Automation in Construction 71 (2016) 49–61

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Translating building legislation into a computer-executable format for evaluating building permit requirements

Hyunsoo Lee [a], Jin-Kook Lee [a,*], Seokyung Park [a], Inhan Kim [b]

[a] Department of Interior Architecture Design, Hanyang University, Seoul, Republic of Korea
[b] Department of Architecture, Kyunghee University, Suwon, Republic of Korea

ARTICLE INFO

Article history:
Received 30 September 2015
Received in revised form 19 April 2016
Accepted 25 April 2016
Available online 16 May 2016

Keywords:
Building information modeling (BIM)
Automated design assessment
Rule checking
Logic rule
Computer-executable form

ABSTRACT

In this paper, we describe an approach to translating the written content of the Korean Building Act into a computer-executable format for the purpose of evaluating building permit requirements. Among the various applications of building information modeling (BIM), we focus on automated design assessment and its rule-making process, which has been performed by both architects and software developers. Compared to the conventional rule-making approach, which is integrated with rule-checking software, our suggested mechanism (KBimLogic) of converting rule sentences from the Korean Building Act into computer-executable code (KBimCode) is a software-independent approach that separates the rule-making and rule-checking processes. We use the rule-making approach to translate the Korean Building Act into an explicit code that focuses on building permit requirements.

Building permit-related regulations are defined in the Korean Building Act, but some of the sentences are ambiguous, and some implicit definitions hinder translation into an explicitly defined computer-executable form. Some building permit-related requirements vary by building type, administrative district, permitted date, and as-built date; thus, it is critical to design computable rules independent of specific proprietary software. The building permit system in Korea changes (as it does in other countries), making it critical that code compliance rules be kept up-to-date. Our work is motivated by such fundamentals and suggests a logic rule-based mechanism for use by non-programmers and a user-friendly approach to the rule-making process.

The scope and major components of our research are as follows: (1) noun phrases classify building objects and associated properties from the Korean Building Act; (2) verb phrases derive high-level methods to construct the actual rule-checking body; and (3) the logic rule-based mechanism processes natural language sentences. In the actual implementation for translation, this approach has been demonstrated by each database and a GUI-based application named KBimLogic, for generating KBimCode. The KBimCode, which contains a set of specific building permit requirements translated from sentences, can be imported into our code-checking software (KBimAssess) using a specific file format.

1. Introduction

To qualify for building permits, architectural design firms must satisfy a series of requirements defined in the Korean Building Act, which contains design rules and regulations that are often interpreted in various ways by domain experts. The verification of multiple requirements with given building design models can be time consuming, and the results can be incomplete and arbitrary. The makers of various building information modeling (BIM) applications have attempted to reengineer natural language-based design regulations into machine-readable formats to improve the logical structure of the regulatory codes and thereby automate the design evaluation process [1].

BIM applications in the architecture, engineering, construction, and facility management industries support computer-interpretable and information-rich building models and automate some parts of the design assessment process. As has been demonstrated by some challenging projects using this approach, BIM reduces design errors and eventually improves the overall design quality [2,3]. Among the various tasks required to automate the design assessment process for acquiring building permits, this study focuses on the approach and mechanism for generating computer-readable explicit forms from the implicit natural language sentences specific to the Korea Building Act [4].

The scope of this study is to describe major findings and distinguished features developed to translate sentences from the Korean Building Act into computer-executable forms that can be directly used in a rule-checking application. To design the logical rule-based mechanism, we refined the objects/properties and predicates of sentences from the Korean Building Act into a database called the "logic meta

* Corresponding author.
E-mail addresses: hyunsoolee120@gmail.com (H. Lee), designit@hanyang.ac.kr (J.-K. Lee), seokyung.park529@gmail.com (S. Park), ihkim@khu.ac.kr (I. Kim).

http://dx.doi.org/10.1016/j.autcon.2016.04.008
0926-5805/

database." Initially, the logic meta-database contained more than 15,000 building permit-related sentences (e.g., 12 types of higher-level laws including content, such as required specifications for elevators, fire-retarding divisions, and evacuation facilities, and 49 types of related lower articles or standards). The logic rule-based mechanism, shown in Fig. 1, is as follows: (1) classify each object and its related properties in the Korean Building Act, rather than using a standard BIM model such as IFC or the Revit schema; (2) classify a method for verifying the objects and properties of the target building; and (3) translate the natural language sentences in the Korean Building Act according to a logical process. Through a logical rule-based mechanism, sentences are stored in the logic meta-database as intermediate code, called "KBimCode," which is reusable and executable according to the purposes of the Korean Building Act [5]. The KBimCode is managed in a web-based database interface. An application for generating and managing KBimCode is also implemented as "KBimLogic." Exported KBimCode can be used in a BIM assessment tool called "KBimAssess" to check designs against building permit requirements in Korea. Using this series of application systems, this study focuses on the translation mechanism, including the logic meta-databases. Actual building permit-related code compliance verification is demonstrated using the suggested mechanism and KBimAssess applications with existing building models. (See Fig.1.) (See Fig. 2.)

2. Background and research motivation

From the perspective of facilitating computer-executable rules from natural language regulations (a so-called rule-making process), several parties have explored various applications of automated design reviews on actual projects.

1. CORENET (2005)
   The Construction and Real Estate Network (CORENET) is Singapore's major IT initiative, led by the Ministry of National Development and driven by the Building and Construction Authority [6,7]. It aims to streamline the business processes of the construction industry in order to improve turnaround time, productivity, and quality [8]. Among its functionalities, the e-PlanCheck module performs automated checks against Singapore codes on building control, barrier-free access, fire prevention, environmental health, households, public housing, and vehicle parking. It consists of an e-submission system and integrated review of plans for IFC-based files. The e-PlanCheck module is a cutting-edge system that integrates expert knowledge in regulations, artificial intelligence, and BIM technologies [9,10]. To verify code compliance, this project developed the FORNAX platform to calculate each required condition. The FORNAX platform is an object library programmed in C++ language [11]. Each FORNAX object contains hard-coded rules that assess themselves.
2. Statsbygg Project (2009)
   Norway undertook several projects for automated code checks. CORENET's e-PlanCheck was adopted and tested in the Selvaag Group's "Munkerud" housing project and the Akershus University Hospital project [12]. The HITOS (Tromsø University College) project driven by Statsbygg (Norwegian Directorate of Public Construction and Property) performed spatial requirement and accessibility checks using dRofus software [13] and the Solibri Model Checker (SMC) [14]. Although those projects demonstrated the increasing feasibility of BIM-based code checking, it remains a "black-box" process in terms of how commercial software implements related codes, standards, and regulations for computable rules. The Statsbygg project developed methods to translate and transform building-related codes in standard documents, national codes, and regulations for

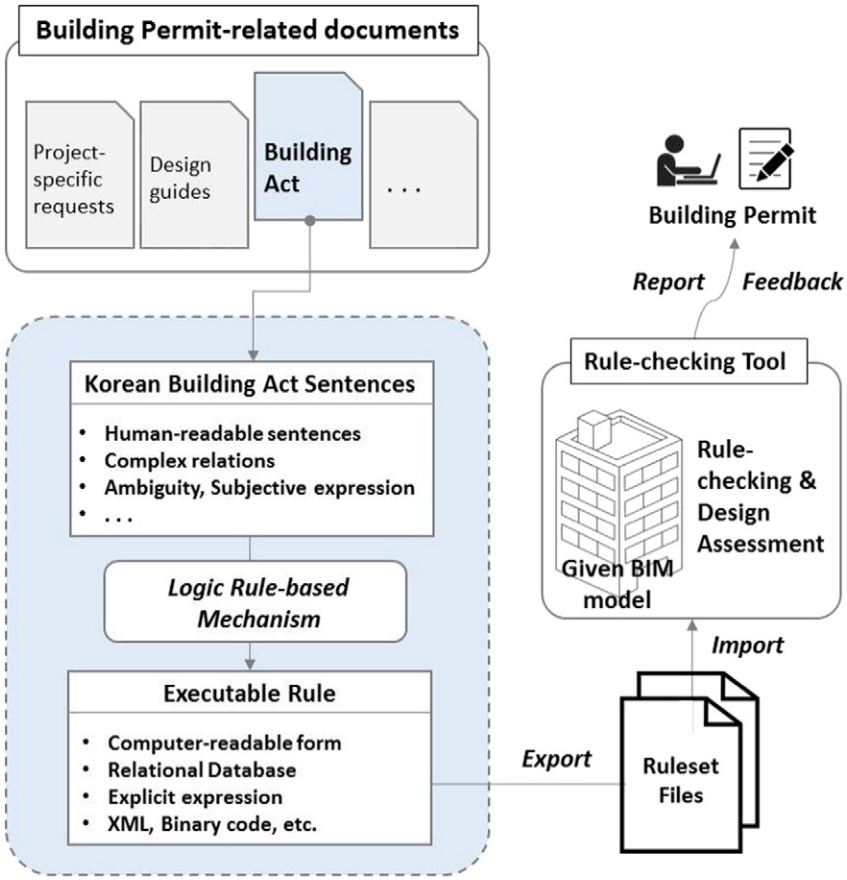

Fig. 1. Overview diagram of the research scope and flow: an approach to translating Building Permit-related documents into a computer-executable format.

