016). *Industrial Automation Systems and Integration -- Product Data Representation and Exchange -- Part 1: Overview and Fundamental Principles* (ISO Standard No. 10303-1:1994). Retrieved from https://www.iso.org/standard/20579.html

Ismail, A.S, Ali, K.N., & Iahad, N.A. (2017). A Review on BIM-based automated code compliance checking system. In Proceedings of the IEEE International Conference on Research and Innovation in Information Systems, ICRIIS, Langkawi, Malaysia, 16–17 July 2017.

Kasim, T. (2016). BIM-Based Smart Compliance Checking to Enhance Environmental Sustainability (Doctoral thesis). Retrieved from http://orca.cf.ac.uk/78325/

Kerrigan, S., & Law, K. (2003). Logic-based regulation compliance-assistance, Proceedings of the Ninth International Conference on Artificial Intelligence and Law (ICAIL 2003), Edinburgh, Scotland, UK.

Kiliccote, H., Garrett Jr., J.H., Chmielenski, T.J., & Reed, K. A. (1994). The Context-Oriented Model: An Improved Modeling Approach for Representing and Processing Design Standards. Paper presented at the First ASCE Congress on Computing in Civil Engineering, Washington, D.C.

Kiliccote, H., & Garrett, J. H. (1998). Standards Modeling Language. *Journal of Computing in Civil Engineering*, 12(3), 129-135.

Lee, H., Lee, J.-K., Park, S., & Kim, I. (2016). Translating building legislation into a computer-executable format for evaluating building permit requirements. *Automation in Construction*, 71, 49–61. https://doi.org/10.1016/j.autcon.2016.04.008

Lee, J. M. (2010). Automated Checking of Building Requirements on Circulation Over a Range of Design Phases (Doctoral thesis). Retrieved from https://smartech.gatech.edu/bitstream/handle/1853/34802/lee_jae_m_201008_phd.pdf?sequence=1&isAllowed=y

Level of Development Specification. (n.d.). Retrieved from https://bimforum.org/lod/

Liebich, T., & Wix, J. (1999) Highlights of the development process of Industry Foundation Classes, in Lacasse, M. and Vanier, D. (eds) 8th International Conference on Durability of Building Materials and Components, Vol 4: IT in Construction, CIB W78 Workshop, National Research Council of Canada, pp. 2758–75.

Liebich, T. (2014). IFC overview presentation. Retrieved from https://www.bre.co.uk/filelibrary/events/BRE%20Events/BIM%20Conference%20Season/Delivery%20of%20IFC/2-Thomas-Liebich.pdf

Macit, S. (2014). Computer Representation of Building Codes For Automated Compliance Checking (Doctoral thesis). Retrieved from http://openaccess.iyte.edu.tr/bitstream/handle/11147/4296/T001336.pdf?sequence=1&isAllowed=y

Malsane, S., Matthews, J., Lockley, S., Love, P. E. D., & Greenwood, D. (2015). Development of an object model for automated compliance checking. *Automation in Construction*, 49, 51–58. https://doi.org/10.1016/j.autcon.2014.10.004

Martins, J.P., & Monteiro, A. (2013). LicA: A BIM based automated code-checking application for water distribution systems. Automation in Construction, 29(2013), 12–23. https://doi.org/10.1016/j.autcon.2012.08.008

National Institute of Building Sciences buildingSMART alliance®. (2015). *National BIM Standard - United States® Version 3*. Retrieved from https://www.nationalbimstandard.org/files/NBIMS-US_V3_4.2_COBie.pdf

Nawari, N. O. (2018). *Building information modeling: Automated code checking and compliance processes*. 6000 Broken Sound Parkway NW, Suite 300 Boca Raton, FL 33487-2742: CRC Press

novaCITYNETS. (2000). Retrieved http://www.novacitynets.com/news.htm

novaCITYNETS. (n.d.). About Fornax™ - Plancheck Expert. Retrieved from http://www.novacitynets.com/fornax/about.htm

Nufus ve Idari Durum. (n.d.). Retrieved from http://www.ankara.gov.tr/nufus-ve-idari-durum

Nyman, D. J., Fenves, S. J., & Wright, R. N. (1973). Restructuring Study of the AISC Specification Civil Engineering Studies SRS-393 (Vol. 1). Urbana-Champaign: Department of Civil Engineering, University of Illinois Engineering Experiment Station.

Nyman, D. J., & Fenves, S. J. (1975). An Organization Model for Design Specifications. *Journal of structural Division ASCE*, 101(4), 697-716.

Occupational Safety and Health Administration. (2015). *Fall Protection in Construction* (Standard No. 3146-05R 2015). Retrieved from https://www.osha.gov/Publications/OSHA3146.pdf OSHA 3146-05R 2015

OWL Features. (2009, November 12). Retrieved from https://www.w3.org/TR/owl-features/

Pauwels, P., Van Deursen, D., Verstraeten, R., De Roo, J., De Meyer, R., Van De Walle, R., & Van Campenhout, J. (2011). A semantic rule checking environment for building performance checking. *Automation in Construction*, 20(5), 506–518. https://doi.org/10.1016/j.autcon.2010.11.017

Projects International Code Council – AEC3. (n.d.). Retrieved from http://www.aec3.com/5/5_013_ICC.htm

RDF Primer. (2014, February 25). Retrieved from https://www.w3.org/TR/rdf-primer/

RDF Schema. (2014, February 25). Retrieved from https://www.w3.org/TR/rdf-schema/

Salama, D., & El-Gohary, N. (2011). Semantic modeling for automated compliance checking. *Journal of Computing in Civil Engineering*, 641–648. https://doi.org/10.1061/41182(416)79

Sjøgren, J. (2005). Governmental and industry experience in Norway. buildingSMART Norwegian Experiences presentation slides.

Sjøgren, J. (2007). BuildingSMART—a smart way for implementation of standards. buildingSMART presentation slides, European technical data event. Netherlands, Jul 08-11.

Solihin, W., & Eastman, C. (2015). Classification of rules for automated BIM rule checking development. *Automation in Construction*, 53, 69 –82. doi:10.1016/j. autcon.2015.03.003

Solihin, W. (2016). A simplified BIM data representation using a relational database schema for an efficient rule checking system and its associated rule checking language (Doctoral thesis). Retrieved from https://smartech.gatech.edu/handle/1853/54831

Tan, X., Hammad, A., & Fazio, P. (2010). Automated Code Compliance Checking for Building Envelope Design. *Journal of Computing in Civil Engineering*, 203–211. https://doi.org/10.1061/(ASCE)0887-3801(2010)24:2(203)

The American Institute of Architects. (2013). *Building Information Modeling and Digital Data Exhibit* (Standard No. E203-2013). Retrieved from https://www.aiacontracts.org/contract-documents/19026-building-information-modeling-and-digital-data-exhibit

Vassileva, S. (2000). An approach of constructing integrated client/server framework for operative checking of building code, Construction Information Technology 2000, Taking the Construction Industry into the 21st Century, Reykjavik, Iceland.

W. Solihin. (2004). Lessons learned from experience of code-checking implementation in Singapore. Retrieved from http://www.buildingsmart.com/files/u1/ned_20from_20 experience_20of_20code_checking.pdf

Yabuki, N. (1992), An Integrated Framework for Design Standards Processing (Doctoral thesis). Department of Civil Engineering, Stanford University.

Yabuki, N. and Law, K.H, (1993). An Object-Logic Model for the Representation and Processing of Design Standards. *Engineering with Computers*, 9, pp 133-159.

Yang Q.Z., Li X. (2001) Representation and Execution of Building Codes for Automated Code Checking. In: de Vries B., van Leeuwen J., Achten H. (eds) Computer Aided Architectural Design Futures 2001. Springer, Dordrecht. https://doi.org/10.1007/978-94-010-0868-6_24

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

Zhang, S., Teizer, J., Lee, J. K., Eastman, C. M., & Venugopal, M. (2013). Building Information Modeling (BIM) and Safety: Automatic Safety Checking of Construction Models and Schedules. *Automation in Construction*, 29, 183–195. https://doi.org/10.1016/j.autcon.2012.05.006

Zhong, B. T., Ding, L. Y., Luo, H. B., Zhou, Y., Hu, Y. Z., & Hu, H. M. (2012). Ontology-based semantic modeling of regulation constraint for automated construction quality compliance checking. *Automation in Construction*, 28, 58–70. https://doi.org/10.1016/j.autcon.2012.06.006

APPENDICES

A. Analysis of AMHZC

ID	Textual Expression of Rule	Type
Madde 9 - Bina yol cephe(leri) genişliği		
1	İmar durumu (yapılaşma şartları), Taban Alanı Katsayısı (TAKS) ve/veya Kat Alanı Katsayı (KAKS=Emsal) ve yapı yaklaşma sınırları ile belirlenen parsellerde yapılacak binalarda bina cephe genişlikleri ve derinlikleri serbesttir. Bölge kat nizamı planlı alanlardaki ayrık nizam parsellerde inşa edilecek konut binalarının cephe genişliği 35 metreyi geçemez.	Rule
2	Ayrık yapı nizamına tabi olan yerlerde bu Yönetmelik, plan hükümlerine dayanılarak verilebilecek en fazla bina taban alanını (TA), en az bahçe mesafelerini aşmamak koşulu ile yapı yerini tespite ilgili belediye imar birimleri yetkilidir.	Rule
3	İkili veya üçlü blok yapılması gereken yerlerde, daha uygun çözüm maksadı ile birkaç dar parseli birlikte değerlendirerek o yer için tespit edilen yapı karakterine uyacak bina cepheleri toplamı 35 metreyi aşmayacak olan ikili veya üçlü bloklar teşkil etmeye ilgili belediye imar birimleri yetkilidir.	Rule
Madde 10 - Planla veya planla belirlenmemişse bu yönetmeliğe göre belirlenecek TAKS'ı geçmemek kaydıyla bina derinlikleri		
1	Bina derinliği hesabı;	
1.a	İmar planlarında bina derinlikleri belirlenmemiş yerlerde derinlikler;	
1.a.1	I = L – (K + H / 2) formülü ile hesaplanır. Burada: I = bina derinliğini, L= parsel ortalama derinliğini, K= ön bahçe mesafesini, N=bina kat adedini, ifade eder. H= (3 x N) + 0.50 formülü ile hesaplanır.	Rule
1.a.2	Düzgün geometrik şekillere uymayan parsellerde, alan dengelemesi suretiyle basit ve düzgün bir şekle çevrilmek suretiyle bulunacak derinlik, bu parsellerin ortalama derinliği kabul edilir.	Rule
1.a.3	İkili blok yapı nizamlı parsellerde, ortak kenar ortalama derinlik olarak alınır.	Rule
1.a.4	Ön bahçe mesafesi 5 metreden fazla olan parsellerde hesapta (K) değeri 5 metre olarak kabul edilir.	Rule
1.a.5	(1) numaralı alt benteki formülle bulunacak bina derinlikleri 15 metreden az çıktığı takdirde, en az bahçe mesafeleri uygulanmak şartı ile 15 metre bina derinliği verilir.	Rule

ID	Textual Expression of Rule	Type
1.a.6	Formülle bulunacak derinlik değerlerinin küsuratı 0.50 metreden az olanlar 0.50 metreye; 0.50 metreden fazla olanlar 1 metreye yuvarlanır.	Rule
1.a.7	Bu formülle hesaplanan derinliklere açık ve kapalı çıkmalar dâhil değildir.	Rule
1.b	Değişik parsellerden oluşan adalarda, (a) bendinin (1) numaralı alt bendindeki formüle göre hesap edilecek değişik bina derinliklerinin sebep olacağı düzensizliği gidermek, yapı adasının bir kenarında verilecek bina derinliklerinin aynı sıradaki mevcut ve imar mevzuatına göre muhafaza edilecek binaların derinlikleri ile uyumlu olmasını sağlamak, verilecek bina derinliğinin ada içi boşluğunu bölücü bir nitelik kazanmasını önlemek amaçları ile adanın tamamında etüd yapılarak plan değişikliği ile derinlikler sınırlandırılabilir veya en az arka bahçe mesafesine uymak kaydı ile artırılabilir.	Rule
1.c	Blok, ikili blok, blokbaşı ve bitişik yapı nizamına tabi parsellerde bitişik sınırda derinliklerin aynı olması esastır. Bunu sağlamak üzere ilgili belediye imar birimleri en az bahçe mesafelerini zedelememek koşulu ile derinlikleri azaltabilir/arttırabilir.	Rule
1.ç	İkili blok binalardan bir tanesi imar mevzuatına göre inşa edilmiş ve derinliği bu Yönetmelikteki ölçüyü aşıyorsa diğeri derinlik bakımından mevcut binaya uyabilir.	Rule
2	En fazla bina derinlikleri için dikkate alınacak esaslar;	
2.a	Konut, ticaret yolu ve bölgelerinde; ayrık nizamda 22 metreden, bitişik, blok, blok başı ve ikili blok nizam binalarda 20 metreden fazla bina derinliği verilemez.	Rule
2.b	Yüksek inşaat bölgelerinde; bu bölgelerde yapılacak binaların derinliği bu Yönetmelikteki bahçe mesafelerine uyulması şartıyla en fazla derinlik kayıtlarına tabi değildir.	Rule
2.c	Küçük sanayi bölgelerinde; bu bölgelerde yapılacak binaların derinliği, derinlik formülüne bağlı olmayıp parsel arka hududuna 5 metre kalana kadar verilir.	Rule
2.ç	Sanayi bölgelerinde; KAKS verilen parsellerde bina derinliği bu Yönetmelikteki bahçe mesafelerine uyulması şartıyla en fazla derinlik kayıtlarına tabi değildir. KAKS verilmeyen parsellerde derinlik planla belirlenir.	Rule
3	Yol genişlemelerinde bina derinlik hesabına dair esaslar;	
3.a	Bölge kat nizamı planları ile tespit edilmiş olan yolların genişletilmesi halinde yola giden kısmın teknik ve sosyal altyapı gereksinimi olarak bedelsiz terkin edilmesi şartıyla, bu yollar üzerindeki parsellerde yapılacak binaların derinlikleri parselin değişiklikten evvelki özgün haline göre hesaplanabilir; aksi halde, parselin yeni derinliği esas alınır. Cephe hatları ise zorunluluk olmadıkça aynen muhafaza edilir.	Rule
3.b	Emsalli parsellerde yol genişlemesi olması halinde yola giden kısım teknik ve sosyal altyapı gereksinimi olarak bedelsiz terkin edilmek ve planda hüküm bulunmak koşuluyla terkten önceki parsel alanı üzerinden emsal (KAKS) belirlenebilir.	Rule

ID	Textual Expression of Rule	Type
4	Köşebaşı parsellerde denk taban alanı (TA) sabit kalmak, en az yan ve arka komşu mesafeleri kayıtlarına uyulmak ve planla ya da planda belirlenmemişse bu Yönetmelikle belirlenen emsal alanı aşılmamak şartıyla derinlik sınırlamasına tabi olunmadan, bina boyutları çevre adalardaki yapılaşmalar dikkate alınarak ilgili belediyenin imar birimince tespit edilebilir. Bu belirlemede parselin yüz aldığı yollar üzerindeki komşu parsellerin derinliklerinin aşılmaması esastır.	Rule
5	Mevcut binalara kat ilavesinde derinlik hesabında;	
5.a	Bu Yönetmeliğin yürürlüğe girmesinden önce yürürlükte olan mevzuat uyarınca inşa edilmiş binalara kat ilavesi halinde, müktesep hak teşkil etmemek koşuluyla mevcut binanın inşaat derinliğine uyulur.	Rule
5.b	2981 sayılı Kanuna göre ruhsat almış yapıların kat ilavesi halinde ise ilave katın derinliği bu Yönetmeliğin derinlik formülüne uymak zorundadır.	Rule
6	Bina derinlikleriyle ilgili bu maddedeki hükümler, bölge kat nizamı planlı alanlarda uygulanır, diğer planlı alanlarda bina derinlikleri serbesttir.	Rule
Madde 11 - En fazla bina yükseklikleri (H max)		
1	Verilen kat sayısına göre saçak seviyesi (H=bina yüksekliği, n=zemin kat hariç kat sayısı olmak üzere);	
1.a	Bölge kat nizamı planlı alanlardaki konut ile ticaret parsellerinde; H=(3.50) + n x (3.00)	Rule
1.b	Diğer planlı konut alanlarında H=(5.00) + n x (3.60)	Rule
1.c	Diğer planlı konut dışı kullanım alanlarında H=(5.50) + n x (4.00) formülüne göre hesaplanır.	Rule
1.ç	Plan notlarında verilen saçak seviyelerinin güncellenmesi için, plandaki saçak seviyesinin 0.50 metre eksiğinin 3'e bölünmesi ile elde edilecek kat adedi üzerinden güncel formüle göre hesaplama yapılır.	Rule
2	İkili blok binalarda, binalardan bir tanesi imar mevzuatına göre inşa edilmiş ve yükseklik bakımından bu Yönetmelikteki ölçüleri aşıyorsa, diğer bina yükseklik bakımından mevcut binaya uyabilir.	Rule
3	Küçük sanayi bölgelerinde yapılacak binaların yükseklikleri 17.50 metreyi geçemez. Bu bölgelerdeki umumi binalara planla veya planla belirlenmemiş ise bu Yönetmelikle belirlenen TAKS ve KAKS içinde kalınmak şartı ile ihtiyaca göre ilgili belediyenin imar birimince daha fazla yükseklik verilebilir.	Rule
4	Eğitim, sağlık, dini tesis, sanayi yapıları, imalathane, katlı otopark, sinema, tiyatro ve konferans salonları, kongre merkezi, alışveriş merkezleri, spor salonları, tarım ve hayvancılık amaçlı tesisler gibi özellik arz eden binaların yükseklikleri, binanın özelliği ile en az bahçe mesafeleri şartları göz önünde bulundurulmak suretiyle mimari estetik komisyonunca belirlenir.	Rule
5	Ahşap binalara; ayrık nizamda olmak, bahçe mesafeleri şartlarına uymak ve bölge kat nizamı içinde kalmak kaydı ile en fazla iki kat yükseklik verilebilir.	Rule
6	Kerpiç vb. binaların kat adedi ve bina yüksekliği 1 bodrum ve 1 normal kat olmak üzere 3.50 metreyi geçemez.	Rule

