Visual Language-based approach for the Definition of Building Permit related rules

Hayan Kim[a] , Jin-Kook Lee[a] , Jaeyoung Shin[a] , Jinsung Kim[a]

[a]Department of Interior Architecture Design, Hanyang University, Seoul, Republic of Korea
E-mail: hayaan92@gmail.com, designit@hanyang.ac.kr, jjyoung311@gmail.com, wlstjd1320@gmail.com

Abstract –

This paper aims to describe the visual language-based approach to define building permit related rules in the much intuitive way and support high applicability. The advent of BIM (Building Information Modeling) enables an automatic rule-checking process in the early phase of design by translation and interpretation of regulation about building design assessment. For the precise rule-checking process, a logic rule-based mechanism for translating is needed as prerequisite step and the rule-making process have to be proceeded based on this mechanism. Accordingly, Korea government also proceeding the KBim project for employing BIM to adapt automatic approach to the overall process of building permission system. As the part of the project, KBimLogic offers logic rule-based mechanism which in charge of rule-making by translating natural language regulation sentences in Korea Building Act into the computer-readable form-KBimCode.

This paper introduces the KBim Visual Language (KBVL), the extended way of composing the rule module of building related regulations by the visual elements for the intuitive and immediate identifying the process and result of rule-making. The language supports user to combine visual symbols and their connection according to the user-defined nesting relation and generate KBimCode automatically at the background of the visual language. Thus, intuitive composing shorten the time of generating rule modules and defining the logical relation between the modules. As the result, the user can handily query the components of KBimCode and define relation of them to compose tailored ruleset for their own rule-checking process with various type of software.

Keywords –

BIM (Building Information Modeling); Building permit related rules; Rule-making; Visual language; Rule definition

1 Introduction

The conventional building design assessment of legislations mandatorily requires handling of human resources. Since the sentences of building regulation are written in natural language, it includes expressions with ambiguity and vagueness [10]. Therefore, each sentences requires precise interpretation by the experts, and rule-checking phase spends a plenty of money and time. The introduction of BIM (Building Information Modeling) enables those manual process to be automatized by computer-based 3D and information modeling. Especially in the field of evaluating building related regulation based on BIM model, establishing logic for translating natural language sentences into the computer-readable format became the forefront of the assessment [1, 2]. Especially, the Korea Building Act has multiple clauses and phrases with intricate relations such as upper-lower, reference, and commission [3]. To read natural language in the building design assessment program, Korea government includes the phase of research about translating and interpreting Korea Building Act in the KBim project for the e-submission system with BIM model [11,12].

The overall project is summarized with three features-1)developing overall process for design assessment, 2) developing standard and practical technology for applying BIM-based design plan, 3) developing cooperation working system to integrate design and engineering. As the part of the project, KBimLogic is specifically devised to translate Korea Building Act into the KBimCode – intermediate code for executing regulation in the rule-checking software.

Although the KBimCode supports the computer-based format of the building code, there are still high thresholds for domain workers who are non-expert about programming knowledge to write code directly. For the much higher usability of KBimLogic and KBimCode, this paper aims to introduce a visual language-based approach to define building permit related regulation much intuitive way. Consequently, this paper includes the approach based on the backdrop

research – KBim project that has been led by the government of South Korea. By the utilizing preceded research, this paper suggests the graphical approach to practically use the KBimLogic and KBimCode.

2 Related work

The state of the rule-making approaches for building design requirements in recent studies are typically with two methods, one is hardcoding-based scripting language and another is visual language. By the reviewing main feature of representative researches, the necessity of visual approach acquires validity. Through the recent researches using visual language-based approach for rule-making, KBVL establishes its specific characteristic different with existing approach and acquires validity for utilizing visual representation.

2.1 Hardcoding-based rule-making approach

The advent of BIM in the field of architecture, engineering, construction and facility management (AEC-FM) enables computer-readable and analyzable building model in the computer-based environment. Automating rule-checking also became possible by the swift development of BIM application and related technologies [13,14]. For structuring the checking process, the prerequisite phases are needed - interpreting and translating regulation into the computer-readable format. In the early 2000s, several pacesetting projects had been proceeded at the government level for their own software system or language framework based on their national regulations and design guideline.

