BIM-based Automated Design Checking for Building Permit in the Light-Frame Building Industry

H. Narayanswamy[a], H. Liu[b], and M. Al-Hussein[a]

[a]Department of Civil and Environmental Engineering, University of Alberta, Canada
[b]Department of Civil and Construction Engineering, Western Michigan University, USA
E-mail: hnarayan@ualberta.ca, hexu.liu@wmich.edu, malhussein@ualberta.ca

Abstract –

Automation of the code compliance checking has been explored extensively, particularly in recent years with the emergence of building information modelling (BIM). Still, automated code compliance checking has not yet been fully realized, as there is no standardized method for rule interpretation and building model preparation for code compliance. Manual verification of design code compliance, meanwhile, requires significant effort and time and is error-prone, while uncertainty and inconsistency in assessment lead to delays in the construction process. In this paper, the development of a prototype to automate municipal bylaw and wall framing code compliance checking for residential building is presented. The building rules have been classified into three groups based on the complexity involved in translation into computer-readable format and complexity in retrieving the required information, and they are represented based on building objects, which makes the regulations easier to understand and assists in translating the regulations into a computer-readable format. By creating a model view based on the required element's parameters extracted from a model for checking purposes, the prototype application offers automated code compliance checking functionality to validate designs based on building code requirements and construction engineering specifications. The implementation of the prototype and its benefits compared to manual checking is demonstrated.

Keywords –

Building Information Modeling (BIM); Automated code compliance checking; Building permit; Building rules

1 Introduction

Technological advancements in the architectural, engineering, and construction (AEC) industry have digitized nearly every stage of the building lifecycle, and this digitization has been a significant advancement in the industry over the past several decades. In this context, building information modelling (BIM) technology has been utilized for a wide variety of applications across the building lifecycle. In particular, automated code compliance checking saw a major leap with the advent of BIM in the late nineties [1]. An automated code compliance checking system is one of the processes for verifying the design in accordance with building codes, regulations, and bylaws. For many years, authorities and researchers have been working on an automated code compliance checking process, yet it is still only a semi-automated process. Many researchers have developed applications for safety, egress, and design checking, but still no application is in use for code compliance checking, even in countries where the BIM model for the design checking process was made mandatory. Many applications have not been updated since they were first developed, and an even fewer number of those have survived. This is partially due to the fact that there is no standardized method for rule interpretation and building model preparation for code compliance.

This research develops a BIM-based automated design checking prototype, in the form of an add-on for Autodesk Revit (i.e., DCheck), for automated checking Edmonton zoning bylaws, lot design for residential houses construction, and wood framing for walls of the residential building in accordance with Alberta building code 2014 part 9. To develop the prototype, the building rules have been classified into three groups based on the complexity involved in translation into computer-readable format and complexity in retrieving the required information, and they are represented based on building objects, which makes the regulations easier to understand and assists in translating the regulations into a computer-readable format.

In the subsequent section, a review of state-of-the-art existing software and plug-in applications with respect to automated compliance checking is presented. Detailed explanations pertaining to the methodology are presented in Section 3. Section 4 illustrates the system architecture of the prototype. A case study is presented as a test-bed to verify the developed prototyped system in Section 5.

Finally, findings are summarized, particularly as they pertain to potential future research.

2 Existing Design Checking Applications

With the use of CAD tools for design purposes by AEC professionals in the 1990s, the automation of design checking has gained more interest among researchers. Exhaustive studies have been performed in automating the building code by many researchers in this field, each exploring different techniques in interpreting the rules from his or her perspective [2]. Typical examples include the development of the following: logic-based approaches for the organization of design standards [3]; computer representation of design standards [4]; and knowledge-based expert systems capable of reviewing building design [5]. An overview of software applications developed by different countries and government authorities for automated compliance of each country's building code is described in the subsection that follows and a summary is presented in Table 1.