ID	Textual Expression of Rule	Type
7	Bu Yönetmelikte gösterilen yüksekliklerin; herhangi bir anıtın, korunması gerekli tescilli taşınmaz kültür ve tabiat varlıklarının, tarihi ve mimari değeri olan eserlerin görünüşünü bozması halinde Kültür Varlıklarını Koruma Bölge Kurulu görüşü alınarak uygulama yapılır.	Rule
Madde 12 - Farklı yükseklikte bina yapabilme		
1	Sahiplerinin toplu müracaat ve muvafakatleri halinde; bir yapı adasının tamamı üzerinde tüm parselleri tevhiden o adanın parsellerinin KAKS ve TAKS toplamını ve planla belirlenen kat adedini aşmayan değişik yükseklikte vedüzende kitleler yapılabilir.	Rule
2	Umumi binalara ayrık nizamda olmak, parselin planla veya planla belirlenmemiş ise bu Yönetmelikle belirlenen KAKS ve TAKS değerleri aşılmamak, şehircilik ve planlama esaslarına uyulmak kaydı ile bölge kat nizamı dışında yapı düzeni ve yükseklik tespit edilebilir.	Rule
3	Birinci ve ikinci fıkralarda belirtilen düzenlemeler, ilgili ilçe belediye meclisinin kabulü ve Ankara Büyükşehir Belediyesinin onayı ile kesinleşir. Birinci ve ikinci fıkralar ile ilgili tekliflerde açıklayıcı maket, perspektif ve kentsel tasarım projesi istenebilir.	Rule
4	Mevcut yapılaşmalarla ilgili olarak;	
4.a	Bu Yönetmeliğin yürürlüğe girmesinden önce yürürlükte olan mevzuata uygun olarak yapılmış ayrık nizam binalara kat ilavesi halinde, planda aksine bir hüküm bulunmamak kaydıyla Ankara bölge kat nizamını gösteren plandaki bina yüksekliğine bağlı olmaksızın bölgenin kat adedine uyulur. Bu kabil ilave katlarda kat yüksekliği en az ölçüden fazla olamaz.	Rule
4.b	Bitişik nizam binalarda bloğun tamamının rölevesi yapılmak, planla veya planla belirlenmemiş ise bu Yönetmelikle belirlenen yapılaşma şartları aşılmamak suretiyle ilgili belediye imar birimince uygun görülecek etüde göre işlem yapılır.	Rule
4.c	Bir adada bu Yönetmeliğin yürürlüğe girmesinden önce teşekkül etmiş binaların saçak seviyesi ile bölge kat nizamı planının bu ada için tayin ettiği en fazla saçak seviyesi arasında estetik mahzur doğuran farklar bulunması halinde, bu adada yeni yapılacak binalara verilecek saçak kotu, mevcut binaların röleveleri yapılmak ve kat nizamı planı ile ahenk teşkil edecek şekilde ayarlanmak suretiyle ilgili belediyenin imar birimince tespit edilebilir.	Rule
5	Eksik katlı bina yapma şartları;	
5.a	Uygulama imar planında veya planda olmaması halinde bu Yönetmelikte gösterilen kat adedi ve bina yüksekliği aşılamaz. Ancak, planla veya bu Yönetmelikle belirlenen kat adedine veya bina yüksekliğine uygun olarak bahçe mesafesi bırakılmak ve ilgili belediyenin uygun görmesi koşuluyla daha az katlı bina yapılabilir. Uygulama imar planlarında bu uygulamanın yapılmasına ilişkin hüküm olması halinde ilgili belediyenin uygun görmesi koşulu aranmaz.	Rule
5.b	Eksik katlı yapılan binalarda yapı ruhsatı, yapı kullanma izin belgesi ve enerji kimlik belgesi, yapılan kısım için düzenlenir. Daha sonradan tamamlanmak istenmesi halinde, yürürlükteki plan ve mevzuat hükümlerine	Rule

ID	Textual Expression of Rule	Type
	uygun olarak ilave ruhsat düzenlenmek ve binanın tamamı için enerji kimlik belgesi onaylanmak zorundadır.	
5.c	Eksik katlı binalara imar planına aykırı olmamak koşuluyla kat ilavesi yapılabilmesi için temel ve statik çözümlerinin, yangın tedbirlerinin, enerji verimliliğinin, otoparkının, sığı-nağının, merdiven, asansör yeri, ışıklık ve diğer yapı elemanlarının, plan ve bu Yönetmelikte gösterilen azami yüksekliğe göre hesaplanması ve bırakılması zorunludur.	Rule
5.ç	Eksik katlı inşa edilen binanın mevcut haliyle veya tadilat yapılarak yürürlükteki plana ve mevzuata uygunluğunun sağlanamaması halinde bina yıkılmadan kat ilavesi yapılmasına izin verilmez. Eksik katlı binalara yapılacak ilavelerde fenni mesuliyet, temel ve statik hesapları, yangın tedbirleri ve enerji verimliliği konuları da dahil mevcut yapı ve ilave yapılan kısımları kapsayan teknik rapor da düzenlemek suretiyle yapı denetim kuruluşlarınca üstlenilir. Zemin katta ticari faaliyet yürütülebilen binalarda ve birden fazla yapı yapılabilen parsellerde de bu hüküm uygulanır.	Rule
Madde 12 - Farklı yükseklikte bina yapabilme		
1	Yoldan kotlandırılan binalara yol cephelerinde, girişin hizasındaki bordür taşı üst seviyesinin altında giriş yapılamaz.	Rule
2	Yola bakmayan cephelerden, köprü veya giriş şeridi eksenindeki bordür seviyesinden en fazla 1,50 metre inilmek veya çıkılmak suretiyle de giriş yapılabilir.	Rule
3	Tabii zeminden kotlandırılan parsellere girişler birinci ve ikinci fıkralardaki şartlara tabi değildir. Giriş tabii zemine uyumlu olarak düzenlenecek merdiven ve rampalarla sağlanabilir.	Rule
4	Yoldan ve/veya tabi zeminden kotlandırılan ve üzerinde birden fazla bina yapılan ada/parsellerdeki bina girişleri, her kitlenin kendi ±0,00 röper kotu ve birinci, ikinci ve üçüncü fıkralardaki esaslar göz önüne alınarak düzenlenir.	Rule
Madde 24 - Bina girişleri ve yaya rampaları		
1	Bina giriş koridoru genişliği, ana merdivene ve asansöre ulaşıncaya kadar dış kapı genişliğinden az olmamak koşuluyla umumi binalarda en az 2.20 metre, diğer binalarda ise en az 1.50 metredir.	Rule
2	Yoldan doğrudan giriş alan binalarda, girişin hizasındaki bordür taşı üst seviyesinin altında giriş yapılamaz.	Rule
3	Tabi zeminden kotlandırılan parseller birinci ve ikinci fıkralardaki şartlara tabii de-ğildir. Girişin, tabi zemine uyumlu olarak düzenlenen merdiven ve rampalarla sağlanması gerekir.	Rule
4	Bölge kat nizamını bozacak şekilde tesviye yapılamaz.	Rule
5	Konut binalarının zemin katlarının dükkân veya mağaza olarak düzenlenmesi halinde dükkân ve mağaza girişlerinin sadece yol cephesinden yapılması gerekir.	Rule
6	Döşeme kaplamalarında kaymayı önleyen, tekerlekli sandalye ve koltuk değneği hareketlerini güçleştirmeyen, yüzeyleri sert, stabil, düzgün ve standardına uygun malzeme kullanılması zorunludur.	Rule

ID	Textual Expression of Rule	Type
7	Binalarda ve girişlerinde engellilerin erişimine yönelik TSE standartlarına uyulması zorunludur.	Rule
8	Rampaların kenar korumaları, genişlikleri, sahanlıkları, korkuluk ile küpeşte ve kaplama malzemeleri engellilerin de dolaşımına olanak sağlayacak şekilde TSE standartlarına uygun yapılmak zorundadır. Rampaların genişliği 1.50 metreden az olmamalıdır.	Rule
9	Bina girişlerinde engellilere yönelik ön bahçede parsel sınırına kadar giriş rampası veya merdivene bitişik dar kenarı en az 0.90 metre ve alanı en az 1.20 m² engelli asansörü yeri ya da mekanik kaldırma iletme platformu yapılır. Aile ve Sosyal Politikalar Bakanlığının görüşü alınmak suretiyle engellilerin kullanımı için farklı uygulama yapılabilir.	Rule
10	Bina girişinde ve bina içinde bulunan rampaların eğimleri aşağıdaki değerlere uygun olmak zorundadır: En fazla yükseklik En fazla eğim 15 cm ve daha az 1:12 (% 8) 16-50 cm arası 1:14 (% 7) 51-100 cm arası 1:16 (% 6) 100 cm üzeri 1:20 (% 5)	Rule
11	Rampalarda ve ara sahanlıklarda kesintisiz olarak 0.90 metre yükseklikte 1. düzey ve 0.70 metre yükseklikte 2. düzey, elle tutulduğunda kolay kavranabilecek şekilde 32-45 mm çapında küpeşte bulunmak zorundadır.	Rule
12	Birbirini takip eden rampaların başında ve sonunda sahanlıklar bulunmalıdır. Bu sahanlıkların boyu 1.80 metreden az olamaz. Sahanlıkların genişliği 1.50 metreden az olamaz. Sahanlık eğimleri en fazla 1/50= 0.02 olabilir.	Rule
Madde 25 - Merdivenler		
1	En az merdiven genişliklerinde;	
1.a	Tek bağımsız bölümlü konutlardaki dâhili merdivenlerin genişliği 1 metreden,	Rule
1.b	Birden fazla bağımsız bölümlü konutlardaki genel merdivenlerin genişliği ise 1.20 metreden, az olamaz	Rule
1.c	Konut kullanımı harici binalardaki merdiven genişliği en az 1.50 metredir.	Rule
2	Basamak yükseklik ve genişliklerinde;	
2.a	Basamak genişlik ve yüksekliği; 2h + b =61- 65 eşitliğini sağlayacak şekilde saptanır. (h) cm olarak basamak yüksekliği, (b) cm olarak basamak genişliğidir.	Rule
2.b	Basamak yüksekliği, asansörü olmayan binalarda 0.16 metreden, asansörlülerde 0.175 metreden, bina dışındakilerde ise 0.15 metreden fazla olamaz.	Rule
2.c	Basamak genişliği konut merdivenlerinde 0.28 metre, diğer binalarda 0.30 metre, dış merdivenlerde ise 0.35 metreden az olamaz.	Rule
2.ç	Dönel merdivenlerde basamak genişliği en dar kenardan itibaren 0.15 metre uzaklıkta 0.10 metreden, basamak ortasında ise 0.28 metreden az olamaz.	Rule
3	Merdiven kolu, kat ve ara sahanlıklar ile ölçüleri hakkında;	
3.a	Ortak merdivenlerde rıht sayısının 17'den fazla olması durumunda, iki kat arasında bir ara sahanlık yapılması zorunludur.	Rule

ID	Textual Expression of Rule	Type
3.b	Ara ve kat sahanlıklarının genişliği iki kollu merdivenlerde merdiven genişliğinden az olamaz.	Rule
3.c	Tek kollu ve dönel merdivenlerde ara sahanlık, basamak ortasında; (n x 63+b) toplamı kadar olacaktır. Örneğin; 1x 63 + 29 = 92 cm. 2x 63 + 29 = 155 cm.	Rule
4	Kullanım değişikliğinde ve karma kullanımlı binalarda merdivenler için;	
4.a	Mevcut binalarda umumi binaya dönüştürmek gibi kullanımı değiştirmeye yönelik taleplerde, mevcut merdiven ölçülerinin yeni kullanıma dair merdiven ölçülerini sağlaması zorunludur.	Rule
4.b	Karma kullanımlı binalarda her kullanım için ayrı merdiven evi düzenlenmesi zorunludur.	Rule
5	Servis merdivenleri ve ölçüleri hakkında;	
5.a	İskan edilmese de çatı ve bodrum katlara ulaşan ortak merdivenler, merdiven en az ölçülerine uyarlar.	Rule
5.b	Tüm binalarda ana merdivenle bodrum katlarda tertiplenen ortak yerler arasında bağ-lantı kurulması zorunludur.	Rule
5.c	Tek bağımsız bölümlü konut binalarında bodrum katlar iskan edilmediği takdirde dı-şarıdan ulaşım sağlanabilir.	Rule
6	Yangın önlemleri ve kaçış merdivenleri ile ilgili olarak Binaların Yangından Korunması Hakkında Yönetmelik hükümleri geçerlidir.	Rule
7	Konut dışı kullanımlarda birden fazla kata yayılan ve herkesin yararlandığı bağımsız bölüm merdivenleri de bu madde hükümlerine tabidir.	Rule
8	Merdivenlerin her iki tarafında da engelliler ile ilgili TSE standartlarına uygun korkuluk ve küpeşte yapılması ayrıca basamak, kat ve ara sahanlıkların kaplamalarında da standartlara uyulması zorunludur.	Rule
Madde 26 - Asansörler		
1	Uygulama imar planına göre kat adedi 3 olan binalarda asansör yeri bırakılmak, 4 ve daha fazla olanlarda ise asansör tesis edilmek zorundadır. Kat adedi hesabına iskan edilen bodrum katlar da dahil edilir. Daha az katlı yapılarda da asansör yapılabilir.	Rule
2	Asansör mecburiyeti bulunan binalarda; asansör sayısı ve kapasiteleri trafik hesabıyla belirlenir, ancak giriş katından başlamak üzere, 10`dan fazla kat bulunduğu veya toplam bağımsız bölüm sayısı 20'yi geçtiği takdirde en az 2 adet, toplam bağımsız bölüm sayısı 50'yi geçtiği takdirde en az 3 adet, toplam bağımsız bölüm sayısı 100'ü geçtiği takdirde en az 4 adet asansör yapılması zorunludur.	Rule
3	Tek asansörlü binalarda; asansör kabininin dar kenarı 1.20 metre ve alanı 1.80 m2 'den, kapı net geçiş genişliği ise 0.90 metreden az olamaz. Asansör kapısının açıldığı sahanlıkların genişliği, asansör kapısı sürgülü ise en az 1.20 metre, asansör kapısı dışa açılan kapı ise en az 1.50 metre olmak zorundadır. Birden fazla asansör bulunan binalarda, asansör sayısının yarısı kadar asansörün bu fıkrada belirtilen ölçülerde yapılması şarttır. Tek sayıda asansör bulunması durumunda sayı bir alta yuvarlanır. TSE standartlarının bu fıkrada belirtilen ölçü ve miktarlardan küçük olması halinde; taban alanında yapılaşma hakkı 200 m 2'nin altında olan parseller ile tek bağımsız bölümlü	Rule