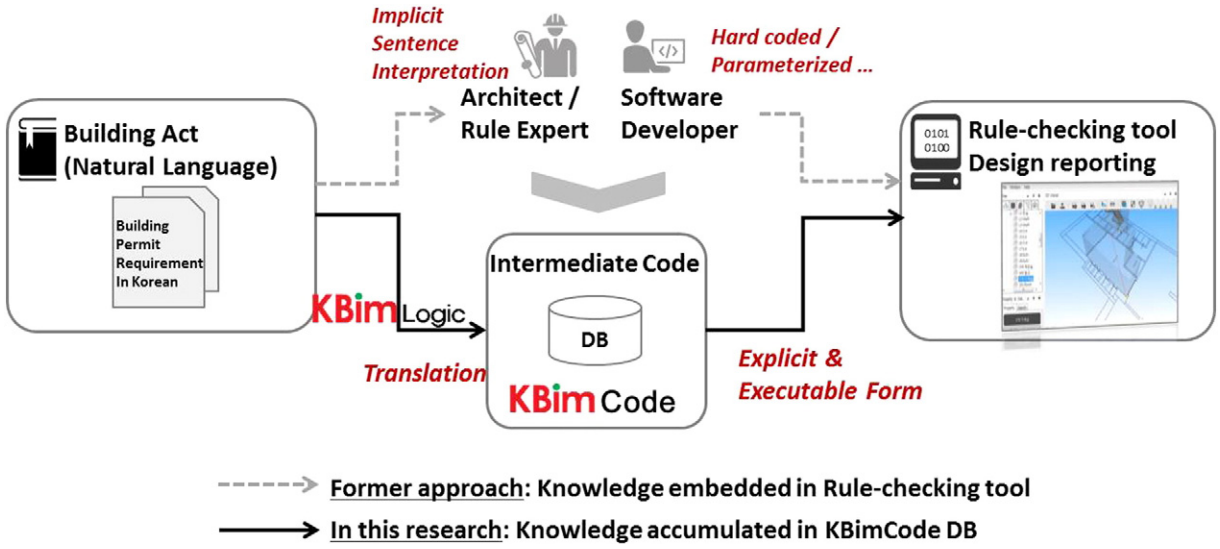

Fig. 2. Comparison between the alternative approach (above) and the method proposed in this research: Rule-making and rule-checking processes are separated using the KBimCode database.

use in digital rule-checker software. This project suggested a six-stage standardization process: (1) definition of scope and source for the rule set, (2) computability assessment, (3) committee assessment, (4) logic rule notation, (5) selection of rule format, and (6) implementation of the rule in rule-checker software.

3. DesignCheck (2006)
   DesignCheck is an automated code-checking system for the Building Code of Australia [15]. It employs a shared object-oriented database approach using the EXPRESS Data Manager (EDM) platform [16]. The EDM contains data models and schemas, including model schemas, rule schemas, and query schemas. In particular, rule schemas define rules to validate data models using the EXPRESS language and define entities, rules, functions, and procedures based on the building code. For the initial feasibility assessment, the Australian Standard 1428.1 "Design for access and mobility" was encoded with EDM rule schemas. Object-based interpretation was first performed for pre-implementation specification and used descriptions, performance requirements, objects, properties, relationships, and domain-specific knowledge for interpretation. Then, the object-based interpretation was encoded into EDM rule schemas.
4. SMART Code (2006)
   The International Code Council's SMARTCode project was developed to automate code compliance checks for I-Codes and federal and state codes. The SMARTCode builder software provides an interface to mark the entities required for a code check [17]. Information about the extracted entity is collected in a STEP file and converted using XML schemas. The XML extracted from STEP and from the document information can be compared, and the legality can be verified [18].
5. US GSA Project (2007)
   The US GSA (General Services Administration) and US Courts have supported the development of design rule checks for United States federal courthouses, which is an early example of a rule check applied to automated design guides [19]. In this project, a research team from the Georgia Institute of Technology developed a BIM application that automated rule checks based on the Courts Design Guide, producing a prototype of an assessment tool for architects and clients. The team parameterized circulation and security rules to develop parametric rules executable in an SMC plug-in. The assessment tool provided design assessment results that included appropriate feedback and data to designers [20].

Start	Required	Destination	Transition Conditions	...
Name: Press / Media Room		Name: Bankruptcy courtroom,U...	Security Level: public,Usage: circulation,Route Length: 100.00 m,Verti...	...
Name: Press / Media Room		Name: Magistrate courtroom,Us...	Security Level: public,Usage: circulation,Route Length: 100.00 m,Verti...	...
Name: Trial Jury Room	ALL Name: Soundlock	Name: -	Security Level: public,Usage: circulation,Vertical Access: allowed	...
Name: Grand Jury Hearing Room	ALL Name: Entry Security Station	Name: -,Usage: circulation	Security Level: restricted,Usage: circulation,Vertical Access: allowed	...
Name: Circuit Librarian		Name: Library Entry/Lobby	Security Level: restricted,Usage: circulation,Route Length: 100.00 m,...	...
Name: Circuit Librarian		Name: Library Circulation Area	Security Level: restricted,Usage: circulation,Route Length: 100.00 m,...	...
Name: Circuit Librarian		Name: Reference/Card Catalog ...	Security Level: restricted,Usage: circulation,Route Length: 100.00 m,...	...
Name: Circuit Executive's Office		Name: -,Usage: circulation	Security Level: restricted,Usage: circulation,Vertical Access: allowed	...
Name: Senior Staff Attorney		Name: -,Usage: circulation	Security Level: restricted,Usage: circulation,Vertical Access: allowed	...
Name: Senior Staff Attorney		Name: Library Entry/Lobby	Security Level: restricted,Usage: circulation,Route Length: 100.00 m,...	...
Name: Senior Staff Attorney		Name: Court of Appeals Clerk o...	Security Level: restricted,Usage: circulation,Route Length: 100.00 m,...	...

On/Off	Condition	Value
☑	Security Level	restricted
☑	Usage	circulation
☑	Route Length	100.00 m
☐	Direct Access	
☐	Vertical Access	

Fig. 3. An example screenshot of the SMC rule manager for customizing rule sets using a pre-defined rule library.

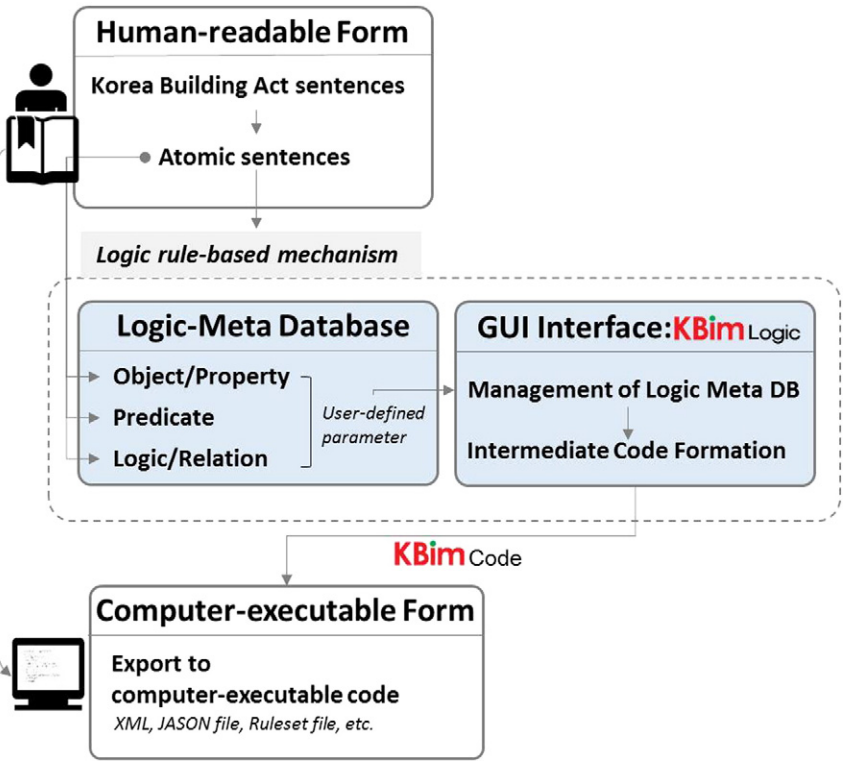

Fig. 4. Moving from Korean Building Act sentences to an explicit, computer-executable form (KBimCode) via the KBimLogic application and the KBimCode database.

The rule-based system for design review helps architects effectively validate building designs. However, current rule-checking software usually requires knowledge of computer programming techniques in order to make new types of rules to meet architects' demands. That is, the current rule-making process is more rule-checking and software-centered than platform-independent. Making computable rules independent of specific proprietary software is critical to users. For the building permit system in Korea or elsewhere, it is also critical that given code compliance rules be up-to-date. Moreover, some building permit-related requirements vary by building type, administrative district, permitted date, and as-built date. Reusable rules are necessary because it is inefficient to write such rules for every project. Accordingly, this study is motivated by such fundamentals and suggests a logic rule-based mechanism for use by non-programmers and a user-friendly approach to the rule-making process based on the KBimCode database and its authoring tool, KBimLogic. This study also demonstrates ongoing development of the design rule-checking software KBimAssess.