2.1 CORENET (Singapore)

In 1995, the Building Construction Authority (BCA) of Singapore initiated CORENET (Construction and Real Estate Network) as a comprehensive network system with a series of IT systems for exchange of information between government agencies and parties involved in construction and real estate [6]. CORENET for approval process provides electronic web-based submission system incorporating in-house building plans (BP) expert system to check 2D plans for any technical irregularities with reference to the building regulations [7]. e-PlanCheck, as part of CORENET, was the first initiative developed for automated code-checking. CORENET consists of three platforms: e-submission, e-PlanCheck, and e-info. E-PlanCheck was used as a pilot project in Norway and New York with replacement of rules required by Norway and by using ICC (International Code Council) codes for New York.

2.2 DesignCheck (Australia)

DesignCheck was developed by Australian authorities for automated building code compliance for Australia, focusing on accessible design regulations [8]. In fact, code checking efforts by Australia involved development in two phases. The first phase was to assess the capabilities of existing rule checking systems to find out which would be the best one for computerization of Australian standards [8]. Both SMC and Express Data Management (EDM) were considered as possible platforms for automated code checking. EDM was considered as the more suitable one because of its ability to provide a publicly accessible definition language to represent building codes. After the first stage of checking for a feasible solution, different domain-specific knowledge can be encoded to EDM rule base and can be applied to check a building model.

2.3 Statsbygg (Norway)

The Norwegian government organization, Statsbygg, acts as the Norwegian government's key advisor in construction, building commissioning, property management and property development [9]. The CORENET e-PlanCheck system has been used for a couple of industry foundation class (IFC) based BIM building projects as an early effort by Norwegian authorities. Multiple platforms, such as e-PlanCheck, SMC (Solibir Model Checker), dRofus, and Express Data Management (EDM) model checkers were also adopted for the purpose of experimenting for finding a better checking system. "HITOS" is a BIM project managed by the Statsbygg government organization and Tromso University since 2005, for which several software applications have been used for modelling architectural, structural, MEP (mechanical, electrical and plumbing), cost estimation, and energy simulation, and an EDM model server was also used for storing and accessing the model data in IFC format [10]. dRofus as rule based system was used for spatial program validation: which acts as a database system used for managing architectural programs, technical functional requirements and equipment's for early stage planning[9]. Solibir model checker (SMC) was used for checking accessible design in the building model. SMC was developed in 2000 in Finland as quality assurance and validation tool.

2.4 International Code Council (ICC) and General Service Administration (GSA) Design Rule Checking (The United States)

GSA, an independent agency of the United States government, issued BIM-guidance in 2006, and starting in 2007, made it mandatory to have a BIM model for validation for all projects seeking permission for spatial planning projects[11]. The application uses the SMC platform and design assessment tool for extending rules, developed by Georgia Institute of Technology. The most interesting initiative in this area is SMARTcode, which was started in 2006 and handled by ICC, a US-based association that develops the master building codes for residential and commercial buildings and most institutional buildings [12]. SMARTcodes is a project for transforming natural language code into computer inter-

Table 1. Summary of typical BIM-based Automated Design Checking

Article	Checking Platform	Focus	Research Theme		
			Code representation	MVD	Checking algorithms
Khemlani 2005[13]	FORNAX	Rules in Building plans and services (Singapore)	Computer code	IFC based (FORNAX)	Object-based approach
Preidel & Borrmann 2015[14]	CodeBuilder plugin	German fire code	VCCL (visual code checking language)	VCCL graph	Flow-based visual Language
Pauwels et al. 2011[15]	Semantic web	Acoustic Performance Checking Occupational circulation rules	Semantic rule language	N3Logic rules	Semantic Web Ontology language
Martins & Monteiro, 2013[16]	LicA	Portuguese Domestic Water system	XML-based parametric tables	IFC based	Structured Query Language
Sjøgren 2007[17]	SMC (Solibri model checker)	Norway's Building accessibility rules	Parametric tables	IFC based with adding geometric data	Object-based parameters checking
Zhang et al. 2013[18]	Rule checking Process	OSHA Fall protection and safety	Parametric tables	Object-based parametric model	Object-based and logic approach
See 2008[19]	DA's SMART codes for SMC, AEC3 XABIO	United States: ICC Building code	SMART builder	IFC based	RASE
Ding et al. 2006[8]	EDM	Australian Design checking for disabled access code.	Rule-based language	IFC based using internal model schema.	Object-based approach using ExpressX language
Tan et al. 2010[20]	Rule Engine	Building Envelop (Canada)	XML based decision tables	EBIM, XML based model	Decision table
Lee et al. 2015[21]	KBIMLogic program	Korean Building code	Parametric table	Object based parametric model	Logic-based query