ID	Textual Expression of Rule	Type
	müstakil konut binalarında TSE standartlarına uyulmasına ilgili idaresi yetkilidir.	
4	Mevcut binalarda yapılacak tadilatlarda, bu madde hükümlerinin ya da TSE standartlarının uygulanmasında idaresi yetkilidir.	Rule
5	Kullanılabilir katlar alanı tek katlı olan binalar hariç 800 m 2'den veya kat adedi birden fazla olan umumi binalarda en az bir adet asansör yapılması zorunludur. Ayrıca kat alanı 800 m 2 'den ve kat adedi 3'ten fazla olan umumi binalarla yüksek binalarda üçüncü fıkrada belirtilen asgari ölçülere uygun ve en az 2 adet olmak üzere binanın tipi, kullanım yoğunluğu ve ihtiyaçlarına göre belirlenecek sayıda asansör yapılması zorunludur. Bu asansörlerden en az bir tanesinin herhangi bir tehlike anında, arıza veya elektriklerin kesilmesi halinde zemin kata ulaşıp kapılarını açacak, yangına dayanıklı malzemeden yapılmış, kuyu içinde, duman sızdırmaz nitelikte, kesintisiz bir güç kaynağından beslenecek şekilde tesis edilmesi gerekmektedir.	Rule
6	10 kat ve üzeri binalarda asansörlerden en az bir tanesi yük, eşya ve sedye taşıma amacına uygun olarak dar kenarı 1.10 metre ve alanı 2.31 m^2 'den, kapı genişliği ise net 1.10 metreden az olmayacak şekilde yapılır.	Rule
7	Binalarda usulüne göre asansör yapılmış olması, bu Yönetmelikte belirtilen şekil ve ölçülerde merdiven yapılması şartını kaldırmaz.	Rule
8	Asansörün yapılması ve işletilmesi ile ilgili hususlarda; bu madde hükümleri de dikkate alınarak, 29/6/2016 tarihli ve 29757 sayılı Resmî Gazete'de yayımlanan Asansör Yönetmeliği (2014/33/AB), Binaların Yangından Korunması Hakkında Yönetmelik ve TSE standartları hükümlerine uyulur.	Rule
9	Asansörlerin, bodrum katlar dâhil tüm katlara hizmet vermesi zorunludur.	Rule
10	Asansörlere bina girişinden itibaren erişilebilirlik standartlarına uygun engelsiz erişim sağlanması zorunludur.	Rule
11	Asansörler, erişilebilirlik standartlarına uygun gerekli donanımlara sahip olmak zorundadır.	Rule
12	Özellik arz eden binalarda, binanın kat adedi, yapı inşaat alanı, kullanma şekli göz önünde tutularak asansör sayıları ile asgari ölçüleri ilgili idaresince artırılabilir.	Rule
13	Mevcut binaların merdiven boşlukları, ışıklıkları veya havalandırma bacaları, 634 sayılı Kat Mülkiyeti Kanununda belirlenen çerçevede kat maliklerinin muvafakatlerini almak ve havalandırmaya bakan banyo/helâların havalandırılması için en az 0.60x0.60 metre kesitinde yer ayırmak veya cebri havalandırma yapmak kaydı ile asansör yeri olarak değerlendirilebilir. Ancak mimari projede tadilat yapılıp Asansör Yönetmeliğine göre asansör uygulama projeleri çizilerek ilgili belediyece onaylanıp, 24/6/2015 tarihli ve 29396 sayılı Resmî Gazete'de yayımlanan Asansör İşletme, Bakım ve Periyodik Kontrol Yönetmeliği çerçevesinde tescil edilmesi gerekir.	Rule
14	Talep halinde, engellilerin bulunduğu mevcut binalarda asansör yapma zorunluluğu olmasa dahi engellilerin erişiminin sağlanabilmesi için, bina içinde uygun bir yer olmaması durumunda, ön, yan ve arka bahçe mesafeleri içinde parsel sınırına en az 1.50 m. mesafe bı-rakmak kaydıyla asgari	Rule

ID	Textual Expression of Rule	Type
	ölçülerde panoramik asansör veya ulaşılacak katın yüksekliğinin uygun olması halinde mekanik platform yapılabilir.	
Madde 27 - Bodrum Katlar		
1	Bodrum katların iskân edilebilme şartları;	
1.a	Bodrum katlar, en az bir cephesinde en düşük doğal/düzeltilmiş bahçe kotunda veya üzerinde kalmak, hiçbir noktada hizasındaki bahçe kotlarına göre 0.90 metreden fazla gömülmemek, sel, taşkın ve su baskınlarına karşı gerekli tedbirleri alınmak şartları ile konut olarak iskan edilebilir. Ancak bir cephesinde tümü ile bahçe kotlarında/üzerinde kalmakla birlikte diğer cephelerinde 0.90 metreyi aşan gömülmeler bir üst kattaki plan tertibine uyan bölümler boyunca kabul edilebilir.	Rule
1.b	Bölge kat nizamı planlı bölgelerde; iskan edilebilir bodrum katların tamamı konut olarak değerlendirilebilir.	Rule
1.c	Ticari alanlarda yapılan binaların ticari amaçla kullanılan bodrum katlarında döşemenin doğal/düzeltilmiş bahçe kotunda/en fazla 0.90 m gömülü olma şartı aranmaz. Bu tür binalarda suni havalandırmanın sağlanması ile engellilerin dolaşımına olanak sağlayan önlemler alınır.	Rule
2	Kat sayıları aşağıdaki hükümlere göre belirlenir:	
2.a	Konut bölgelerinde; binalarda zemin kat veya iskan edilebilir konut şartını sağlayan katın altında bir bodrum kat daha düzenlenebilir. Ancak istenilmesi halinde müştemilat için ilave bodrum katlar yapılabilir.	Rule
2.b	Ticaret ada/parsellerinde; birinci fıkranın (a) bendindeki iskan edilebilme şartını sağ-layan bodrum katın altında olmak üzere iki kat daha yapılabilir. Ancak istenilmesi halinde müştemilat için ilave bodrum katlar yapılabilir.	Rule
2.c	Ticaret ada/parselleri dışında; ticaret ada/parselleri dışındaki tüm kullanımlarda da imar planı/plan notlarına aykırı olmamak ve imar birimince uygun görülmek kayıtlarıyla (b) bendi hükümleri uygulanır.	Rule
2.ç	Ticari ve sanayi depolama, sanayi, organize sanayi, küçük sanayi ve konut dışı kentsel çalışma vb. alanlarda; birinci fıkranın (a) bendindeki iskan edilebilme şartını sağlayan bodrum katın altında ancak bir kat yapılabilir. Ancak istenilmesi halinde müştemilat için ilave bodrum katlar yapılabilir.	Rule
3	Bodrum katlar, aşağıda belirtilen şartlarda çeşitli işlevler için kullanılabilir:	
3.a	İskan edilebilme şartına uymak kaydıyla konut,	Rule
3.b	Her türlü zorunlu veya ihtiyari ortak yerler ve eklentiler,	Rule
3.c	Ticaret ada/parsellerinde yol cephesinden ve hizasındaki bordür kotundan girmek ve en az net alanın 1/3'ü dükkan giriş kotu ve üzerinde tertiplenmek kaydı ile dükkan,	Rule
3.ç	Köşe başı veya köşe başından başka iki yola cephesi olan parsellerde yapılacak binaların yola cephesi bulunan bodrum katlarına ticari kullanımlıbağımsız bölüm,	Rule
3.d	Ticaret ada/parsellerinde kapalı çarşı ve pasaj,	Rule

ID	Textual Expression of Rule	Type
3.e	Bölge kat nizamı planlı alanlarda yoldan, ticaret ada/parsellerinde yoldan veya komşu ticaret parseline bakan bahçelerden geçiş yapmak suretiyle ticari kullanımlar,	Rule
3.f	Konut binalarında bağımsız bölüm ve dışarı ile bağlantılı olmamak üzere depo,	Rule
3.g	Bölge kat nizamı planlı bölgelerdeki zemin/bodrum katlarında ticari kullanım bulunan binalarda ticari kullanımlara mahsus olmak üzere bu bölümle bağlantılı ve ait olduğu bölüm alanının iki mislini aşmayan depo,	Rule
3.ğ	Bu Yönetmeliğin diğer hükümlerine de uymak kaydı ile bina ana girişleri, yangın kaçışları vb.,	Rule
3.h	Otopark,	Rule
3.ı	Sanayi, depolama, küçük sanatlar ve konut dışı çalışma alanlarında amaca uygun tesisler.	Rule
4	Bodrumların tevsinde aşağıdaki esaslara uyulur.	
4.a	Konut bölgelerinde;	
4.a.1	Bölge kat nizamı planlı alanlardaki konut parsellerinde yapılacak 19 uncu maddenin birinci fıkrasının (f) bendinde sayılan umumi binalarda tabii/tesviye edilmiş bahçe kotları altında kalmak, komşu parseller için sakınca oluşturmadığı tüm komşu parselleri de kapsayacak şekilde civarı ile birlikte yapılacak incelemeye göre saptanmak kayıtları ile gerekli görülen hallerde yan ve arka bahçelerin tamamında bodrum yapılmasına, hiçbir sınıfta gayri sıhhi müessese olarak projelendirilmemek/kullanılmamak üzere ilgili belediyenin imar birimince izin verilebilir.	Rule
4.a.2	Bu şekilde tevsi edilen bodrum katın altında otopark ihtiyacı ile en az ölçülerdeki zorunlu ortak alanları karşılayacak şekilde bodrum/bodrumlar tevsi edilebilir.	Rule
4.a.3	Emsalli parsellerde ise imar planı/plan notlarında belirtilmese bile yan/arka bahçelerde tevsi işlemine ancak bu bendin (1) ve (2) numaralı alt bentlerinde belirtilen şartlarla ve zorunlu bina otoparkı yapmak amacı ile ilgili belediyesince uygun görülen şekilde izin verilebilir.	Rule
4.b	Ticaret parsellerinde bodrum katlar, arka ve yan bahçelerde imar planı kararı ile tevsi edilebilir.	Rule
Madde 28 - Bodrum Katlar		
1	Konut bölgelerinde zemin katların kullanma şekilleri;	
1.a	Bölge kat nizamı planlı konut bölgelerinde ilgili idare meclisince yol boyu ticari te-şekkül kararı alınmak kaydıyla zemin katlar, bu Yönetmelik ve ilgili diğer mevzuat hükümlerine uymak kaydı ile konut ve ticari amaçla kullanılabilir.	Rule
1.b	Diğer planlı alanlarda zemin katlar, plan hükmü veya ilgili idare meclisince yol boyu ticari teşekkül kararı olmadıkça ticari amaçla kullanılamaz.	Rule
1.c	Zemin katlarda ticari kullanımların yer alması halinde, bu kullanımların bodrum katlarında içten bağlantılı piyesleri olabilir. Bu kullanımlar binanın ortak alanları ve müştemilatıyla irtibatlandırılamaz.	Rule
2	Planda aksine bir hüküm yoksa zemin kat döşeme seviyeleri, binaların kot aldığı cephesinde ± 0.00 kotunun altına düşürülemez ve +1.20 kotunun	Rule

ID	Textual Expression of Rule	Type
	üstüne çıkarılamaz. Meyilli yollar üzerinde yapılacak dükkan ve benzeri yerlere girişlerin yaya kaldırımı kotlarına uydurulması amaçlı döşeme kademeleri ve yol cephesi esas alınmak, bina yüksekliği aşılmamak kaydı ile en fazla dükkan alanının 1/3 oranında düzenlenebilecek döşeme kademeleri bu sınırlamaların dışındadır.	
3	Zemin katlar;	
3.a	Binanın kot aldığı yol cephesi tarafında yapılmamak	Rule
3.b	Yükseltilen veya alçaltılan kısım, ± 0.00 kotuna göre 1.50 metreden daha yukarıda/aşağıda ve bina alanının, açık çıkmalar hariç 2/3'ünden fazla olmamak,	Rule
3.c	İmar durumunda belirlenen saçak seviyesini aşmamak ve bu Yönetmeliğin diğer bağ-layıcı hükümlerine uymak, kaydı ile kademelendirilebilir.	Rule
4	Ancak binanın kot aldığı yolun meyilli olması halinde bu yol üzerinde bulunan bölümler de gömülmeyi önlemek amacı ile ± 0.00 kotunu hiçbir noktada +1.50 metreden fazla aşmamak ve imar durumunda verilen saçak seviyesini ihlal etmemek kaydıyla kademelendirilebilir. Bu şekilde yapılabilecek kısımlar da yukarıda belirtilen 2/3 oranı içinde mütalâa edilir.	Rule
Madde 29 - Konutlarda bodrum ve zemin katlara ait bazı hususlar		
1	Konut binalarının bodrumlarında ve zemin katlarında hiçbir surette ticari amaçlı genel depolara veya bağımsız kullanılabilecek özel depolara ve ardiyelere müsaade edilmez.	Rule
2	Konut binalarında ana merdivenle veya ortak alanlarla irtibatlı, en az 1.00 metre genişliğinde bahçe çıkışı aranması zorunludur.	Rule
3	Binaların ticari amaçla kullanılan bodrum ve zemin katlarında, öncelikli olarak eş düzey dolaşımın sağlanmasına yönelik çözümlere gidilmesi, bunun mümkün olmadığı hallerde engellilerin dolaşımına olanak sağlayan rampa, yürüyen bant vb. düzenlemelerle önlem alınması zorunludur.	Rule
4	Su taşkın riski bulunduğu DSİ veya ASKİ tarafından belirlenen parsellerde, iskân edilen katın taban kotu ile bina, otopark gibi giriş-çıkış kotlarının, su seviyesine esas dere kret kotunun en az 1.50 m. üzerinde olması zorunludur. Tereddüde düşülen konularda DSİ veya ASKİ görüşüne göre uygulama yapılır.	Rule
Madde 30 - Asma katlar		
1	Zemin katı ticari olarak kullanılmayan konut alanları haricindeki binaların zemin katlarında asma katlar düzenlenebilir.	Rule
2	Zemin katta asma kat yapma şartları:	
2.a	Asma kat yapılan bölümün iç yüksekliği temiz 5.50 metreden, asma katın iç yüksekliği ise temiz 2.40 metreden az olamaz.	Rule
2.b	Asma kat alanı, içinde bulunduğu bağımsız bölümün, merdivenler hariç temiz alanının 2/3'ünü geçemez, 1/3'ünden az yapılamaz.	Rule
2.c	Asma katlar yola bakan cephe/cephelerde merdiveni de dâhil 3.00 metreden fazla yaklaşmamak kaydıyla tertiplenir.	Rule
Madde 31 - Tabii zeminden kotlandırılan parsellerin ön bahçesinde kapalı otoparklar yapabilme		

