

## 서울부의 함수체계화를 통한 인허가관련 건축법규의 자동검토 응용방안

박서경<sup>1</sup> · 이진국<sup>1†</sup> · 김인한<sup>2</sup>

<sup>1</sup>한양대학교 실내건축디자인학과, <sup>2</sup>경희대학교 건축학과

### Development of High-level Method for Representing Explicit Verb Phrases of Building Code Sentences for the Automated Building Permit System of Korea

Seokyoung Park<sup>1</sup>, Jin-Kook Lee<sup>1†</sup>, and Inhan Kim<sup>2</sup>

<sup>1</sup>Dept. of Interior Architecture Design, Hanyang Univ.

<sup>2</sup>Dept. of Architecture, Kyung Hee Univ.

Received 12 February 2016; received in revised form 18 April 2016; accepted 25 April 2016

#### ABSTRACT

As building information modeling (BIM) is expanding its influence in various fields of architecture, engineering, construction and facility management (AEC-FM) industry, BIM-based automated code compliance checking has become possible prospects. For the automated code compliance checking, requirements in building code need to be processed into explicit representation that enables automated reasoning. This paper aims to develop high-level methods that translate verb phrases into explicit representation. The high-level methods represent conditions, properties, and related actions of the building objects and clarify the core content of the constraints. The authors analyze building permit requirements in Korea Building Code and establish a standardized process of deriving the high-level methods. As a result, 60 kinds of the high-level methods were derived. In addition, method classification, analysis, and application are introduced. This study will contribute to the representation of explicit building code sentences and establishment of the automated building permit system of Korea.

**Key Words:** Automated code checking, BIM (building information modeling), Building permit system, High-level methods, Korea Building Code, KBimCode, Rule-making

## 1. 서 론

### 1.1 연구의 배경과 목적

건축물의 설계품질에 대한 요구사항은 설계지

침서, 제안 요청서(RFP: request for proposal), 법규 등에 명시되어 있으며, 설계품질검토는 전문가의 지식에 의존하여 이루어졌다. 최근 건물정보모델링(BIM: Building Information Modeling)의 영향력이 건설산업 전반에 걸쳐 확대됨에 따라 그 응용 기술 중 하나로 BIM기반의 자동화된 설계검토에 대한 연구가 이루어지고 있다<sup>1)</sup>. 설계품질검

<sup>†</sup>Corresponding Author, designit@hanyang.ac.kr

토 자동화는 수작업으로 이루어지던 기존의 검토 프로세스의 효율성을 증대하고 결과의 신뢰도를 높일 것으로 국내의 선행연구를 통하여 보고된 바 있다<sup>[2-4]</sup>. 국제적으로 BIM기반 설계품질검토의 이점을 반영한 인허가 시스템에 대한 필요성이 증대하고 있으며 이와 관련하여 정부 주도의 프로젝트들이 실시되었다. 일례로, 싱가포르 정부는 웹 기반의 자동 인허가 시스템 CORNET(Construction and Real Estate Network)을 구축하였으며, 그 효과로 인허가에 소요되는 시간을 102일에서 26일로 절감하였다고 보고하였다<sup>[5]</sup>. 국내에서도 건축행정시스템 세움터<sup>[6]</sup>를 기반으로 BIM기반 건축인허가 시스템을 구축하기 위한 연구개발 과제가 진행 중에 있다. 해당 연구과제는 1) 개방형 BIM기반 설계품질 검증기술 개발 및 인증체계 구축, 2) BIM기반 설계도서 최적화 기준 및 적용기술 개발, 3) 설계·엔지니어링 인허가 통합연계 협업 시스템 개발 등으로 요약될 수 있다<sup>[7]</sup>. 본 연구는 설계품질 검증기술 개발과 관련된 다양한 연구 분야 중, 자연어로 기술된 인허가 관련 건축법규를 컴퓨터에서 실행 가능한 명시적 형태로 변환하는 것을 목적으로 하며, 이를 위한 핵심적인 요소로서 서술부에 대한 함수 체계를 정의 및 개발하였다.

## 1.2 연구의 범위 및 방법

대한민국 건축법규는 한글로 명문화되어 있다. 따라서 건축법규를 자동으로 검토하기 위해서는 자연어로 기술된 문장을 컴퓨터에서 해석될 수 있는 명시적 형태로 변환하는 과정이 필요하다<sup>[8,9]</sup>. 특히 건축법규는 양이 방대하고 해석이 난해하여 명시적으로 표현하는 것이 쉽지 않다. 따라서 법규를 명시적 표현으로 변환하는데 있어 논리적이고 체계적인 프로세스를 수립하는 것이 중요하다. 이러한 논리적 변환 프로세스에 대한 접근방법으로써 이현수 외(2015)의 연구<sup>[10]</sup>에서는 논리규칙화 프로세스를 제시하였다. 논리규칙화 프로세스는 다음과 같은 세 가지 핵심 요소로 구성되어 있다: 1) 건축법규 기반 객체 및 속성 분류, 2) 건축법규 문장 서술부의 유형화, 3) 건축법규 문장의 단일 구문 프로세싱. 본 논문은 논리규칙화 프로세스의 필수 요소 중 하나인 서술부의 유형화를 연구의 범위로 하며, 그 구체적인 내용으로 논리규칙화 함수를 정의하였다. 즉, 논리규칙화 함수는 법규문장 서술부의 명시적 형태라고 할 수 있

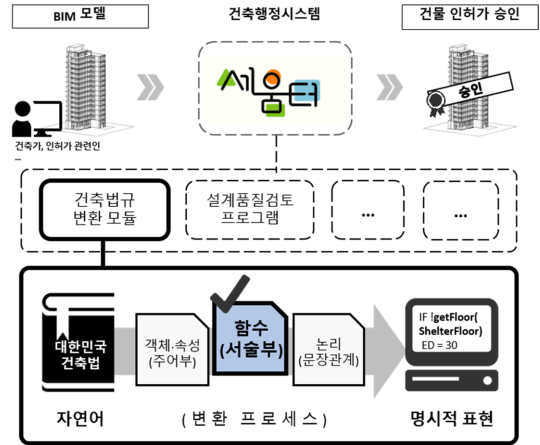


Fig. 1 Overview and the scope of the research

다. 다음 Fig. 1을 통하여 본 연구의 범위를 세움터 기반의 인허가 시스템 구축의 종합적인 관점에서 파악할 수 있다.

논리규칙화 함수체계를 정의 및 개발하기 위하여 본 연구는 다음과 같은 방법 및 절차에 따라 연구를 수행하였다.

첫째, 법규검토에 관한 국내의 선행연구 사례를 개발에 초점을 맞추어 조사하고 시사점을 도출하였다.