pretable format, and a dictionary of the properties found within the building codes have been developed in SMARTcodes. The dictionary is helpful in communication between SMARTcodes model checking system and the IFC building model [22]. The rule interpretation process is the most vital stage in automated code compliance checking, where various technologies have been investigated and employed. With so many technologies there is no standardized way for translating the complete building rules and regulations into computer-readable format. Out of many applications developed the Singapore CORENET project is an early initiative started by a government organization about 23 years ago for checking of 2D building plans submitted online for code compliance. Subsequently, the Solibri company developed an application called Solibri model-checker (SMC) for checking 2D plans around 1999. And even after over two decades since the start of the CORENET project, there has not been much progress in the development of an automated checking process. Solibri is the only commercial software available for checking some aspects of building design like clash detection, and space validation. The applications developed should provide easy way for future updates with change of building rules and regulations and should be user-friendly for continuous use of code compliance. In terms of BIM model preparation, the building models should be developed with BIM technology-enabled software with a certain level of detail (LoD). LoD is details that are included in model objects related to dimensional, special, qualitative, quantitative, and other data to support required purposes. Models should be developed with LoD 300 or more for efficient extraction of information needed for compliance checking process.

It should be noted that previous research classified design rules into four groups, based on the complexity involved in extracting the required information from the BIM model. The present research further considers the complexity of rule translation to classify rules, as the major problem in the process of automated checking is the rule interpretation, where human-written codes are translated to computer interpretable format. However, building codes are not self-contained and make reference to many other documents. We proposed the strategies of the rule interpretation for each rule category. The translation of the human-readable natural language code into computer interpretable code is complete, only if when the logical representation of regulations gives a clear understanding of the building regulations.

3 Methodology

Figure 1 shows an overview of the process of automated design checking of building code and municipal bylaws for residential buildings. This process can be divided into four main steps [11]: (1) Rule translation, which is the interpretation of natural language building rules into computer-interpretable format; (2) BIM model preparation, which involves the design of the building model in Autodesk Revit software and creating model views from the given BIM model; (3) Rule checking, which involves the checking of the designed model with the encoded rules; and (4) Checking report, where the compliance check result is obtained.

3.1 Rule Translation

The translation of the context and content of the building code and municipal bylaws for residential buildings into a machine-readable format is one important step in the automated rule checking process. Each regulation has a different level of complexity involved in the translation into a machine-readable format. In this research, all the building rules are classified into three groups based on complexity in translation into machine-readable format and also with complexity involved in extracting the required information from the BIM model, These include (1) Easy: These are rules that are classified as easy to translate from natural human-readable language into computer-processable codes and where the information required can be directly extracted from the BIM model; (2) Intermediate: These are rules that are classified as difficult to translate and where the information required needs to be derived. The complexity level of translating these rules into computer-processable codes and getting that information from BIM model involves introducing some new attribute values for defining some properties of building objects for compliance; (3) Difficult: These are rules that are classified as needing to be simplified in order to translate them, and where the information required needs an extended data structure. Some rules need clarification depending on the building design and specifications, and some rules need the building model to be analysed to get the required value to be checked, and the information required from the BIM model is extracted using an extended data structure. Compared with previous studies in which building rules are classified into 4 groups, only based on complexity in extracting the information from BIM model, the proposed three groups make it easier to implement. This classification of building rules facilitates the development of the checking add-on software application in later stages, giving the user a clear understanding of the checking process. The