- As the part of CORENET (Construction and Real Estate NETwork) project (2005), e-PlanCheck takes charge of automatic rule-checking process with the IFC building model and their own platform, FORNAX. This platform employs C++ based ruleset library and both expert of Building Act and computer programmer cooperate for composing code [9].
- SMARTcode project (2006) utilizes Markup method to distinguish noun phrase, verb phrase, and the context in the sentences. As the next step, components of analyzed sentences are translated as structured XML file maintaining original meaning of sentences [8].
- GSA project (2007) is for the federal courthouses building design assessment project using BIM application and led by the USA government. The part of the design guidelines for building plan are translated as computer-readable form for rule-checking program. The rules of indoor walkability and circulation are checked automatically by the generated circulation graph [4].

2.2 Visual language-based approach

The preceded projects about rule-making process in the early 2000s are typically based on the specific software dependent algorithm or require expert knowledge about computer programming. Since the most of domain expert in architecture field are not familiar with programming language, previous process is difficult to approach for the building related workers. This is why it may be less accessible and useful to the architects.

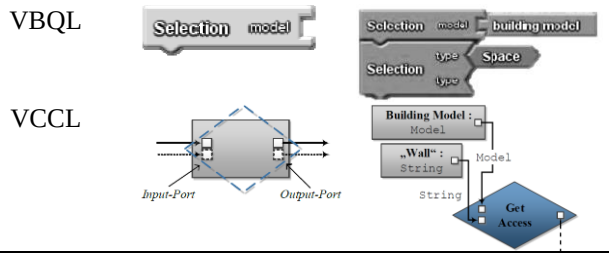
As the solution, visual language is effectively employed for the supportive and higher usable language framework because of its advantages – intuitive, real-time modification, and parametric outcome as the consequences. The early phase of introduction of visual language in the field of the building has been focused on the transformation of building geometry. The Dynamo for Revit or Grasshopper for Rhino has been role as the frontier for introducing geometry focused visual language in the BIM authoring tools.

As the higher level in the AEC-FM fields, the visual language activates the rule composing by domain workers who are not familiar with computer programming language for building permission system. Especially the visual language effectively works for querying and combining the components for composing ruleset. Querying specific code component of objects and composing them as ruleset file for computer-readable format demands language-specific knowledge about the language's own grammar, vocabulary, and exceptional rule. Therefore, there steadily have been researching to visually represent the process of rule-making and checking.

- VBQL (Visual BIM Query Language) has been developed for the craftsmen who are not familiar with text-based programming knowledge to formulate model queries [7]. This language considers not only for the building model but also cost and time model to deals with 5D BIM model for the different fields and phases.
- The VCCL (Visual Code Checking Language) represents information using various type of methods and tested with the BIM+ [6, 15]. As the main attributes of VCCL, genericity and finest granularity are typically dealt as important aspects.

Table 1 Comparing the visual representation of the general node and node combination of VBQL and VCCL

Visual Language	Visual characteristic of the general node	Connected component



2.3 Script language-based rule-making

As the research for adapting BIM technology in the building permission system, several countries are proceeding government-based research project [4,8,9]. The Korea government also proceeding the KBim project that for establishing BIM-enabled building design assessment system based on the Korea Building Act[11,12]. As the part of overall project, logic rule-based mechanism called KBimLogic is developed to translate natural language of Korea Building Act into computer-readable format called KBimCode [5]. This paper aims to introduce KBim Visual Language(KBVL) for the improvement of the KBimCode usability.

3 Development Approach of KBVL

3.1 Fragmentizing building permit related rules

The building permit related regulations are descripted in not only the Building Act but also in various type of document such as Building Act, Request for proposal (RFP), and international design guideline with their own grammar and vocabulary. As the minimum requirement for the building rule-checking, Korea Building Act is finally checked for the building permission.

For the precise representing of information as visual symbol without contents omission, sentences have to be fragmentized as atomic sentences to seize the logical relation between multiple clauses. The sentences of the building related regulation essentially include the key statement (KS) appointing the statement of the object. Also, there is the condition statement (CS) appointing the conditional statement for the key statement.

In the building regulation, building object is the target of verb phrase. Also, there are common verb phrases that used frequently for checking building object such as the measuring physical geometry, identifying position relation between other object, or whether specific object is installed or not.