둘째, 인허가관련 건축법규를 분석하고 논리규칙화 함수 도출 프로세스를 제시하였다.

셋째, 연구의 대상에서 논리규칙화 함수를 도출하고 그 종류와 분류체계를 수립하였다. 또한 함수 빈도조사를 실시하여 함수의 구현 및 법규문장의 명시적 표현을 위한 시사점을 도출하였다.

넷째, 논리규칙화 함수를 활용하여 인허가 관련 법규를 명시적 표현으로 변환하는 방안을 제시하였다.

## 2. 선행연구 및 개발 사례 분석

자동화된 설계검토에 대한 연구는 1960년대부터 이루어져왔다<sup>[11]</sup>. 컴퓨터 성능의 향상과 BIM의 도입으로 최근 10년 동안 설계검토 자동화에 관한 연구 및 개발이 활발히 이루어지고 있다. 국내의 관련 프로젝트 및 연구를 정리하면 Table 1과 같다.

국의 프로젝트의 경우, 정부 주도의 법규 및 제기준에 대한 자동검토가 주를 이루었다. 자동검토를 위하여 설계 요구사항을 명시적 표현으로 변환

**Table 1** Projects and researches on the automated design assessment

구분	프로젝트 및 연구사례	
국외	싱가포르 CORENET e-PlanCheck (2005)	
	호주 DesignCheck (2006)	
	미국 SMARTcode (2006) GSA Courthouse project (2007)	
국내	2D 기반 연구 건축법규의 형식화에 관한 연구 (송영규 외, 1989) 건축기본계획서 건축법규 해석을 위한 전문가 시스템의 개발(이한석 외, 1990) CALs/EC 표준 기반의 건축도면 자동 법규검토 방안에 관한 연구(김인한 외 2004)	
	BIM 기반 연구	국내 BIM지침의 품질검토 항목을 활용하기 위한 Rule 기반 BIM Model Checker의 Rule 패턴 개발에 관한 연구(송종관 외, 2012)
		BIM기반의 건축법규검토를 위한 룰셋정의서 개발- 장애인,노인,임산부 등의 편의증진 보장에 관한 법률 대상으로 (김유리 외, 2012)
		덕트설비의 BIM 기반 자동검토를 위한 규칙개발에 관한 연구 (송종관 외, 2013a)
		BIM 모델의 완성도를 높이기 위한 품질검토항목의 룰 개발 (송종관 외, 2013b)
		BIM기반 설계품질 자동검토 구현을 위한 건축법규 문장의 논리규칙 체계화 접근방법 (이현수 외, 2015)

하는 프로세스, 즉 룰의 개발 측면에서 각 프로젝트를 살펴보면 다음과 같다.

싱가포르 정부 및 건설청의 주도로 추진된 CORENET(CONstruction and Real Estate Network) 프로젝트는 웹 기반 건설 행정처리 시스템을 구축하고 건설산업의 정보 공유 및 건축허가 프로세스를 간소화하는 것을 목적으로 한다. CORENET e-PlanCheck는 법규검토 모듈로, 자체적인 FORNAX 플랫폼을 사용하여 싱가포르 건축법규에 대한 자동 법규검토를 수행한다. 싱가포르 건축법규는 프로그래밍 언어를 이용하여 FORNAX 프로그램에 통합적으로 반영되었다<sup>5)</sup>.

DesignCheck는 호주 건축법(BCA; Building Code of Australia)에 대한 법규검토 시스템이다. 법규를 명시적인 룰로 변환하기 위하여 프로젝트

팀은 요구사항, 객체, 속성, 관계 등의 내용을 담은 객체기반의 사전 구현 설명서를 작성하였다. 작성 내용은 EDM(Express Data Manage)의 룰 스키마를 사용하여 EXPRESS 언어로 변환되었다<sup>12)</sup>.

국제 코드위원회(ICC; International Code Council)는 건축도면에 대한 국제 코드(International Code)와 미국 연방, 주, 지역별 법안을 자동 검토하기 위하여 SMARTcode 프로젝트를 추진하였다. SMARTcode는 마킹업(marking up) 방식을 이용하여 법규를 컴퓨터에서 실행 가능한 형태로 변환하였다. 사용자가 SMARTcode Builder 소프트웨어의 GUI 인터페이스를 사용하여 법규 문장의 요소를 사전 정의된 분류에 따라 태그하면, 해당 사항을 반영한 IFC constraint 파일이 자동으로 생성될 수 있도록 하였다<sup>13)</sup>.

미국 연방 조달국(GSA; General Services Administration)은 3D-4D BIM 프로젝트를 수립하고 그 일환으로 BIM기반 설계품질검토 연구를 수행하였다. 법원건물을 대상으로 PBS(Public Building Service)에서 명시하는 공간프로그램과 동선분석에 대한 자동검토 모듈이 개발되었다. 동선에 관한 모든 규정은 분석을 통해 매개변수화 되어 룰로 제작되었다<sup>6)</sup>.

국내의 법규검토에 대한 연구는 2D 기반의 법규검토와 BIM기반의 법규검토로 나누어 살펴볼 수 있다. 송영규 외(1989)의 연구<sup>14)</sup>와 이한석 외(1990)의 연구<sup>15)</sup>는 설계초기단계에서 설계자의 법규 검토를 지원하는 시스템에 대한 연구를 수행하였다. 이를 위하여 서술문식으로 기술된 법규를 논리적으로 형식화하고 2차원 CAD 시스템에 적용하고자 했다. 또한 김인한 외(2004)의 연구<sup>16)</sup>에서는 CALS/EC 도면 교환 표준을 기반으로 한 법규검토를 제안하였다.

BIM기반 설계품질검토에 대한 국내 연구는 설계지침서, RFP, BIM 지침, 법규 등 다양한 요구사항을 기반으로 수행되었다. 송종관 외(2012)의 연구<sup>17)</sup>는 BIM지침을 대상으로 품질검토 항목별 패턴을 정의하여 룰을 생성하는 방법을 제안하였다. 품질검토 항목은 구조화 과정을 통하여 함수의 형태로 표현되었다. 김유리 외(2012)의 연구<sup>18)</sup>는 장애인, 노인, 임산부 등의 편의 증진 보장에 관한 법률을 대상으로 룰셋정의서를 개발하였다. 법규 원문은 분석과정을 거친 뒤 C# 언어 기반의 룰로 변환되었다. 송종관 외(2013a)의 연구<sup>19)</sup>는 건

축기계설비기준을 대상으로 공기조화 설비의 덕트 시스템에 관한 룰을 개발하였다. 룰 구성을 위한 요소로써 정보모델과 함수가 개발되었다. 정보 모델은 품질검토항목에서 요구되는 객체와 속성을 나타내며, 함수는 설계검토 프로그램에서 BIM 모델의 객체와 속성을 활용하여 검토 기능을 구현하기 위한 개념함수이다. 또한 이와 유사한 방법으로, 송종관 외(2013b)의 연구<sup>[20]</sup>는 국내 BIM 지침에 명시된 BIM 모델의 품질검토항목에 대한 룰을 함수와 변수를 사용하여 나타내었다.

