

Article

Fire Safety in Tall Timber Building: A BIM-Based Automated Code-Checking Approach

Kristina Kincelova ^{1,2,*}, Conrad Boton ², Pierre Blanchet ¹ and Christian Dagenais ^{1,3}

¹ NSERC Industrial Research Chair on Ecoresponsible Wood Construction (CIRCERB), Forest and Wood Sciences Department, Université Laval, 2425 rue de la Terrasse, Quebec City, QC G1V 0A6, Canada; pierre.blanchet@sbf.ulaval.ca (P.B.); Christian.Dagenais@fpinnovations.ca (C.D.)

² Department of Construction Engineering, École de Technologie Supérieure, Montréal, QC H3C 1K3, Canada; conrad.boton@etsmtl.ca

³ FPInnovations, Quebec City, QC G1V 4C7, Canada

* Correspondence: kristina.kincelova.1@ens.etsmtl.ca

Received: 11 May 2020; Accepted: 1 July 2020; Published: 8 July 2020



Abstract: Fire safety regulations impose very strict requirements on building design, especially for buildings built with combustible materials. It is believed that it is possible to improve the management of these regulations with a better integration of fire protection aspects in the building information modeling (BIM) approach. A new BIM-based domain is emerging, the automated code checking, with its growing number of dedicated approaches. However, only very few of these works have been dedicated to managing the compliance to fire safety regulations in timber buildings. In this paper, the applicability to fire safety in the Canadian context is studied by constituting and executing a complete method from the regulations text through code-checking construction to result analysis. A design science approach is used to propose a code-checking method with a detailed analysis of the National Building Code of Canada (NBCC) in order to obtain the required information. The method starts by retrieving information from the regulation text, leading to a compliance check of an architectural building model. Then, the method is tested on a set of fire safety regulations and validated on a building model from a real project. The selected fire safety rules set a solid basis for further development of checking rules for the field of fire safety. This study shows that the main challenges for rule checking are the modeling standards and the elements' required levels of detail. The implementation of the method was successful for geometrical as well as non-geometrical requirements, although further work is needed for more advanced geometrical studies, such as sprinkler or fire dampers positioning.

Keywords: BIM; fire safety; compliance checking; building code; visual programming

1. Introduction

To encourage the interest in Canada for mid-rise and high-rise timber buildings, different organizations and initiatives take place at federal and provincial levels, including the Wood First Act in British Columbia [1] and Wood Charter in Quebec [2]. The Wood First Act requires, since 2009, to use wood as a primary material in all provincially funded buildings. Likewise, the Wood Charter is an initiative of the Quebec government containing measures, which aim to encourage the use of wood in all aspects of construction. The prescriptive provisions of the Canadian and provincial building codes, however, continue to limit this type of construction to buildings of no more than six stories in height, with the exception of the province of Quebec where up to 12 stories of mass timber construction is allowed [3]. Tall buildings need to provide very strict fire safety conditions; it is therefore necessary to provide efficient tools for designing and verifying compliance of the fire safety regulations.

In the context of the Quebec construction industry, fire safety is ensured by two regulatory documents: the National Building Code of Canada (NBCC) [4], which is modified at the provincial level in the Quebec Construction Code, and the technical guide entitled “Mass timber buildings up to 12 floors” established specifically for Quebec [3]. Fire protection design regulations in Canada are today still very prescriptive [5]. The verification of code compliance is done manually and thus is prone to errors [6]. With the growing building information modeling (BIM) integration in the design practices [7], emerging BIM-based approaches to regulation compliance checking can be seen. There are tools and approaches that specialize in automated compliance checking of different regulations (early concept design [8] or sustainability [9]). However, the BIM approach takes a more prominent place in the design process and can contribute to automation of the compliance checking practice. In terms of fire protection, several reports show that its integration in BIM in general is limited [10,11]. As of today, there exist mainly tools for clash detection of the sprinkler systems (i.e., AgaCAD [12] or AutoSPRINK [13]), for the use of the general geometry for fire and smoke simulations [14], for evacuation simulations [15], or the calculation of the fire load checking [16]. It is concluded that the BIM-related fire protection requirements are considered late in the design process [10]. Very few research works were dedicated to BIM-based automated code checking in the fire safety domain. These works are limited to isolated parts of the rule construction process and do not address the passive fire protection area. Moreover, there is virtually no proposal or research work dedicated to the particular context of Quebec. To be effective and to ensure a good appropriation by the practitioners, such a proposal should take into account not only the nature of the regulations (at federal and provincial levels) but also be based on the practices and modeling tools widely used in the industry. As shown by the recent Annual BIM Report, “Autodesk products dominate the Canadian market space”, and Revit appears to be by far the most used BIM design authoring tool [17].

The aim of the research presented in this paper is to propose a BIM-based automated code-checking approach, to improve fire safety in tall timber buildings, by reducing the time dedicated to checking architectural models during the design phase.

A global method for code-compliance checking is developed specifically for the domain of fire protection in the regulative context of Quebec and its sector of timber building. It is noted that it can easily be adapted to other regulatory environments, such as other provinces or countries. The method aims to support the manual construction of rules, which will then enable automated execution of compliance checking of BIM models for specific prescriptions. In this research, the challenges of information transfer and integration in particular are addressed, specifically the fire-protection-related data. The present article is organized into four main sections. The Section 2 presents the literature review, including the main fire safety challenges, the BIM-based automated code-checking approaches, and the recent innovations in the domain. The Section 3 describes the research method, including the theoretical framework and the main steps of the research approach. The Sections 4 and 5 present the results, including an overview of the proposed method, the details of each of its steps, and the tools and the artefacts proposed to support them. The Sections 6 and 7 discuss the results with regards to the recent similar works and concludes the paper.