3. Rule-making approaches

The software for automated design rule checks is implemented primarily by experts from the two different domains of software development and architecture as an integrated process for rule-making and rule-checking [21]. This section provides an overview of the current state of rule-checking software based on built-in rule sets and methods similar to the proposed approach but less user-friendly. In addition, this section describes a building object-oriented approach to code-compliance checks rather than an alternative geometry or lower-level rule set-oriented approach. The SMC and EDM are two example systems that provide customized rule-making approaches [16].

3.1. SMC approach

For developer-based coordination of rule sets, SMC [14] is one of the most well-known programs for BIM-enabled design assessment. It includes various functions, such as model/name convention pre-checking and object existence/path checking [22]. In SMC, the user can configure rules using a parametric table structure. SMC and its rules are based on JAVA, indicating the need for JAVA programmers if users require new rules.

Most of the rule sets in SMC are hard-coded into the software, and it is difficult to specify new types of rules [23]. Users must combine existing rule sets and configure each parameter of the rules themselves, as shown in Fig. 3; otherwise, SMC developers must add new rule sets to the software. Moreover, not all rule sets built into SMC can be adopted in other rule-checking software.

3.2. EDM and the EXPRESS approach

The EDM server [24] operates with an object database and requires EXPRESS and EXPRESS-X, which are object-oriented query languages. It executes model and rule validation according to schema and exchange requirements. It can facilitate access to the semantics of IFC, ifcXML, and IDM using open standards. EXPRESS is a data modeling language defined in ISO10303–11 as an international standard, so it supports the open development of rule checks [22]. Rules are written in this format, and they can be used in any software that understands EXPRESS-X. Similar to SMC, the EDM server consists of a set of built-in rules. The EDM server supports the flexible encoding of wide domain-specific knowledge [16]; however, a high level of expertise is required to modify the rules in the EDM server. To develop new types of rules, EDM experts must develop the necessary rules using EXPRESS.

3.3. Building object-oriented approach to generating computer-readable forms

Given the limitations of the current rule-making approaches explained above, we suggest a building object-oriented approach that is comparatively easy-to-use and independent of proprietary software. Design-related regulations, including the Korean Building Act, should be interpreted from human-readable sentences into computer-

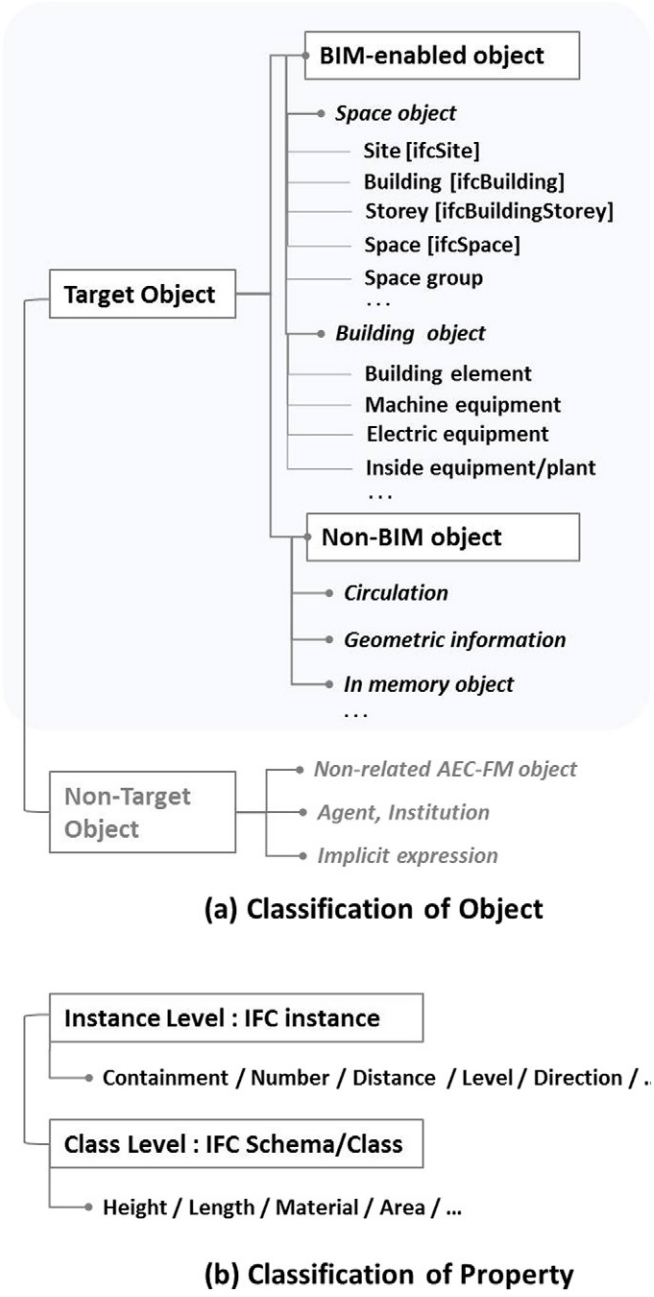

Fig. 5. An example of classification by object/property in the Korean Building Act.

implementable rules for automated checks [11]. Building Act sentences cause particular difficulty for rule checks that use a combination of rule sets and parameters because of their complex referred relations within sentences and occasional use of subjective expressions. We therefore focus on the "building object" defined in the Korean Building Act and suggest a logic rule-based mechanism for creating a rule set file from natural language sentences. As described in Fig. 4, human-readable language from the Korean Building Act can be the raw data for a logical process as it is translated into a rule set file using a logic rule-based mechanism.

Each building object and its associated properties in the Building Act can be handled intuitively by a user who is unfamiliar with computer programming. In addition, the user can manage the predicates of the sentences by interpreting the articles' purposes. The human-related and building object-oriented parts are included within the logic meta-database in the form of a table. The database contents of object/property, predicate, and logic can be managed in the KBimLogic GUI interface. In the KBimLogic environment, users can define, manage, and modify each content item according to the intent of the Korean Building Act. The result of the KBimLogic application is the generation of a type of 'rule set file' for use in rule-checking software.

4. Logic rule-based mechanism for natural language sentences in the Korean Building Act

Natural language technically includes infinite expressions and phrase lengths; thus, ambiguity and vagueness are inevitable issues in any translation approach [22]. Translating human-readable language into a computable format therefore requires a logical, reliable rule-making process. In other words, for automated design assessment with a BIM model, it is important to translate sentences into a computer-executable format using a restructuring process. In this section, we describe the proposed three-part logic rule-based mechanism in terms of sentence structure: (1) noun phrases (object/property) (See Fig. 5), (2) verb phrases (predicate; high-level method), and (3) logic (code generation).

4.1. Noun phrases: building objects and properties

In the sentences of the Korean Building Act, most nouns are used in the nominative case to describe building objects and their properties. Most often, the definition of the noun is connected by other clauses that relate it to content.

Therefore, the definition of a noun considers the connectivity of relevant regulations and other clauses, although exceptional items exist. The importance of providing a name for each object and setting up a relationship has been demonstrated in a previous study [25]. Moreover, an English name is required as a name set (Korean–English) for the computable form because there are currently no Korean-supported programming languages. This database contains an ID, names in Korean and English, classification type, definition, and Act clause reference ID for each noun.

4.2. Verb phrases (predicates): classification of rules

The verb phrase (predicate) in each sentence specifies the condition and associated action for objects and their properties, providing the salient content of the rule. Therefore, this part is intimately connected with how the building code is translated into a computer-implementable format. We approach this issue with a high-level method-based representation. Using this approach, the sentence predicate is the function, and the building objects and their properties are the argument. The main issue in our rule-checking method is querying the object and then verifying its properties in a given building model.

The proposed methods are high-level types derived from the predicates of the target sentences to represent the components of the sentences in an implementable format. The method name and parameter are mapped directly to the predicate and objects/properties of the sentence. From its shape, users can intuitively infer the function of the method. Combinations of low-level methods and algorithms, along with mapping mechanisms between regulation-specific objects/properties and those in standards such as the IFC, are required to successfully implement the methods.