선행연구에서 제시한 룰 개발 방법은 다음과 같이 분류될 수 있다: 1) 프로그래밍 언어, 모델링 언어 등을 활용하여 하드코딩하는 방법, 2) 검토항목을 파라미터화하여 설계검토 프로그램에 구현하는 방법, 3) IFC constraint 모델을 이용하는 방법, 4) 검토항목으로부터 개념함수와 변수 등의 요소를 정의하고 논리적으로 조합하여 자연어와 컴퓨터 실행 가능한 의 중간단계의 표현으로 변환하는 방법 등. 룰 개발에 관한 다양한 접근 방법 중, 본 연구는 중간단계의 표현으로 룰을 나타내고자 하며, 그 구성요소로써 함수 체계를 개발하였다. 이러한 접근방법은 특정 설계품질검토 소프트웨어나 BIM 환경에 종속적이지 않고 확장성의 측면에서 유연하다는 이점이 있다<sup>[21]</sup>. 기존 연구는 BIM 지침, 설계기준, 또는 한정된 범위의 특정 법규만을 대상으로 한 반면, 본 연구는 다양한 건축법규를 아우르는 인허가 관련 건축법규를 대상으로 한다. 건축법규는 그 양이 방대하고 예외규정이 많으며, 복잡한 상하위 위임 및 참조관계를 형성한다. 따라서 전체 법규를 명시적으로 표현하는데 어려움이 있다. 본 연구에서 다루는 함수 체계는 이러한 건축법규의 특성을 반영하였으며, 특히 인허가와 관련된 사항들에 초점을 맞추어 개발되었다.

### 3. 논리규칙화 함수 개발 접근방법

#### 3.1 인허가 관련 건축법규 분석

건축 인허가에 관련된 규정은 건축법, 건축법 시행령, 건축법 시행규칙 등을 비롯하여 국토의 계획 및 이용에 관한 법률, 주차장법 등 관련법에도 명시되어 있다. 본 연구는 인허가 관련 46개 법규를 대상으로, BIM 기반 검토 가능성이 높은 113개 조문을 논리규칙화 함수 도출을 위한 연구 대

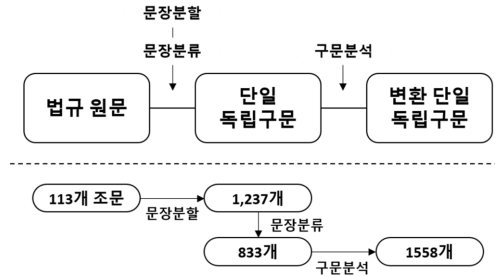


Fig. 2 Process of the law sentence to develop the high-level methods

상으로 설정하였다. 해당 조문은 건축물의 피난시설 및 용도제한, 내화구조와 방화벽, 마감재료, 건축선의 지정, 건축선에 따른 건축제한, 그리고 지하층 및 건축물의 대지에 대한 인허가 요구사항을 다룬다. BIM 기반 검토 가능성이 높은 조문은 검토의 기준이 모호하거나 정성적이지 않고, 수치적으로 정량화 할 수 있으며, 객체의 존재 유무, 속성 등을 규정한다.

일반적으로 사람은 법규의 내용을 다수의 문장으로 구성된 조문 단위로 인식한다. 그러나 컴퓨터는 사람과 같은 방식으로 법규의 구성 및 내용을 이해하고 처리할 수 없다. 따라서, 건축법규를 논리적으로 체계화하여 컴퓨터에서 실행 가능한 단위로 처리하여야 한다. 다음 Fig. 2는 법규 원문을 최소 문장단위로 분해하고 명시적 형태로 변환하는 프로세스를 도식화 한 것이다. 각 단계에 대한 설명은 다음과 같다.

#### 1) 법규 원문

대한민국 법규는 조문, 항, 호, 목의 위계로 구성된다. 특정 항목에 대한 종합적인 설계 요구사항은 조문 단위에서 명시되며, 일반적으로 조문은 복수의 항, 호, 목, 즉 다수의 단일 문장으로 구성된다. 따라서 최소 문장단위를 도출하기 위해서는 조문을 단일 문장으로 분할해야 한다. 본 연구의 대상이 되는 113개 조문을 문장 단위로 분해하여 1,237개의 단일 독립구문을 도출하였다.

#### 2) 단일 독립구문(AS: atomic sentence)

본 연구에서 단일 독립구문은 법규 검토를 위한 최소 문장단위를 의미한다. 단일 독립구문은 명확한 주어부와 서술부로 구성되어 있으며, 의미상 검토의 조건과 내용을 명시하여 검토 결과를 참, 거

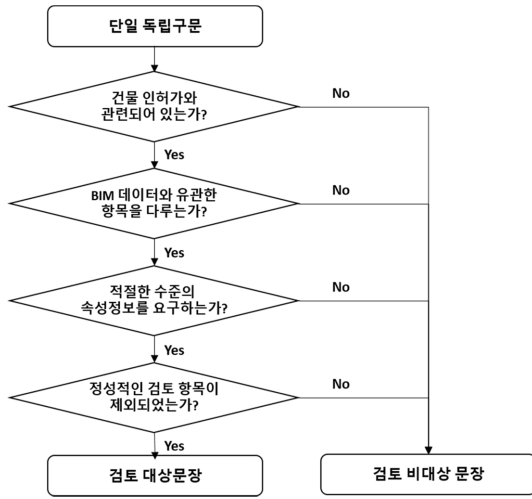


Fig. 3 Filtering process of the atomic sentence to figure out appropriate sentences for the BIM-enabled code compliance checking of building permit

것으로 나타낼 수 있다. 단일 독립구문은 법규 원문의 문장 분할을 통하여 도출되며, 이중 일부는 건축인허가와 무관하거나 BIM기반 검토에 부적합하다. 따라서 이러한 문장을 여과하고 적합한 검토 대상 문장을 판별하기 위하여 문장분류가 수행되어야 한다. 건축물 인허가 관련된 설계품질 검토 대상 문장을 판별하기 위한 문장분류 프로세스는 다음 Fig. 3과 같다.