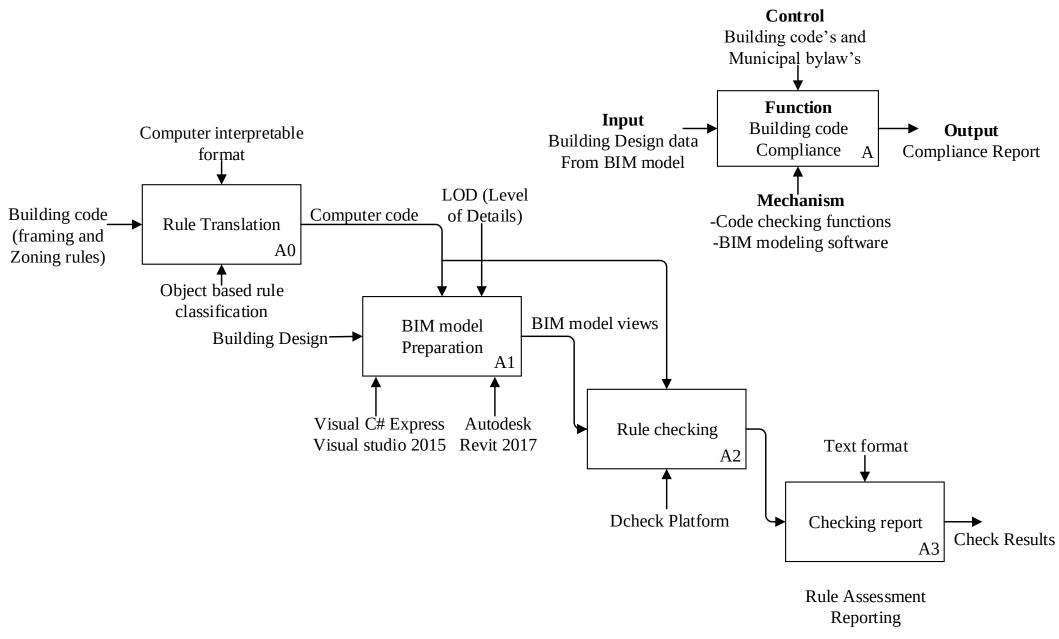

Figure 1. Methodology

representation of building codes based on building objects with conditions to be satisfied, attribute values to be driven from BIM models, and the threshold values to be checked for object-oriented representation, provides a clear understanding of building rules and makes easy for knowing the required parameter value from the model. By using this rule classification, building codes can be efficiently and comprehensively translated into computer interpretable format. In this study, an object-oriented programming language (i.e., C#) is used for compliance checking because of its flexibility and consistency in encoding building rules from the Alberta building code and municipal bylaws related to different zones in accordance with Edmonton municipal bylaws.

3.2 BIM Model Preparation

The BIM model can be defined as a digital representation of physical characteristics, such as architectural designs, and functional characteristics, like structural analysis, energy analysis, or a myriad of other simulations with semantically rich information. Building objects modelled in BIM enabled software's have parametric properties. For example, a wood stud member in the model possess types and properties like dimensions (length, width, depth), material properties, location of element member (XYZ coordinates), and so forth. The building model can be developed with level of details (LoD) above 300 for better extraction of information from a given model. Autodesk Revit provides a way to define and export extensible and interoperable BIM model data with use of Autodesk Revit's API (application program interface). The structure of APIs for exchanging information is object-based, where the geometry and properties of objects, such as name, size, location, finishes, faces and abstract information like cost, quantities, and so forth, can be accessed. In this research, C# language has been used to extract building information for compliance checking from the BIM solution, Autodesk Revit.

3.3 Rule Checking

The fundamental aspect of automated compliance checking is the information exchange from the BIM model to the rule checking platform. The checking platform (i.e. DCheck) applies a set of related rules to a model view. Prior to applying rule checking, syntactic checking of model is needed, to determine that the building model carries the properties, names, objects needed for the complete checking task. With object-based building information extraction, the geometric and functional information of objects, such as faces of building components, vertices, edges, location, and some derived information is extracted for compliance checking. Basically, mapping between the BIM model and the building rules is done by DCheck platform in this step.