2. Literature Review

2.1. Main Fire Safety Challenges

The main goal of fire safety engineering is to limit the probability of death and injury, but also of property damage [18]. Several domains are impacted in the fire safety design, including electricity, mechanics, architecture, and structure. More specifically, one can distinguish two types of fire safety systems: active and passive controls [18]. Preventing ignition and fire spread is an important part of the passive protection for timber buildings. This can be controlled through the choice of materials and architectural decisions but also through the fire resistance of fire separations or façade flame spread control [19–21].

The passive protection system is an essential element of the design and highlights the necessary cooperation of the various domains during the design phase. For example, the interference between firewalls and technical conduits can create an unplanned fire path. In addition, for vertical construction, the main risk is represented by the vertical ducts [18,19] (lifts, ventilation, and stairs) that cross several floors and therefore can create a space for rapid fire spread between different floors. In light of these works, it can be concluded that if the construction material is a combustible material (e.g., wood), it is necessary to give more attention to the aspect of passive protection in the fire safety, and the following work will be therefore restricted to the study of passive systems.

2.2. BIM and Automated Code Compliance Checking

The BIM approach consists of using a multidisciplinary object-oriented tridimensional (3D) model of the constructed facility to document its design and to simulate different aspects of its construction or its operation [22]. This includes a more multidisciplinary approach to the design phase of the project. Code compliance checking is defined as the process of reviewing the building design against building requirements from a building code [23] and does not alter the original building design [6]. The BIM offers a platform for implementation of automated processes, including code compliance checking [24]. The data contained in the building models allows for the development of parametric rules that are more interoperable [6,24].

In 2009, Eastman et al. [8] examined five applications of the existing commercial tools to present the state of the art at the time in the automated rule-checking domain. In order to do so, he defined four stages of rule checking process, using detailed functional criteria. The process is comprised of rule interpretation, building model preparation, rule execution, and rule check reporting. This framework has set a solid foundation for the following research works. Since Eastman et al. [8], several works were carried out to address one or more of these steps.

As of today, the main solutions commercially available for BIM-based automated checking are three software programs: Solibri Model Checker (SMC) [6,9,25,26], Fornax [27], and EDM Jotne [28]. In reality, based on a more detailed study conducted in the context of this research project, the current commercially available solutions seem unable to cover the essential elements of passive fire protection adequately. The rules in SMC do not allow to freely manipulate data to generate new properties [29]. However, to address the issue of automated checking of fire safety regulations, it is necessary to be entirely free to manipulate the data and to have absolute transparency in terms of the construction of rules, because there is a lot of implicit information. This explains why the current commercially available solutions are often criticized [30,31] for their lack of flexibility and black-box aspect. Therefore, recent research works started to approach this domain from different points of view, proposing new avenues to address the issue.

2.3. Recent Innovations in Research

To improve the clause classification, Nawari [32] proposed a definition of four different categories of clauses: The *provisory* class concerns clauses where it is possible to directly transform the text into a set of rules based on objects. *Dependent* clauses define the dependent regulations on one or more propositions and define specific conditions. These two categories are similarly defined as declaration clauses by Malsane, et al. [33]. Finally, *ambiguous* provisions are subjective and can be automated only with special approaches such as artificial intelligence (AI) or require a rewrite. After filtering further usable clauses, Nawari [32] and Malsane et al. [33] defined the necessary type of information to associate with each object studied per prescriptive article in a building code. A manual approach is also presented in the project called KBIMCode [34] for regulation analysis compliance to the Korean Building Act. Finally, there is a mark-up approach to extract information from the regulation texts. The developed language RASE (Requirement–Applicability–Selection–Exception) project [35,36] is one of the most cited methods in BIM-based rule automated analysis. The RASE methodology merely captures the target values and explicit data and does not consider the interpretation of the rule meaning

and its possible ambiguities or subjectivities [37,38]. Hence, it must be considered that, according to Solihin and Eastman [37], in order to make a complete and meaningful translation of a regulation rule, it can only be done manually, as the computer will not be able to substitute the domain's expert knowledge. In this research, the manual approaches from the literature will be adapted for the present needs.

Rule construction is generally challenging in terms of computing expertise. Nawari's framework based on subsets of the Industry Foundation Classes (IFCs), referred to as Model View Definition (MVD), requires ifcXML and LINQ manipulations, that are most likely not common for a construction professional [9,37]. A similar challenge is encountered in the approach for sustainability verification [9] and in early design assessment [39]. The logic RASE tags needed to be converted into a computable format by SWRL (Semantic Web Rule Language). Finally, Ghannad et al. [39] used visual programming through Marionette (Vectorworks) to transfer rules in the XML format. Hard-coded code-checking tools were also developed, in form of a plug-in to a BIM modeling software [40,41] or as a stand-alone code checker [36,40]. However, the hard-coded construction is not viable in the context of ever-changing standards in the building industry context. Then, the visual programming for rule construction was explored in the KBIM Project [34,42] and by Preidel [43]. Visual programming is a broadly used tool by architecture professionals [44] since its interface "is more in line with architects' design habits" [45]. Both Preidel and Kim defined a new rule-construction and -checking tool that seems to answer the necessary requirements of IT accessibility and the possibility of new rulesets construction. However, they did not address the link between the rule-checking tool and the model information requirements. In this work, the construction of rules will be therefore more oriented on the specific challenges of fire safety.

In the context of the present research, the fire-safety-specific information requirements are needed. Some of the works [10,11] only provide general recommendations for fire safety BIM modeling, but there is a lack of consistency in the inventory of building information requirements. This was addressed in the context of performance-based design [15,46], where input data for fire simulation needs were identified and then associated with the IFC format. Similar effort is therefore expected with adaptation for the code compliance checking approach.