문장분류 기준에 대한 자세한 설명은 다음과 같다. 첫째, 법규의 개정, 시행일 등과 같이 건물 인허가와 무관한 내용을 고지하는 문장은 건물 인허가와 무관하므로 제외된다. 둘째, 외부 도로, 하천 등 BIM 데이터와 무관한 내용을 담고 있는 문장들 또한 제외된다. 셋째, 과도한 속성정보를 요구하는 검토내용 또한 제외된다. 회전문의 분당 회전 수, 지나치게 LOD(level of detail) 높은 형상정보 등이 이에 속한다. 해당 내용은 경제적 측면에서 자동 검토의 효율성이 없다고 판단되었다<sup>[22]</sup>. 마지막으로, 정성적인 검토 항목을 나타내는 문장들이 비대상 문장으로 분류된다. 해당 항목은 “적당한 거리를 두고 설치하여야 한다”와 같이 검토의 기준이 모호하고 정량화 할 수 없는 내용을 나타낸다. 결과적으로, 1,237개 단일 독립구문의 68%에 해당하는 833개 문장이 검토 대상문장으로 선별되었다.

### 3) 변환 단일 독립구문(TAS: translated atomic sentence)

변환 단일 독립구문은 더 이상 쪼개지지 않는 구문으로, 주어(S) + 동사(V) + 목적어(O)의 문법적 구조로 표현될 수 있다. 변환 단일 독립구문은 단일 독립구문에 대한 구문분석을 통하여 얻어진다. 단일 독립구문은 일반적으로 복수의 변환 단일 독립구문으로 구성되며, 각각의 변환 단일 독립구문은 검토의 조건 또는 내용을 독립적으로 명시한다. 하나의 변환 단일 독립구문은 더 이상 나누어지지 않는 하나의 서울부를 나타내며 이는 하나의 함수와 대응된다. 결론적으로, 변환 단일 독립구문을 통하여 검토의 최소단위를 파악할 수 있으며, 이로부터 서울부를 명료화하고 최종적으로

Table 2 Type of verbal phrases in the target building code sentences

서울부 유형		서울부 예시
논리적 처리	위임/참조	~한 바에 의한, ~로 정한다 등
	예외	~는 제외한다 등
	강도	~할 수 있다, ~하여야 한다 등
	계사	~이다, ~을 말하다, ~로 보다 등
함수적 처리	존재하다	~을 설치하여야 한다 등
	개수확인	~개 이상/이하이다 등
	포함하다	~을 포함한다, ~에 설치하는 등
	연결하다 (물리적)	~에 닿게 하여야 한다, 외기에 접해야 한다 등
	연결하다 (접근성)	연결하여 설치한다, 직접 연결되도록 한다, ~를 통하여 등
	인접하다	~에 인접하다, ~와~를 이웃하지 아니하도록 배치한다 등
	구획하다	~으로 구획하다 등
	거리	보행거리/수직거리가 ~ 이상/이하이다 등
	방향확인	피난의 방향으로 열다 등
	너비확인	폭/너비가 ~이상/이하이다 등
	마감재료 확인	~로 마감하다, ~재료로 하다, ~을 마감재료로 쓰다 등
	용도확인	~의 용도로 쓰는 등
	소방관련 속성 확인	내화구조/방화구조/방화구획으로 하다, 방화벽을 설치하다 등
기본속성 확인	블락이 창이다, 돌음계단으로 하여서는 아니된다 등	

논리규칙화 함수를 도출할 수 있다.

Table 3는 법규원문으로부터 변환 단일 독립구문까지의 변환 프로세스를 법규 예시를 통하여 보여준다. 앞서 문장분류를 통해 검토 대상문장으로 파악된 833개 단일 독립구문으로부터 1,558개 변환 단일 독립구문을 도출하였다.

### 3.2 서술부 추출 및 유형화

다음의 Table 2는 1,558개 변환 단일 독립구문으로부터 서술부를 추출하고 의미론적으로 동일한 서술부를 유형화한 표이다. 서술부 유형은 논리적으로 처리될 수 있는 서술부와 함수적으로 처리될 수 있는 서술부로 나뉜다. 논리적으로 처리되는 서술부는 연산자로 처리 가능하거나 혹은 함수로는 처리가 불가능하여 복잡한 논리적 관계가 고려되어야 하는 유형이다. 반면 함수적으로 처리 가능한 서술부는 객체의 존재유무, 속성, 관계 등을 나타내며, 서술부는 함수(function)로, 대상 객체는 함수의 매개변수(argument)로 치환된다.

### 3.3 논리규칙화 함수 도출

논리규칙화 함수는 함수적 처리 유형으로 분류된 서술부를 기반으로 개발되었다. 서술부와 논리규칙화 함수는 일대일 관계이다. 논리규칙화 함수는 고급(high-level) 함수로, 설계검토 소프트웨어 구현 단계의 저수준(low-level) 함수와 구별된다. 논리규칙화 함수의 목적은 자연어의 서술부를 직관적인 형태로 변환함에 있다. 따라서 설계

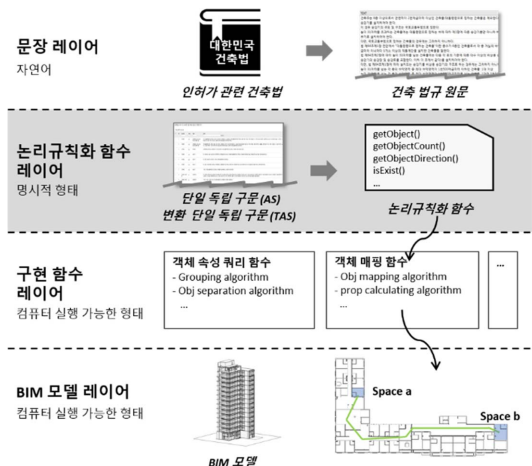
**Table 3** Process of deriving the high-level methods from Korea Building Code

1) 법규 원문	
[건축법 제64조 (승강기) 제1항] “건축주는 6층 이상으로서 연면적이 2천 제곱미터 이상인 건축물을 건축하려면 승강기를 설치하여야 한다. 이 경우 승강기의 규모 및 구조는 국토교통부령으로 정한다.” <sup>23)</sup>	
2) 단일 독립구문 (AS)	
6층 이상으로서 연면적이 2천 제곱미터 이상인 건축물에는 승강기를 설치하여야 한다	
3) 변환 단일 독립구문 (TAS)	
구문 분석	주어부: 6층 이상으로서 연면적이 2천제곱미터 이상인 건축물 서술부: 승강기를 설치하여야 한다
TAS	TAS 1-1. 건축물이 6층 이상이다. TAS 1-2. 건축물의 연면적이 2천 제곱미터 이상이다. TAS2. 건축물에 승강기가 존재한다.
4) 서술부 추출	
V1. 층의 개수가 ~이다 V2. 연면적이 ~이다 V3. 객체 <b>a</b> 가 객체 <b>a</b> 에 존재한다	
5) 함수 도출	
M1. getBuildingStoriesCount() M2. getTotalFloorArea() M3. isExist(elevator)	

검토 단계에서 논리규칙화 함수를 구현하기 위해서는 해당 설계검토 소프트웨어의 저수준 함수 라이브러리와 매핑되어야 한다. Fig. 4는 논리규칙화 함수와 구현 단계 함수의 차이점을 보여주며, 자연어로 기술된 요구사항이 설계검토 단계의 BIM모델과 매핑되는 전체적인 프로세스를 보여준다.