As shown in Fig. 6, a high-level method can be applied as an intermediary in the mapping between sentence components and a BIM model's objects and properties. The sentence components are restructured into a method-based representation and then executed to return specific numeric values, collections of objects, or Booleans from the given building model. Because the result of executing a method gives only a return value and not a determination of model conformance, we borrow the concept of an arithmetic logic unit (ALU) [26], taken as an atomic unit of rule checking that contains explicit and basic left operand, right

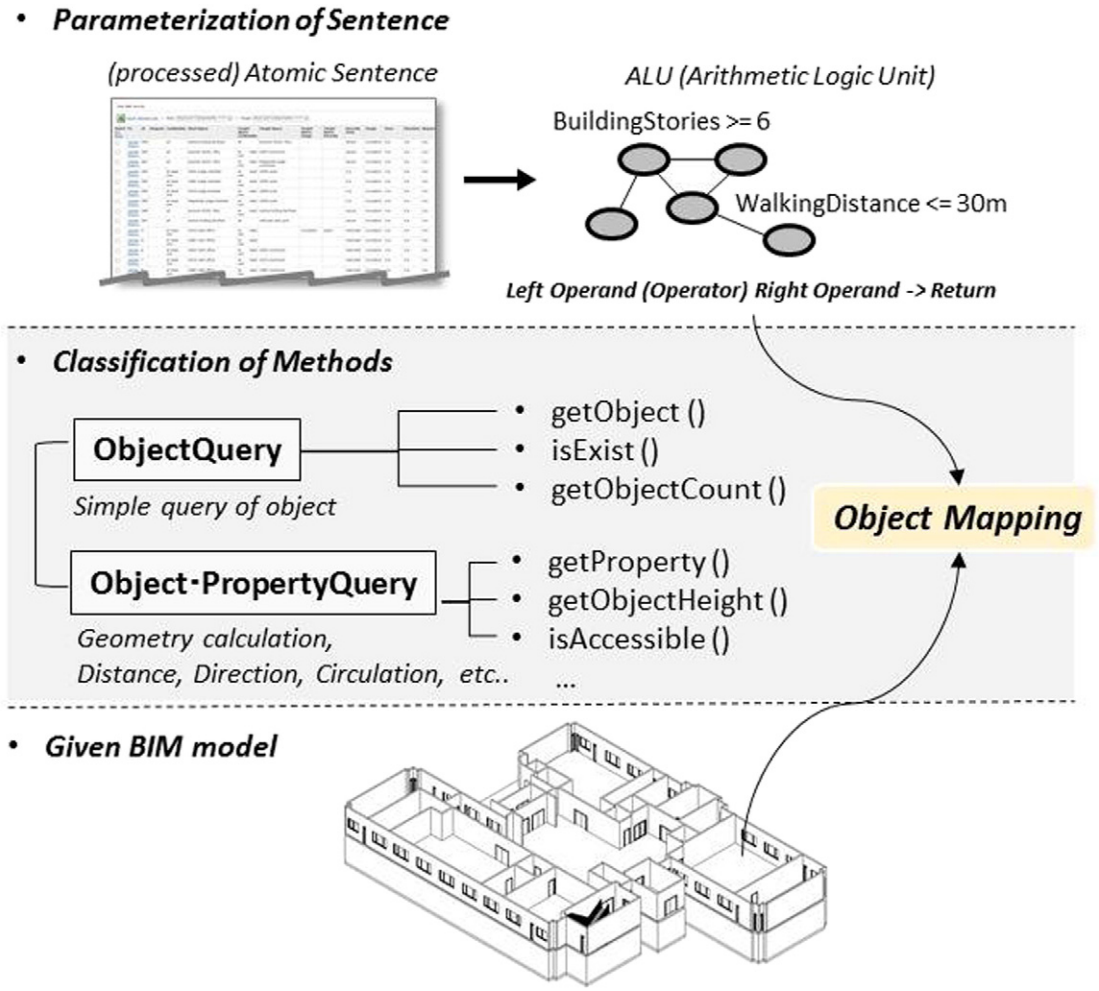

Fig. 6. Association of a high-level method with ALU and BIM model instances.

operand, value, operator, result, etc. It compares the return values with constraints stated in the original sentence (as discussed in detail in the following section). We classified the high-level methods (Table 1) into two groups according to the type of target instance: (1) object query—the return value can be obtained with the existence of the object, and (2) object-property query—the result of this method requires both the object and the property. The second stage of classification is based on the property types, which we categorize into four groups: (1) basic property, (2) geometry-related property, (3) complex property, and (4) relational property.

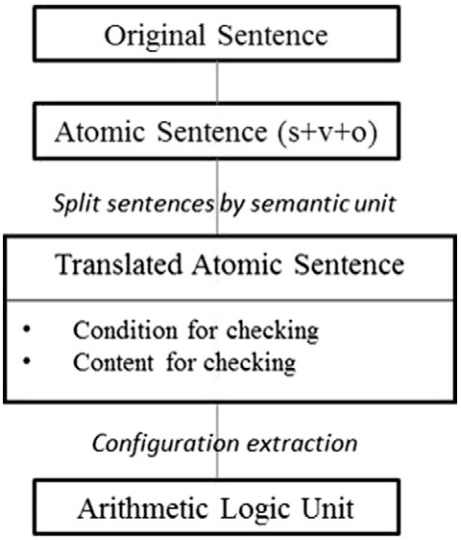

Fig. 7. KBimCode generation process: from an original sentence to atomic ALUs.

Complex properties denote regulation-specific properties whose definitions are stated in the building code. The third stage involves specific types derived from the previous stages. Each type is specified with a representative method, which forms the fourth stage. The representative methods are subdivided according to the objects and their properties. From the target sentences, a total of 60 different methods were derived. As the target sentences expand to all building permit-related codes in South Korea, new types of predicates, objects, and properties will appear. Consequently, the number of methods and method classifications will expand. At this stage, representative rule methods are defined according to the property's attributes. The methods can be expanded by various parameters and eventually be implemented as general rule-checking tools using complex, low-level methods. Development of the method described in this section will be discussed in further detail [27].

4.3. Logic: from sentences to executables

Typically, a single Korean Building Act sentence contains one condition clause and one content clause. In this case, the target building

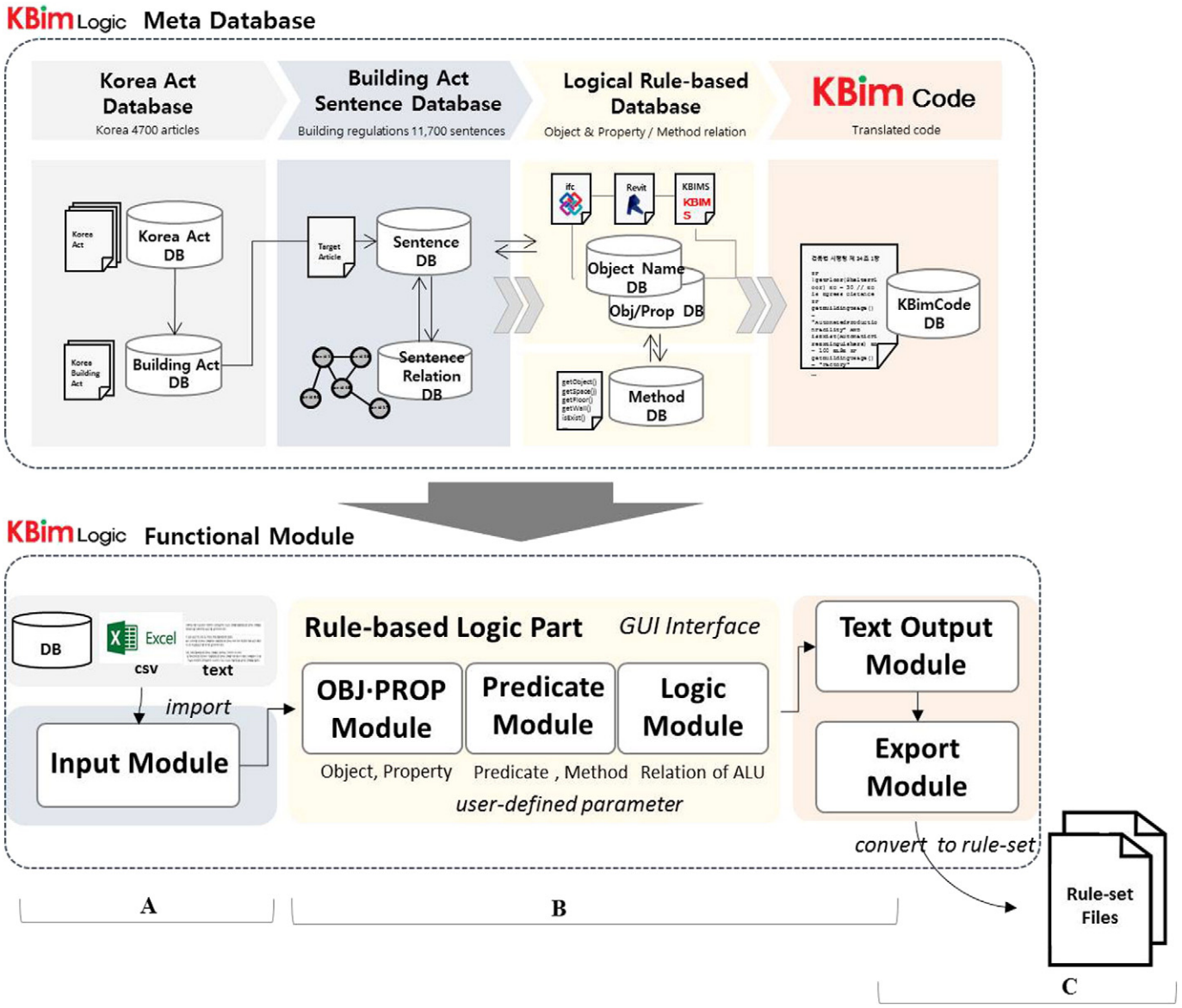

Fig. 8. Overview of the KBimLogic application with associated functional modules. (A) Input of human-readable language (original sentences from the Korean Building Act). (B) Logic rule-based mechanism for generating KBimCode. (C) Actual executable files.

model should satisfy the conditional clause before the content clause is checked. Each clause is generally composed of a noun (subject and object) and a verb (predicate). In the logic rule-based criteria, an atomic sentence is a type of declarative sentence that is either true or false and cannot be broken down into other simpler sentences [28]. In other words, an atomic sentence can be expressed in a single S (subject) + V (verb) + O (object) structure. We thus convert natural language sentences to ALU form (independent syntactical units). Restructured ALUs will be combined with IF/THEN/ELSE and other necessary operators to logically represent the meaning of the original sentence. We separate each atomic sentence's "check condition" and "check content" based on semantics. Fig. 7 describes the logic rule-based KBimCode generation process for natural language sentences.

