Logging Modeling Events to Enhance the Reproducibility of a Modeling Process

Suhyung Jang[a], Sanghyun Shin[a], Ghang Lee[a*]

[a] Building Informatics Group, Department of Architecture and Architectural Engineering, Yonsei University, Republic of Korea
E-mail: rgb000@yonsei.ac.kr, spicysh03126@yonsei.ac.kr, glee@yonsei.ac.kr
*Corresponding author

Abstract

This study was aimed at developing a building information modeling (BIM) process logger that can capture modeling process information as an event log file. BIM log mining is a research area that focuses on utilizing a massive amount of data created from BIM software usage. Several studies have monitored, analyzed, improved, and predicted modeling process based on BIM log mining. However, the BIM logs recorded using current BIM authoring tools do not offer explicit information about modeling commands and object-related parameters. The BIM logger proposed in the current work was developed using the application programming interface (API) of BIM authoring software to create an enhanced log file that can represent the modeling process. The reproducibility of the modeling process in the log file created using the developed logger was evaluated via quantitative and qualitative methods.

Keywords

Building Information Modeling (BIM); BIM Log Mining; Modeling Process Representation; Data Enhancement

1 Introduction

With a massive volume of data created from digitalized industry platforms, a data analysis strategy that utilizes the data to monitor, analyze, improve, and optimize a process is currently in the limelight [1]. Building information modeling (BIM) has become an operational core of the architectural, engineering, and construction (AEC) industries, working as the digitalized platform of the industry. The growing utilization of BIM applications has resulted in the accumulation of a massive volume of computer-generated data regarding the process [2]. One type of such data is BIM logs. BIM logs automatically record BIM usage data in a BIM authoring tool.

BIM log mining is a young discipline in a research area that focuses on utilizing the event logs created from the BIM usage process. Extensive research has been conducted to perform BIM log mining for various purposes, such as design pattern analysis, design productivity analysis, social network analysis, modeling process visualization, and modeling command prediction.

The data maturity of BIM logs, however, has not been questioned in most previous research [3–11], even though it very much affects the results of analyses in event log mining. The quality of an event log varies according to the purpose of the analysis [12]. For reliable and precise modeling process analysis, the modeling information in BIM event logs needs to be self-descriptive to the level at which it can reproduce the process [1]. During a literature review, we verified that the modeling process data created in the current BIM log file is not explicit enough to reproduce the modeling process for process analysis [13]. In this regard, we focused on increasing the process reproducibility of the modeling log using a custom-developed BIM logger.

In this paper, we propose a BIM logger to create a modeling log that can reproduce the modeling process with the log, itself. In the next section, previous research on BIM log mining and its data usage is reviewed. In the third section, the research method used in this study is described. In the fourth section, the development sequence and algorithm of the modeling logger are explained. In the fifth section, the reproducibility of the log created from the logger is evaluated. Section 6 concludes the paper, along with its contributions, limitations, and future research.

2 Literature Review

Parametric modeling is a process that enables designers to translate and embed domain knowledge about original modeling processes as geometric expressions that can automatically generate modeled objects in modifiable form [14]. Designers can manipulate parametrically modeled objects during or after a modeling process by modulating geometrical attributes and relationships between objects [15]. Parametric modeling is distinguished from BIM logs in that it is implemented on the basis of a designer's knowledge of the modeling process. Conversely, BIM log files record each and every process during modeling process.

Recently, several researchers have made efforts to utilize BIM logs to solve issues during the design phase. Yarmohammadi et al. proposed a BIM log mining method to attain insights into the design process [2][13] by analyzing Revit journal files and BIM log files provided by Revit from Autodesk [16].

Zhang et al. proposed methods to evaluate the modeling productivity of project participants by analyzing the design patterns retrieved from the Patricia tree structure [4]. Pan et al. conducted research on design productivity analysis by identifying implicit command execution patterns from BIM logs using the Efficient Fuzzy Kohonen Clustering Network (EFKCN) algorithm [6] and the Adaptive Efficient Fuzzy Kohonen Clustering Network (AEFKCN) [9].