다음 Table 3은 법규 원문을 예시로 전반적인 논리규칙화 함수 도출 프로세스를 보여준다.

이와 같은 방법으로 연구의 대상이 되는 인허가 관련 46개 법규, 113개 조문으로부터 BIM기반 검토에 적합한 833개 단일 독립구문, 1,558개 변환 단일독립구문을 도출하였으며, 최종적으로 60종의 논리규칙화 함수를 도출하였다. 다양한 논리규칙화 함수는 동일한 함수 명명 규칙을 기반으로 개발되었다. 함수 명명 규칙은 해당 함수의 기능을 직관적으로 파악할 수 있는 것을 목표로 한다. 이를 위하여 가장 많이 사용되는 고급 프로그래밍



**Fig. 4** Description of the difference between high-level methods and the low-level methods in terms of four layers

언어 중 하나인 자바 언어<sup>[24]</sup>의 함수 명명규칙을 참고하였다. 함수의 명명 규칙을 EBNF(Extended Backus-Naur Form)<sup>[25]</sup>로 나타내면 다음과 같다.

```
Method = ( mVerb, mObj, [mProp], [mCond]
          | mVerb, mCond ),
          (“, {mParam}, “)
;
mVerb : get | is
;
mObj : buildingObject
;
mProp : buildingObjectProperty
;
mCond : condition
;
mParam : mObj, {“, ”, mObj} [“, ”, mType]
;
```

## 4. 논리규칙화 함수 체계

### 4.1 논리규칙화 함수의 종류와 분류체계

논리규칙화 함수체계는 Table 4에 나타난 바와 같이 네 단계로 구성된다. 대분류는 함수를 크게 두 가지로 구분한다. 첫 번째는 객체 인스턴스의 존재유무만으로 검토 가능한 경우로 객체 자체를 쿼리하거나 유무를 확인 그리고 개수를 확인하는 함수가 이에 속한다. 두 번째는 객체의 존재유무 뿐만 아니라 속성 또한 검토하는 경우로, 이에 속하는 함수는 속성의 유형에 따라 중분류 단계에서 다시 1) 기본속성, 2) 계산속성, 그리고 3) 관계속성으로 분류된다. 기본속성은 객체명, 유형, 종류 등과 같이 객체가 기본적으로 내포하는 속성을 의미한다. 계산속성은 객체의 길이, 너비, 높이 등과 같은 형상정보와 관련된 속성과 법규에서 규정하는 마감재 유형, 방화관련 기준과 같은 특정 기준

**Table 4** Classification of the logic rule-based methods

대분류	중분류	소분류	논리규칙화 함수		
			대표함수	확장함수	
객체		쿼리	getObject()	getStair()...	
		유무	isExist()		
		객체 개수	getObjectCount()	getBuildingStoriesCount()	
객체 속성	기본 속성	쿼리	getProperty()	getDoorType(),...	
		마감재 확인	getObjectMaterial()		
		용도 확인	getObjectUsage()	getBuildingUsage(),getFloorUsage(), ...	
	계산 속성	형상 속성	높이	getObjectHeight()	getBuildingHeight(),...
			너비	getObjectWidth()	getDoorWidth(),...
			면적	getObjectArea()	getBuildingArea(),getFloorArea(),...
			경사도	getObjectGradient()	
	복합 속성	방화 관련	마감재 유형	getMaterialType()	
			방화 관련	isFireResistantStructure()	
				isFireProofStructure()	
	isFirePartition()				
	관계 속성	포함	포함	hasObject()	hasElevator(),...
거리			getObjectDistance()	getVerticalDistance(),...	
물리적 연결성		물리적 연결성	isConnectedTo()		
			isExternal()		
			isAdjacent()		
동선		isAccessible()	isDirectlyAccessible(),...		
방향	getObjectDirection()				

과 관련된 복합속성으로 세분화되며, 기본속성과 달리 추가적인 연산을 통해 도출된다. 마지막으로 관계속성은 복수 객체간 관계에 대한 속성으로, 포함관계, 거리, 물리적 연결성, 동선, 방향 등에 관한 속성이 이에 속한다. 소분류는 검토의 구체적인 내용을 분류한 단계이다. 소분류는 논리규칙화 함수와 직접적으로 연관된다.

논리규칙화 함수는 대표함수와 확장함수로 구분된다. 대표함수는 검토의 대표성을 띄는 함수로, 다양한 객체 및 속성을 수용할 수 있는 함수이다. 대표함수는 소분류의 각 항목을 대표한다고 볼 수 있다. 반면 확장함수는 세분화된 객체 및 속성 검토에 특화된 함수이다. 예를 들어, 객체의 넓이를 쿼리하는 대표함수 `getObjectArea()`는 대상 객체 및 속성의 종류에 따라 `getSiteArea()`, `getBuildingArea()`, `getFloorArea()`, `getTotalFloorArea()` 등으로 확장된다. 총 60개의 함수 중, 대표함수는 27개, 확장함수는 33개이다.

본 연구는 대한민국 전체 건축 법규의 일부를 대상으로 논리규칙화 함수를 도출하였으며, 추후 함수 도출 대상의 범위가 확장됨에 따라 함수체계 또한 확장될 수 있다.