ID	Textual Expression of Rule	Type
1	Tabii zeminden kot alan ada/parsellerde, yapı yaklaşma sınırları dı-şında, yol ile parsel sınırı arasındaki kot farkı +2.50 metre ve üzerinde ise tabii zemin veya planla önerilen tesviye kotları altında kalmak kaydıyla bu kısımlar binaya ait kapalı otopark olarak düzenlenebilir.	Rule
Madde 32- Binalarda bulunması zorunlu odalar/bölümler ve en az ölçüleri		
1	Her bağımsız konutta en az 1 oturma odası, 1 yatak odası, 1 mutfak veya yemek pişirme yeri, 1 banyo veya yıkanma yeri, 1 tuvalet bulunması zorunludur. Ancak 3 veya daha az odalı konutlarda banyo/yıkanma yeri ile tuvalet aynı yerde düzenlenebilir. Oturma ve yemek kısmı gibi birleşik kullanma imkânı sağlayan mahaller 1 oda kabul edilir.	Rule
1.a	Konut binalarındaki tüm oda/bölümlerin iç ölçüleri engellilerin de kullanımını sağ-lamak üzere standartlara uygun olmalıdır. Konutlarda dar kenar ölçüsü ve net alanı en az: Oturma odasında 3.00 metre, 12.00 m², Yatak odasında 2.50 metre, 9.00 m², Mutfak veya yemek pişirme yerinde 1.50 metre, 4.00 m², Yıkanma yeri veya yıkanma yeri ile birlikte tuvalette 1.50 metre, 3.00 m2, Ayrı tertiplendiğinde tuvalette 1.00 metre, 1.40 m², Hol ve koridorda 1.20 metreden az olamaz.	Rule
1.b	Giriş, hol, kiler, soyunma yeri gibi yerler oda sayılmaz.	Rule
1.c	Odaların/bölümlerin düzgün geometrik şekilde düzenlenmemesi durumunda içerisinde bu fıkranın (a) bendinde zikredilen en az dar kenar ölçülerinde bir kare sığdırabilecek büyüklükte olması gerekir.	Rule
2	Dükkan ve bürolarda;	
2.a	Pasajlar ve kapalı çarşılar hariç her dükkan için 1 adet tuvalet düzenlenecektir.	Rule
2.b	16.00 m² 'nin altında ve her üç dükkana bir ortak tuvalet ve lavabo oluşturulması halinde dükkanlarda ayrı ayrı tuvalet aranmaz.	Rule
2.c	Dükkan ve büroların dar kenarı 2.00 metreden az olmamak üzere alanı 8.00 metrekareden az olamaz.	Rule
2.ç	Özellik arz eden binalarda bu Yönetmeliğin kendileri için belirtilen hükümleri geçerlidir.	Rule
3	Umumi binalarda;	
3.a	Tüm yapı, tesis ve açık alan düzenlemeleri, engellilerin de ulaşımını ve kullanımını sağlamak amacı ile TSE standartlarına uygun olarak yapılır.	Rule
3.b	Dar kenar genişliği, koridor uzunluğu 20.00 metreye kadar en az 2.00 metre, bundan sonra ise en az 2.50 metre olacaktır.	Rule
3.c	50 kişiye en az 1 kadın, 1 erkek olmak üzere standardına uygun tuvalet, pisuar ve lavabo yapılır. Binada engelliler için de en az 1 kadın, 1 erkek olmak üzere standardına uygun tuvalet ve lavabo yapılır. Özellik arz eden durumlarda söz konusu sayıları arttırma veya azaltmaya imar birimleri yetkilidir.	Rule
3.ç	İlgili mevzuatında aksine bir hüküm bulunmadığı takdirde; otel ve motellerde oda sayısının %3'ü ve en az her otelde 1 oda, hijyenik bakımı da dahil tekerlekli sandalyelerin dolaşımına cevap verecek şekilde düzenlenir.	Rule
Madde 33 - Kat yükseklikleri		
1	Uygulama imar planlarında aksine bir hüküm bulunmadığı takdirde kat yükseklikleri; bu Yönetmeliğin 11 inci maddesine göre belirlenecek saçak	Rule

ID	Textual Expression of Rule	Type
	seviyesini aş-mamak, planla veya bu Yönetmelikle belirlenen kat adedini ihlal etmemek ve bu madde hükümlerine aykırı olmamak kaydıyla serbesttir.	
2	Tesisat katının yüksekliği normal kat yüksekliğini aşamaz. Özellikli olan yapılarda bu hüküm uygulanmayabilir.	Rule
3	İskân edilen katların iç yüksekliği, asma katlar hariç 2.60 metreden, bölge kat nizamı planlı alanlarda 2.40 metreden az olamaz. Ancak hava maniası veya güvenlik nedeniyle kısıtlamaları olup planla kat adedi belirlenen parsellerde bu yükseklik 2.40 metreye düşürülebilir. Minimum kat yüksekliğinin altına düşmemek, kat adedi arttırılmamak ve en fazla bina yüksekliğini aşmamak kaydıyla kat iç yükseklikleri arttırılabilir/azaltılabilir.	Rule
4	Yıkanma yeri, banyo, duş, lavabo yeri, tuvalet, kiler, merdiven altı, her türlü iç ve dış geçitler ve iskân edilmeyen bodrum katları ile müştemilât binalarında, iç yükseklik 2.20 metreye kadar düşürülebilir.	Rule
5	Garaj, kalorifer dairesi, odunluk, kömürlük, bodrum katlarda yer alan otoparklar ve benzeri özellik arz eden yerlerin yükseklikleri bu maddede yer alan hükümlere tabi olmayıp, hizmetin gerektirdiği şekilde 2.20 metrenin altında olmamak kaydıyla idaresince tespit ve tayin olunur.	Rule
6	Eğitim, sağlık, dini tesis, sanayi yapıları ile sinema, tiyatro ve konferans salonları, katlı otoparklar, alışveriş merkezleri, düğün salonu, resmi kurum ve kuruşlara ait binalar ve spor salonları gibi özellik arz eden yapılarda iç yükseklikler, teknolojik ve mimari gereklere göre mimari estetik komisyon kararı ile belirlenir.	Rule
Madde 34 - Çatılar		
1	Çatıların, çevresindeki cadde ve sokakların karakterine uygun olması ve inşa edilecek yapının gereksinimlerine cevap vermesi şarttır. Çatı eğimi için aşağıdaki esaslar dikkate alınır:	Rule
1.a	Binalarda çatı eğimi % 40'ı geçemez.	Rule
1.b	Bu eğim binanın dar cephesinden hesaplanır. Saçaklar hesaba katılmaz. Ancak kapalı çıkma bulunan ve bu çıkma bina yüksekliğince devam eden kısımlarda, çatı eğimi saçak ucundan hesaplanır.	Rule
1.c	1 ve 2 katlı konutlarda, tek bağımsız bölümlü yapılarda veya eksik katlı yapılan binalarda çatı şekli serbest olmak kaydıyla çatı eğimi azami %50 olabilir.	Rule
1.ç	Blok ve ikili bloklarda çatı örtüsü şekli birlikte değerlendirilmelidir.	Rule
2	Çatı aralarının kullanımına dair hükümler aşağıdaki gibidir:	
2.a	Çatı aralarında bağımsız bölüm yapılamaz. Bu kısımlarda ancak en az ölçülerdeki su deposu, asansör kulesi, iklimlendirme ve kaskat sistemleri de içerebilen tesisat odası vb. ile son kattaki bağımsız bölüm ile kendi içinden irtibatlandırılmak, ait olduğu bağımsız bölüm sınırlarını aşmamak koşulu ile piyesler yapılabilir.	Rule
2.a.1	Çatıların yukarıdaki şekilde düzenlenmesi durumunda bu kattaki bölüm önleri teras olarak kullanılabilir.	Rule

ID	Textual Expression of Rule	Type
2.a.2	Mimari proje ile önerilen saçak seviyesi, imar planı ile belirlenen seviyenin altında olması halinde, parapet yapılarak çatı bu parapetin üzerine oturtulabilir.	Rule
2.a.3	Çatı arasındaki mekânlarda, çatı eğimi içerisinde kalmak ve fonksiyonunu sağlamak şartıyla asgari yükseklik şartı aranmaz. Ancak üst kat tavan döşemesi ile çatı örtüsü arasında kalan hacimler, uygulama imar planında aksine bir hüküm yoksa ilave kat döşemeleri yapılmak suretiyle bölünemezler.	Rule
2.b	Tamamı umumi, ticari ve sanayi amaçlı binalarda teras çatılar, 22/2/2018 tarihli ve 30340 sayılı Resmî Gazete'de yayımlanan Otopark Yönetmeliği hükümlerine aykırı olmamak kaydı ile binaya ait açık otopark olarak kullanılabilir.	Rule
2.c	Son kat tavan döşemesi en yüksek mahya kotunu aşmayacak şekilde eğimli olarak tertip edilebilir. Üst kat tavan döşemesi ile çatı örtüsü arasında kalan hacimler ek kat döşemeleri yapılmak sureti ile kat oluşturacak şekilde bölünemez.	Rule
2.ç	Teras çatılarda çatı bahçesi olarak düzenleme yapılabilir. Bahçe düzenlemesi yapı-labilmesi için gerekli olan 0.50 m. toprak dolgu, parapet yüksekliğine dâhil edilmez.	Rule
2.ç.1	Ortak alan olarak kullanılan teras çatılarda; bahçe düzenlemesi yapılması halinde merdiven evi yanında, bina sakinleri tarafından kullanılmak üzere, tuvalet, lavabo, çay ocağı, bahçe düzenlemesinde kullanılacak malzemeleri depolamak için merdiven evine bitişik, toplam teras alanının %10'unu geçmeyen ve en fazla 3 metre yüksekliğinde ve 20 m² alanında kapalı mekân oluşturulabilir. Kapalı mekan bina ön cephesine 3 metreden fazla yaklaşamaz.	Rule
2.ç.2	Ayrıca rezidans, otel, apart otel gibi konaklama tesislerinin teras çatılarında bina cephelerine 3 metreden fazla yaklaşmamak, en fazla 1.50 m. derinliğinde olmak ve parapet kotunu aşmamak koşuluyla açık havuz yapılabilir.	Rule
3	Çatı dışına taşmaya dair hükümler aşağıdaki gibidir:	
3.a	Merdiven evleri, ışıklıklar, hava bacaları, alın ve kalkan duvarları çatı örtüsünü en fazla 0.60 metre aşabilir.	Rule
3.b	Ayrıca zorunlu olan tesisatla ilgili hacimlerin, çatı örtüsüne paralel olarak düzenlenmek kaydıyla güneşle su ısıtıcılarının ve çatı pencerelerinin çatı örtüsünü aşmasına ilgili belediyenin imar birimince teknik gereklere göre uygun görülecek ölçü ve şekilde izin verilebilir.	Rule
3.c	TSE standartlarına göre projelendirilecek asansör kulelerinin en az ölçülerdeki bölümlerinin de çatı örtüsünü aşmasına izin verilir.	Rule
3.ç	Bu fıkranın (a), (b) ve (c) bentlerinde belirtilenler dışında bu maddenin beşinci fıkrasına göre belirlenecek çatı örtüsü düzlemleri dışına taşılamaz.	Rule
4	Binalarda çatı arasına/teras çatıya çıkmak için en az 0.80x0.80 metre ebatlarında kapaklı bir boşluk bulundurulur veya genel merdiven gerekli önlemleri almak koşulu ile çatıya kadar devam ettirilebilir.	Rule
5	En yüksek mahya kotuna dair hükümler aşağıdaki gibidir:	Rule

ID	Textual Expression of Rule	Type
5.a	En yüksek mahya kotu, imar durumunda verilen saçak üst kotuna çatı yüksekliği eklenerek bulunur.	Rule
5.b	En fazla çatı yüksekliği; ayrık binalarda kırma, ikili blok binalarda bloğu ile müşterek kırma, iki taraftan da bitişik binalarda ise ön ve arka cepheye akıntılı beşik çatı kurulacağı varsayılarak hesap edilir.	Rule
6	Tescilli kültür varlığı binalar, anıtsal yapılar, dini yapılar, mimari eleman niteliğindeki kuleler, elçilikler, spor tesisleri, konsolosluklar, plan ile korunan vb. binaların çatıları bu madde hükümlerine tabi değildir.	Rule
Madde 35 - Çıkmalar		
1	Binalarda ada/parsel sınırını aşmamak kaydıyla en fazla taban alanı/yapı yaklaşma sınırları dışında;	
1.a	Yol cephelerinde;	
1.a.1	1.50 metreden geniş olmamak,	Rule
1.a.2	Bölge kat nizamı planlı alanlar dışında; 5 metreye kadar olan ön bahçelerde, parselin yol sınırına 3.50 metreden fazla yaklaşmamak,	Rule
1.a.3	Cephe boyunca bordür taşı üst seviyesi ile çıkma altı arasındaki en yakın düşey mesafe hiçbir noktada 2.30 metreden aşağı düşmemek, şartlarıyla açık ve kapalı çıkma yapılabilir.	Rule
1.b	Arka cephelerinde;	
1.b.1	1.50 metreden geniş olmamak,	Rule
1.b.2	Çıkmanın en yakın noktası arka komşu parsel sınırına 5 metreden fazla yaklaşmamak,	Rule
1.b.3	Tabii veya tesviye edilmiş zemin ile çıkma altı arasındaki en yakın düşey mesafe hiçbir noktada 2.20 metreden aşağı düşmemek, şartlarıyla açık veya kapalı çıkma yapılabilir.	Rule
1.c	Yan cephelerinde;	
1.c.1	1 metreden geniş olmamak,	Rule
1.c.2	Çıkmanın en yakın noktası yan komşu parsel sınırına 2 metreden fazla yaklaşmamak,	Rule
1.c.3	Tabii veya tesviye edilmiş zemin ile çıkma altı arasındaki en yakın düşey mesafe hiçbir noktada 2.20 metreden aşağı düşmemek, şartlarıyla yalnız açık çıkma yapılabilir.	Rule
1.ç	Yeşil veya iskan harici (yerleşme dışı) alanlara bakan cephelerinde;	
1.ç.1	1.50 metreden geniş olmamak,	Rule
1.ç.2	Tabii veya tesviye edilmiş zeminle çıkma altı arasında en yakın düşey mesafe hiçbir noktada 2.20 metreden aşağı düşmemek,	Rule
1.ç.3	Çıkmanın en yakın noktası, yeşil ve iskan harici alan sınırına 3.50 metreden fazla yaklaşmamak, şartları ile kapalı çıkma,	Rule
1.ç.4	Çıkmanın en yakın noktası, yeşil ve iskan harici alan sınırına 2 metreden fazla yaklaşmamak, şartları ile açık çıkma yapılabilir.	Rule
2	Çıkmalara ait sınırlamalar ve istisnalar;	
2.a	Bu maddenin birinci fıkrasında belirtilen hükümler dahilinde binaların bitişik olmayan cephelerinde yapılacak çıkmalar, cepheler boyunca devam edebilir.	Rule

ID	Textual Expression of Rule	Type
	Bitişik olan cephelerde çıkmalar bitişik sınıra en fazla 2 metreye kadar yaklaşabilir. Ayrıca bitişik nizam parsellerden birinde bina inşa edilmiş ve çıkmalar ortak sınıra kadar yanaştırılmış ise aynı şartlarda olmak üzere komşu parselde inşa edilecek binanın çıkmaları da bu sınıra yanaştırılabilir.	
2.b	Bina arka cephelerinde bu Yönetmelikten önce yürürlükte olan Yönetmelik hükümleri dahilinde yapı ruhsatı alınarak bitişik sınıra kadar çıkma yapılmış olan ikili veya blok binalarda, bu maddenin birinci fıkrasının (b) bendindeki hükümlere bağlı kalınmaksızın ilgili belediye imar biriminin etüdüne göre açık veya kapalı çıkma yapılmasına izin verilebilir.	Rule
2.c	Bina taban alanının tümü kullanılmadan yapılan binalarda bu maddede belirlenen diğer hükümlere uyulmak şartı ile en fazla taban alanı ve yapı yaklaşma sınırları içinde kalan yerlerden yararlanılarak çıkma genişliklerinin artırılması mümkündür.	Rule
2.ç	Bahçenin tümü yol eğiminde tesviye edilen, arka yoldan yüz alan arka yol cephelerinde, bu maddenin birinci fıkrasının (a) bendindeki hükümleri uygulanır.	Rule
2.d	Bitişik ve blok nizama tabi yerlerde, ilgili belediyece sakınca görülmediği takdirde, çıkmaların yan komşu sınıra kadar yaklaştırılmalarına izin verilebilir. Ancak komşu ortak kı-sımlar duvarla kapatılır.	Rule
2.e	İlgili belediyeler, binaların cephelerine konulmak istenen klima yerlerinin tespitine ve son kattan sonra düzenlenen tesisat alanlarının da binanın mimarisine uygun bir şekilde kapatılmasına yetkilidirler.	Rule
2.f	Ayrık nizam emsalli parsellerde; zemin katta bina taban alanına katılmayan boşluklar oluşturulduğunda, normal katlarda bu boşluklar üzerinde yapılacak çıkmalar arasındaki uzaklık 3 metreden az olamaz.	Rule
3	Aşağıda yapılanma şartları belirtilen bina çıkma ve çıkıntıları, çıkmalar ile ilgili hükümlere bağlı değildir.	
3.a	Döşemesi tabii/tesviye edilmiş zemin kat seviyesinde olan kat da dahil kendi parsel sınırını taşmamak, kapalı bir hacim oluşturarak inşaat alanını büyütecek nitelikte olmamak üzere; cepheden itibaren genişliği en fazla 0.40 metre olmak kaydı ile binaların tüm katlarının cephelerinde motif çıkmalar ve denizlikler yapılabilir.	Rule
3.b	Parsel sınırları içinde veya dışında, çıkma yapılamayan yerlerde bina cephe hattından itibaren, kapalı çıkma tertip olunabilen yerlerde kapalı çıkmalara ilaveten, kapalı bir hacim oluşturarak inşaat alanını büyütecek nitelikte olmamak üzere genişliği 0.40 metreyi aşmayan güneş kırıcıları yapılabilir. Açık çıkma yapılabilen yerlerde güneş kırıcıları çıkma ucunu 0.20 metreden fazla aşmamak kaydı ile tertip edilebilir.	Rule
3.c	Yola bakmayan cephelerde giriş köprüleri;	
3.c.1	Binalara girişin/çıkışın köprü şeklinde yola bağlanması zorunluluk olduğu takdirde, köprü genişliği 1.50 metre olacaktır. Bu köprülerin bina yan cephesine bitişik ve konsol olarak inşaları zorunludur.	Rule
3.c.2	Köprülerin altı hiçbir surette kapalı hacim yapılarak kullanılamaz.	Rule
3.c.3	Tabii zeminden kotlandırılan parsellerde köprülü giriş yapılamaz. Giriş tabii zemine uyumlu olarak düzenlenecek merdiven ve rampalarla sağlanır.	Rule