An analysis of collaborations among designers and the relationships between collaborative` characteristics using a social network analysis was another research topic suggested by Zhang and Ashuri [3], and Pan et al. [10].

Modeling command prediction was the most recent research topic in BIM log mining. It was conducted with the application of recent deep learning techniques, such as the long short-term memory neural network (LSTM NN) and recurrent neural network (RNN) by Pan et al. [7] [8].

In all previous research reviewed, the Autodesk Revit journal file was the only source of the analysis. As shown in 'Figure 1', a command executed on the Revit system is specified by the type of general command, event, and specific command. *General command* explains how an activity is executed in the system. *Event* explains whether the command is creating, deleting, or modifying an object. *Specific command* discriminates a command from other commands by a given identifier (ID). From the example, however, the geometric attributes of the modeled object, thickness, or height of the wall, in this case, cannot be verified. Moreover, the object affected by the activity is not specified in the example. The modeling information provided by the Autodesk Revit journal is not sufficiently specific to reproduce the whole modeling process.

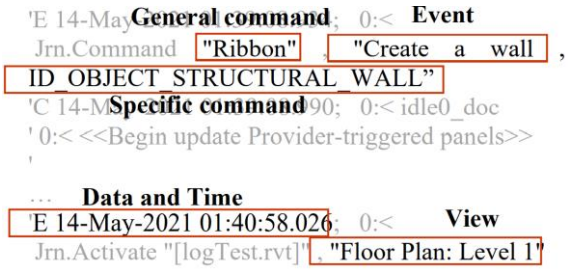

Figure 1. Example of the Revit journal file

Further reinforcement of information about modeling commands and modeled object attributes is required to reproduce an already-created model. With sufficient modeling information that can reproduce modeling processes and results, the accuracy and reliability of BIM log mining results can be highly increased. The objective of the study is to enhance the modeling information in the BIM log to where the representation of the original modeling process is possible.

3 Research Method

The research method of this paper comprises two steps, as illustrated in 'Figure 2':

1. Development of the BIM logger to enhance the modeling information described in Section 4.

2. Evaluation of the enhanced log described in Section 5.

The development of the BIM logger consists of two phases, including real-time logging and capturing the object information. Through the two phases, the modeling process logger is developed. The details of each phase of the development are explained in the next section.

In the evaluation section, the enhanced log created from the logger is evaluated. The goal of BIM log enhancement in this study was set to the level at which repeating the original modeling process is possible. To validate the equality of the process between the original modeling result and the represented modeling result, the equality of the modeling process is defined. A standard to evaluate the reproducibility of the modeling process is introduced in Section 4.