3.4 Checking Report

The final step of the automated compliance checking process is to provide the user with a final compliance checking report by notifying the user checking results, such as success or failure and associated reasons and suggestions for failed regulation. This report is displayed to the user in textual format, and those building objects related to the rules which they have failed to satisfy are highlighted in the model, which helps to spot those objects and make corrections. The checking process can be run at any time during the design by the user, so it is easy for the architects or the draftspersons to check models in a parallel manner while designing.

4 Prototype Application

The automated building design checking is implemented as an add-on software application (i.e., DCheck) for the Autodesk Revit software, developed in C# language using Revit API's. Figure 2 shows the architecture of the prototyped Revit-based automated design checking software application. The inputs for the system include: (1) building design of the project and BIM model of the building intended to be constructed containing the architectural and structural framing information; (2) the project applicant information, regarding the applicant, architect's and builder information; and (3) zoning and framing information, regarding site location, plot number, type of wood used for framing and so forth. Criteria for this project are: (1) building code, in this case Alberta building code 2014 part 9, housing and small buildings, containing regulations related to framing of residential building; (2) municipal bylaws, Edmonton zoning bylaws, containing regulations related to different types of zones and residential building; (3) Level of details (LoD), development of model with LoD above 300 serves the required purpose for automation of building rules checking in this prototype.

The core processor of this prototype has main components: (1) object-based representation of building code, where building rules are represented based on the building objects so that required information from building model related to particular building object can be known clearly; (2) BIM model view definition, where required model views of BIM data for compliance checking are extracted; (3) BIM model extension, where some information which cannot be accessed directly from the BIM model needs to be derived and data structure platform extended by DCheck will provide those values; and (4) code compliance with model, where extracted information from the model will be checked

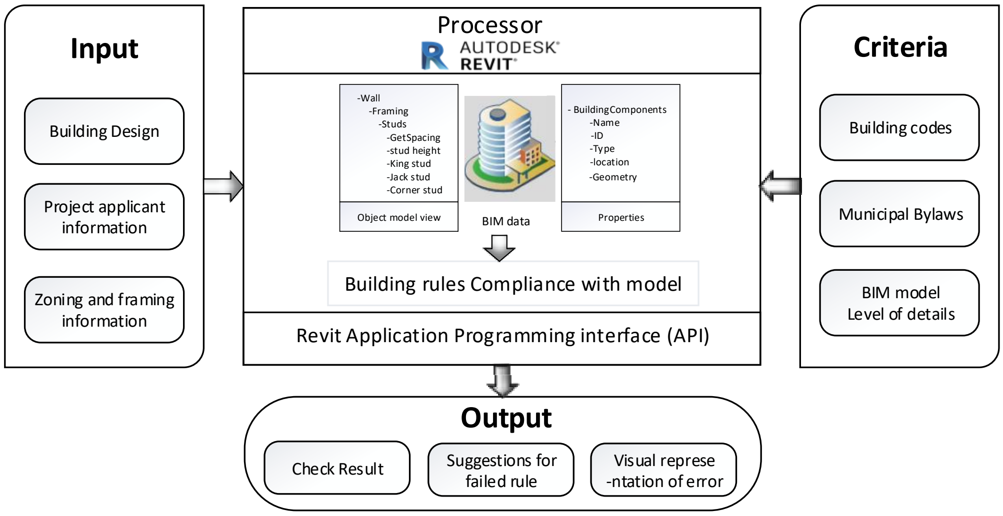

Figure 2. System Architecture

against the already-encoded building rules. These four components are compiled into Autodesk Revit as an add-on through using C# language.

5 Case Study

A residential building design has been modelled in Autodesk Revit as required, with a level of details more than LoD 300 for efficient automated-checking process. Before starting checking, users are required to provide some initial information related to the project, which is the same information house owners or contractors used to provide while submitting 2D CAD drawings for approval.