The mapping of information requirements for code checking is addressed in general with the IFC format, such as an extension of the format for fire safety requirements in Great Britain [33] or a proposed template for Danish fire safety requirements [26]. In other domains, the data exchange requirements can also be stored in the MVD format [32,47]. These formats can only encompass explicit objects and do not take into consideration that some required information for code checking can be derived. The concept of derived information was specifically formalized by Solihin and Eastman [37]. The goal is to keep the manual input of new properties to a minimum. It is supposed that the fire-protection-required attributes can often be derived and, therefore, the IFC representation is not sufficient. The model element table enables a more comprehensive communication between actors [7]. It will be considered as a possible solution.

Another challenge is related to the rule execution and reporting. Two possible ways of reporting are defined by Eastman [6]: graphical reporting and building code reference stating. The results are defined as a pass, fail, or unknown and can be accompanied by placing a viewing camera onto the concerned elements or giving the statement of the original instance [6].

3. Materials and Methods

3.1. Methodology Overview

The literature review showed that there is a certain number of constructs or tools for automated code checking that focus on one or more steps of the whole process. Among the four types of artifacts defined by Hevner et al. [48] in design science research, this work focuses on a method proposition.

That means the steps for the whole process are described. This includes the positioning from the fire protection point of view.

The research approach is presented in Figure 1 and is comprised of four parts. In the understanding of the needs, the fire protection concepts were studied as well as their integration in the design phase. In parallel, the existing solutions for automated code checking were listed based on the literature review and critically assessed against the identified fire protection needs. The safety concepts and challenges were studied in order to understand the regulatory context. The critical elements of fire safety regulation could, therefore, be identified.

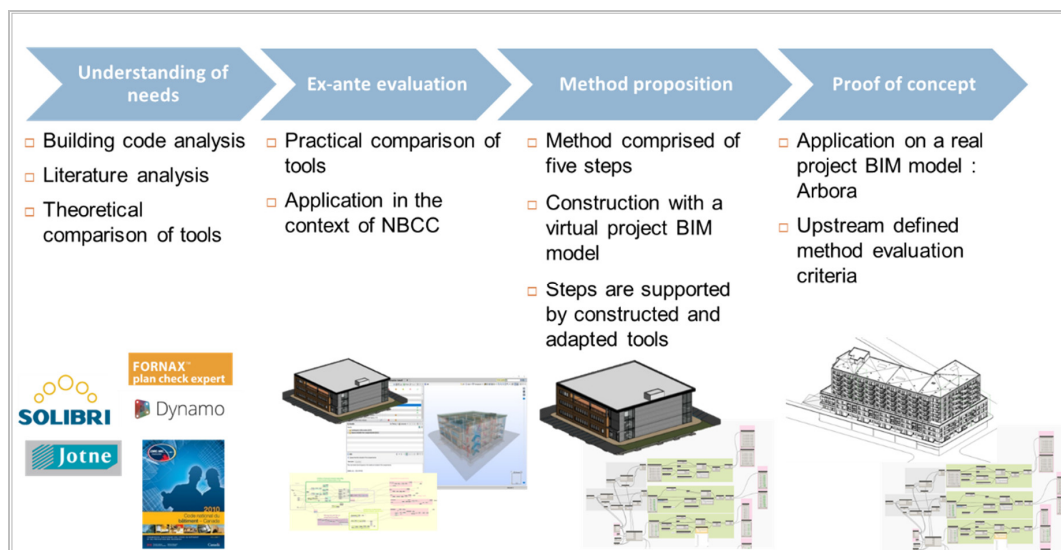


Figure 1. Research approach.

Then several tools for rule construction were identified in the ex-ante evaluation. Example rules addressing different aspects of passive fire protection were tested to identify in practice the fire safety needs in terms of building information modeling. In this step, visual programming with Dynamo was selected [29].

The complete method was constructed and its fire protection challenges discussed. Finally, the proposed method was tested and evaluated through a proof of concept using a real project building architectural model and upstream defined evaluation criteria in order to study the different styles of modeling.

3.2. BIM and Prescriptive Environment Definition

The compliance checking tool was constructed in the Revit environment, using Dynamo 1.2.2. The following packages containing specific nodes were required for the script construction: Archi-lab 2018.22, SteamNodes 1.2.4, Clockwork 1.31.2, LunchBox for Dynamo 2015.11.28, and Spring nodes 132.2.

Finally, the presented framework was applied to a set of example rules to validate the method. The rules were taken from Division B of the 2010 National Building Code of Canada (NBCC) [4], as well as the technical guide for mass timber buildings up to 12 stories [3], and they concern specifically responsibilities of an architect in the present case. The used model is therefore the architectural model that is modified.

For the needs of the proof of concept, clauses addressing essential passive fire protection concepts for tall timber buildings were selected. This study implemented elements of fire-resistance rating, flame spread rating, and continuity. Elements on the same story will have different required values of fire resistance rating or flame spread rating, based on the occupancy in each compartment. Furthermore, occupancy on different stories can change. This is especially true for the case of tall buildings.

3.3. Definition of Evaluation Criteria for the Proof of Concept

In the context of the experimental evaluation, the completeness, consistency, usability, and accuracy will be evaluated as defined by Hevner et al. [48]. These criteria are defined more specifically in the context of this work as follows:

- *Completeness*: In the context of the reduced scope, it will be considered how the constructed rules have answered to the challenge of automating human written building code regulations.
- *Consistency*: The use of an independent BIM model will show how the constructed rules can be adapted to different building model constructions. The completeness of the rules is verified regarding the selected rules to check and not in regard to the complete NBCC.
- *Usability*: This criterion concerns the implementation of the proposed workflow, as well as its integration to the designing process (e.g., the effort of minimizing the need for manual input).
- *Accuracy*: The accuracy of the constructed rules has to be evaluated by manual verification. Because the results are presented in form of a list of non-compliant elements, false negatives will be detected easily. However, the false positives are harder to point out, because of the high number of verified elements.