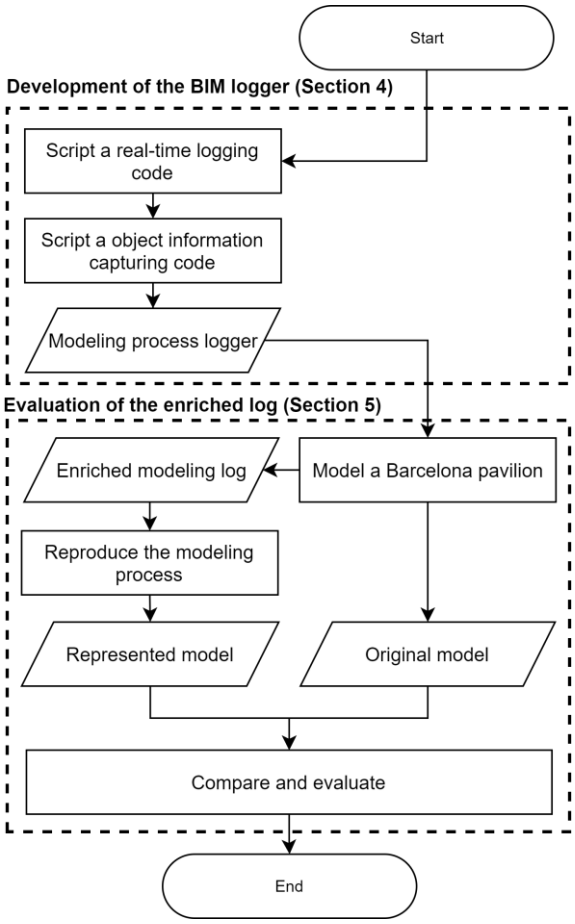

Figure 2. Research flowchart

4 Development of a BIM Logger

Rhinoceros (*Rhino*, for short) [17] is a widely used software for architectural designers in the early stages of the design process. This study used Rhino and Grasshopper—a visual programming module of Rhino—to develop a BIM logger with enhanced modeling information.

To conduct event log mining, the activity, timestamp, and case ID are required [1]. Activity in the event log specifies which activity has taken place. A timestamp records the precise time of each activity. A case ID is a numeric identifier assigned to a unique case.

In the case of modeling event log mining, activity is the modeling-related command. The timestamp is the record of the time when a modeling command is executed. Case ID is defined by the globally unique identifier (GUID) of each modeled object. The following subsections describe how these information items were recorded.

4.1 Real-Time Logging

Rhino provides a modeling history that works as a user-system communication board. The modeling history shows the process of each command execution. As shown in 'Figure 3', the modeling history of Rhino includes a command, subcommand, command options, parameters, system message, and result message. The modeling history shows the ongoing command after "Command:" and reveals the result of the command. The user can easily understand the status of the actions running on the system. It is also possible for users to verify the options and input parameters from the modeling history so they can decide which action to take. We built an algorithm to extract and record a modeling history through the Python-based Rhino API [18] while a modeler is creating a design. Grasshopper executes the script in real-time and stacks each command event. The BIM logger captures commands, subcommands, and parameters from the modeling history, timestamp, and object information. The BIM log can be exported as a comma-separated values (CSV) file. More details on an algorithm to capture object information are described in the next section.

4.2 Capturing Object Information

During the modeling process, a modeled object is affected by the modeling command being created, deleted, or modified. Thus, object information is core information that needs to be clarified to reproduce a modeling process based on a BIM log. Nevertheless, neither the Rhino modeling history nor the Revit journal file includes object data.

The BIM logger developed in this study captures an object's GUIDs to record objects created or changed through a modeling process. Each object in all BIM authoring tools, including Rhino, has a GUID to discriminate a modeled object from other objects [19].

To add GUID information to the BIM log, a Python script was written. The script identifies the GUID of currently modeled objects by comparing the total list of GUIDs of modeled objects. The total list of GUIDs is recorded based on real-time updates from the Rhino system on command execution. If a command creates an object, the GUID of the created object will be added to the list of GUIDs. The script, therefore, can verify which object is either created, deleted, or modified on command executions.

The developed algorithm of enhanced BIM logger is presented as pseudocode in 'Figure 4'.