ID	Textual Expression of Rule	Type
	Eğimden kaynaklı zorunluluk halinde ilgili belediyesince giriş köprüsüne izin verilebilir.	
3.c.4	Tabii zeminden kotlandırılan parsellerde, engellilerin bodrumdan/zemin katlardan bina girişini sağlamak üzere komşu parsellere zarar vermeyecek şekilde en az rampa ölçülerinde tesviye yapılabilir.	Rule
3.c.5	Binalara girişi sağlayan köprü ve giriş şeridi ile yaya kaldırımı ve bina girişi arasında kot farkı olduğu durumlarda veya binalara girişin merdivenle sağlanmasının zorunlu olduğu hallerde, merdivenlerin yanı sıra, engellilerce de kullanılmak amacıyla, kaymayı önleyen dö-şeme kaplamalı, tekerlekli sandalye ve koltuk değneği hareketlerini güçleştirmeyen standardına uygun rampa yapılması zorunludur.	Rule
Madde 36 - Saçaklar		
1	Parsel sınırları içinde kalmak kaydıyla blok/bitişik nizam binalarda saçak genişliği 1 metreyi, ayrık nizam binalarda ise 1.50 metreyi aşamaz.	Rule
2	Binaların saçaksız olması veya saçak genişliği hakkında çevresindeki ve bitişiğindeki binalar ile mimari ahenk temin edecek şekilde karar almaya ilgili belediyenin imar birimi yetkilidir.	Rule
3	Tescilli binalar ile sefaret, cami vb. binaların saçak ölçü ve nitelikleri bu madde hükümlerine bağlı değildir.	Rule
4	Mevcut saçaklı binalara ek kat yapıldığı takdirde, bu saçaklar bu Yönetmeliğin çıkmalarla ilgili hükümlerine göre düzeltilmedikçe yapı kullanma izni verilmez.	Rule
5	Bina cephesinden itibaren uzunluğu parsel sınırını, genişliği 2.50 metreyi geçmemek kaydıyla en alçak noktası tretuar kotundan en az 2.50 metre yükseklikte giriş saçakları yapılabilir.	Rule
Madde 37 - Işıklık, havalık ve hava bacaları		
1	Işıklık ve hava bacalarından yararlanma esasları aşağıdaki gibidir:	
1.a	Konutlarda;	
1.a.1	Oturma ve yatak odaları ile mutfak, yıkanma yeri ve merdiven evinin doğrudan dı-şarıdan ışık ve hava almaları genel esastır.	Rule
1.a.2	Her bağımsız bölümde en az 1 oturma odası, 1 yatak odası ve mutfak doğrudan dış cepheden hava ve ışık almak zorundadır. Diğer odalar ile merdiven boşluğunun ışıklıklardan; yıkanma yeri, tuvaletler ile yatak odası ve mutfak nişlerinin nitelik ve ölçüler açısından işlevine uygun ışıklık, havalık ve hava bacalarından faydalanmaları mümkündür.	Rule
1.b	Umumi binalarda;	
1.b.1	Umumi binalarda doğalgaz ısıtma cihazı konan oda dışındaki her türlü mahallin doğ-rudan ve dışarıdan ışık ve hava alması esası aranmaz.	Rule
2	En az ışıklık ölçüleri;	
2.a	Binalarda ışık ve havayı ışıklıktan sağlayan odaların olması halinde, yararlanan kat sayısına göre gereken ışıklık en az ölçüleri aşağıdaki gibi olacaktır. Bu ölçüler emsalli parsellerde en fazla %20 arttırılabilir. Yararlanan Kat Sayısı Dar Kenarı Alan (En çok) (En az, metre) (En az, metrekare) 2 1.00 3.00	Rule

ID	Textual Expression of Rule	Type
	3 ila 4 1.50 4.50 5 ila 7 1.50 6.00 8 ve daha yukarı 2.00 9.00	
2.b	Cephelerle irtibatlı olarak düzenlenen ışıklıklar en az alan kaydına bağlı olmayıp yalnız en az genişlik şartı aranır.	Rule
2.c	Yararlanılan kat sayısından; ışıklıktan yararlanan en alt kattan itibaren üstündeki (asma kat da dahil) pencere açılsın açılmasın tüm katların toplamıanlaşılmalıdır.	Rule
3	En az havalık bacası ölçüleri aşağıdaki gibidir.	
3.a	Işıklığa bakmayan helâ ve banyolar asgari 0.60 x 0.60 metre kesitinde bir havalığa veya şartları bu maddenin dokuzuncu fıkrasının (a) bendinde belirtilen hava bacalarına bağlanabilir.	Rule
3.b	Bütün binalarda helâ ve banyoların, suni ışıklandırma ve havalandırma tesisatından yararlanmak suretiyle ışıklık, havalık veya hava bacasına bağlantı yapılmadan düzenlenmeleri mümkündür. Ancak, bu gibi tesisata gerekli yerler ayrılmış olmalı ve bu tesisata ait proje mimari projeyle birlikte onanarak uygulanmalıdır.	Rule
4	Ortak merdiven ışıklıkları;	
4.a	İskan edilebilir bodrum katlar dahil, üç katlı binalarda merdivenler için merdiven boşluğunun üzeri tamamı açılmak kaydı ile çatıdan ışık temin edilebilir. İskan edilmeyen bodrum katlar hariç üç kattan fazla binalarda, ışıklığa başka oda/bölüm açılmayıp sadece merdiven penceresi açıldığı takdirde, ışıklığın dar kenarı 1 metreden az olmamak şartı ile alanı en az 2.50 metrekare olabilir.	Rule
4.b	Ayrıca asansör boşluğunu merdiven kovasında yapmamak, merdiven boşluğunun tamamı cam sathı ile aydınlatılmak şartı ile dar kenarı 0.50 metre ve alanı 2 metrekareden az olmayan merdiven kovası olan merdivenlerin ışığı merdiven boşluğuüzerinden verilebilir.	Rule
4.c	Bir ailenin oturduğu bağımsız bölüm içindeki merdivenlerde ışıklık zorunluluğu yoktur.	Rule
5	Işıklık, havalık ve hava bacalarında aranan nitelikler;	
5.a	Işıklık, havalık ve hava bacalarının üzeri havalandırmayı önleyecek şekilde veya aydınlığı kesici bir malzeme ile örtülemez. Işıklık, havalık ve hava bacalarının en az alanı içinde merdiven, asansör, balkon, baca gibi şeyler yapılamaz. Işıklıkların içerisine açık renk boya veya badana yapılması zorunludur.	Rule
5.b	Işıklığa başladığı kattan giriş temini şarttır.	Rule
6	Işıklık, havalık ve hava bacalarından faydalanma şartları;	
6.a	Bir havalıktan her katta en çok dört oda/bölüm faydalanabilir. Ancak, sandık odası, yatak holü, giriş gibi ışık ve hava alması zorunlu olmayan veya gerekli ışık ve havayı esasen bu Yönetmelikte tarif edilen şekilde alan bölümlerden herhangi birinden ışıklık ve havalığa fazladan pencere açılması bu ışıklık ve havalığın ölçülerinin arttırılmasını gerektirmez.	Rule
6.b	Kapalı merdiven ışıklığına bir havalıktan yararlanabilecek sayıda veya daha az helâ ve banyo penceresi açıldığı takdirde, ışıklık alanı en az 0.36 metrekare arttırılır.	Rule

ID	Textual Expression of Rule	Type
6.c	Işıklık ve havalıklar bunlara ihtiyaç olan kattan; havalıklar, istenilen en az iç yükseklikler ve pencere imkânı sağlanmak şartı ile tavandan da başlayabilir.	Rule
7	Mevcut binalara kat ilavesinde ışıklık ölçüleri; bu Yönetmeliğin yürürlüğe girmesinden önce yürürlükteki mevzuata uygun olarak inşa edilmiş yapılara bu Yönetmelik hükümlerine göre en çok iki kat ilavesi mümkün olan hallerde mevcut ışıklık ölçüleri ilave katlar için de aynen uygulanabilir.	Rule
8	Bitişik, blok ve ikili blok binalarda ışıklık;	
8.a	Bitişik, blok ve ikili blok binalar, ışıklıklar karşı karşıya getirilmek ve müşterek proje tasdik ettirilmek şartıyla bir bina gibi değerlendirilir. Ancak bunların aralarına çekilecek duvar, yararlanılan en alt katın tavan seviyesini aşamaz. Ayrıca, müşterek düzenlenen ışıklık komşu parsel sınırına 1 metreden daha az yaklaşamaz.	Rule
8.b	Binaların bitişik olması gereken komşu tarafında boydan boya ve ön veya arka cepheler ile irtibatlı ışıklık yapıldığı takdirde, civarın yapı nizamına aykırı bir görünüş meydana getirmemek için ışıklığın önündeki sokak cephesi ilgili belediyenin imar birimince uygun görülecek şekilde bina ile uyumlu bir cephe elemanı ile kapatılabilir.	Rule
9	Hava bacaları;	
9.a	İmar durumuna göre kat sayısı en fazla 2 ve saçak seviyesi 7.50 metre olan binalarda en az ölçüleri 0.45 x 0.45 metre olan hava bacaları yapılabilir. Bu tür hava bacalarından her katta en fazla iki bölüm yararlanabilir.	Rule
9.b	İmar durumuna göre kat sayısı ne olursa olsun dar kenarı en az 0.60 metre olan hava bacaları yapılabilir. Bunlardan yalnızca, her katta en fazla dört olmak üzere, doğalgaz ısıtma cihazı konulmamak kaydıyla, doğrudan hava almayan mutfak nişleri yararlanabilir.	Rule
Madde 38 - Çöp Bacaları		
1	Binalarda çöp bacası yapma zorunluluğu yoktur. İstenildiğinde çöp bacaları yapılabilir. En az ölçüleri zedelememek kaydı ile aydınlıklar içerisinde de düzenlenebilecek çöp bacalarının zemin veya bodrum katlarındaki çöp toplama yerlerinin bağımsız bağ-lantılarının bulunması ve kapak iç kısımlarının, hiçbir maddenin sızmasına olanak vermeyecek şekilde yapılması zorunludur.	Rule
2	Çöp bacalarının projelendirme ve düzenlenmesinde TS 2166'ya uyulacaktır.	Rule
Madde 39 - Kuranglezler		
1	Parsel sınırı içinde kalmak ve binaya bitişik olmak şartıyla binaların tabii zemin altında kalan bölümlerine doğal ışık ve havalandırma sağlamak amacıyla en az 0.80 m en fazla 1.20 m genişlikte, boyu yapıldığı pencere genişliğini 0.50 m fazla geçmeyen, derinliği azami 2.00 m olan, kuranglez yapılabilir.	Rule
2	Bina etrafında mütemadi kuranglez tesis edilemez. Kuranglezlerden giriş çıkış yapılamaz. Ancak, yol cephesinde bulunmayan kuranglezlerden kaçış amacıyla çıkış tertiplenebilir. Kuranglezler, havalık ve en az alan şartını sağlamak kaydıyla konutlarda helâ ve yıkanma yeri, ticari amaçlı	Rule

ID	Textual Expression of Rule	Type
	kullanışlarda ise bunlara ilaveten mutfağın havalandırılması için kullanılabilir.	
3	Sanayi bölgelerinde bodrum katların imalathane olarak tertiplenmesi halinde, yol cephesi dışındaki cephelerde kuranglez genişlikleri 1.20 m olarak düzenlenebilir.	Rule
Madde 40- Duman Bacaları		
1	Baca yapma zorunluluğu;	
1.a	Merkezi ve ferdi ısıtmalı binaların konut olarak kullanılan bağımsız bölümlerinde mutfak ve banyo dışında kalan en az bir odasında/bölümünde ve konut dışı amaçlı düzenlenen her bağımsız bölümünde ve umumi binaların ise her katında en az bir baca yapılması zorunludur.	Rule
1.a.1	Ancak, binaların konut olarak kullanılan bağımsız bölümlerinde; banyoya, şofben, kombi veya merkezi sistemle sıcak su sağlanması halinde burada sıcak su sistemine ait ayrı bir bağımsız baca yapılmayabilir.	Rule
1.a.2	Doğalgazla ısıtılan konut binalarının her bağımsız bölümünde, mutfakta yapılan baca haricinde bir mekânda en az bir baca yapılması zorunludur.	Rule
1.b	Sobalı binaların konut olarak kullanılan bağımsız bölümlerinde mutfak ve banyo dı-şında en az 2, konut dışı amaçlı kullanılan bağımsız bölümlerde ise en az 1'er adet duman bacası yapılması zorunludur.	Rule
1.c	10 kat ve üzeri yükseklikteki merkezi ısıtmalı binalar için duman bacası yapılması isteğe bağlıdır.	Rule
2	Bacalarda uyulması zorunlu hükümler aşağıdaki gibidir:	
2.a	Adi bacalar: Tek kolon halinde zeminden çatıya kadar yükselen, birden fazla birimin kullanabileceği şekilde tasarlanmış bacalara adi baca denir. Bu tip bacalara doğalgaz cihazları bağlanmaz.	Rule
2.b	Ortak (Şönt) bacalar: Zeminden çatıya kadar yükselen ana baca ve buna bağlanan her birime ait branşmanlardan meydana gelen bacaya ortak (şönt) baca denir. Bu tip bacalara doğalgaz cihazları bağlanmaz.	Rule
2.c	Müstakil (Ferdi) bacalar: Tek kolon halinde hitap edeceği birimden çatıya kadar yükselen ve sadece bir birimin kullanımına göre tasarlanmış bacalara müstakil baca denir. Bacalı cihazlar, sadece müstakil bacalara bağlanabilir. Asgari etkili baca yüksekliği 4 metre olmalıdır. Hızlandırma parçasının, 1 metre ve üstünde olabildiği durumlarda bu mesafenin 1.5 katına eşit bir etkili yükseklik yeterlidir. Atık gaz boruları başka kat hacimleri içerisinden ve başka oturma mahalleri içerisinden geçirilmemelidir. Bacalar ısı, yoğuşma ve yanma ürünlerinden etkilenmeyecek malzemeden ilgili standartlara (TS EN 1856-1, TS EN 1856-2, TS EN 1447, TS EN 13063-1, TS EN 13063-2 veya TS EN 14471) uygunluk belgesine sahip malzemeden imal edilmelidir. Yoğuşmalı tip doğalgaz yakıcı cihazlara ait bacalar, ilgili standartlara uygun olmalıdır.	Rule
2.ç	Kat sayısına bakılmaksızın mutfaklarda kokuların atılması amacıyla yapılacak bacalar şönt olabilir.	Rule
2.d	Aspiratörler gaz ve duman bacalarına bağlanamaz.	Rule
2.e	Baca detayları mimari proje ve betonarme kalıp planları üzerine ölçüleri ve uygun ölçekte detayları ile birlikte işlenecektir.	Rule