The building rules are represented based on building objects. The logical representation of rules based on building objects gives a clear understanding and provides an efficient way to accomplish the interpretation of rules into a computer-readable format. Figure 3 gives details about minimum setback distances required for different house types. Building rules from municipal bylaws like "If it is Single Detached Housing: Minimum site/ Lot Dimensions should be, Area: 250.8 m^2. Width: 7.6 m^2. Depth: 30 m^2" and so on are represented in a logical format for better understanding. As an example, the above explained rule related to lot checking can be represented as shown in Eq. (1), which represents two-dimensional matrix with threshold values to be satisfied for each type of house (j) and for different zone types (i). Where i is different residential zones present in city, j is different types of houses built in those residential zones. Finally, if the equation (2) returns the result of 1, then the rules have failed to satisfy, else it is correct.

In the City of Edmonton, there are ten different zones (Z_{bylaw}): Single Detached Residential Zone (RF1),

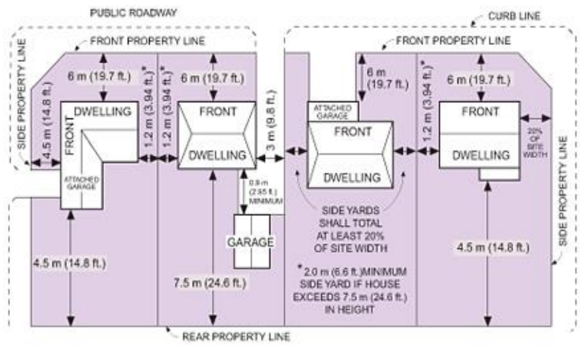

Figure 3. Illustration of Setback Distances Requirements for RF1 Zone (Adopted from City of Edmonton Website [23])

Residential small Lot Zone (RSL), Low Density Infill Zone (RF2), Planned Lot Residential Zone (RPL), Small Scale Infill Development Zone (RF3), Semi-detached Residential Zone (RF4), Residential Mixed Dwelling Zone (RMD), Row Housing Zone, Urban Character Row Housing Zone (UCRH), and Medium Density Multiple Family Zone (RF6). All the residential houses built in these zones are classified as single-detached housing (sdh), semi-detached housing (ssh), duplex housing (dh), limited group homes (lgh), garden suite (gs), secondary suits (ss), or minor home-based business (mhb).

$$i = Z_{bylaw} = \begin{bmatrix} \text{RF1, RF2, RPL, RF3, RF4,} \\ \text{RMD, Row housing zone,} \\ \text{UCHR, RF6.} \end{bmatrix}$$

$$j = HT_{bylaw} = \begin{bmatrix} \text{sdh, ssh, dh, lgh, gs,} \\ \text{ss, mhb.} \end{bmatrix}$$

$$L_{bylaw} \in M_{9X7} = A = \left| \begin{array}{l} \text{for i} \in 1 \dots 9 \\ \quad \text{for j} \in 1 \dots 7 \\ M_{i,j} \leftarrow [i\_j] \end{array} \right.$$

$$where, A = \begin{pmatrix} RF1_{sdh} & \cdots & RF1_{mhb} \\ \vdots & \ddots & \vdots \\ RF6_{sdh} & \cdots & RF6_{mhb} \end{pmatrix} \quad (1)$$

$$\text{get the user inputs:} \begin{cases} \sum Z_i = 1\ (i) \\ \sum HT_j = 1\ (j) \end{cases} \rightarrow find\ A(i,j)$$

$$\rightarrow L_{bylaw(i,j)} = \begin{bmatrix} LA^{bylaw} \\ LD^{bylaw} \\ LW^{bylaw} \end{bmatrix}$$

$$\delta_{lot} = \begin{cases} 1 & \text{if } L_{bylaw(i,j)} > L_{BIM} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The user needs to provide the project information on housing and zoning type through the main user interface of DCheck add-on application as shown in Figure 4. By running the application for checking municipal bylaws, it provides a textual report related to compliance checking in accordance with municipal bylaws based on building design and user input (see Figure 4). The textual report displays municipal bylaws failed to satisfy with the building objects attribute name, reason for the failure and details about that rules, so that user can easily understand the error in design and make changes as suggested. Once the errors have been corrected, the user can run the checking application again. If there are no further errors in design, it will display a textual report indicating checking successful.