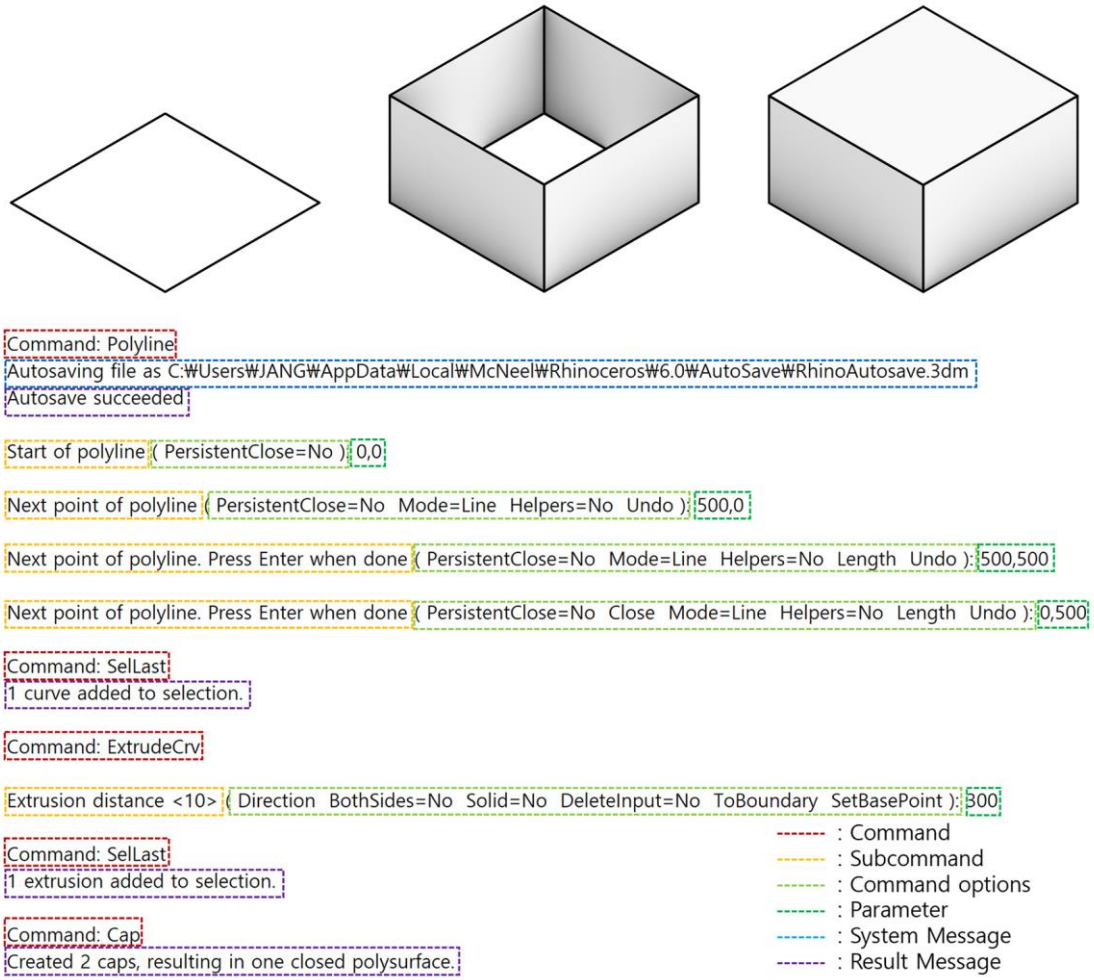

Figure 3. A modeling process and the modeling history of the process

5 Evaluation

This section describes how the reproducibility of the created log was evaluated. Since there is no standard or common method to evaluate the reproducibility of a model, this study proposes the level of reproducibility (LOR) and the intersection of union (IoU) as measurements. Section 5.1 describes LOR. Section 5.2 reports the LOR of the enhanced BIM logger through a test case. Section 5.3 compares the modeling information included in the existing BIM log and the enhanced BIM log.

5.1 Level of Reproducibility

This study proposes LOR to quantitatively measure the extent to which a model is reproduced from BIM log data. The algorithm for calculating the LOR is as follows:

For the i^{th} object (o_i) from the original modeling process, if the object o_i exists in the represented model, the value of a variable X_i is 1 and otherwise 0.

Additionally, IoU is used to measure the geometrical similarity between an original object and a reproduced object. IoU is a commonly used metric for measuring the overlap between two bounding boxes or masks by calculating the intersected area divided by the unified area of the boxes or masks.