4. Results: Method Formalization

The literature review approaches the general challenges of code checking in terms of software needs or regulation classification. However, the BIM needs are different for each individual domain (sustainability, construction site, accessibility, or fire protection among others). Therefore, the code checking approaches must take into consideration the particularity of each domain. The application to fire safety in timber construction is studied in the present work. The ex-ante evaluation brings up the challenge of representing specific information needed for fire safety code checking; however, the preparation must not burden the modeler with heavy data input [38]. At the same time, the constructed tool must verify rules efficiently. The preparation and the standardization of the information is an essential element in code-checking. The method is emphasizing in particular these two aspects during the whole method presented herein (Figure 2).

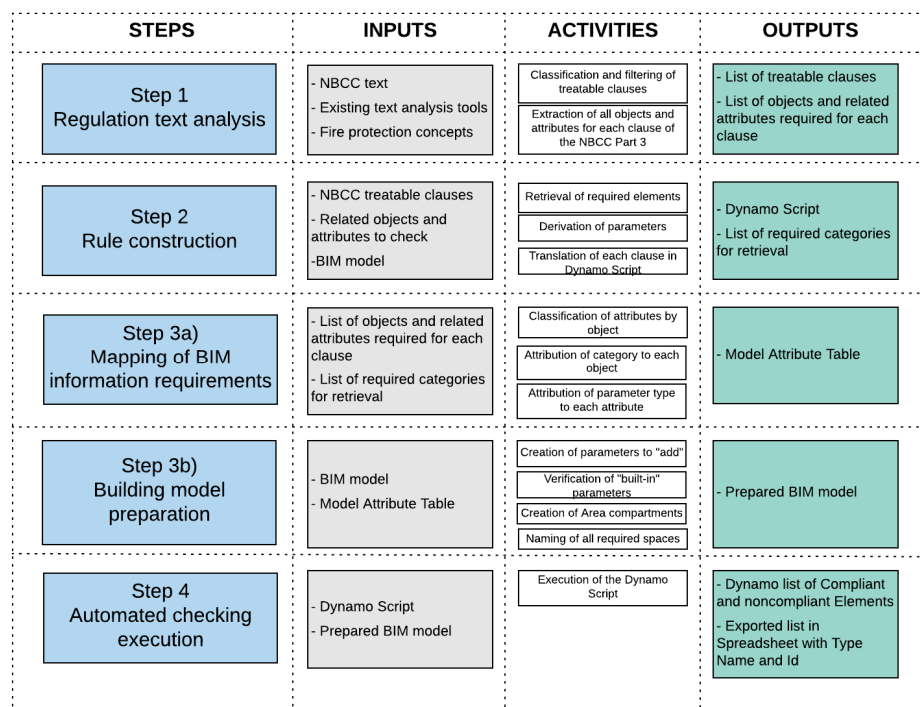


Figure 2. General overview of the suggested method.

4.1. Regulation Text Analysis

First, all the necessary concepts and attributes that are of regulation's concern must be extracted. Several methods exist in the literature with different degrees of automation. However, the manual approach was chosen in order to maintain complete control over manipulated information. As fire safety regulations are often formulated implicitly, the terminology use can be inconsistent and the information is very specific and concerns detailed matters. The model of table that is used for the data extraction (Table 1) is adapted from the methods of Nawari [32] and Malsane et al. [33].

Table 1. Proposed table for extraction of semantics (adapted from Nawari [32] and Malsane et al. [33]).

Section		Classification	Concepts & Attributes	Dependencies
Clause number	Clause text	Provisory/Dependent/Ambiguous/Content	Extracted concepts and attributes contained in the clause text	Clause numbers that the clause text mentions

All the clauses are first classified in order to select the categories that can be treated. Given the previous analysis of different clauses classifications, it was decided to classify all the regulation clauses, according to [32], into provisory, dependent, ambiguous, and content categories. Only provisory and dependent clauses were treated completely here. In fact, the content clauses did not contain any rule check material; then the ambiguous category needs more interpretation, such as fuzzy logic treatment [32] that will not be addressed in the scope of this paper.

Once the clauses that will be further explored are chosen, the related objects and attributes are listed. The objects and attributes are not presented yet in a BIM-compatible format.

4.2. Rule Construction

It was seen in the preliminary test, that the fire protection requirements do not expose the same challenges as the accessibility or sustainability prescriptions to which code-checking tools are adapted today. The intent is to give a solid base as a starting point with different methods, which will contribute to the creation of rules regarding fire safety in timber construction.

A more detailed study [29] allowed to identify specific challenges in the fire protection domain, and it was concluded that in comparison to the existing proprietary solutions, visual programming allows for complete liberty in accessing the information that is contained in the building models, which is crucial for this type of work. As of today, visual programming exists in the BIM environment for parametric design in architecture and for code checking approaches at the research stage. The main providers are Grasshopper for Rhino (Abvent), Dynamo for Revit [49], or Marionette for Vectorworks (Nemetschek). In the present work, the method is illustrated with Dynamo for Revit.

4.2.1. Rule Script Implementation

The visual programming consists of creating scripts with nodes. The scripts are grouped by domain, each representing a rule check, for example, a verification of fire-resistance rating for a group of elements. In order for elements to be checked in the same script, the information in the building model must be accessible. The rule construction in the visual programming script is intuitive and accessible for users with limited knowledge of programming, and therefore, the structure of the script is not very rigid. In order to construct clear and re-employable scripts, however, a general framework is set in Figure 3.