As an easy-to-use example case of this logic process, we chose the Korean Building Act Article 64, (1) clause, which pertains to the conditions for elevator installation. Table 2 describes each step in the process.

Translating the natural language sentence is a necessary (if tedious) process for obtaining a trustworthy computable form. This is currently a manual process, but the eventual goal is automated sentence translation.

5. Implementation of the KBimLogic application

This section provides an overview of the implementation of the GUI application (KBimLogic) used to generate the computer-executable format (KBimCode). KBimLogic is responsible for the translation process, with direct links between the original code sentences and a series of parings in the KBimCode. Fig. 8 shows an overview of the KBimLogic application and its functional modules, which customize the logic rule-based process explained in Section 4.

5.1. KBimLogic meta-database

As shown in Fig. 8, the KBimLogic application is composed of a meta-database derived from the Korean Building Act and the logical rule-based mechanism. Based on the logical mechanism described in Section 4, the following meta-databases were implemented: (1) Korean Building Act database, (2) Korean Building Act sentence database, (3) logical rule-based database, and (4) KBimCode database. To develop the project beyond experimental status into a sustainable, comprehensive management tool, the overall system will continue to be refined

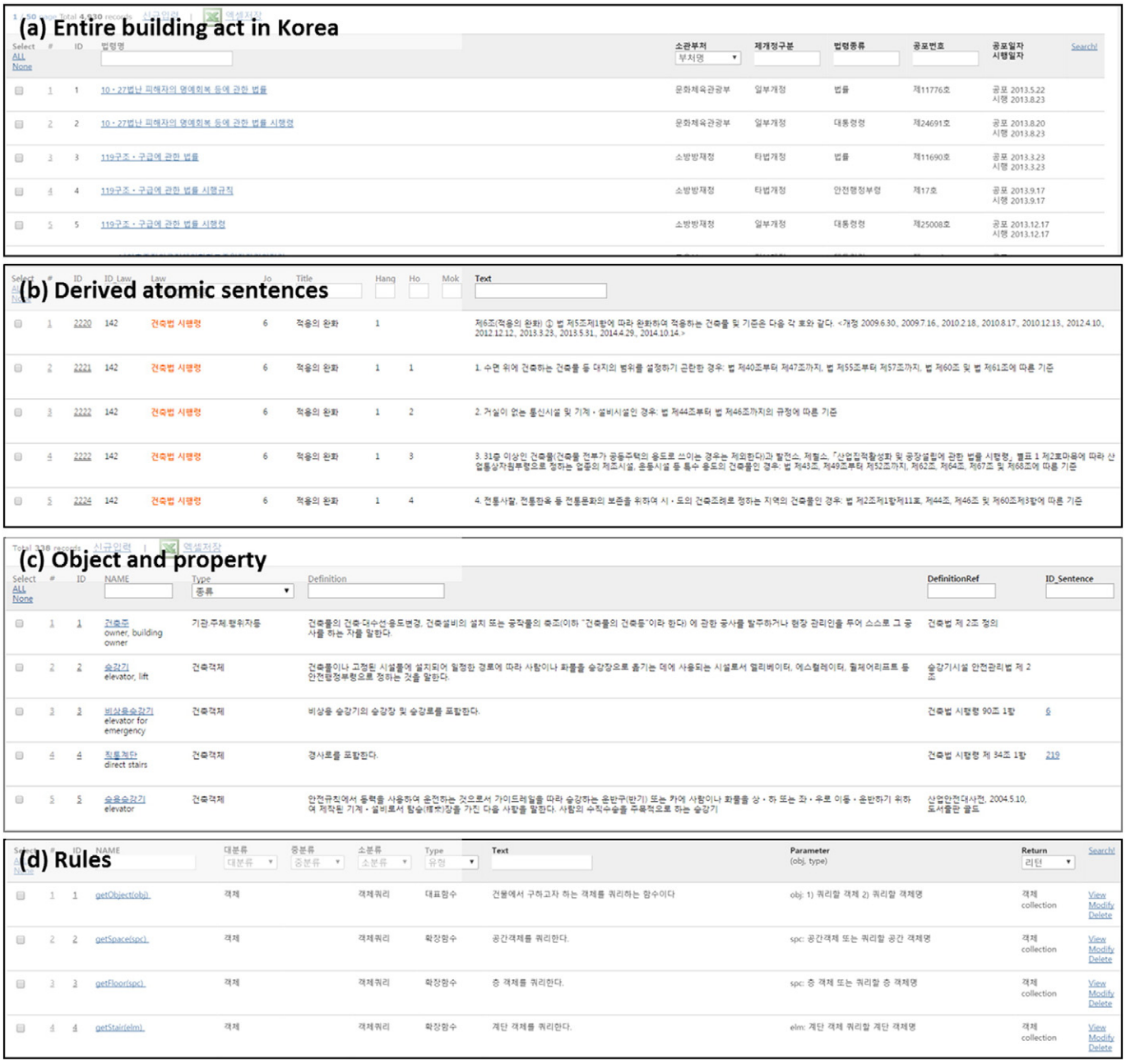

Fig. 9. Images from the KBimLogic meta-databases.

and provided constantly renewed code data. Each meta-database is developed in a web environment [5], as described below.

5.1.1. Korean Building Act database

We imported all Korean Building Act data from the National Law Information Center (law.go.kr) [4] to our own KBimLogic meta-database as the resource for Korean Building Act sentences. Each article of the Korean Building Act is widely related to other Korean Acts; therefore, 4390 types of articles were stored in order to retrieve Korean Building Act-related content. Fig. 9(a) shows a portion of this database, which contains the entire Korean Building Act.

5.1.2. Korean Building Act sentence database

There are 15,260 atomic sentences derived from the previous database, which are managed in another web-based database. Each atomic sentence has its own ID (identity), along with its reference Act clause. Because complex delegation and reference relationships among sentences are inherent in the Korean building code, clarifying those relationships is crucial for accurate rule checks. Using a web-based interface, the atomic sentence database allows users to track related sentences. Fig. 9(b) shows a portion of the database of atomic sentences derived from the Korean Building Act.

5.1.3. Logic rule-based database

Based on the proposed logical rule-based mechanism, a noun and predicate database was developed. First, we extracted nouns from sentences and connected them with a reference clause from the Korean Building Act, as shown in Fig. 9(c). Those nouns refer to building objects and the properties considered for code compliance. We set up a relationship between connected nouns and entered the appropriate IDs. Using the rule classifications from the sentence predicates, we developed a method database with type and definition, as shown in Fig. 9 (d).