The fragmentized sentence element collected as the component database for the establishment of standardized library comprising the objects, their properties, and the high-level methods.

3.2 KBVL Grammar

The grammatical logic structure of the KBVL is based on the connecting relation of visual element.

Figure 1. The sentence elements and they are covered by a) object node b) method node c) logic node d) rule node

The code in (1) shows the example of KBimCode generated by composing nodes explained in Figure 1.

[getObjectCount(Window) = 5] OR (1)
[hasSpace(Room, Window) = TRUE]

Unlike uncomplicated checking process of single building object, the checking regulation with constrain relation with other object needs logical statement and nesting process as necessary phase. If the method nodes define the relation between objects, the logic nodes decide the relation between rule nodes. The nesting process prominently removes the repetitive node combination by binding several user defined properties in the common object and offers reusability of the node including specific condition.

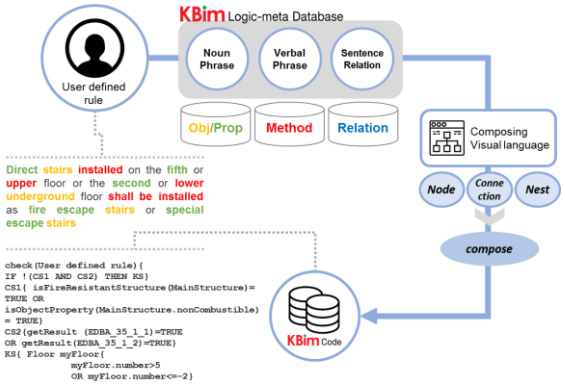


Figure 2. The process of utilizing KBimLogic meta database and generating KBimCode

Also, the KBVL basically utilizes the KBimLogic meta database as the semantic library of building object, property, and method. This database includes largely 5 main databases that established by the KBimLogic. Based on the Korea Building Act, database is composed of sentence DB, object and property DB, method DB, logical relation DB, and KBimCode DB.

In this paper, KBVL is related with object and property DB and method DB. The main visual element – node – queries each nodes' semantic information and function from the database.

4 Component of KBVL

KBVL intuitively displays the flow of sentence component with the basic visual language component. The each visual components of KBVL includes the fundamental and physical connecting function and also supports users to generate KBimCode much easily than just describing the script-based code with hard-coded method.

4.1 Node

The visual language basically represent the main structural part with node component. Node generally corresponds to the specific class and creates semantic attributes by the connection between other nodes. Also through the importation and exportation of the data through the port, visual program transfer and accumulate the information over the node combination and finally deduces the result.

According to the type, the nodes have specific function to handle input data or hold the condition of object in the runtime of the visual language process. The KBVL nodes are divided as four type according to the function – 1) object 2) method 3) logic 4) rule.

4.1.1 Object Node

Object and property node queries building object class which commonly defined in the KBimLogic meta database considering 'IfcBuildingElement' structure. The object class primarily includes the main components of the building such as "Window", "Stair", and "Wall" etc. like shown in Figure 3. Property is inseparable from the object. In other words, each object has none or many properties but each property has to be attached at an object. Object nodes are connected with method node to generate minimum unit of rule node.

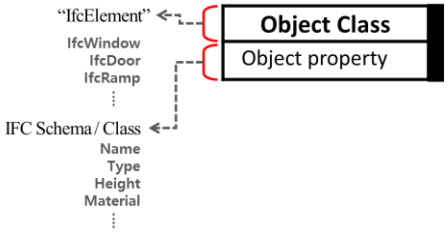


Figure 3. Semantic structure based on the IFC framework of the object node

4.1.2 Method Node

Method node perform specific action on the inputted object node. Depending on the type of method, one or many object and property nodes have to be connected as parameter of method at input port of the node. By the specific action, result value is exported through output port.

Method node is composed of four types of contents lines – method name, parameter type, operator, result. To reflect the feature of rule-making that result have to be defined as specific type, method node includes operator and result. Basic three type of method node is descripted in Table 2.