#### 4.2 논리규칙화 함수 빈도분석

변환 단일독립구문 1,558개에 대한 논리규칙화 함수의 빈도분석을 실시한 결과, Fig. 5와 Fig. 6의 결과를 얻을 수 있었다. 전체 60종 함수 중에서 객

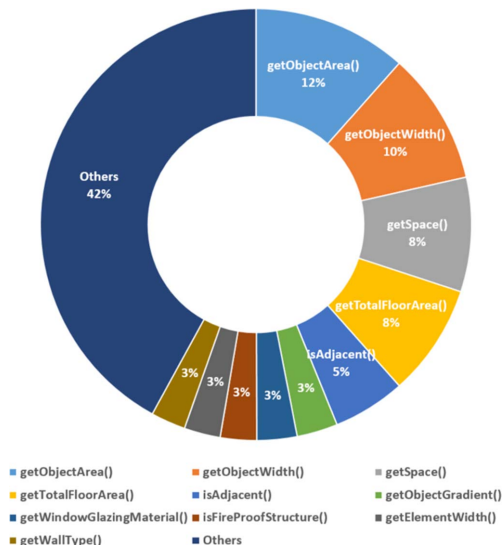


Fig. 5 Frequency analysis of the entire 60 methods

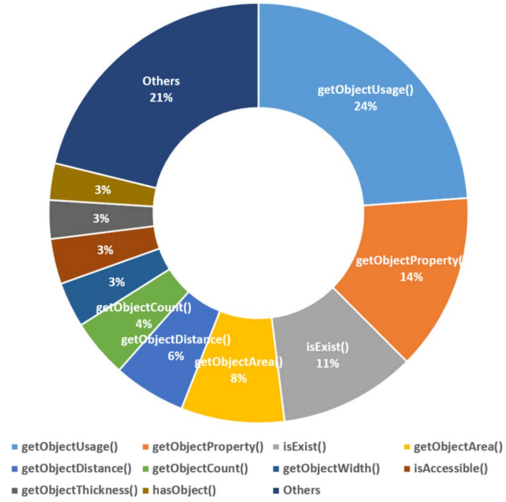


Fig. 6 Frequency analysis of the 27 representative methods

체의 면적을 확인하는 함수 `getObjectArea()`의 빈도가 가장 높았으며, 그 뒤를 객체의 너비를 확인하는 함수 `getObjectWidth()`와 공간객체를 쿼리하는 함수 `getSpace()`가 뒤따랐다. 반면 확장함수의 빈도수를 합산하여 대표함수로 함수 분류체계의 빈도수를 조사한 결과, 객체의 용도를 확인하는 함수 `getObjectUsage()`의 빈도수가 가장 높았으며, 그 뒤를 객체의 속성을 확인하는 `getObjectProperty()`가 뒤따랐다. 즉, 객체의 기본속성을 확인하는 함수가 가장 높은 비율을 차지했다. 이외에, 객체의 존재유무를 확인하는 함수 `isExist()`와 넓이를 확인하는 함수 `getObjectArea()`, 객체 간 수직, 수평 거리를 확인하는 함수 `getObjectDistance()`, 그리고 객체의 개수를 확인하는 함수 `getObjectCount()`가 높은 빈도로 사용되었다.

함수의 빈도를 분석한 결과, 객체의 속성을 검토하는 함수가 대다수를 차지하며, 그 중에서도 기본속성을 확인하는 함수가 높은 비중을 차지하였다. 함수의 빈도는 검토 대상 법규의 내용에 종속적이다. 즉, 대상 법규는 객체의 기본 속성에 대한 요구사항을 주로 다루며, 해당 법규의 명시적 표현을 위해서는 함수 체계뿐만 아니라 객체 및 그 속성에 대한 체계화된 분류체계 수립이 동반되어야 한다는 것을 파악할 수 있었다. 또한 함수의 빈도분석 결과는 설계품질검토 소프트웨어와의 연동 단계에서 구현의 우선순위를 정하는 데 유용한 자료로 활용될 수 있다.



### 5. 논리규칙화 함수의 활용

#### 5.1 산술논리단위 기반의 명시적 표현

산술논리단위(ALU: Arithmetic Logic Unit)는 변환 단일 독립구문을 명시적으로 표현한 형태이다. 즉, 명시적으로 표현된 설계검토의 최소단위이다. 산술논리단위는 변환 단일 독립구문의 주어 및 목적어에 해당하는 객체와 속성, 동사에 해당하는 함수, 그리고 법규문장에서 규정하는 제약조건에 해당하는 명료한 값과 비교연산자로 구성된다. Table 4의 예시를 산술논리단위로 표현하면 다음과 같다.

```
getBuildingStoriesCount() >= 6
getTotalFloorArea() >= 2000m2
isExist(elevator) = TRUE
```

다음 Fig. 7은 법규문장의 명시적 표현을 위한 변환 프로세스의 선상에서 산술논리단위를 보여준다. 최종적으로 법규원문으로부터 하나 이상의 산술논리단위가 도출되며, 동일한 법규원문에서 파생된 복수의 ALU는 검토의 조건과 내용, 문장 간 관계 등이 반영된 논리적 형태로 조합된다. 본 연구는 이러한 조합을 KBimCode라 명명하였다.

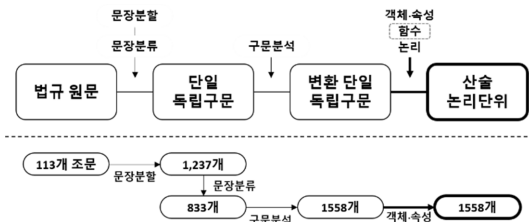


Fig. 7 Process of deriving ALU

Table 5 Examples of sentence to KBimCode. The high-level methods are presented in bold

건축법규 원문 및 KBimCode	
1	<p>[건축법 제64조 (승강기) 제1항] “건축주는 6층 이상으로서 연면적이 2천 제곱미터 이상인 건축물을 건축하려면 승강기를 설치하여야 한다...”</p> <pre>Check(BA_64_1){   IF (getBuildingStoriesCount() &gt;=6       AND getTotalFloorArea() &gt;=2000m2)     THEN isExist(elevator) = TRUE   END IF }</pre>

Table 5 Continued

건축법규 원문 및 KBimCode	
2	<p>[건축법 시행령 35조 (피난계단의 설치) 제 1항] “법 제49조제1항에 따라 5층 이상 또는 지하 2층 이하인 층에 설치하는 직통계단은 국토교통부령으로 정하는 기준에 따라 피난계단 또는 특별피난계단으로 설치하여야 한다. 다만, 건축물의 주요구조부가 내화구조 또는 불연재료로 되어 있는 경우로서 다음 각 호의 어느 하나에 해당하는 경우에는 그러하지 아니하다.”</p> <pre>check(EDBA_35_1){   IF (!CS1 AND !CS2 AND CS3) THEN KS } CS1 {   isFireProofStructure(MainStructurePart) = TRUE   OR   isObjectProperty(MainStructurePart.nonCombustibility) = TRUE } CS2 {   getResult(EDBA_35_1_1) = TRUE   OR getResult(EDBA_35_1_2) = TRUE } CS3 {   Floor myFloor {     getFloorNumber() &gt; 5     OR getFloorNumber() &lt;= -2   }   Stair myStair {     isObjectProperty(Stair.isDirect) = TRUE   }   hasElement(myFloor, myStair) = TRUE } KS {   isObjectProperty(myStair.isEscape) = TRUE   OR isObjectProperty(myStair.isSpecialEscape) = TURE }</pre>
3	<p>[건축법 시행령 35조 제 1항 제1호] “1. 5층 이상인 층의 바닥면적의 합계가 200제곱미터 이하인 경우”</p> <pre>check(EDBA_35_1_1){   Floor myFloor {     getFloorNumber() &gt;= 5   }   getTotalFloorArea(myFloor) &lt;= 200m2 }</pre>
4	<p>[건축법 시행령 35조 제 1항 제2호] “2. 5층 이상인 층의 바닥면적 200제곱미터 이내마다 방화구획이 되어 있는 경우”</p> <pre>check(EDBA_35_1_2){   Floor myFloor {     getFloorNumber() &gt;= 5   }   isFirePartition(myFloor, 200) = TRUE }</pre>

즉, KBimCode는 법규문장의 온전한 의미를 전달하는 명시적 형태이다.