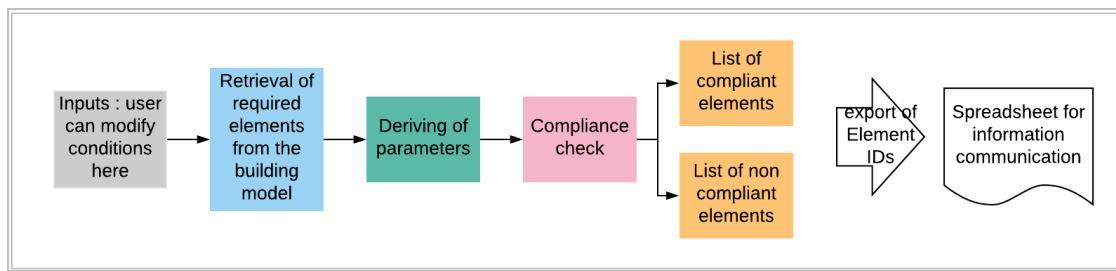


Figure 3. Visual programming script structure for any compliance check.

First, the script presents a set of inputs, where the user is required to set conditions. In order for the script to be useful and adaptable in different design firms, it has to be flexible enough to work with different modeling habits. As each firm has its own standards of modeling, the user must be able to modify the entry conditions in the input. This is illustrated in Figure 4, where different material names are listed by flame spread rating and can be adapted to the present practices of modeling.



Figure 4. Input example of lists for flame spread rating checking.

Then necessary data are extracted from the model. In Dynamo, all items for a given category or family type can be retrieved from a model. Elements of the same category or of the same family type in the building model may have different fire protection constraints depending on their location. The more general approach to extract elements by category is therefore chosen. If an open standard format is used, the retrieval of information in Dynamo is done from a Revit model based on an imported IFC file. Then a spatial study is carried out in order to classify these elements according to the compartment or room to which they belong as part of the parameter derivation step.

During the compliance check of a given concept, it is necessary to list an exhaustive list of possible figures that correspond to a compliance scenario. This was a challenge, especially, given the formulation of certain requirements in the NBCC. A physical relationship such as continuity can be represented in the object-oriented model in several ways (Figure 5). It is, therefore, necessary to retrieve, derive, and then check the data for all the possible cases that represent compliance. Once required elements are checked against all the listed possibilities and the element meets one of them, then the element is assumed as compliant.

This leads to a list of true or false statements, which can be associated to the list of surveyed elements. This list is separated into the list of compliant and non-compliant elements. It can be exported in a tabular format and further exploited.

The goal of the rule construction is to minimize hard-coded solutions and prioritize already present published nodes created by more experienced contributors. Therefore, it is considered that new nodes composed of Python scripts are to be the last resort, and the choice must always be justified. The space classification required specific organization into lists that lead to the creation of specific nodes in the steps of deriving parameters or compliance check.