Table 2. Method node classification based on the type of return value data type

Return value	Left Operand	Operator	Right Operand
Boolean	Methods querying boolean value	=	True, False
Numeric	Methods querying numeric value	<, <=, =, >=, >	Numeric
Collection	Methods querying the name of type	=	String

Transmitted data of the object node is combined with method node to generate basic rule unit. Several rule units arc composed with logic node to generate rule node including result value. Based on the return value type of table 2, method node is composed of method name, parameter type, operator and result value. Method name refers the exact method from the method database. Parameter type, operator and result (return value) is defined in the database. According to the method, parameter type can be one to many.

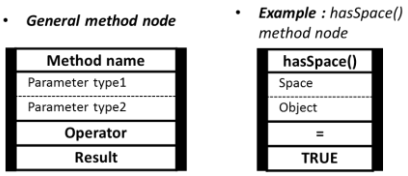


Figure 4. Structure of method node and example

4.1.3 Logic Node

Logic node supports rule node and unit to have logical relation between them. Especially the building permit related regulations are composed of plenty of sentences and difficult to analysis. Therefore, it is important to set up the organized logical relationship between subdivided sentence elements.

Based on the 3.2 section, building permit related regulations are summarized as AS and again subdivided as TAS structure. The TAS is composed of ALU level of semantic combination that represented by the single rule unit. As the logical relation generated in (2) sentences, logic node is necessary for semantically connect the rule units.

IF (CS1 OR CS2) THEN (KS1 AND KS2) (2)

4.1.4 Rule Node

Rule node is composed of multiple nodes and their connection. Because of the complicate logical statement and conditional relationship, rule node bound as nesting box is necessary for brief node combination. Rule node R is defined by (2). N is the set of node composing rule node R and C is the set of connection between N composing R.

$$R = (N, C) \quad (3)$$

$$N = \{ n_i \mid 1 \le i \le n;\ n = \text{number of node composing R} \} \quad (4)$$

$$C = \{ c_i \mid 1 \le i \le n;\ n = \text{number of connection between N composing R} \} \quad (5)$$

4.2 Connection

Multiple nodes are linked by connection component and composed as the node combination. Connection creates the relationship between nodes and exchanges the data of the nodes for their own function. By establishing dependencies between the components, visual program creates possible flow of information through the node combination. Connection also transfer the data of the node with the direction flowing from left to right.

The direction of data flow appears in the characteristics of shape of the nodes. Object node is placed on the left side of the method node and only has the output port on the right side of the node context. Differently with object node, method node has both input and output port to import and export the data through the ports.

4.3 Nesting for Rule node

The complex relation between multiple rule units can be decreased by the degree of nesting. By the nesting function, two visual language components can be generated.

- Rule node
- Predefined object node

Figure 4 shows how to generate logical relationship between the nodes and nesting box. The KBimCode automatically generated at the background of KBVL.

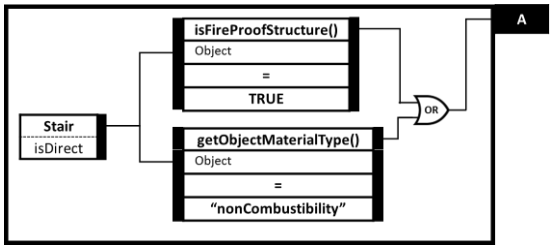


Figure 4. Apprearance of nesting box that is related with generation of rule node.

5 Demonstration

Based on the approach explained in previous section, simple rule and complex regulation are represented with KBVL. By analyzing rule sentence and composing nodes with connection, user easily write KBimCode.

5.1 Target regulation and sentence analysis

The design regulations of building include the usage of building, site, structure or facility. Among the various type of regulation, the demonstration uses two type of regulation. First sentence is restructured design regulation composed with the common building element such as floor, stair, or window etc. This simple rule is checked for the general and common demand occurred among the domain workers. By this demonstration, usability of KBVL for general and simple KBimCode generation is proved.

- **(Rule A)** Fifth or upper floor or the second or lower underground floor shall be installed direct stairs as escape stairs.

As the next step, much complex regulation referred in Korea Building Act that has reference to other regulation or including object with complex property are represented with KBVL. Regulation is the part of EDBA (Enforcement Decree of the Building Act) 34-3 notifying the standards of installation of direct stairs. Among the original texts of regulation, the definition of specific object is eliminated with the assumption that the object database already includes the definition.

- **(Rule B)** In cases of installing a window that can be opened or closed less than 1.2 meters high over the surface of the living room in an officetel, a safety facility designed to prevent falling accidents shall be installed in conformity with the standards prescribed by Ordinance of the Ministry of Land, Infrastructure and Transport.