ID	Textual Expression of Rule	Type
2.f	İlgili doğalgaz idaresinden onaylı doğalgaz projeleri ile mimari/statik projelerinin uygunluğu denetlenmeden yapı izni verilemez.	Rule
2.g	Şofben, kombi vb. doğalgaz cihazları hayati tehlike arz edecek şekilde yerleştirilemez ve banyo, yatak odaları ile hacmi 8 metreküpten az olan yerlerekonulamaz.	Rule
2.ğ	Doğalgaz cihazlarının bulunduğu yerlerin dış cepheden havalandırılması zorunludur.	Rule
2.h	Bacaların ölçülendirilmesinde, projelendirilmesinde ve yapımında TSE standartları, 5/12/2008 tarihli ve 27075 sayılı Resmî Gazete'de yayımlanan Binalarda Enerji Performansı Yönetmeliği, Binaların Yangından Korunması Hakkında Yönetmelik, 18/9/2002 tarihli ve 24880 sayılı Resmî Gazete'de yayımlanan Doğal Gaz Piyasası İç Tesisat Yönetmeliği ve ilgili kamu kuruluşlarının şartname hükümlerine uyulması zorunludur.	Rule
2.ı	Bacaların üzerine temizlenmesine engel teşkil etmeyecek şekilde sökülebilir ve standardına uygun şapka takılacaktır.	Rule
2.i	Müstakil doğalgaz bacaları döşemeden başlamalı, dip kısımlarında temizlik kapağı bulunmalı ve bu kapaklar tam sızdırmaz olmalıdır.	Rule
2.j	Her bir şönt baca koluna sadece bir cihaz bağlanabilir.	Rule
3	Baca ölçü ve nitelikleri;	
3.a	Bacalar gerek yangına karşı korunmuşluk, gerek çekiş bakımından teknik ve fen kurallarına uygun şekilde yapılacaktır.	Rule
3.b	İki baca birbirine bağlanamayacağı gibi her ateş kaynağının ayrı bir bacası olacaktır.	Rule
3.c	Duman bacalarının dış duvarı en az 0.19 metre iç bölmeleri en az 0.135 metre olacaktır.	Rule
3.ç	Duman bacaları çatı örtüsünü en az 1 metre, mahyayı 0.80 metre aşacaktır.	Rule
3.d	Çatı konstrüksiyonu ister ahşap, ister çelik olsun, bacalara en çok 0.05 metre yakla-şacak şekilde projelendirilecektir.	Rule
3.e	Baca geçen mekanlarda, baca ile temas edecek şekilde ahşap kaplama ve dolap yapılacak ise aralarında gerekli yalıtım sağlanacaktır.	Rule
3.f	Bitişik, blok ve ikili blok binalarda aşağıda kalan ve komşu binaya 6 metreden yakın bacaların, yüksek olan binanın mahyasından 0.80 metre yukarı çıkarılması zorunludur.	Rule
3.g	Etkili baca yüksekliği en az 4 metredir.	Rule
3.ğ	Baca kesiti dairesel veya kare kesit seçilecek, dikdörtgen kesitli ise küçük kenarın büyük kenara oranı 2/3'ten büyük olacaktır.	Rule
3.h	Duman bacaları 300oC, kazan baca duvarları 500oC sıcaklığa dayanıklı malzemeden yapılmalı, baca duvarlarında delikli tuğla ve briket kullanılmamalıdır.	Rule
3.ı	Bütün doğalgaz cihazlarının bağlandığı bacaların, ısıtılmayan dış şartlara maruz duvarlarında yoğuşmaya karşı ısı yalıtımı yapılacaktır.	Rule
3.i	Tüm bacalar paslanmaz çelikten veya yangına, aşınmaya ve sızdırmaya karşı dayanıklılığı kanıtlanmış malzemeden yapılacak; yangına, özellikle çatıda ısıya karşı yalıtılacak ve tuğla veya benzeri malzeme ile koruma altına alınacaktır.	Rule

ID	Textual Expression of Rule	Type
3.j	Şönt bacanın kol boyu en az 1.20 metre olmalı, iki şönt baca bir ana bacaya bağlanmamalıdır. Şömine bacaları mutlaka müstakil olacaktır.	Rule
Madde 41 - Korkuluk ve Parapetler		
1	Korkuluk ve parapet yapılacak yerler;	
1.a	Her türlü yapıda, balkon ve terasların etrafında, 5'ten fazla basamaklı açık merdivenlerde veya bu durumdaki rampalarda, istinat duvarları üzerinde, her ne sebeple olursa olsun bu Yönetmeliğin yürürlüğe girmesinden önce ön bahçeleri yaya kaldırımı seviyesine göre 0.50 metreden daha aşağıda teşekkül etmiş parsellerde yol kenarına bahçe duvarı yapılmadığı takdirde korkuluk/parapet yapılması zorunludur.	Rule
1.b	Denizlik yüksekliği 0.90 metreden az olan pencerelerde bu yükseklikte fen ve teknik gereklere uygun korkuluk yapılacak veya doğramada sabit bir kayıt geçirilecektir.	Rule
1.c	İstenildiğinde binaların saçak seviyesi üzerine bina yüksekliğine dahil olmayan parapetler yapılabilir.	Rule
1.ç	Denge bacalı doğalgaz sobalarının monte edildiği duvar bölümlerinde denizlik yüksekliği 0.90 metreden az olamaz. Doğalgaz baca çıkışı olan pencere parapetlerinin önüne isabet eden açık çıkmaların ön ve yan cephelerinde duvar veya parapet olmayacak, bunların etrafı hava hareketini engellemeyecek şekilde korkulukla çevrilecektir.	Rule
2	Korkuluk veya parapet yükseklikleri ve nitelikleri;	
2.a	Korkulukların/parapetlerin en az yüksekliği; döşeme, basamak veya istinat duvarının tuttuğu toprak seviyesinden itibaren 1.20 metredir.	Rule
2.b	Çatı parapetinin yüksekliği 1.20 metreyi aşamaz. Çatı altında yapılan kalkan ve alın duvarları bu kayda tabi değildir.	Rule
2.c	Yığma duvar malzemesi ile yapılan parapet/korkulukların duvar yüksekliği 0.60 metreden fazla olamaz.	Rule
2.ç	Korkuluklar, kırılmaz veya kırıldığında dağılmayan malzemeden ve insan çarpması dâhil, tasarım yüklerini karşılayacak taşıyıcı malzeme ve montaj sistemleri ile yapılır.	Rule
2.d	Korkuluklar düşme, kayma, yuvarlanma gibi sebeplerle insanların özellikle çocukların can güvenliğini tehlikeye atacak boşluklar içermeyecek şekilde düzenlenir. Boşluklarda, yük altındaki deformasyonlar da dâhil, en fazla 0.10 metre çapında geçişe izin verilir.	Rule
Madde 42 - Kapı ve Pencereler		
1	Bütün yapılarda;	
1.a	Kapı yükseklikleri 2.10 metreden,	Rule
1.b	Kapı net (temiz) genişlikleri bina giriş kapılarında 1.50 metreden, kapıların çift kanatlı olması halinde bir kanat 1.00 metreden,	Rule
1.c	Bağımsız bölüm giriş kapılarında 1.00 metreden, diğer mahallerin kapılarında 0.90 metreden, az olamaz. Balkon ve tuvalet kapıları 0.80 metreye düşürülebilir.	Rule
1.ç	Döner kapılar, belirtilen ölçülerde yapılacak normal kapıların yanında ilave olarak bulunabilir.	Rule

ID	Textual Expression of Rule	Type
2	Kapılarda eşik yapılamaz. Eşik yapılması zorunlu hallerde engellilerin hareketini, yangın çıkışlarını ve benzeri eylemleri engellemeyecek önlemleralınır.	Rule
3	Pencerelerde, Binalarda Enerji Performansı Yönetmeliğine ve TSE standartlarına uyulması zorunludur.	Rule
4	Bitişik ve blok nizama tabi binalarda komşu parsel sınırı üzerindeki bitişik duvarlarda pencere ve kapı açılamaz.	Rule
5	Asansörlü eşya taşımacılığı için, 3 kattan fazla 10 kattan az katlı binalarda her bir bağımsız bölümün en az bir balkonunun kapısının eni net 0.90 metreden düşük olamaz.	Rule
6	Bağımsız bölümün tuvalet, banyo ve benzeri ıslak hacimlerinde mekanik havalandırma yapılmadığı takdirde yapılacak havalandırma penceresinin ölçüsü net 0.30 metre x 0.30 metreden az olamaz.	Rule
7	Atriumlu, galeri boşluklu veya iç bahçeli tasarlanan binalarda, bu mekânlara bakan pencere veya camekânların camlarının kırıldığında dağılmayan özellikli olması zorunludur.	Rule
8	Bodrum katlardaki mekânların gün ışığından faydalandırılması ve havalandırılması amacı ile yapılan pencerelerde sel, taşkın ve su baskınlarına karşı tedbirlerin alınmış olması ve bunların zemin seviyesinden en az 0.10 metre yukarıdan başlaması zorunludur.	Rule
Madde 43 - Bahçe Duvarları		
1	Bahçe duvarları yapılabilen parsellerde ön bahçe duvarları;	
1.a	Ön bahçe duvarlarının yol üstüne isabet eden kısımlarının yükseklikleri harpuşta dahil 0.50 metreyi geçemez. Bu duvarların üzerine ayrıca görüşü kapatmayacak, can güvenliğini tehdit etmeyecek şekilde 1 metre yüksekliğinde parmaklık veya şeffaf korkuluklar yapılabilir.	Rule
1.b	Eğimli yerlerde ön bahçe duvarlarının yüksekliği yaya kaldırımından 1.50 metreyi aşmamak üzere uygun görülecek şekilde kademelendirilebilir.	Rule
1.c	Bahçe duvarlarının yola bakan ön, yan ve arka bahçelerdeki diğer bölümleri de yol üstüne isabet eden kısım gibi devam eder.	Rule
2	Yan ve arka bahçe duvarları;	
2.a	Yan ve arka bahçe duvarlarının tabii veya tesviye edilmiş zeminden yüksekliği 1 metreyi geçemez. Ayrıca bu duvarların üzerine görüşe engel olmayacak şekilde 1 metre yüksekliğinde parmaklık veya şeffaf korkuluklar yapılabilir.	Rule
3	Ön bahçe duvarı yapılması yasak / sınırlı parseller;	
3.a	Yaya bölgelerindeki parsellerin yollara bakan bahçelerinde hiçbir suretle bahçe duvarı yapılmaz.	Rule
3.b	Zemin veya bodrum katlarında dükkan da dahil ticari amaçlı kullanım olan binalarda yola bakan bahçelerde bahçe duvarı veya yayaların can emniyetini tehlikeye düşürecek manialar yapılamaz.	Rule
3.c	Ön bahçe mesafesi 7 metre ve daha dar olup otopark olarak kullanılan parsellerde, ön bahçede bahçe duvarı ile benzeri mania yapılamaz; 7 metreden daha fazla ön bahçe mesafesi bulunan yerlerde ise parsel hududundan itibaren 5 metre bırakılarak geriye kalan bölümde sadece bitkisel çit yapılabilir.	Rule

ID	Textual Expression of Rule	Type
4	Devletin güvenlik ve emniyeti ile harekât ve savunma bakımından gizlilik veya önem arz eden bina ve tesisler ile okul, hastane, cezaevi, ibadet yerleri, elçilik ve benzeri özellik arz eden bina ve sanayi tesislerinin çevresindeki bahçe duvarları çevreye uyum sağlamak ve komşu parseller için sakınca doğurmamak üzere ilgili belediyenin imar birimince uygun görülen şekil ve yükseklikte bu madde hükümlerine bağlı olmadan yapılabilir.	Rule
5	Bahçe duvarları parselin ruhsat eki mimari projesine işlenerek yapılabilir.	Rule
6	Bahçe duvarları ve temelleri istinat duvarı olarak yapılsalar dahi, yola tecavüz edemezler.	Rule
7	Yığma duvar malzemesi ile yapılan bahçe duvarlarının yüksekliği doğal/tesviye edilmiş zeminden 1 metreden fazla olamaz.	Rule
Madde 44 - Kapıcı dairesi ve bekçi odası		
1	Kapıcı dairesi ve bekçi odası yapılacak binalar;	
1.a	Konut kullanımlı bağımsız bölüm sayısı 40'tan fazla olan ve katı yakıt kullanan kaloriferli veya kalorifersiz binalar için bir adet kapıcı dairesi yapılması zorunludur. Birden fazla yapı bulunan ve toplam konut kullanımlı bağımsız bölüm sayısı 40'tan fazla olan parsellerde de bu hüküm uygulanır, ancak konut kullanımlı bağımsız bölüm sayısının 80'i aşması halinde ikinci bir kapıcı dairesi yapılır. İlave her 80 daire için ek bir kapıcı dairesi yapılır. Ayrıca, birden fazla yapının bulunduğu parsellerde 60'tan fazla konut kullanımlı bağımsız bölümü olan her bir bina için mutlaka ayrı bir kapıcı dairesi yapılır.	Rule
1.b	Katı yakıt haricindeki diğer ısıtma sistemleri kullanılan konut kullanımlı binalar için konut kullanımlı bağımsız bölüm sayısının 60'tan fazla olması halinde bir, 150'den fazla olması halinde 2 kapıcı dairesi yapılması zorunludur. İlave her 150 daire için ek bir kapıcı dairesi yapılır.	Rule
1.c	Sıra evler düzeninde, ayrık, ikiz nizamda tek bağımsız bölümlü 1'den fazla müstakil konut binası bulunan parsellerde kapıcı dairesi yapılması mecburiyeti aranmaz.	Rule
1.ç	Yapı inşaat alanı 2000 m^2 'den fazla olan işyeri ve büro olarak kullanılan binalarda bekçi odası yapılması şarttır.	Rule
2	Kapıcı dairelerinin ve bekçi odalarının ölçü ve nitelikleri;	
2.a	Kapıcı daireleri, doğrudan ışık ve hava alabilecek şekilde düzenlenir.	Rule
2.b	Taşkın riski taşıyan alanlarda kalan binalarda düzenlenecek kapıcı dairelerinin kapı ve pencere boşluklarının alt seviyesi su taşkın seviyesinin en az 1.50 metre üzerinde olmak zorundadır.	Rule
2.c	Kapıcı dairelerinin toprağa dayalı ve iskân edilebilen bodrum katlarda yapılması halinde, oturma odası ve bir yatak odasının dış mekâna açılması, bu mekânların taban döşemesinin üst seviyesinin tabii veya tesviye edilmiş zemine gömülü olmaması, kapı ve pencere açılmak suretiyle, doğal aydınlatma ve havalandırmasının sağlanması, sel, taşkın ve su basmasına karşı önlem alınmış olması zorunludur. Kapıcı dairelerinin toprağa gömülü duvarlarında kuranglez yapmak suretiyle kapı ve pencere açılamayacağı gibi, bu duvarlarda pencere açılabilmesi için pencere denizliğinin tabii zeminden veya tesviye edilmiş zeminden en az 0.90 metre yukarıda konumlanması gerekir.	Rule

ID	Textual Expression of Rule	Type
2.ç	Bina içinde düzenlenen kapıcı daireleri, en az brüt 60 m²'dir. Kapıcı dairelerinde, her birisi en az 9 m² ve dar kenarı en az 2.50 metre olmak üzere 2 yatak odası ve 12 m²'den az olmamak üzere 1 oturma odası, en az 3.30 m² mutfak ve banyo veya duş yeri ve tuvalet bulunur.	Rule
2.d	Bir parselde birden fazla yapı bulunması halinde, yapılması zorunlu kapıcı daireleri orantılı bir şekilde parseldeki binalar arasında dağıtılır.	Rule
3	İşyeri ve büro olarak kullanılan binalarda yapılacak bekçi odası en az 4 m² büyüklüğünde, doğrudan ışık ve hava alabilecek şekilde düzenlenir. Bekçi odasında en az 1.5 m² 'lik bir tuvalet yer alır.	Rule
Madde 45 - Kontrol kulübeleri		
1	Üzerinde birden fazla yapı yapılması mümkün ve yüzölçümü en az 1000 m² olan parsellerde, istenmesi halinde, trafik emniyeti bakımından tehlike arz etmemek ve hiçbir şartta parsel sınırını aşmamak kaydıyla bahçe mesafeleri içinde kontrol kulübesi yapılabilir.	Rule
2	Kontrol kulübesi 9 m² 'yi geçemez.	Rule
3	Kontrol kulübesinin yüksekliği tabii veya tesviye edilmiş zeminden itibaren en fazla 4.00 metredir.	Rule
4	Kontrol kulübesi ile esas bina arasındaki mesafe 2.00 metreden az olamaz.	Rule
5	Devletin güvenliği bakımından özellik arz eden parsellerde bu maddede belirtilen ölçülere uyulma şartı aranmaz.	Rule