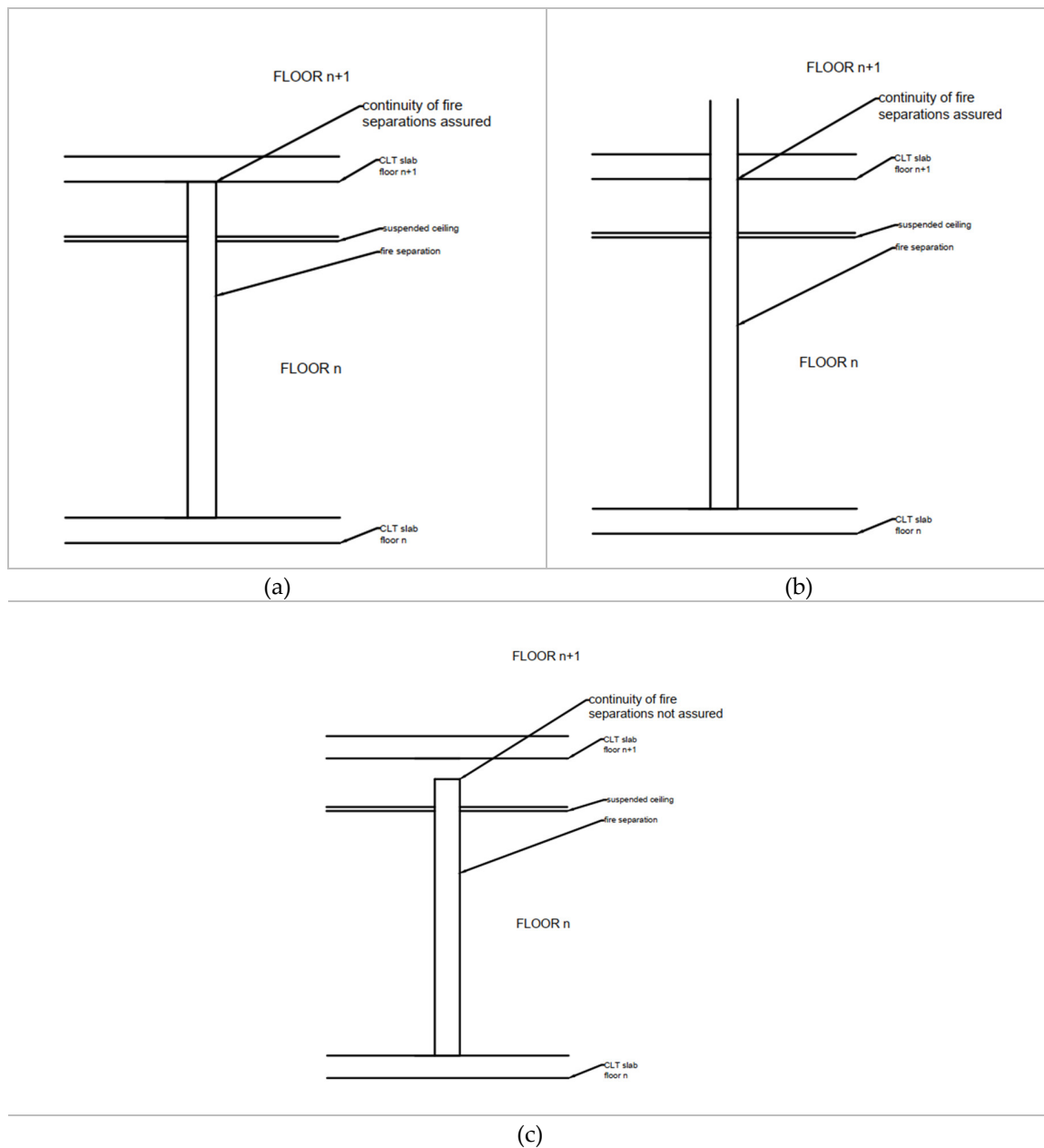


Figure 5. Geometry representation of continuity of a vertical fire separation. Vertical separation intersection: (a) with the underface of the slab above, (b) with the lateral faces of the slab above, (c) no intersection with a fire separation.

4.2.2. Space Classification

The NBCC building code was analyzed in terms of implementation needs. The global overview showed that passive protection requirements of different building elements are based on the characteristics of the related space and, more specifically, on the compartmentation. For example, the walls' fire-resistance rating will be different depending on whether the space is a suite or a common corridor leading towards an exit.

It is possible to distinguish three categories of compartments that will be approached differently from the BIM point of view: (1) Room, single entities; (2) Compartments is a group of several rooms restricted to a single floor; (3) Vertical service spaces that go vertically through several floors. A script is implemented for each type of compartment to group the different rooms of one compartment together. This then enables the grouping of different elements belonging to the same compartment.

1. The single room can be retrieved from the model directly, as it is a separate category.
2. The compartments of several rooms are not directly represented in the building model. This requires a preliminary preparation of the BIM model. The Area tool is used to draw a closed line to create limits of the area representing a compartment. The shared Boolean parameter “apartment” is created and then set for true. The developed Dynamo script classifies individual rooms in a set of sub-lists by compartment (Figures 6 and 7b).
3. The vertical service spaces are the most complex class of space as the compartment is a unique space that is going through several floors. However, in the BIM models, the staircases for example are represented as a set of rooms, one for each floor. The parameters of level, name, and location point for each object are exploited to obtain a set of sublists that group the objects by corresponding vertical service spaces (Figure 8b).

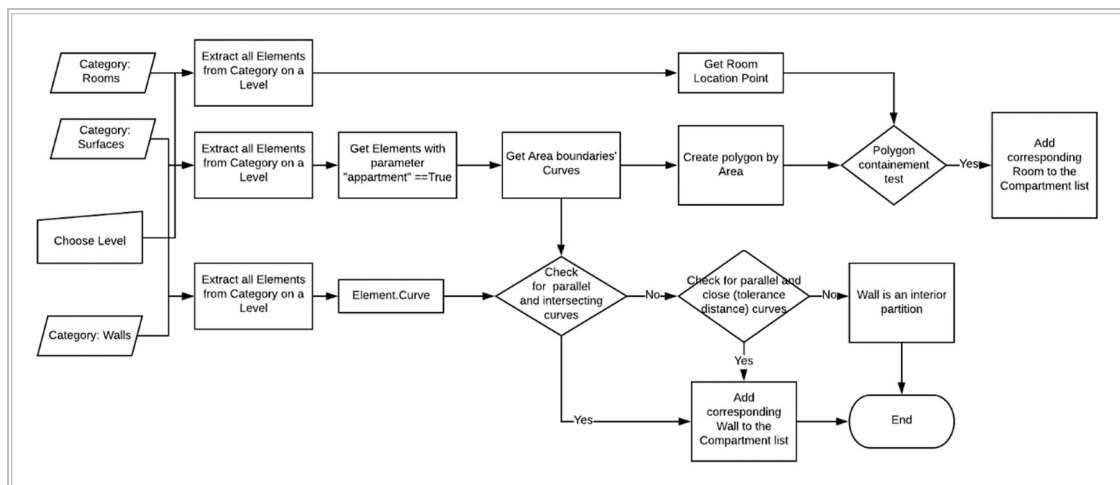
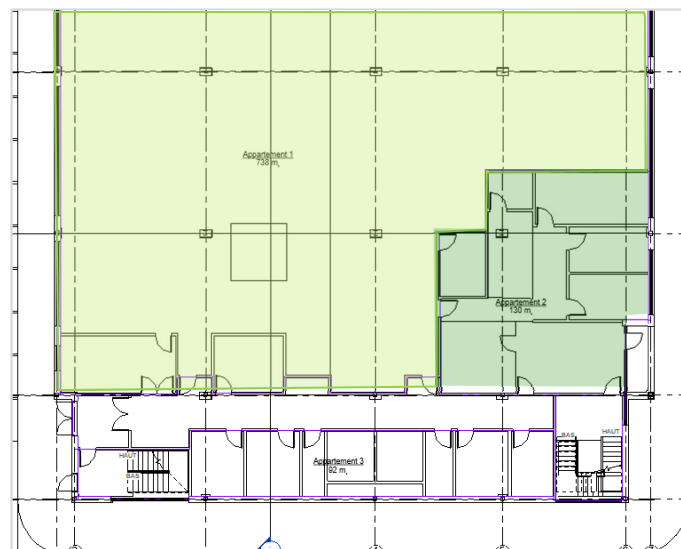
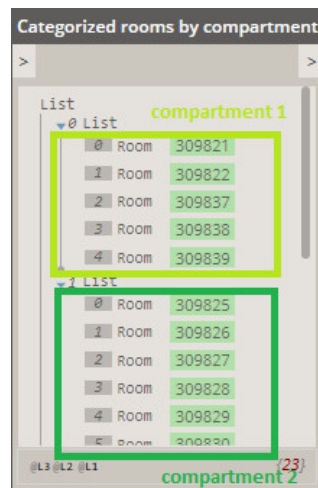


Figure 6. Horizontal compartmentation method.