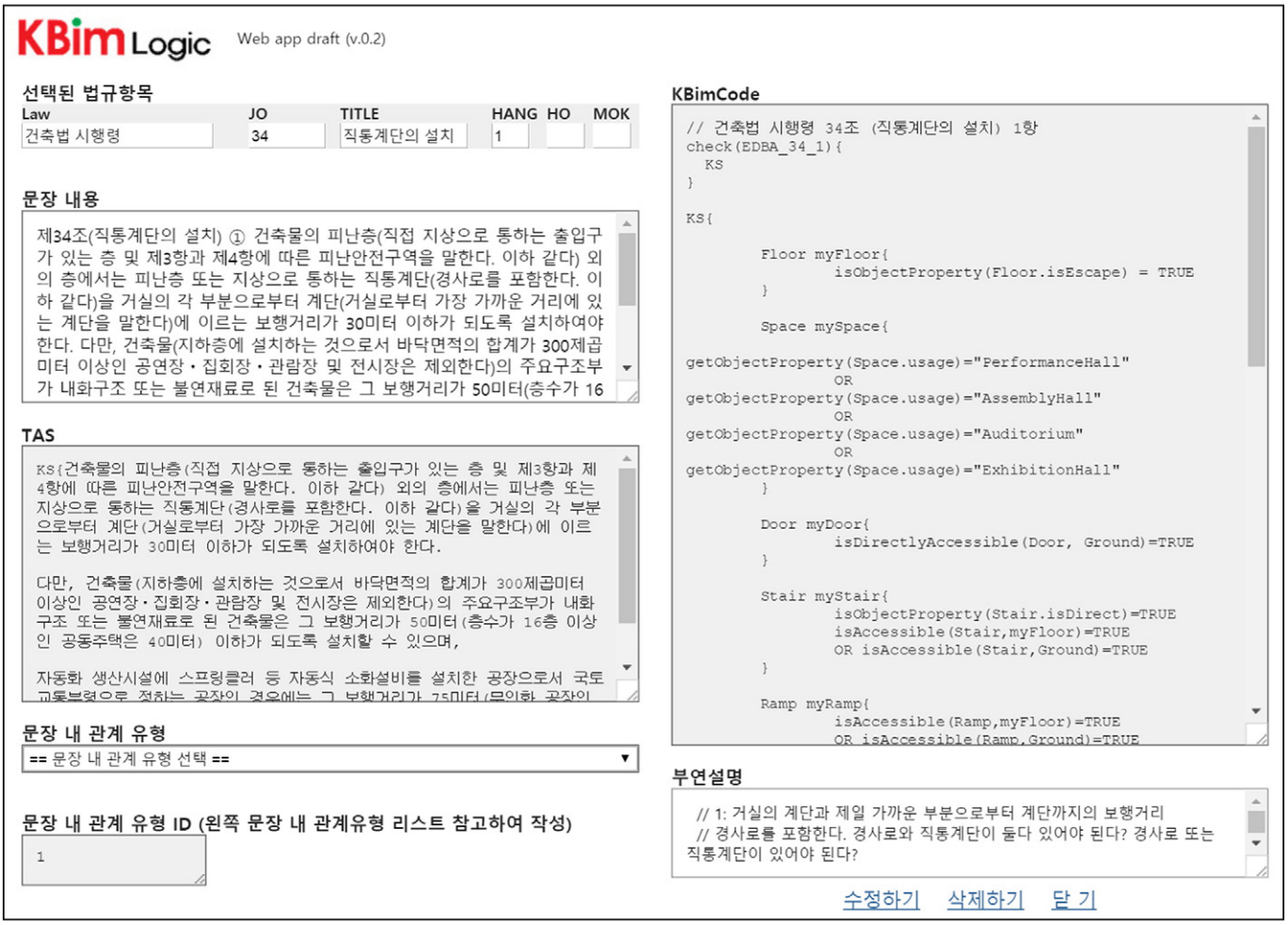

Fig. 10. An example web interface for KBimLogic: An editor of KBimCode can translate and modify each line of KBimCode using its database.

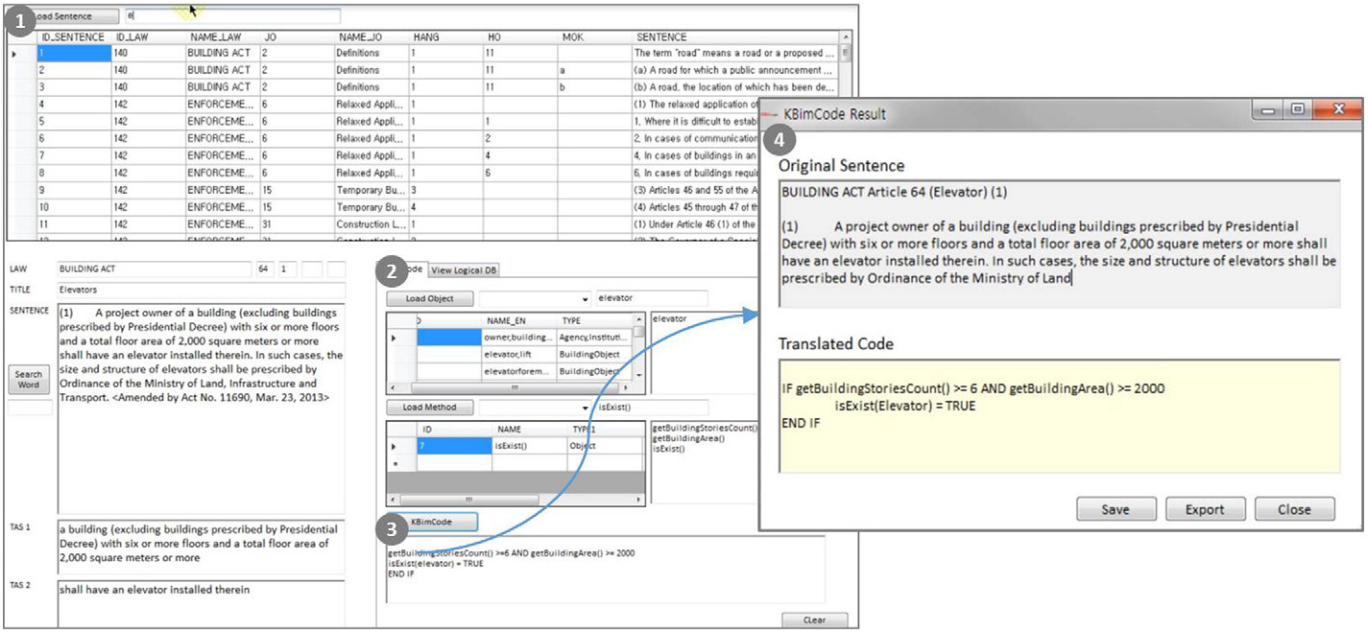

Fig. 11. Sample images from KBimLogic (an early draft version); the following numbered modules in the figure indicate (1) input module; (2) OBJ·PROP module, predicate module, and logic module; (3) text output module; and (4) export module.

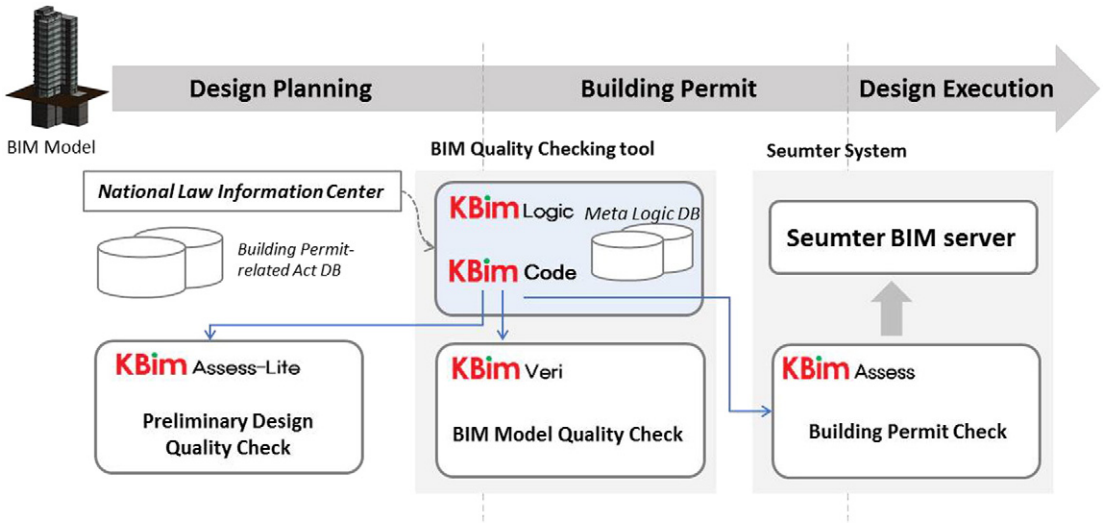

Fig. 12. A simplified overview of the KBim applications in research and development within Seumter, an e-submission and building permit system of the Korean government.

5.1.4. KBimCode database

Using each meta-database, a user can generate KBimCode, as shown in Fig. 10. The left side of the interface shows the original Korean Building Act article from the sentence database, and the right side describes those sentences translated into KBimCode via the logic rule-based mechanism, a related meta-database, and the KBimLogic application. The KBimCode database is also connected with its related article ID; therefore, each KBimCode entry is reusable and can be updated along with the Korean Building Act.

5.2. KBimLogic functional modules

5.2.1. Input module (human-readable sentences)

The input module imports and manages natural language sentences in the form of a database, csv, or text file. In this module, we link the web-based database and data source to ensure that we have the most up-to-date sentences, which are maintained through the public data portal of the Korean government. Atomic sentences in the Korean Building Act that a user wants to translate can be imported using this module.

5.2.2. KBimLogic module (to generate intermediate language)

1) OBJ·PROP Module: The object and property module imports the objects and properties from the user-selected clause in the input module. In addition, the user can choose or enter sentences for specific purposes. This module plays a critical role in object mapping between the original law sentences and the actual IFC models.
2) Predicate Module: In this module, the user can select the appropriate high-level method according to the condition and content of the sentences. This module cooperates with the OBJ·PROP module to derive the objects and properties that define the method parameters.
3) Logic Module: Based on the type of method selected in the predicate module, the user can define the type of relation between the condition and content of a clause.

5.2.3. Text output module (to organize KBimCode)

This module organizes the KBimCode output and manages sets of already defined KBimCode entries as a reusable code database. This is a specific type of repository that provides executable files to various rule-checking applications.