B. AMHZC Rule Classifications and Interpretability Conditions

Clause	Statement	Rule	Rule Properties	
			Rule Class	Interpretability
C9	ST1	R9.1	Class 1	Interpretable
	ST2	R9.2	Class 1	Semi-Interpretable
	ST3	R9.3	Class 1	Semi-Interpretable
C10	ST1	R9.1		
		R9.1.a		
		R9.1.a.1	Class 2	Interpretable
		R9.1.a.2	Class 3	Semi-Interpretable
		R9.1.a.3	Class 1	Interpretable
		R9.1.a.4	Class 1	Interpretable
		R9.1.a.5	Class 1	Interpretable
		R9.1.a.6	Class 1	Interpretable
		R9.1.a.7	Class 1	Interpretable
		R9.1.b	Class 1	Semi-Interpretable
		R9.1.c	Class 1	Semi-Interpretable
		R9.1.ç	Class 1	Semi-Interpretable
	ST2	R9.2		
		R9.2.a	Class 1	Interpretable
		R9.2.b	Class 1	Interpretable
		R9.2.c	Class 1	Interpretable
		R9.2.ç	Class 1	Semi-Interpretable
	ST3	R9.3		
		R9.3.a	Class 1	Semi-Interpretable
		R9.3.b	Class 1	Semi-Interpretable
	ST4	R9.4	Class 1	Semi-Interpretable
	ST5	R9.5		
		R9.5.a		Non-Interpretable
		R9.5.b	Reference	
	ST6	R9.6	Class 1	Interpretable
C11	ST1	R9.1		
		R9.1.a	Class 2	Interpretable
		R9.1.b	Class 2	Interpretable
		R9.1.c	Class 2	Interpretable
		R9.1.ç	Class 2	Semi-Interpretable
	ST2	R9.2	Class 1	Semi-Interpretable
	ST3	R9.3	Class 1	Semi-Interpretable
	ST4	R9.4		Non-Interpretable
	ST5	R9.5	Class 1	Interpretable

Clause	Statement	Rule	Rule Properties	
			Rule Class	Interpretability
	ST6	R9.6	Class 1	Interpretable
	ST7	R9.7		Non-Interpretable
C12	ST1	R9.1		Non-Interpretable
	ST2	R9.2		Non-Interpretable
	ST3	R9.3		Non-Interpretable
	ST4	R9.4		
		R9.1.a	Class 1	Interpretable
		R9.1.b	Class 1	Semi-Interpretable
		R9.1.c	Class 1	Semi-Interpretable
	ST5	R9.5		
		R9.5.a	Class 1	Semi-Interpretable
		R9.5.b		Non-Interpretable
		R9.5.c	Class 1	Interpretable
		R9.5.ç		Non-Interpretable
C15	ST1	R9.1	Class 2	Interpretable
	ST2	R9.2	Class 2	Interpretable
	ST3	R9.3		Non-Interpretable
	ST4	R9.4	Class 2	Semi-Interpretable
C24	ST1	R9.1	Class 1	Interpretable
	ST2	R9.2	Class 2	Interpretable
	ST3	R9.3		Non-Interpretable
	ST4	R9.4		Non-Interpretable
	ST5	R9.5	Class 1	Interpretable
	ST6	R9.6	Class 1	Semi-Interpretable
	ST7	R9.7	Reference	
	ST8	R9.8	Reference	
	ST9	R9.9	Class 1	Semi-Interpretable
	ST10	R9.10	Class 1	Interpretable
	ST11	R9.11	Class 1	Interpretable
	ST12	R9.12	Class 2	Interpretable
C25	ST1	R9.1		
		R9.1.a	Class 1	Interpretable
		R9.1.b	Class 1	Interpretable
		R9.1.c	Class 1	Interpretable
	ST2	R9.2		
		R9.a	Class 2	Interpretable
		R9.b	Class 1	Interpretable
		R9.c	Class 1	Interpretable
		R9.ç	Class 2	Interpretable
	ST3	R9.3		
		R9.a	Class 3	Interpretable

Clause	Statement	Rule	Rule Properties	
			Rule Class	Interpretability
		R9.b	Class 1	Interpretable
		R9.c	Class 2	Interpretable
	ST4	R9.4		
		R9.a	Class 1	Interpretable
		R9.b	Class 1	Interpretable
	ST5	R9.5		
		R9.a	Class 1	Interpretable
		R9.b	Class 2	Interpretable
		R9.c	Class 1	Interpretable
	ST6	R9.6	Reference	
	ST7	R9.7	Class 1	Interpretable
	ST8	R9.8	Reference	
C26	ST1	R9.1	Class 2	Interpretable
	ST2	R9.2	Class 2	Interpretable
	ST3	R9.3	Reference	
	ST4	R9.4		Non-Interpretable
	ST5	R9.5	Class 2	Interpretable
	ST6	R9.6	Class 2	Interpretable
	ST7	R9.7		Non-Interpretable
	ST8	R9.8	Reference	
	ST9	R9.9	Class 1	Interpretable
	ST10	R9.10		Non-Interpretable
	ST11	R9.11		Non-Interpretable
	ST12	R9.12		Non-Interpretable
	ST13	R9.13	Reference	
	ST14	R9.14	Class 1	Semi-Interpretable
C27	ST1	R9.1		
		R9.1.a	Class 2	Semi-Interpretable
		R9.1.b	Class 1	Interpretable
		R9.1.c		Non-Interpretable
	ST2	R9.2		
		R9.a	Class 1	Semi-Interpretable
		R9.b	Class 1	Semi-Interpretable
		R9.c	Class 1	Semi-Interpretable
		R9.ç	Class 1	Semi-Interpretable
	ST3	R9.3		
		R9.a	Class 1	Semi-Interpretable
		R9.b	Class 1	Interpretable
		R9.c	Class 2	Interpretable
		R9.ç	Class 2	Interpretable
		R9.d	Class 1	Interpretable

Clause	Statement	Rule	Rule Properties	
			Rule Class	Interpretability
		R9.e	Class 1	Interpretable
		R9.f	Class 1	Interpretable
		R9.g	Class 1	Interpretable
		R9.ğ	Class 1	Interpretable
		R9.h	Class 1	Interpretable
		R9.ı	Class 1	Interpretable
	ST4	R9.4		
		R9.a		
		R9.a.1	Class 1	Semi-Interpretable
		R9.a.2	Class 1	Semi-Interpretable
		R9.a.3	Class 1	Semi-Interpretable
		R9.b	Class 1	Semi-Interpretable
C28	ST1	R9.1		
		R9.1.a	Reference	
		R9.1.b	Class 1	Semi-Interpretable
		R9.1.c	Class 2	Interpretable
	ST2	R9.2	Class 2	Interpretable
	ST3	R9.3		
		R9.a	Class 2	Interpretable
		R9.b	Class 2	Interpretable
		R9.c	Class 1	Interpretable
	ST4	R9.4	Class 2	Interpretable
C29	ST1	R9.1	Class 1	Interpretable
	ST2	R9.2	Class 2	Interpretable
	ST3	R9.3	Class 3	Semi-Interpretable
	ST4	R9.4	Class 2	Semi-Interpretable
C30	ST1	R9.1	Class 1	Interpretable
	ST2	R9.2		
		R9.2a	Class 1	Interpretable
		R9.2b	Class 2	Interpretable
		R9.2c	Class 2	Interpretable
C31	ST1	R9.1	Class 2	Interpretable
C32	ST1	R9.1	Class 1	Interpretable
		R9.1.a	Class 2	Interpretable
		R9.1.b	Class 1	Interpretable
		R9.1.c	Class 3	Interpretable
	ST2	R9.2		
		R9.a	Class 1	Interpretable
		R9.b	Class 2	Interpretable
		R9.c	Class 1	Interpretable
		R9.ç	Class 1	Semi-Interpretable

Clause	Statement	Rule	Rule Properties	
			Rule Class	Interpretability
	ST3	R9.3		
		R9.a	Reference	
		R9.b	Class 1	Interpretable
		R9.c	Class 2	Semi-Interpretable
		R9.ç	Class 2	Semi-Interpretable
C33	ST1	R9.1		Non-Interpretable
	ST2	R9.2	Class 1	Semi-Interpretable
	ST3	R9.3	Class 1	Interpretable
	ST4	R9.4	Class 1	Interpretable
	ST5	R9.5	Class 1	Semi-Interpretable
	ST6	R9.6		Non-Interpretable
C34	ST1	R9.1		Non-Interpretable
		R9.1.a	Class 1	Interpretable
		R9.1.b	Class 1	Interpretable
		R9.1.c	Class 1	Interpretable
		R9.1.ç	Class 1	Interpretable
	ST2	R9.2		
		R9.2.a	Class 1	Interpretable
		R9..a.1	Class 1	Interpretable
		R9..a.2	Class 1	Interpretable
		R9..a.3	Class 1	Interpretable
		R9.2.b	Reference	
		R9.2.c	Class 2	Interpretable
		R9.2.ç	Class 1	Interpretable
		R9..ç.1	Class 2	Interpretable
		R9..ç.2	Class 2	Interpretable
	ST3	R9.3		
		R9.3.a	Class 2	Interpretable
		R9.3.b	Class 1	Semi-Interpretable
		R9.3.c	Reference	
		R9.3.ç	Reference	
	ST4	R9.4	Class 1	Semi-Interpretable
	ST5	R9.5		
		R9.5.a	Class 1	Interpretable
		R9.5.b	Class 1	Interpretable
	ST6	R9.6	Class 1	Interpretable
C35	ST1	R9.1		
		R9.1.a		
		R9.1.a.1	Class 1	Interpretable
		R9.1.a.2	Class 2	Interpretable
		R9.1.a.3	Class 2	Interpretable

Clause	Statement	Rule	Rule Properties	
			Rule Class	Interpretability
		R9.1.b		
		R9.1.b.1	Class 1	Interpretable
		R9.1.b.2	Class 2	Interpretable
		R9.1.b.3	Class 2	Interpretable
		R9.1.c		
		R9.1.c.1	Class 1	Interpretable
		R9.1.c.2	Class 2	Interpretable
		R9.1.c.3	Class 2	Interpretable
		R9.1.ç		
		R9.1.ç.1	Class 1	Interpretable
		R9.1.ç.2	Class 2	Interpretable
		R9.1.ç.3	Class 2	Interpretable
		R9.1.ç.4	Class 2	Interpretable
	ST2	R9.2		
		R9.2.a	Class 2	Semi-Interpretable
		R9.2.b	Class 1	Semi-Interpretable
		R9.2.c	Class 2	Interpretable
		R9.2.ç	Class 2	Interpretable
		R9.2.d	Class 1	Semi-Interpretable
		R9.2.e		Non-Interpretable
		R9.2.f	Class 2	Interpretable
	ST3	R9.3		
		R9.3.a	Class 1	Interpretable
		R9.3.b	Class 1	Interpretable
		R9.3.c		
		R9..c.1	Class 1	Semi-Interpretable
		R9..c.2	Class 1	Interpretable
		R9..c.3	Class 1	Semi-Interpretable
		R9..c.4	Class 1	Semi-Interpretable
		R9..c.5	Reference	
C36	ST1	R9.1	Class 1	Interpretable
	ST2	R9.2		Non-Interpretable
	ST3	R9.3	Class 1	Interpretable
	ST4	R9.4	Class 1	Interpretable
	ST5	R9.5	Class 1	Interpretable
C37	ST1	R9.1		
		R9.1.a		
		R9.1.a.1	Class 1	Interpretable
		R9.1.a.2	Class 1	Semi-Interpretable
		R9.1.b		
		R9.1.b.1	Class 1	Interpretable

Clause	Statement	Rule	Rule Properties	
			Rule Class	Interpretability
	ST2	R9.2		
		R9.2.a	Class 1	Interpretable
		R9.2.b	Class 1	Interpretable
		R9.2.c	Class 2	Interpretable
	ST3	R9.3		
		R9.3.a	Class 1	Interpretable
		R9.3.b	Class 1	Semi-Interpretable
	ST4	R9.4		
		R9.4.a	Class 1	Interpretable
		R9.4.b	Class 2	Interpretable
		R9.4.c	Class 1	Interpretable
	ST5	R9.5		
		R9.5.a	Class 1	Semi-Interpretable
		R9.5.b	Class 1	Interpretable
	ST6	R9.6		
		R9.6.a	Class 1	Interpretable
		R9.6.b	Class 1	Interpretable
		R9.6.c	Class 1	Interpretable
	ST7	R9.7	Class 1	Interpretable
	ST8	R9.8		
		R9.8.a	Class 1	Interpretable
		R9.8.b	Class 1	Semi-Interpretable
	ST9	R9.9		
		R9.9.a	Class 1	Interpretable
		R9.9.b	Class 1	Interpretable
C38	ST1	R9.1	Class 1	Semi-Interpretable
	ST2	R9.2	Reference	
C39	ST1	R9.1	Class 2	Interpretable
	ST2	R9.2	Class 1	Interpretable
	ST3	R9.3	Class 1	Interpretable
C40	ST1	R9.1		
		R9.1.a	Class 1	Interpretable
		R9.1.a.1	Class 1	Interpretable
		R9.1.a.2	Class 1	Interpretable
		R9.1.b	Class 1	Interpretable
		R9.1.c	Class 2	Interpretable
	ST2	R9.2		
		R9.2.a		Non-Interpretable
		R9.2.b		Non-Interpretable
		R9.2.c		Non-Interpretable
		R9.2.ç	Class 1	Interpretable

Clause	Statement	Rule	Rule Properties	
			Rule Class	Interpretability
		R9.2.d	Class 2	Interpretable
		R9.2.e	Class 1	Semi-Interpretable
		R9.2.f		Non-Interpretable
		R9.2.g	Class 1	Semi-Interpretable
		R9.2.ğ	Class 1	Interpretable
		R9.2.h	Reference	
		R9.2.ı	Class 1	Semi-Interpretable
		R9.2.i	Class 1	Interpretable
		R9.2.j	Class 1	Interpretable
	ST3	R9.3		
		R9.3.a		Non-Interpretable
		R9.3.b	Class 1	Interpretable
		R9.3.c	Class 1	Interpretable
		R9.3.ç	Class 2	Interpretable
		R9.3.d	Class 2	Interpretable
		R9.3.e	Class 1	Semi-Interpretable
		R9.3.f	Class 2	Interpretable
		R9.3.g	Class 1	Interpretable
		R9.3.ğ	Class 2	Interpretable
		R9.3.h	Class 1	Semi-Interpretable
		R9.3.ı	Class 1	Semi-Interpretable
		R9.3.i	Class 1	Semi-Interpretable
		R9.3.j	Class 1	Interpretable
C41	ST1	R9.1		
		R9.1.a	Class 3	Interpretable
		R9.1.b	Class 1	Semi-Interpretable
		R9.1.c	Class 1	Interpretable
		R9.1ç	Class 1	Semi-Interpretable
	ST2	R9.2		
		R9.a	Class 1	Interpretable
		R9.b	Class 1	Interpretable
		R9.c	Class 1	Interpretable
		R9.ç	Class 1	Semi-Interpretable
		R9.d	Class 1	Semi-Interpretable
C42	ST1	R9.1		
		R9.1.a	Class 1	Interpretable
		R9.1.b	Class 1	Interpretable
		R9.1.c	Class 1	Interpretable
		R9.1ç	Class 1	Interpretable
	ST2	R9.2	Class 1	Semi-Interpretable
	ST3	R9.3	Reference	

Clause	Statement	Rule	Rule Properties	
			Rule Class	Interpretability
	ST4	R9.4	Class 1	Interpretable
	ST5	R9.5	Class 2	Interpretable
	ST6	R9.6	Class 1	Interpretable
	ST7	R9.7	Class 1	Semi-Interpretable
	ST8	R9.8	Class 2	Semi-Interpretable
C43	ST1	R9.1		
		R9.1.a	Class 2	Interpretable
		R9.1.b	Class 2	Semi-Interpretable
		R9.1.c	Class 2	Interpretable
	ST2	R9.2		
		R9.2.a	Class 2	Interpretable
	ST3	R9.3		
		R9.3.a	Class 1	Semi-Interpretable
		R9.3.b		Non-Interpretable
		R9.3.c	Class 2	Interpretable
	ST4	R9.4		Non-Interpretable
	ST5	R9.5	Class 1	Semi-Interpretable
	ST6	R9.6	Class 1	Semi-Interpretable
	ST7	R9.7	Class 2	Interpretable
C44	ST1	R9.1		
		R9.1.a	Class 1	Interpretable
		R9.1.b	Class 1	Interpretable
		R9.1.c	Class 1	Interpretable
		R9.1.ç	Class 1	Interpretable
	ST2	R9.2		
		R9.2.a	Class 1	Interpretable
		R9.2.b	Class 1	Semi-Interpretable
		R9.2.c	Class 1	Semi-Interpretable
		R9.2.ç	Class 1	Interpretable
		R9.2.d	Class 1	Semi-Interpretable
	ST3	R9.3	Class 1	Interpretable
C45	ST1	R9.1	Class 1	Semi-Interpretable
	ST2	R9.2	Class 1	Interpretable
	ST3	R9.3	Class 1	Interpretable
	ST4	R9.4	Class 2	Interpretable
	ST5	R9.5	Class 1	Semi-Interpretable