This study used the volume of the intersection and union of the object's volume instead of the area, (see Equation (1)). After the i^{th} IoU (IoU_i) is calculated, the LOR of the model total sum can be calculated, (see Equation (2)).

```
Input: modeling history
Output: modeling log

log_ord_list: list of event log orders
cmd_list: list of commands
time_list: list of timestamp
guid_list: list of GUID
cmd_ord_list: list of command orders
prm_list: list of parameters

for i in modeling history do
        cmh EQUALS i;
        log_ord EQUALS log_ord plus 1;
        if the first word of string cmh is "Command" then
                cmd EQUALS string after ": " ;
                cmd_ord EQUALS cmd_ord plus 1;
        else if the string cmh is not command related message then
                go to next i;
        end if
        if the second last word of cmh includes ": " then
                parameter EQUALS string after ": ";
        else
                parameter EQUALS "-";
        end if
        time EQUALS current time;
        guid EQUALS current object GUID;
        APPEND log_ord to log_ord_list;
        APPEND cmd to cmd_list;
        APPEND time to time_list;
        APPEND guid to guid_list;
        APPEND cmd_ord to cmd_ord_list;
        APPEND parameter to prm_list;
end for
modeling log EQUALS [log_ord_list, time_list, guid_list, cmd_list, cmd_ord_list, prm_list]
return  modeling log
```

Figure 4. Pseudo code of the modeling logger

$$IoU_i = V_{Inter_i} / V_{Uni_i} \quad (1)$$

$$LOR = \frac{\sum_{n=1}^{i} X_i * IoU_i}{i} * 100\ (\%) \quad (2)$$

5.2 Evaluation of Model Reproducibility

For comparison, the Barcelona Pavilion, designed by Mies van der Rohe, was modeled. A total of 288 instances of commands and subcommands were used, including Polyline, Box, Cylinder, Boolean Difference, Extrude, Cap, SelLast, and Delete. As shown in 'Figure 5', the original modeling process created 27 modeled objects. The modeling process was represented based on the enhanced log from the logger developed in the previous section through execution of modeling command by code. Among 27 modeled objects, the reproduced model included 24 model objects with 21 correctly created model objects and three incorrectly created model objects. The modeling process reproducibility of the enhanced log shows 64.08% LOR with an average of 0.98 IoU_i.

5.3 Comparison of Log-Containing Information

'Table 1' shows modeling information included in the Revit journal file. Although the file stores data on