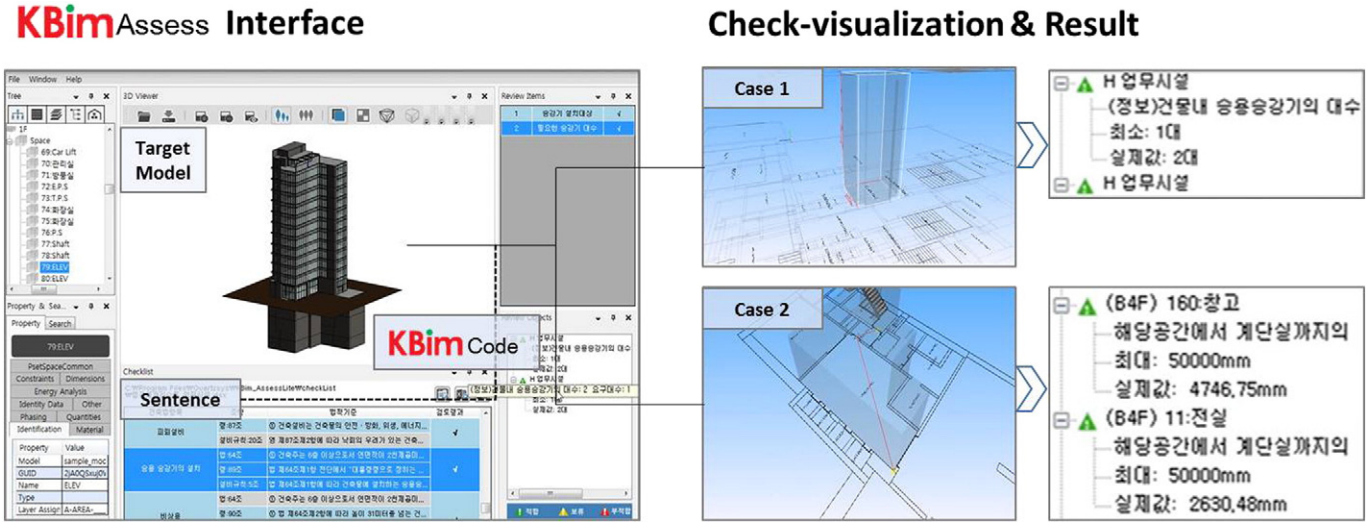

Fig. 13. Example screenshots from the KBimAssess software that demonstrates a check for code compliance with the Korean Building Act (Case 1: "Act Article 64, (1), Elevator" and Case 2: "Article 34, (1), Installation of direct stairs") based on the appropriate KBimCode.

Table 1
Example of high-level classification methods for representing verb phrases in the Korean Building Act.

Instance	Property type	Property character	Specified method
Obj query		Object query	getObject()
		Existence	isExist()
		Number	getObjectCount()
Obj, Prop query	Basic property	Property	getProperty()
		Material	getMaterial()
		Usage	getObjectUsage()
	Geometry	Height	getObjectHeight()
		Length	getObjectLength()
		Area	getObjectArea()
		Gradient	getObjectGradient()
	Complex	Material type	getMaterialType()
		Illumination	getSpaceIlluminance()
		Structure	getObjectStructure()
		Fire resistant	isFireResistant()
		Fireproof	isFireProof()
		Fire compartment	isFireCompartment()
	Relation	Inclusion	hasObject()
		Distance	getObjectDistance()
		Connection	isConnectedTo()
		Circulation	isAccessible()
		Direction	getDoorDirection()

5.2.4. Export module (to export executable files)

This module exports files converted in the text output module into a computer-executable rule-set file. The form of the rule set can be XML, a script language such as BERA [30], or any other extension because a rule-maker is not dependent on proprietary software. As an example, we demonstrate a set of KBimCode in the JSON file format in this paper.

Fig. 11 is a sample image of the implemented KBimLogic application, which is composed of the logic rule-based functional modules described in Fig. 8. KBimLogic is a dot net based stand-alone application that uses an SQL server to retrieve the KBimLogic meta-databases. Users can browse and choose the Korean Building Act clause for translation (as shown in circle number 1 of Fig. 11). Users also intuitively select objects, properties, and high-level methods. In other words, users can restructure a sentence with a combination of components from the logic meta-database. Consequently, each selected option is transformed into atomic KBimCode that is reviewed as a series of scripts that can be exported as actual executable files. The KBimLogic application is an interface application for accessing the logic meta-database, a particular implementation of the approach described in this paper. If users have specific building permit-related issues and the corresponding laws or their article numbers in the Korean Building Act, they can browse the law articles and determine whether to reuse existing KBimCode or amend it for the given conditions.

Table 2
KBimCode generation process for the Korean Building Act, Article 64, (1), Elevators.

1. Original sentence
A project owner of a building (excluding buildings prescribed by Presidential Decree) with six or more floors and a total floor area of 2000 square meters or more shall have an elevator installed therein. In such cases, the size and structure of elevators shall be prescribed by the Ordinance of the Ministry of Land, Infrastructure, and Transport. (Building Act, Article 64, (1), Elevators) [29].

2. Atomic sentence
A building with six or more floors and a total floor area of 2000 square meters or more shall have an elevator installed therein.

3. Translated atomic sentence (TAS)

TAS 1 (Condition)	A building has six or more floors and a total floor area of 2000 square meters or more
TAS 2 (Content)	A building shall have an elevator installed therein.

4. Configuration extraction from TAS

TAS 1 (Condition)	Number of Floors (more than 6) Total Floor Area (more than 2000 m^2)
TAS 2 (Content)	Install (Elevator)

5. Arithmetic logic unit (ALU)
getBuildingFloor() ≥ 6,
getTotalFloorArea() ≥ 2000 m^2
isExist(Elevator)

6. Expression of method and relation

```
IF (getBuildingStoriesCount() >=6
   AND getBuildingArea() >=2000)
       isExist (Elevator) = TRUE
END IF
```

6. Demonstration

This chapter provides a demonstration of how to apply logic rule-based mechanisms and related meta-database with KBimLogic and KBimCode to verify actual Korean Building Act compliance.

6.1. KBimLogic and KBimCode application for the Korean government e-submission system

Seumter is the building administration system run by the Korean Ministry of Land, Transport, and Maritime Affairs [31,32] as one part of the electronic government system. The Korean government has attempted to construct a web-based architecture administrative system that requires submission of 2D architectural plan files. Seumter provides services related to building licenses and maintenance to enable the early establishment of BIM and its applications [33]. Seumter allows applicants to apply for building permits through its e-submission system, and it currently supports IFC as a test phase. In addition, building permit administrators can electronically manage the national standards information system. The Seumter system is now in a transitional stage between conventional 2D CAD and BIM. The Korea government has planned for an IFC-based e-submission system that is currently in an operational test stage.

For further development, a fundamental code-compliance check system will be soon be loaded into Seumter. Additionally, in terms of national BIM infrastructure improvement, a research project (Open BIM-based Technological Environment for Building Design Quality Enhancement) [34] is being carried out by multiple research teams. That project focuses on standardizing code, data formats, standards, and functional requirements for building design quality improvement using BIM. This section shows a brief demonstration of an outcome from a national BIM project in Korea using KBimLogic and KBimCode within the environment of our code-compliance check software (KBimAssess). This outcome is intended to be an application for approval through Seumter.

Fig. 12 briefly outlines a series of KBim applications currently in research and development within the Seumter system. KBimLogic and KBimCode, described in this paper, are intended to provide technical bases of the BIM-enabled building permit system. KBimLogic deals with all building permit-related code using its logic rule-based mechanisms to create KBimCode from the regulations. Exported KBimCode can be used in KBim Assess-lite for design quality checks in the design planning stage. For the assessment of building permits, the target BIM model is processed through KBimVeri, which checks the quality of the BIM model for submission to the Seumter system. In the Seumter system, the BIM server includes KBimAssess to verify the conditions of target building models for building permit consideration. KBimCode translated using KBimLogic can be exported to KBimAssess-lite, KBimVeri, and KBimAssess for rule-checking of the target BIM model.

6.2. Demonstration using actual Korean Building Act sentences

Fig. 13 shows KBimAssess being used to check Korean Building Act sentences via a set of KBimCode entries and the results. The target model is an office building in Seoul, Korea, with a total floor area of

5734 m^2, 4 floors underground, and 15 floors above ground. We chose suitable articles from the Korean Building Act for code checking: Case 1: verify the existence of an elevator in the given model according to the building condition, as shown in Table 2. Case 2: verify the existence of direct stairs and the distance between stairs on evacuation floors and living rooms on non-evacuation floors.

6.2.1. Case 1 (original sentence): Article 64 (Elevators), Korean Building Act

A project owner of a building (excluding buildings prescribed by Presidential Decree) with six or more floors and a total floor area of 2000 square meters or more shall have an elevator installed therein. In such cases, the size and structure of elevators shall be prescribed by Ordinance of the Ministry of Land, Infrastructure and Transport.

6.2.2. Case 2 (original sentence): Article 34 (Installation of Stairs), Enforcement Decree of the Korean Building Act

On each floor of a building, direct stairs leading to the shelter floor or the ground (including slope ways; hereinafter, the same shall apply) other than the shelter floor (referring to a floor having a doorway leading directly to the ground and the shelter safety zone of a skyscraper under paragraphs (3) and (4); hereinafter, the same shall apply) shall be installed in the way that the walking distance from each part of the living room to the stairs (referring to the stair nearest to the living room) is not more than 30 m.

Both of those original sentences from the Korean Building Act sentence database should be divided into TAS1 (condition) and TAS2 (content) for translation. Through the logical meta-database, we manually translated the Korean Building Act sentences into KBimCode using the KBimLogic application. We then used the KBimCode generated from the KBimLogic software, as shown below, in the check process using the implemented KBimAssess software. The KBimCode examples for each case are described in a simplified format below.