### 5.2 KBimCode기반의 법규검토

다음 Table 5는 건축법규 문장을 KBimCode로 변환한 예시이다. 각 문장은 논리규칙체계화 과정을 통하여 ALU로 변환되고 원문에서 명시하는 검토의 조건과 내용에 적합하게 논리적으로 조합되었으며, KBimCode의 문법<sup>[20]</sup>에 맞게 작성되었다.

예시 1은 건축물의 승강기 설치에 대한 규정을 나타내는 문장으로, Table 4 및 5.1의 예시로 사용되었다. 예시 2,3,4는 서로 연결된 KBimCode로, 복잡한 위임 및 참조관계를 형성하는 대한민국 건축법규 문장의 특성을 보여주는 예시이다. 예시 2의 법규 원문에서 밑줄로 표시된 위임관계는 KBimCode 상에서 컨트롤 함수 getResult()와 하위 법규 문장의 식별자(EDBA\_35\_1\_1, EDBA\_35\_1\_2)로 표현된다. 예시 3과 4는 예시 2로부터 위임 받은 하위 법규문장에 대한 KBimCode예시이다.

KBimCode는 자연어로 쓰여진 법규와 컴퓨터 실행 가능한 형태의 중간 단계 언어이다. KBimCode는 JSON, XML 등 형식으로 변환되어 룰체킹 소프트웨어에서 사용된다. KBimCode는 추가적인 파서를 통해 어떠한 종류의 품질검토 소프트웨어

에서도 직접 실행 가능한 룰셋 파일의 형태로 변환된다. 파서 개발을 위하여 EBNF 언어 기반의 KBimCode 언어 정의, 논리규칙화 함수의 사양을 설명한 매뉴얼<sup>[27]</sup>, 법규 기반의 객체 및 속성 사전 등 KBimCode의 사양에 대한 정보가 소프트웨어 개발사에 제공된다. Fig. 8은 건축 법규문장이 KBimCode로 변환되는 룰 개발 과정과 설계품질 검토 소프트웨어에서 활용되는 룰체킹 과정의 일련의 시나리오를 나타낸 것이다. 논리규칙화 함수는 KBimCode를 구성하는 핵심요소 중 하나으로써 KBimCode와 품질검토 소프트웨어의 연동 단계에서 저수준 함수 라이브러리 등과의 매핑을 통해 구현된다.

## 6. 결 론

본 논문은 BIM기반 설계품질 자동검토 구현을 위해 자연어 형태의 법규 문장을 컴퓨터에서 실행 가능한 형태로 변환하기 위한 논리규칙화 함수를 개발하였다. 논리규칙화 함수는 법규 문장의 서술부를 명시적 형태로 변환하며, 검토 제약조건을 나타내는 ALU의 구성요소로 사용되어 최종적으로 인허가 관련 요구사항을 명시적 형태에 해당하는 KBimCode를 구성한다. 연구의 대상이 되는 46개 법규 113조문을 분석한 결과, 833개 단일 독립구문, 1,558개 변환 단일 독립구문을 도출하였다. 변환 단일 독립구문은 주어, 동사, 목적어로 구성된 명료한 서술문으로, 이로부터 서술부를 추출하고 유형화 하여 60종의 논리규칙화 함수를 개발하였다. 논리규칙화 함수의 확장성을 고려하여 함수체계를 수립하였으며, 함수체계에 따른 빈도분석을 실시하여 건축법규 자동검토를 위한 시사점과 구현단계에서의 사안을 도출하였다.

본 연구는 인허가 관련법의 일부를 대상으로 하여 논리규칙화 함수체계를 개발하였다. 추후 대상 법규의 범위를 확대하여 광범위한 인허가 관련 법규를 포괄할 수 있도록 연구를 진행할 것이다. 또한 건축법규뿐만 아니라 다양한 설계품질 요구사항을 반영할 수 있도록 함수의 종류와 분류체계를 확장해 나갈 것이다.

## 감사의 글

본 연구는 국토교통부 도시건축 연구개발사업

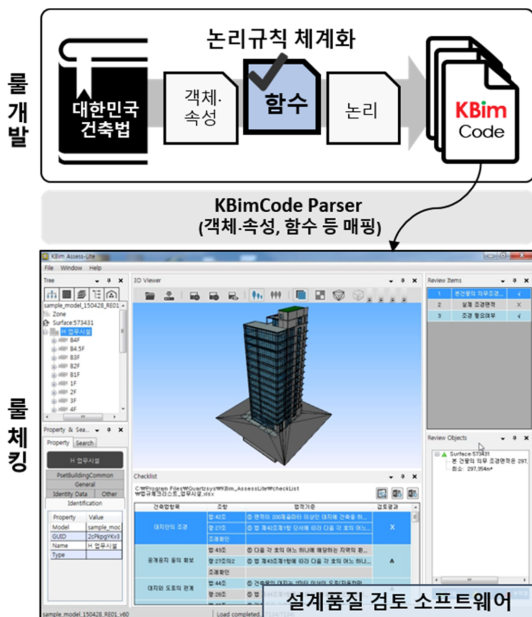


Fig. 8 Scenario of automated code compliance checking using KBimCode

의 연구비지원(15AUDP-C067809-03)에 의해 수행되었습니다.