6 Conclusion

This study presents the automated checking of zoning regulations according to the City of Edmonton municipal bylaws related to residential zoning designs and the Alberta Building Code part 9 related to housing and small buildings for light-frame residential buildings in Autodesk Revit. The representation of the building rules based on building objects and the classification of building rules into three groups based on complexity in translation and also in complexity in extracting required information, makes it easier to understand the rules and helps to translate the rules and regulations into computer-readable format. This kind of knowledge formularization makes it easier to understand the regulation and determine the required threshold value to be checked for that building object in the model with specified conditions. This makes it easier for any changes or for the development of add-on software application for a different province or jurisdiction where the building rules are different. Also, the classification of building regulations supports the development of the extended data structure platform in a step-by-step process, which gives the user a clear understanding of the checking process with rules that are incorporated in the add-on application system.

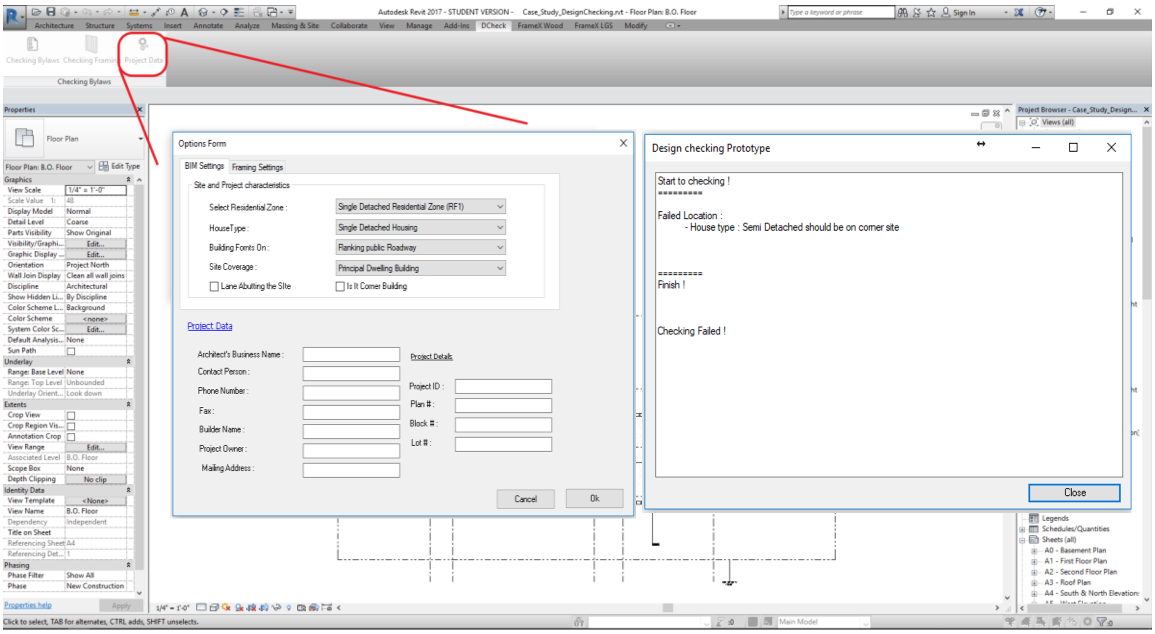

Figure 4. Graphic User Interfaces of DCheck add-on application

References

[1] C. S. Han, J. C. Kunz, and K. H. Law, "A Hybrid Prescriptive-/Performance-Based Approach to Automated Building Code Checking," *Fifth Congr. Comput. Civ. Eng.*, pp. 1–12, 1998.

[2] J. Dimyadi and R. Amor, "Automated Building Code Compliance Checking – Where is it at?," *Proc. 19th World Build. Congr. Constr. Soc. 5-9 May*, pp. 172–185, 2013.

[3] B. W. J. Rasdorf and S. Lakmazaheri, "Logic-Based Approach For Modeling Organization Of Design Standards In general , a design standard is composed of a set of provisions , where each provision defines a set of rules (constraints) that have to be satisfied in a give," vol. 4, no. 2, pp. 102–123, 1990.