Case 1. KBimCode

```
IF (getBuildingStoriesCount() >=6
    AND getBuildingArea() >=2000)
        isExist (Elevator) = TRUE
END IF
```

Case 2. KBimCode

```
IF (!getFloor(ShelterFloor))
    ED = 30  // Egress Distance will be determined by given condition
    getSpaceDistance(LivingRoom, Stairs, MRP) <= ED
END IF
```

The KBimAssess program using KBimCode is an ongoing development project that will eventually be officially incorporated into the national system (Seumter) for building permits. The demonstration shown in Fig. 13 does not represent the final development outcome. Additionally, the logical rule-based mechanism and its database will be elaborated to cover all building-related sentences in order to ensure the accuracy and integrity of KBimCode.

7. Conclusions and future research

This study introduced a logic rule-based mechanism for translating the natural language text of the Korean Building Act into a computable form and demonstrated its implementation as part of an ongoing project funded by the Korean government. We suggest classifications of building objects mentioned in the Korean Building Act, derived properties, and high-level methods for processing the text. Distinguishing features of this approach can be summarized as follows: (1) We carefully reviewed more than 15,000 law sentences in the Korean Building Act. (2) We then filtered and classified them according to the mechanism described in this paper in order to generate KBimCode. (3) The logic meta-database and its implementation can be separately managed by the relevant domain experts. That is, the task of establishing KBimCode using KBimLogic is independent of any specific proprietary software on the market. In this paper, we also provided an overview of an actual government project to verify the applicability of KBimLogic and KBimCode with other software for building-permit assessment. For demonstration and implementation, we provided an example of the Korean Building Act code and translated it into KBimCode using the rule-checking software KBimAssess to evaluate a target building model.

The research and development described herein, including the before and after sequences of the rule-making approach, is an ongoing project as of 2015, one of Korea's government-led efforts to enhance the overall quality of the built environment. It is important to secure the integrity of flawless logical processing for the text of the Korean Building Act because this project is intended to carry legally binding weight in the future. Our structured logic rule-based mechanism shall be further improved and refined through continuous and intensive research and development. To validate this approach with up-to-date regulations of the building permit system, a test server system has been established inside the Korean government network. Our work described here could significantly benefit the development of automated design assessment tools that meet a wide range of design assessment requirements, not only those in national legislation. This paper describes only a small part of a larger picture of BIM use. Other significant BIM advances are also being developed for publication.

Acknowledgment

This research was supported by a grant (15AUDP-C067809-03) from the Architecture and Urban Development Research Program funded by the Ministry of Land, Infrastructure, and Transport of the Korean government.

References

[1] C.M. Eastman, J. Lee, Y.-s. Jeong, J.-K. Lee, Automatic rule-based checking of building designs, Autom. Constr. 18 (2009) (2009) 1011–1033.
[2] C.M. Eastman, P. Teicholz, R. Sacks, K. Liston, BIM Handbook—A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Constructors, John Wiley & Sons, Inc., New Jersey, 2007 5.2–5.3 (chapter1.6).
[3] J.-K. Lee, C.M. Eastman, Y. Lee, Implementation of a BIM domain-specific language for the building environment rule and analysis, J. Intell. Robot. Syst. (2014), http://dx.doi.org/10.1007/s10846-014-0117-7.
[4] Korean Building Act in National Legislation CenterOn line: http://law.go.kr/lsOrdinAstSc.do?menuId=9&p1=&subMenu=1&nwYn=1&section=&tabNo=10&query=%EA%B1%B4%EC%B6%95%EB%B2%95&x=0&y=0# (Accessed: 01/01/2015).
[5] Open BIM based technological environment for building design quality enhancement. On-line: www.designitlab.kr/bim (Accessed: 22/01/2015).
[6] T.F. Sing, Q. Zhong, Construction and real estate NETwork (CORENET), Facilities 19 (2001) 419–428.
[7] A.L. Teo, T.F. Cheng, A strategy for implementing BIM solution in Singapore, Synth. J. 5 (2006) 117–124 (Building smart).
[8] CORENET, On-Line: http://www.corenet.gov.sg (Accessed: 09/12/2014).
[9] J. Choi, I. Kim, An approach to share architectural drawing information and document information for automated code checking system, Tsinghua Sci. Technol. 13 (S1) (2008) 173–178.
[10] CORENET, E-PlanCheck: Singapore's automated code checking system, AECbytes "Building the Future" Article 2005, pp. 1–8.
[11] S. Malsane, J. Matthews, S. Lockley, P.E.D. Love, D. Greenwood, Development of an object model for automated compliance checking, Autom. Constr. 49 (2015) (2015) 51–58.
[12] Use of OpenBIM Denmark Norway FinlandOn-line: http://bips.dk/files/article_files/9a_2010.pdf (Accessed: 01/03/2016).
[13] Nosyko AS's dRofus user manual, dRofusOn-Line: http://www.drofus.no (Accessed: 28/12/2014).
[14] SMC: Solibri model checker, SolibriOn-Line: http://www.solibri.com (Accessed: 15/11/2014).
[15] L. Ding, R. Drogemuller, M. Rosenman, D. Marchant, J. Gero, Automating code checking for building designs, Clients Driving Construction Innovation: Moving Ideas into Practice, Cooperative Research Centre (CRC) for Construction Innovation, Brisbane, Ausralia 2006, pp. 1–16 (ISBN: 1741071283).
[16] L. Ding, R. Drogemuller, J. Jupp, M. Rosenman, J. Gero, Automated Code Checking, Clients Driving Innovation International Conference, CRC for Construction Innovation, Surfers Paradise, Australia 2004, pp. 1–17 (ISBN 0–9750977–3-3).
[17] J. Wix, N. Nisbet, T. Liebich, Using constraints to validate and check building information models, EWork and EBusiness in Architecture, Engineering and Construction, ECPPM, London 2008, pp. 467–475.

[18] ICC, Smart Code Project DefinitionOn-line: http://web.stanford.edu/group/narratives/classes/0809/CEE215/ReferenceLibrary/International%20Code%20Council%20%28ICC%29/SMARTcodes%20fact%20sheet%2011-01-07.pdf (Accessed: 10/01/2015).
[19] The National 3D-4D-BIM Program Office of the Chief Architect Public Buildings Service U.S. General Services Administration, "BIM Guide Series 02 v096", General Services AdministrationOn-line: http://www.gsa.gov/bim (Accessed: 28/12/2014).
[20] GSA project: BIM enabled design guide automationOn-Line: http://dcom.arch.gatech.edu/gsa/ (Accessed: 20/01/2015).
[21] E. Hjelseth, Foundation for development of computable rules, Managing It in Construction/Managing Construction for Tomorrow 2009, pp. 1–10.
[22] N.O. Nawari, Automated code checking in BIM environment, 14th International Conference on Computing in Civil and Buildng Engineering, Moscow, Russia, 2012.
[23] H. Bell, L. Bjørkhaug, E. Hjelseth, Standardized Computable Rules, Standards Norway, Strandveien 18 P.O.Box 242, NO-1326 Lysaker, Norway, 2009.
[24] EDMmodelServer(ifc)™One-line: http://www.epmtech.jotne.com/edmmodelserver-ifc (Accessed: 29/11/2014).
[25] J.-K. Lee, J. Lee, Y.-s. Jeong, H. Sheward, P. Sanguinetti, S. Abdelmohsen, C.M. Eastman, Development of space database for automated building design review systems, Autom. Constr. 24 (2012) (2012) 203–212.
[26] Definition of arithmetic logic unit (ALU)On-line: http://techterms.com/definition/alu (Accessed 10/03/2015).
[27] S. Park, J.-K. Lee, J. Choi, I. Kim, G. Lee, Development of Implementable High-Level Methods for the Automated Checking of Korea Building Permit Requirements, 2016 (In Review).
[28] Definition of Atomic SentenceOn-line: http://en.wikipedia.org/wiki/Atomic_sentence (Accessed: 14/01/2015).
[29] Statutes of the Republic of KoreaOn-line: http://elaw.klri.re.kr (Accessed: 20/01/2015).
[30] J.-K. Lee, C.M. Eastman, Y. Lee, Implementation of a BIM domain-specific language for the building environment rule and analysis, J. Intell. Robot. Syst. 79 (3) (2015) 507–522.
[31] Korea building administration system, SeumterOn-Line: http://www.eais.go.kr (Accessed: 13/01/2015).
[32] Korea Ministry of Land, Transport and Maritime Affairs, On-Line: http://www.molit.go.kr/portal.do (Accessed: 13/01/2015).
[33] The vision of BIM and Seumter (e-AIS)On-Line: http://webcache.googleusercontent.com/search?q=cache:tH-loBUZM3oJ:cfs7.tistory.com/upload_control/download.blog%3Ffhandle%3DYmxvZzQ4Mzc5QGZzNy50aXN0b3J5LmNvbTovYXR0YWNoLzAvMzEwMDAwMDAwMDA0LnBkZg%3D%3D%26filename%3DBIM(e-AIS)_122-125.pdf+&cd=1&hl=ko&ct=clnk (Accessed: 20/11/2015).
[34] Open BIM based technological environment for building design quality enhancementOn-Line: http://www.ikld.kr/news/articleView.html?idxno=36503 (Accessed: 13/11/2015).