(a)

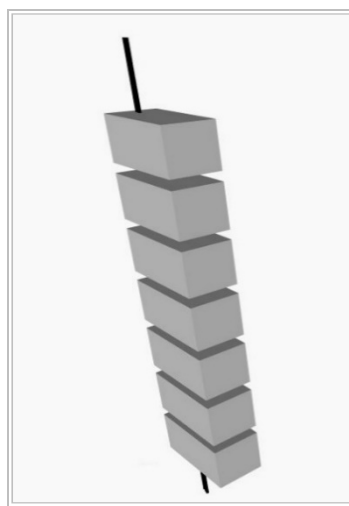
Figure 7. Cont.



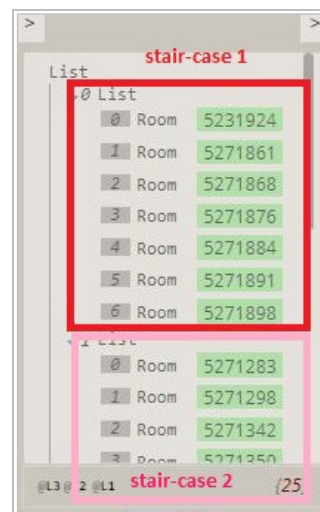
(b)

Figure 7. Multiple rooms compartment representation: (a) Area boundaries drawn in purple in Revit; (b) Dimensions of the Dynamo list and their structuring.

The listing is obtained through analysis of the name parameter and the geometry of each element of the category room. In this developed Dynamo script, the rooms are first filtered with a name constraint, where an input is a string contained by the vertical service spaces (such as “esc” for *escalier*—staircase). In order to group the corresponding volumes together to form the whole compartment, a Dynamo geometry study is conducted. If the vertical line that goes through the center point of the first listed room intersects with the geometry of any other room above or below, then they are grouped in the same vertical compartment.



(a)



(b)

Figure 8. Vertical service spaces representation: (a) Intersection between a room geometry and room location vertical line; (b) Compartmentation list for vertical service spaces.

The presented method of space classification was chosen this way in an attempt to require minimum input into the model. Basic parameters are needed, such as category or geometry. Visual tools such as the area category are preferred to create compartments.

4.3. Mapping of Information Requirements and Building Model Preparation

Information requirements for fire safety were identified in the literature for the performance-based design needs. However, the approach of compliance checking is different, and therefore, the inputs differ as well. The prescriptive requirements of the building code require fewer calculations for example.

Regarding the information requirements extracted from the building code during the rule analysis, it is necessary to represent parameters that are going to be derived from the building model information by the preparatory Visual programming scripts. In order to obtain the most complete information possible, the following work is therefore articulated in the form of a model attribute table with specified categories to extract.

4.3.1. Parameter Types

An individual compliance checking script requires the manipulation of attributes that are related to one or more building model elements. Each parameter required for the check can belong to one of four different cases:

1. Automatic parameter: the element and its required attributes are present in the building model. The information is filled in by the program when the element is drawn (e.g., wall geometry).
2. Built-in parameter: The built-in parameter is part of the modeling software, but its value is not automatically set; it is possible that in an existing project, the parameter is empty (e.g., fire-resistance rating).
3. Add parameter: The element is present, the attribute is absent, and neither are included in the IFC schema or in the modeling software. A new, shared parameter must be created.
4. Derived parameter: The information can be deduced from the information already present in the building model if it is properly manipulated. This will be the selected solution if adding a parameter representation of the info is not adapted. Some attributes that are represented in IFC schema are not used in practice because the input can be too tedious and erroneous or require a mathematical operation.

4.3.2. Element and Property

The checking of fire safety regulation concerns simultaneously simple problems, such as evaluating the fire-resistance rating, or more complex fire separations continuity checking. The value of the fire-resistance rating of an element is obtained by several means. It can be retrieved from generic lists (e.g., in NBCC) or proprietary tests for some assemblies and from structural calculations for other elements. The results of the standard test CAN/ULC-S101, Fire Endurance Tests of Building Construction and Materials, are the input parameter for fire-resistance rating of assemblies such as partitions. For structural calculations, the resistance of the residual section must be determined. This requires notably an extensive structural study, which is done separately by the structural engineer. The data (such as structural loading of elements) that would be required for deriving the fire-resistance rating is considered. Therefore, in the case of the fire-resistance rating, the fire-resistance rating is considered as a parameter that is not derived, and that will be an input by a manufacturer for example. Based on Solihin's categorization [38], the first example is evaluated as directly available information. The second example falls into the category of an extended data structure, as there is a need for more complex information manipulation.

During the study of the NBCC, an inconsistency in the appellation, especially, considering spaces, was found. When only several clauses were isolated, the lack of a clear definition of the above-listed spaces made it impossible to directly study the compliance checking without completely defining the redundancies. All the different names that are used in Part 3 of the NBCC and classified in Dynamo as shown above (cf. Section 4.2.2). It is concluded based on the NBCC analysis that the requirements for fire safety of a specific object will differ depending on which individual space they are related to.

Therefore, the considered spaces are listed in an exhaustive way (Table 2) to be taken into account during the building model preparation.