## References

1. Eastman, C.M., Teicholz, P., Sacks, R. and Liston, K., 2008, *BIM Handbook—A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors*, John Wiley & Sons Inc., Hoboken, NJ, United States of America.
2. Lee, J.-K., Lee, J., Jeong, Y.-S., Sheward, H., Sanguinetti, P., Abdelmohsen, S. and Eastman, C.M., 2012, Development of Space Database for Automated Building Design Review Systems, *Automation in Construction*, 24, pp.203-212.
3. Greenwood, D., Lockley, S., Malsane, S. and Matthews, J., 2010, Automated Compliance Checking Using Building Information Models, *Proceedings of the Construction, Building and Real Estate Research Conference of the Royal Institution of Chartered Surveyors*, Paris, France.
4. Dimyadi, J. and Amor, R., 2013, Automated Building code Compliance Checking Where is it at?, *Proceedings CIB WBC 2013*, Brisbane, Australia, pp.172-185.
5. Eastman, C., Lee, J., Jeong, Y.-S. and Lee, J.-K., 2009, Automatic Rule-based Checking of Building Designs, *Automation in Construction*, 18(8), pp.1011-1033.
6. Seumter, (<https://www.eais.go.kr/>)
7. Kim, I., 2015, *Open BIM based Technological Environment for Building Design Quality Enhancement*, Project report, Kyung Hee Univ.
8. Malsane, S., Matthews, J., Lockley, S., Love, P.E.D. and Greenwood, D., 2015, Development of an Object Model for Automated Compliance Checking, *Automation in Construction*, 49, pp.51-58.
9. Uhm, M., Lee, G., Park, Y., Kim, S., Jung, J. and Lee, J.-K., Requirements for Computational Rule Checking of Requests for Proposals(RFPs) for Building Designs in South Korea, *Advanced Engineering Informatics*, 29(3), pp.602-615.
10. Lee, H., Park, S., Kim, I. and Lee, J.-K., 2015, A Logical Rule-based Approach to the Korea Architecture Code Sentences for BIM-enabled Design Assessment Systems, *Korea Design Knowledge Society*, 34, pp.101-110.
11. Solihin, W. and Eastman, C., 2015, Classification of Rules for Automated BIM Rule Checking Development, *Automation in Construction*, 53, pp.69-82.
12. Ding, L., Drogemuller, R., Rosenman, M., Marchant, D. and Gero, J., 2006, *Automating Code Checking for Building Designs, Clients Driving Construction Innovation: Moving Ideas into Practice*, CRC for Construction Innovation, Brisbane, Australia, pp.113-126.
13. Wix, J., Nisbet, N. and Liebich, T., 2008, Using Constraints to Validate and Check Building Information Models, *eWork and eBusiness in Architecture, Engineering and Construction, ECPPM 2008*, London, UK, pp.467-475.
14. Song, Y.K. and Kim, U., 1989, A Study on the Formalization of Building Codes, *Proceedings of AIK Spring Conference, Planning and Design Section*, 9(1), pp.241-246.
15. Lee, H.S. and Lee, K.H., 1990, The Development of Expert System for the Analysis of the Building Code at the Preliminary Design, *Proceedings of AIK Spring Conference, Planning and Design Section*, 10(1), pp.65-58.
16. Kim, I.H., Choi, J.S. and Cho, M.S., 2004, A Study on an Automatic Building Code Checking System for Architectural Drawings Based on the CALS/EC Standard, *Transactions of the Society of CAD/CAM Engineers*, 9(4), pp.315-324.
17. Song, J., Won, J. and Ju, K., 2012, A Study on Development of Patterns of Rule-based BIM Model Checker Using Quality Check Items of Domestic BIM Guideline, *Design Convergence Study*, 11(5), pp.104-116.
18. Kim, Y., Lee, S.-H. and Park, S.-H. 2012, Development of Rule-Set Definition for Architectural Design Code Checking based on BIM -for Act on the Promotion and Guarantee of Access for the Disabled, the Aged, and Pregnant Women to Facilities and Information-, *KJCEM*, 13(6), pp.143-152.
19. Song, J.-K., Cho, G.-H. and Ju, K.-B., 2013a, A Study on the Rule Development for BIM-based Automatic Checking in a Duct System, *Korean Journal of Air-Conditioning and Refrigeration Engineering*, 25(11), pp.631-639.
20. Song, J.-K. and Ju, K.-B., 2013b, Development of Rule for Quality Checking Items to Raise Quality of BIM Model -Focusing on the Domestic BIM Guidelines-, *KJCEM*, 14(5), pp.131-143.
21. Lee, H., Lee, J.-K., Park, S. and Kim, I., An Approach to Translate the Korean Building Act into a Computer-executable Form for Evaluating Building Permit Requirements, *Automation in Construction*, under review.
22. Kim, I., Kim, Y. and Choi, J., 2014, Building Code Typology and Application for Open BIM based Code Checking, *Transactions of the Society*

- of *CAD/CAM Engineers*, 19(3), pp.224-235.
23. Korea Ministry of Government Legislation, (<http://www.law.go.kr/>)
  24. TIOBE Index for July 2015, (<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>)
  25. Extended BNF, ISO/IEC 14977: 1996(E), (<http://www.cl.cam.ac.uk/~mgk25/iso-14977.pdf>)
  26. Park, S. and Lee, J.-K., Definition of a Domain-

- specific Language to Represent Korea Building ACT Sentences as an Explicit Computable Form, CAADRIA 2016, 2016.
27. Specification of the High-level Methods, Method Manual, (<http://designitlab.kr/bim/methodManual.asp>)



### 박 서 경

2015년 한양대학교 실내건축디자인학과 졸업  
 2015년~현재 한양대학교 실내건축디자인학과 석사과정  
 관심분야: BIM(Building Information Modeling), 디자인컴퓨팅



### 이 진 국

2000년 연세대학교 실내건축학과 졸업  
 2003년 연세대학교 석사  
 2010년 미국 Georgia Tech 건축대학 Ph.D. 디자인컴퓨팅전공  
 2010년~2012년 미국 Georgia Tech DBL(Digital Building Lab) 연구원  
 2012년~현재 한양대학교 실내건축디자인학과 교수  
 2016년~현재 한국CDE학회 이사  
 관심분야: BIM(Building Information Modeling), CAAD, 디자인IT, 디자인컴퓨팅



### 김 인 한

1988년 서울대학교 건축학과 졸업  
 1991년 미국 Carnegie-Mellon 대학 건축학 석사  
 1994년 영국 Strathclyde 대학 건축학 박사  
 1996년~현재 경희대학교 공과대학 건축학과 교수  
 2002년~현재 한국CDE학회 이사  
 2004년~2008년 (사)STEP센터 회장, 지식경제부  
 2008년~현재 (사)빌딩스마트협회 수석 부회장  
 2010년~현재 대한건축학회 이사  
 2014년~현재 대한건축학회 건축정보기술위원회 위원장  
 관심분야: BIM(Building Information Modeling), CAAD, 데이터모델링 및 통합 전산설계환경(STEP, IFC), 건축정보기술, Digital Design Media