Consequently, this section assumes that user have to make two kind of rules including simple condition of building element and complex connection relationship between the multiple target element.

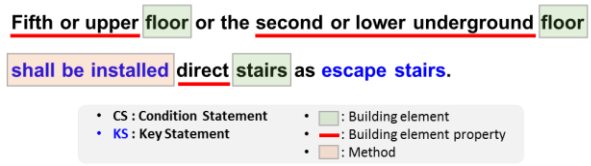


Figure 5. (Rule A) Sentence analysis for arrangement of node and their connection.

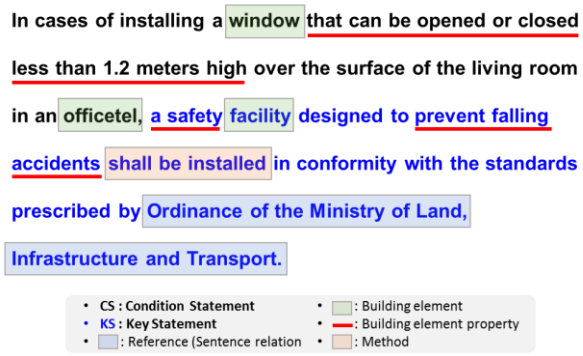


Figure 6. (Rule B) Sentence analysis for arrangement of node and their connection.

The KBVL composing process starts with the step of distinguishing the building elements included in sentence. Also, according to the type of object and requirement descripted in sentence, method for object is chosen. Sentence analysis of [Rule A] and [Rule B] is represented in Figure 5 and 6. Each building element has property as the condition and connected with other element by method node. By the CS and KS, regulation sentences involving condition for the result.

5.2 Node combination

According to the analysis of sentences, (Rule A) is verified as checking property of specified stairs. (Rule B) is much complex than (Rule A) and require more nesting function to generate rule unit.

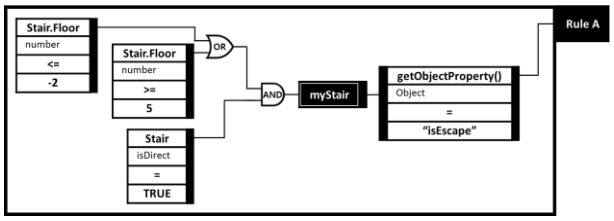


Figure 7. KBVL node composition of (Rule A).

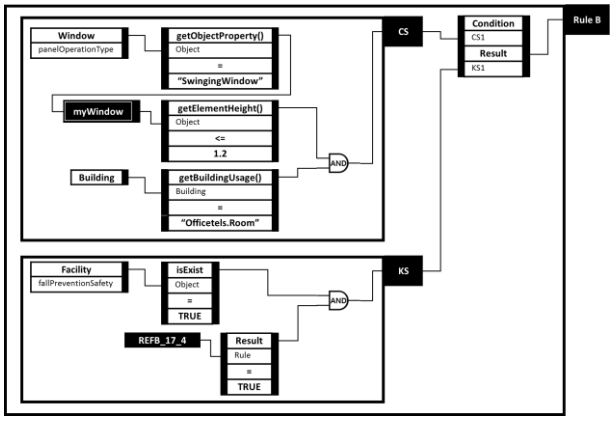


Figure 8. KBVL node composition of (Rule B).

5.3 Result

Through the node combination, (Rule A) can automatically generate KBimCode like Figure 9 and (Rule B) as Figure 10.

```
Check(Rule A)

Stair myStair{
Stair.Floor.number <= -2
OR Stair.Floor.number >= 5
isObjectProperty(Stair.isDirect) = TRUE
}

getObjectProperty(myStair) = "isEscape"
```

Figure 9. Generated KBimCode of (Rule A).

```
Check(Rule B){
        IF CS THEN KS
}
CS{
        Window myWindow{
        getObjectProperty(Window.panelOperationType) = "SwingingWindow"
        }
        getBuildingUsage() = "Officetels.Room"
        getElementHeight(myWindow) <= 1.2 m
}
KS{
        isExist(FallPreventionSafetyFacility) = TRUE
        getResult(REFB_17_4) = TRUE
}
```

Figure 10. Generated KBimCode of (Rule B).