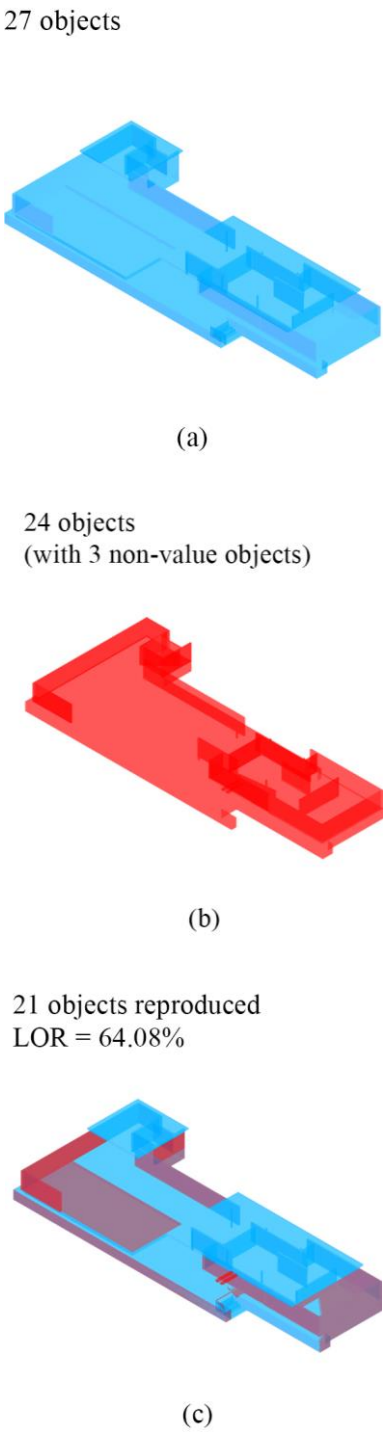

Figure 5. Comparison of the original model and the represented model

modeling commands, it does not sufficiently represent the modeling process because it lacks a discrimination of commands. For example, it is impossible to examine differences between commands if there is a command that creates a wall that is two meters long and one that creates a wall that is three meters long. In addition, because an object ID or attribute is missing, the file is unsuitable for specifying which modeled object is affected by command execution. Meanwhile, 'Table 2' indicates that the modeling information, including GUID, commands, command orders for subcommand discrimination, and parameters, is enhanced. With GUID included in the enhanced log file, the object instance of a modeling command can be specified. The Revit journal file and enhanced modeling log both provide the names of commands, but the enhanced modeling log provides the explicit parameters included in commands. For example, if a command intended to create a box from a width, a depth, and a height is recorded by the enhanced modeling logger, the log created from the process will not only record the box creating the 'Command' and 'GUID' of the created box but also record the width, depth, and height as 'Parameter' in separate event logs marked as the same command by 'Command Order'. An enhanced modeling log therefore provides more explicit data about the modeling process than that generated by an ordinary BIM log file.

6 Conclusion

Current BIM logs do not include sufficient data to reproduce a modeling process. This study proposed a BIM logger that could reproduce a modeling process. Enhanced modeling logs from the BIM logger include specified modeling data. The enhanced log is eligible for reproducing the original modeling process at a reasonable level, as shown in the evaluation section.

This study contributes to the body of knowledge by presenting a BIM logger that creates an enhanced log that can reduce the gap between real-life modeling processes and logged modeling processes. The enhanced log, which provides specific modeling information, can be used as a source of future BIM mining research to attain more reliable and meaningful results.

The results of this study demonstrated the possibility of reproducing a BIM model from a BIM log, which cannot be implemented using current BIM logs. However, this study is limited in that the proposed method was examined only on the basis of a small test case. The reproducibility of a model should be tested through full-scale projects to demonstrate the practicality of the proposed method.

Table 2. Modeling information from the Revit journal file

General Command	Event	Specific Command
Ribbon	Create	ID_OBJECTS_STRUCTURAL_WALL
KeyboardShortcut	Other	ID_TOGGLE_PROPERTIES_PALETTE
Ribbon	Create	ID_FILE_NEW_CHOOSE_TEMPLATE
Ribbon	Create	ID_VIEW_DEFAULT_3DVIEW
Internal	Other	ID_DETAIL_LEVEL_MEDIUM
Internal	Other	ID_IMAGE_SHADING
AccelKey	Other	ID_CANCEL_EDITOR

Table 1. Modeling information from the enhanced modeling log

GUID	Command	Command Order	Parameter
08c83585-f792-4d43-a594-17d2ba855bea	Polyline	0	1
08c83585-f792-4d43-a594-17d2ba855bea	SelLast	1	-
5d01173f-2f3e-4ff8-9943-eb7b9c91d945	ExtrudeCrv	2	-
5d01173f-2f3e-4ff8-9943-eb7b9c91d945	ExtrudeCrv	2	-1200
5d01173f-2f3e-4ff8-9943-eb7b9c91d945	SelLast	3	-
5d01173f-2f3e-4ff8-9943-eb7b9c91d945	Cap	4	-
9e7b8788-0dab-4c69-86b0-571820582c0d	Polyline	5	-
9e7b8788-0dab-4c69-86b0-571820582c0d	Polyline	5	0/0

Even though the aim of developing an enhanced BIM logger is to attain a more reliable result from BIM log mining, the application of enhanced logs was not examined in this study. Future research can involve the application of enhanced logs in tasks such as the analysis or prediction of modeling processes. In addition, a more comprehensive BIM logger can be developed to collect not only geometric attributes but also non-modeling events and non-geometric features.