[4] S. Fenves, J. Garrett, and H. Kiliccote, "Computer representations of design standards and building codes: US perspective," *Int. J. Constr. Inf. Technol.*, vol. 3, no. 1, pp. 13–34, 1995.

[5] C. L. Dym, R. P. Henchey, E. A. Delis, and S. Gonick, "A knowledge-based system for automated architectural code checking," *Comput. Des.*, vol. 20, no. 3, pp. 137–145, 1988.

[6] Global Reporting Initiative, "Construction and Real Estate," 2014.

[7] C. Preidel and A. Borrmann, "Automated Code Compliance Checking Based on a Visual Language and Building Information Modeling," *Proc. 32nd Int. Symp. Autom. Robot. Constr. 15-18 June*, pp. 256–263, 2015.

[8] L. Ding, R. Drogemuller, M. Rosenman, and D. Marchant, "Automating code checking for building designs - DesignCheck," *Coop. Res. Cent. Constr. Innov.*, pp. 1–16, 2006.

[9] "STATSBTGG website," 2018. [Online]. Available: https://www.statsbygg.no/.

[10] S. Malsane, J. Matthews, S. Lockley, P. E. D. Love, and D. Greenwood, "Development of an object model for automated compliance checking," *Autom. Constr.*, vol. 49, no. PA, pp. 51–58, 2015.

[11] "Automated compliance checking using building information models," no. September, pp. 2–3, 2010.

[12] C. Eastman, J. min Lee, Y. suk Jeong, and J. kook Lee, "Automatic rule-based checking of building designs," *Autom. Constr.*, vol. 18, no. 8, pp. 1011–1033, 2009.

[13] L. Khemlani, "CORENET e-PlanCheck : AECbytes ' Building the Future ' Article CORENET e-PlanCheck : Singapore â€™ s Automated Code Checking System CORENET e-PlanCheck : AECbytes ' Building the Future ' Article," *Design*, pp. 1–8, 2005.

[14] W. Solihin, C. Preidel, and A. Borrmann, "Towards code compliance checking on the basis of a visual programming language," *ITcon Vol. 21, Spec. issue CIB W78 2015 Spec. track Compliance Checking, pg. 402-421, http//www.itcon.org/2016/25*, vol. 21, no. 25, pp. 402–421, 2016.

[15] P. Pauwels *et al.*, "A semantic rule checking environment for building performance checking," *Autom. Constr.*, vol. 20, no. 5, pp. 506–518, 2011.

[16] J. P. Martins and A. Monteiro, "LicA: A BIM based automated code-checking application for water distribution systems," *Autom. Constr.*, vol. 29, no. 23, pp. 12–23, 2013.

[17] Sjøgren, "in Norway," 2007.

[18] S. Zhang, J. Teizer, J. K. Lee, C. M. Eastman, and M. Venugopal, "Building Information Modeling (BIM) and Safety: Automatic Safety Checking of Construction Models and Schedules," *Autom. Constr.*, vol. 29, pp. 183–195, 2013.

[19] R. See, "SMARTcodes : Enabling BIM Based Automated Code Compliance Checking AEC-ST Conference Presentation – 21-May-08 SMARTcodes : Enabling BIM Based Automated Code Compliance Checking," no. October 2007, pp. 1–7, 2008.

[20] X. Tan, A. Hammad, and P. Fazio, "Automated Code Compliance Checking for Building Envelope Design," *J. Comput. Civ. Eng.*, vol. 24, no. 2, pp. 203–211, 2010.

[21] H. Lee, S. Lee, S. Park, and J. Lee, "An Approach to Translate Korea Building Act into Computer-readable Form for Automated Design Assessment," *Proc. 32nd ISARC*, pp. 1–8, 2015.

[22] W. Solihin and C. Eastman, "Classification of rules for automated BIM rule checking development," *Autom. Constr.*, vol. 53, pp. 69–82, 2015.

[23] "City of Edmonton." [Online]. Available: https://www.edmonton.ca/residential_neighbourhoods/zoning-regulations-for-houses.aspxå. [Accessed: 15-Jan-2019].