C. List of Interpretable Rules and Their Interpretability Conditions to the SMC Ruleset Manager

No	Clause	Rule	Rule Class	SMC Interpretability
1	9 - Building Road Façade Lengths	R9.1	Class 1	Interpretable
2	10 - Depth of Buildings	R10.1.a.1	Class 1	Non-Interpretable
3	10 - Depth of Buildings	R10.1.a.3	Class 1	Non-Interpretable
4	10 - Depth of Buildings	R10.1.a.4	Class 1	Non-Interpretable
5	10 - Depth of Buildings	R10.1.a.5	Class 1	Non-Interpretable
6	10 - Depth of Buildings	R10.1.a.6	Class 1	Non-Interpretable
7	10 - Depth of Buildings	R10.1.a.7	Class 1	Non-Interpretable
8	10 - Depth of Buildings	R10.2.a	Class 1	Interpretable
9	10 - Depth of Buildings	R10.2.b	Class 1	Interpretable
10	10 - Depth of Buildings	R10.2.c	Class 1	Non-Interpretable
11	10 - Depth of Buildings	R10.6	Class 1	Interpretable
12	11 - Maximum Height of Buildings	R11.1.a	Class 2	Interpretable
13	11 - Maximum Height of Buildings	R11.1.b	Class 2	Interpretable
14	11 - Maximum Height of Buildings	R11.1.c	Class 2	Interpretable
15	11 - Maximum Height of Buildings	R11.5	Class 1	Non-Interpretable
16	11 - Maximum Height of Buildings	R11.6	Class 1	Interpretable
17	12 - Construction of Different Height Buildings	R12.4.a	Class 1	Non-Interpretable
18	12 - Construction of Different Height Buildings	R12.5.c	Class 1	Non-Interpretable
19	15 - Entrance Heights of the Buildings	R15.1	Class 2	Interpretable
20	15 - Entrance Heights of the Buildings	R15.2	Class 2	Interpretable
21	24 - Building Entries and Pedestrian Ramps	R24.1	Class 1	Interpretable
22	24 - Building Entries and Pedestrian Ramps	R24.2	Class 2	Interpretable
23	24 - Building Entries and Pedestrian Ramps	R24.5	Class 1	Interpretable
24	24 - Building Entries and Pedestrian Ramps	R24.10	Class 1	Interpretable
25	24 - Building Entries and Pedestrian Ramps	R24.11	Class 1	Non-Interpretable
26	24 - Building Entries and Pedestrian Ramps	R24.12	Class 2	Non-Interpretable
27	25 - Stairs	R25.1.a	Class 1	Interpretable
28	25 - Stairs	R25.1.b	Class 1	Interpretable
29	25 - Stairs	R25.1.c	Class 1	Interpretable

No	Clause	Rule	Rule Class	SMC Interpretability
30	25 - Stairs	R25.2a	Class 2	Non-Interpretable
31	25 - Stairs	R25.2b	Class 1	Interpretable
32	25 - Stairs	R25.2c	Class 1	Interpretable
33	25 - Stairs	R25.2ç	Class 2	Non-Interpretable
34	25 - Stairs	R25.3a	Class 3	Non-Interpretable
35	25 - Stairs	R25.3b	Class 1	Non-Interpretable
36	25 - Stairs	R25.3c	Class 2	Non-Interpretable
37	25 - Stairs	R25.4a	Class 1	Non-Interpretable
38	25 - Stairs	R25.4b	Class 1	Non-Interpretable
39	25 - Stairs	R25.5a	Class 1	Non-Interpretable
40	25 - Stairs	R25.5b	Class 2	Interpretable
41	25 - Stairs	R25.5c	Class 1	Interpretable
42	25 - Stairs	R25.7	Class 1	Interpretable
43	26 - Elevators	R26.1	Class 2	Interpretable
44	26 - Elevators	R26.2	Class 2	Non-Interpretable
45	26 - Elevators	R26.5	Class 2	Non-Interpretable
46	26 - Elevators	R26.6	Class 2	Non-Interpretable
47	26 - Elevators	R26.9	Class 1	Non-Interpretable
48	27 - Basements	R27.1.b	Class 1	Interpretable
49	27 - Basements	R27.3b	Class 1	Interpretable
50	27 - Basements	R27.3c	Class 2	Non-Interpretable
51	27 - Basements	R27.3ç	Class 2	Non-Interpretable
52	27 - Basements	R27.3d	Class 1	Interpretable
53	27 - Basements	R27.3e	Class 1	Non-Interpretable
54	27 - Basements	R27.3f	Class 1	Interpretable
55	27 - Basements	R27.3g	Class 1	Non-Interpretable
56	27 - Basements	R27.3ğ	Class 1	Non-Interpretable
57	27 - Basements	R27.3h	Class 1	Interpretable
58	27 - Basements	R27.3ı	Class 1	Interpretable
59	28 - Ground Floors	R28.1.c	Class 2	Interpretable
60	28 - Ground Floors	R28.2	Class 2	Non-Interpretable
61	28 - Ground Floors	R28.3a	Class 2	Non-Interpretable
62	28 - Ground Floors	R28.3b	Class 2	Non-Interpretable
63	28 - Ground Floors	R28.3c	Class 1	Non-Interpretable
64	28 - Ground Floors	R28.4	Class 2	Non-Interpretable

No	Clause	Rule	Rule Class	SMC Interpretability
65	29 - Some Subjects About Building Basements and Ground Floors	R29.1	Class 1	Interpretable
66	29 - Some Subjects About Building Basements and Ground Floors	R29.2	Class 2	Interpretable
67	30 - Mezzanine Floors	R30.1	Class 1	Interpretable
68	30 - Mezzanine Floors	R30.2a	Class 1	Interpretable
69	30 - Mezzanine Floors	R30.2b	Class 2	Non-Interpretable
70	30 - Mezzanine Floors	R30.2c	Class 2	Non-Interpretable
71	31 - Construction of Indoor Parking on the Front Yard of Ground Level Elevated Parcels	R31.1	Class 2	Non-Interpretable
72	32 - Required Rooms/Sections in Buildings and Their Minimum Dimensions	R32.1	Class 1	Non-Interpretable
73	32 - Required Rooms/Sections in Buildings and Their Minimum Dimensions	R32.1.a	Class 2	Interpretable
74	32 - Required Rooms/Sections in Buildings and Their Minimum Dimensions	R32.1.b	Class 1	Interpretable
75	32 - Required Rooms/Sections in Buildings and Their Minimum Dimensions	R32.1.c	Class 3	Interpretable
76	32 - Required Rooms/Sections in Buildings and Their Minimum Dimensions	R32.2a	Class 1	Non-Interpretable
77	32 - Required Rooms/Sections in Buildings and Their Minimum Dimensions	R32.2b	Class 2	Non-Interpretable
78	32 - Required Rooms/Sections in Buildings and Their Minimum Dimensions	R32.2c	Class 1	Interpretable
79	32 - Required Rooms/Sections in Buildings and Their Minimum Dimensions	R32.3b	Class 1	Interpretable
80	33 - Floor Heights	R33.3	Class 1	Interpretable
81	33 - Floor Heights	R33.4	Class 1	Interpretable
82	34 - Roofs	R34.1.a	Class 1	Non-Interpretable
83	34 - Roofs	R34.1.b	Class 1	Non-Interpretable
84	34 - Roofs	R34.1.c	Class 1	Non-Interpretable
85	34 - Roofs	R34.1.ç	Class 1	Non-Interpretable
86	34 - Roofs	R34.2.a	Class 1	Interpretable
87	34 - Roofs	R34.2.a.1	Class 1	Interpretable
88	34 - Roofs	R34.2.a.2	Class 1	Non-Interpretable

No	Clause	Rule	Rule Class	SMC Interpretability
89	34 - Roofs	R34.2.a.3	Class 1	Non-Interpretable
90	34 - Roofs	R34.2.c	Class 2	Non-Interpretable
91	34 - Roofs	R34.2.ç	Class 1	Interpretable
92	34 - Roofs	R34.2.ç.1	Class 2	Non-Interpretable
93	34 - Roofs	R34.2.ç.2	Class 2	Non-Interpretable
94	34 - Roofs	R34.3.a	Class 2	Interpretable
95	34 - Roofs	R34.5.a	Class 1	Interpretable
96	34 - Roofs	R34.5.b	Class 1	Non-Interpretable
97	34 - Roofs	R34.6	Class 1	Non-Interpretable
98	35 - Cantilevers	R35.1.a.1	Class 1	Interpretable
99	35 - Cantilevers	R35.1.a.2	Class 2	Non-Interpretable
100	35 - Cantilevers	R35.1.a.3	Class 2	Interpretable
101	35 - Cantilevers	R35.1.b.1	Class 1	Interpretable
102	35 - Cantilevers	R35.1.b.2	Class 2	Non-Interpretable
103	35 - Cantilevers	R35.1.b.3	Class 2	Interpretable
104	35 - Cantilevers	R35.1.c.1	Class 1	Interpretable
105	35 - Cantilevers	R35.1.c.2	Class 2	Non-Interpretable
106	35 - Cantilevers	R35.1.c.3	Class 2	Interpretable
107	35 - Cantilevers	R35.1.ç.1	Class 1	Non-Interpretable
108	35 - Cantilevers	R35.1.ç.2	Class 2	Non-Interpretable
109	35 - Cantilevers	R35.1.ç.3	Class 2	Non-Interpretable
110	35 - Cantilevers	R35.1.ç.4	Class 2	Non-Interpretable
111	35 - Cantilevers	R35.2.c	Class 2	Non-Interpretable
112	35 - Cantilevers	R35.2.ç	Class 2	Non-Interpretable
113	35 - Cantilevers	R35.2.f	Class 2	Non-Interpretable
114	35 - Cantilevers	R35.3.a	Class 1	Non-Interpretable
115	35 - Cantilevers	R35.3.b	Class 1	Non-Interpretable
116	35 - Cantilevers	R35.3.c.2	Class 1	Non-Interpretable
117	36 - Eaves	R36.1	Class 1	Non-Interpretable
118	36 - Eaves	R36.3	Class 1	Non-Interpretable
119	36 - Eaves	R36.4	Class 1	Non-Interpretable
120	36 - Eaves	R36.5	Class 1	Non-Interpretable
121	37 - Light Shafts, Vent Stacks and Air Shafts	R37.1.a.1	Class 1	Interpretable
122	37 - Light Shafts, Vent Stacks and Air Shafts	R37.1.b.1	Class 1	Interpretable

No	Clause	Rule	Rule Class	SMC Interpretability
123	37 - Light Shafts, Vent Stacks and Air Shafts	R37.2.a	Class 1	Interpretable
124	37 - Light Shafts, Vent Stacks and Air Shafts	R37.2.b	Class 1	Non-Interpretable
125	37 - Light Shafts, Vent Stacks and Air Shafts	R37.2.c	Class 2	Interpretable
126	37 - Light Shafts, Vent Stacks and Air Shafts	R37.3.a	Class 1	Non-Interpretable
127	37 - Light Shafts, Vent Stacks and Air Shafts	R37.4.a	Class 1	Non-Interpretable
128	37 - Light Shafts, Vent Stacks and Air Shafts	R37.4.b	Class 2	Non-Interpretable
129	37 - Light Shafts, Vent Stacks and Air Shafts	R37.4.c	Class 1	Interpretable
130	37 - Light Shafts, Vent Stacks and Air Shafts	R37.5.b	Class 1	Interpretable
131	37 - Light Shafts, Vent Stacks and Air Shafts	R37.6.a	Class 1	Non-Interpretable
132	37 - Light Shafts, Vent Stacks and Air Shafts	R37.6.b	Class 1	Non-Interpretable
133	37 - Light Shafts, Vent Stacks and Air Shafts	R37.6.c	Class 1	Non-Interpretable
134	37 - Light Shafts, Vent Stacks and Air Shafts	R37.7	Class 1	Non-Interpretable
135	37 - Light Shafts, Vent Stacks and Air Shafts	R37.8.a	Class 1	Non-Interpretable
136	37 - Light Shafts, Vent Stacks and Air Shafts	R37.9.a	Class 1	Non-Interpretable
137	37 - Light Shafts, Vent Stacks and Air Shafts	R37.9.b	Class 1	Non-Interpretable
138	39 - Areaways	R39.1	Class 2	Interpretable
139	39 - Areaways	R39.2	Class 1	Non-Interpretable
140	39 - Areaways	R39.3	Class 1	Interpretable
141	40 - Smoke Chimneys	R40.1.a	Class 1	Non-Interpretable
142	40 - Smoke Chimneys	R40.1.a.1	Class 1	Non-Interpretable
143	40 - Smoke Chimneys	R40.1.a.2	Class 1	Non-Interpretable
144	40 - Smoke Chimneys	R40.1.b	Class 1	Non-Interpretable
145	40 - Smoke Chimneys	R40.1.c	Class 2	Non-Interpretable
146	40 - Smoke Chimneys	R40.2.ç	Class 1	Non-Interpretable
147	40 - Smoke Chimneys	R40.2.d	Class 2	Non-Interpretable
148	40 - Smoke Chimneys	R40.2.ğ	Class 1	Interpretable

No	Clause	Rule	Rule Class	SMC Interpretability
149	40 - Smoke Chimneys	R40.2.i	Class 1	Non-Interpretable
150	40 - Smoke Chimneys	R40.2.j	Class 1	Non-Interpretable
151	40 - Smoke Chimneys	R40.3.b	Class 1	Non-Interpretable
152	40 - Smoke Chimneys	R40.3.c	Class 1	Non-Interpretable
153	40 - Smoke Chimneys	R40.3.ç	Class 2	Interpretable
154	40 - Smoke Chimneys	R40.3.d	Class 2	Interpretable
155	40 - Smoke Chimneys	R40.3.f	Class 2	Non-Interpretable
156	40 - Smoke Chimneys	R40.3.g	Class 1	Interpretable
157	40 - Smoke Chimneys	R40.3.ğ	Class 2	Non-Interpretable
158	40 - Smoke Chimneys	R40.3.j	Class 1	Non-Interpretable
159	41 - Railings and Parapets	R41.1.a	Class 3	Non-Interpretable
160	41 - Railings and Parapets	R41.1.c	Class 1	Interpretable
161	41 - Railings and Parapets	R41.2a	Class 1	Interpretable
162	41 - Railings and Parapets	R41.2b	Class 1	Interpretable
163	41 - Railings and Parapets	R41.2c	Class 1	Interpretable
164	42 - Doors and Windows	R42.1.a	Class 1	Interpretable
165	42 - Doors and Windows	R42.1.b	Class 1	Interpretable
166	42 - Doors and Windows	R42.1.c	Class 1	Interpretable
167	42 - Doors and Windows	R42.1ç	Class 1	Interpretable
168	42 - Doors and Windows	R42.4	Class 1	Interpretable
169	42 - Doors and Windows	R42.5	Class 2	Non-Interpretable
170	42 - Doors and Windows	R42.6	Class 1	Non-Interpretable
171	43 - Garden Walls	R43.1.a	Class 2	Interpretable
172	43 - Garden Walls	R43.1.c	Class 2	Interpretable
173	43 - Garden Walls	R43.2.a	Class 2	Interpretable
174	43 - Garden Walls	R43.3.c	Class 2	Non-Interpretable
175	43 - Garden Walls	R43.7	Class 2	Interpretable
176	44 - Porter Suite and Watchmen Room	R44.1.a	Class 1	Non-Interpretable
177	44 - Porter Suite and Watchmen Room	R44.1.b	Class 1	Non-Interpretable
178	44 - Porter Suite and Watchmen Room	R44.1.c	Class 1	Non-Interpretable
179	44 - Porter Suite and Watchmen Room	R44.1.ç	Class 1	Interpretable
180	44 - Porter Suite and Watchmen Room	R44.2.a	Class 1	Interpretable
181	44 - Porter Suite and Watchmen Room	R44.2.ç	Class 1	Non-Interpretable
182	44 - Porter Suite and Watchmen Room	R44.3	Class 1	Interpretable
183	45 - Control Sheds	R45.2	Class 1	Interpretable

No	Clause	Rule	Rule Class	SMC Interpretability
184	45 - Control Sheds	R45.3	Class 1	Interpretable
185	45 - Control Sheds	R45.4	Class 2	Interpretable