Acknowledgments

This work was funded by the Korean Ministry of Land, Infrastructure, and Transport (MOLIT) (No. 2019-0-01559-001, Automation of Architectural Design Process Using Artificial Intelligence), Korea Agency for Infrastructure Technology Advancement.

Reference

[1] Reinkemeyer L. *Process Mining in Action: Principles, Use Cases and Outlook*, Springer Nature, 2020.

[2] Yarmohammadi S., Pourabolghasem R., Shirazi A., et al. A Sequential Pattern Mining Approach to Extract Information from BIM Design Log Files. In *33th International Symposium on Automation and Robotics in Construction*Auburn, AL, USA, 2016.

[3] Zhang L. and Ashuri B. BIM log mining: Discovering social networks. *Automation in Construction*, 91:31–43, 2018.

[4] Zhang L., Wen M., and Ashuri B. BIM Log Mining: Measuring Design Productivity. *Journal of Computing in Civil Engineering*, 32:04017071, 2018.

[5] Zhang N., Tian Y., and Al-Hussein M. A Case Study for 3D Modeling Process Analysis based on BIM Log File. *Modular and Offsite Construction (MOC) Summit Proceedings*, :309–313, 2019.

[6] Pan Y. and Zhang L. BIM log mining: Exploring design productivity characteristics. *Automation in Construction*, 109:102997, 2020.

[7] Pan Y. and Zhang L. BIM log mining: Learning and predicting design commands. *Automation in Construction*, 112:103107, 2020.

[8] Pan Y. and Zhang L. Sequential Design Command Prediction Using BIM Event Logs. :306–315, 2020.

[9] Pan Y., Zhang L., and Li Z. Mining event logs for knowledge discovery based on adaptive efficient fuzzy Kohonen clustering network. *Knowledge-Based Systems*, 209:106482, 2020.

[10] Pan Y., Zhang L., and Skibniewski M.J. Clustering of designers based on building information modeling event logs. *Computer-Aided Civil and Infrastructure Engineering*, 35:701–718, 2020.

[11] Ishizawa T. and Ikeda Y. Visual log analysis method for designing individual and organizational BIM skill. 2021.
[12] van der Aalst W., Adriansyah A., de Medeiros A.K.A., et al. Process Mining Manifesto. In *International conference on business process management*, pages 169–194, Berlin, Heidelberg, 2012.
[13] Yarmohammadi S., Pourabolghasem R., and Castro-Lacouture D. Mining implicit 3D modeling patterns from unstructured temporal BIM log text data. *Automation in Construction*, 81:17–24, 2017.
[14] Lee G., Sacks R., and Eastman C.M. Specifying parametric building object behavior (BOB) for a building information modeling system. *Automation in Construction*, 15:758–776, 2006.
[15] Fu F. *Design and analysis of tall and complex structures*, Butterworth-Heinemann, 2018.
[16] Autodesk. Revit Journal File. On-line: https://knowledge.autodesk.com/support/revit-products/getting-started/caas/CloudHelp/cloudhelp/2017/ENU/Revit-GetStarted/files/GUID-477C6854-2724-4B5D-8B95-9657B636C48D-htm.html, Accessed: 07/21/2021.
[17] McNeel. Rhinoceros. On-line: https://www.rhino3d.com/en/, Accessed: 07/13/2021.
[18] McNeel. Rhino API. On-line: https://developer.rhino3d.com/api/, Accessed: 07/13/2021.
[19] Sacks R., Eastman C., Lee G., et al. *BIM handbook: A guide to building information modeling for owners, designers, engineers, contractors, and facility managers*, John Wiley & Sons, 2018.