Unlike script based KBimCode generation, KBVL supports user to find which object is the target object of the regulation and which action is executed for checking result. Also by composing several rule units as the rule node, clear distinguish between condition statement and key statement is offered by nesting function.

6 Conclusion

This paper describes a visual language based approach to support the generation of KBimCode which

translated computer-readable format for building permit related rules. As an approach for visualizing regulations, sentences are precisely analyzed and decomposed into objects, properties, methods and logical relations from noun phrases. Logically restructured components are classified with semantic and syntactic attributes and hierarchically arranged. To structure logical process for visually representation of the regulations, preceded projects are reviewed and the common components used for visual representation are deduced as the visual components of KBVL. By utilizing node and connection, the components extracted from regulation sentences are again composed of a node combination.

The KBimCode is based on the scripting method, therefore it is hard to understand the main target object and context at once. As an updated method, KBVL offers the way of composing node combination to generate KBimCode automatically at the background of the visual language. This approach can be extended to not only for the building permit related regulation but the other types of regulation documents such as design guideline or RFP. Also, the extension of KBimLogic meta database, the usable object, property and method node also can be extended to the new type of components.

Acknowledgement

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2015R1C1A1A01053497)

References

[1] Eastman, C., Lee, J. M., Jeong, Y. S., and Lee, J. K. Automatic rule-based checking of building designs. *Automation in construction*, 18(8):1011-1033, 2009.

[2] Lee, J. K., Eastman, C. M., and Lee, Y. C. Implementation of a BIM domain-specific language for the building environment rule and analysis. *Journal of Intelligent & Robotic Systems*, 79(3-4):507, 2015.

[3] Kim, I.H., Kim, Y.H. and Choi, J.S. Building Code Typology and Application for Open BIM based Code Checking, *Transactions of the Society of CAD/CAM Engineers*, 19(3):224- 235, 2014.

[4] GSA Project- BIM Enabled Design Guide Automation, On-line: http://dcom.arch.gatech.edu, Accessed: 15/12/2016.

[5] Lee, H., Lee, J. K., Park, S., and Kim, I. Translating building legislation into a computer-executable format for evaluating building permit requirements. *Automation in Construction*, 71:49-61, 2016.

[6] Preidel, C., and Borrmann, A. Automated Code Compliance Checking Based on a Visual Language and Building Information Modeling. In ISARC. In *Proceedings of the International Symposium on Automation and Robotics in Construction* Vol. 32, page 1, 2015.

[7] Wülfing, A., Windisch, R. and Scherer, R.J., A visual BIM query language. *eWork and eBusiness in Architecture, Engineering and Construction*: ECPPM 2014:157, 2014.

[8] J. Wix, N. Nisbet, and T. Liebich, Using constraints to validate and check building information models, *eWork and eBusiness in Architecture, Engineering and Construction*, ECPPM, London:467–475, 2008.

[9] T.F. Sing, and Q. Zhong, Construction and real estate NETwork (CORENET), Facilities 19:419–428, 2001.

[10] Nawari O. N. Automated Code Checking in BIM Environment, In *Proceeding of 14th International Conference on Computing in Civil and Building Engineering*, 2012.

[11] Seumter, BIM-enabled e-Submission & Building Act Legality Verification System of Korea, On-line: https://www.eais.go.kr Accessed: 10/12/2016.

[12] I. Kim, Open BIM based Technological Environment for Building Design Quality Enhancement, Kyung Hee Univ., (Project report), 2015

[13] J. Choi, and I. Kim, A Study on the Application of Pre-Processing to Develop the Open BIM-based Code Checking System for Building Administration Process, *Journal of the Architectural Institute of Korea Planning & Design*, 30(9):3-12, 2014.

[14] I. Kim, Y. Kim, and J. Choi, Building Code Typology and Application for Open BIM based Code Checking, *Transactions of the Society of CAD/CAM Engineers*, 19(3):224-235, 2014.

[15] Preidel, C., and Borrmann, A. Integrating Relational Algebra into a Visual Code Checking Language for Information Retrieval from Building Information Models. In *Proceedings of the 16th International Conference on Computing in Civil and Building Engineering*. Osaka, Japan, ICCCBE, 2016.