Table 2. Lists of spaces as mentioned in the National Building Code of Canada (NBCC) classified by the three different uses.

Correspondence with Dynamo Classification of Spaces	One Floor Compartment with Occupants (Room or Area Lists)	Compartments Going Through More Floors (Geometry Lists)	Service Spaces (Not Represented in the Building Model)
NBCC Spaces	Service room	Stairway	Horizontal duct
	Dwelling unit	Staircase	Attic
	Electric equipment chamber	Exit stairway	Roof space
	Public corridor	Emergency stairwell	
	Laundry room		Suspended ceiling void
	Lobby	Vertical duct	Horizontal service space
	Rooftop structure		Horizontal void
	bathroom	Lift	
	Washroom		Vertical service space
	Suite	Laundry chute	Vertical void
	Hallway	Garbage chute	
	Secondary room		Service space
	Mezzanine		Construction void
	Basement		Sanitary void
	Basement garage		Plenum space

4.3.3. Model Attribute Table

A Model Attribute Table (MAT) is constructed from the building code analysis regarding passive fire protection objects. The MAT is needed to define the required data and, therefore, to prepare the building model appropriately (Figure 9). The table was specifically adapted to suit the code compliance approach and the visual programming paradigm. This table integrates required objects and attributes, as well as its parameter type (cf. Section 4.3.1). This creates a link between the objects of the building code and the way it is obtained from the building model through the parameter type. Finally, the category is needed because, in the Dynamo scripts, the objects are first selected from a model based on their category.

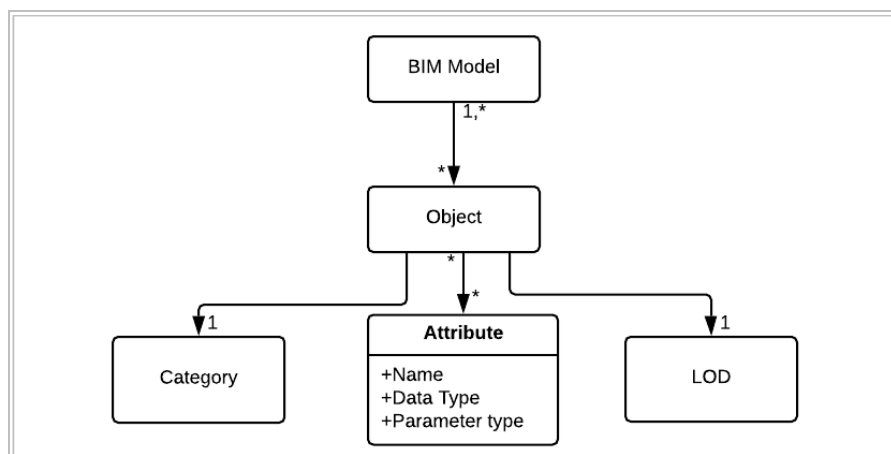
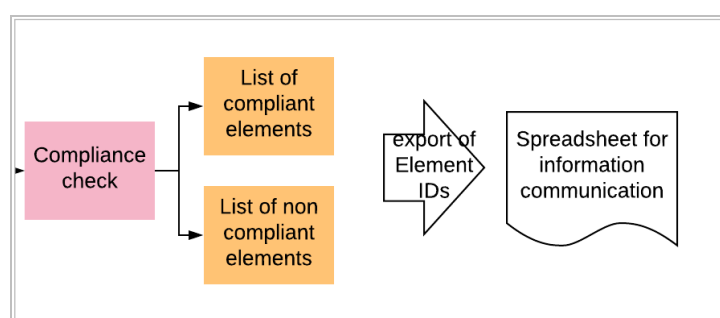


Figure 9. Conceptual model of the information needed for preparing a building model.

4.4. Realization of Rule Checking with Evaluation

As specified in the Section 4.2.1, the implemented script returns two separate Dynamo lists of compliant and non-compliant elements. The rule execution and results must be presented in a way that can be further manipulated and so that the data can be communicated. The visualization inside Dynamo is not sufficient. The information must be retrieved in a rule check report in order to be able to use the data, such as to communicate or save the results to track progression.

Dynamo can export information to spreadsheets; therefore, the element ID and the family type of each element are exported (Figure 10). A search through the model by element ID is then possible in Revit to identify individually the elements that need to be rectified by the functionality “Select by ID”. The family type of an element is complementary information that was added as it contains a more explicit and user-friendly element identification. The export of element IDs is done through an association of several nodes and the final form of the reporting spreadsheet (Figure 10).



(a)

7762891	15C_156mm_PR
5958075	42B_118mm_PR
5958076	42B_118mm_PR
6577332	15C_156mm_PR
7766929	4D_184mm
6577330	15C_156mm_PR
7753681	4D_184mm
6577328	15C_156mm_PR

Walls-Ceilings Walls-Floors

(b)

Figure 10. (a) Visual programming script structure for any compliance check; (b) Spreadsheet visual of the exported non-compliant lists; example of continuity.

5. Results: Proof of Concept

In the methodology, essential passive protection concepts were designated. Elements of each of the following concepts are implemented: fire-resistance rating, flame spread rating, and continuity. The scripts were executed and adapted to a real construction project building model, an eight-story mass timber residential building located in Montreal, Quebec. Its architectural building model was prepared according to the developed method. Then the implemented scripts were evaluated based on the criteria defined beforehand.

5.1. Fire-Resistance Rating

The fire-resistance is the aptitude of an element to resist to a normalized fire based on three criteria of performance: mechanical resistance, integrity, and insulation. The fire-resistance rating

(FRR) represents the time where all the required criteria are respected. The present value in the building model must be compared to the values required by the code (Box 1). It can be seen that the requirements depend on the location of a given element.

Box 1. Excerpt of Mass timber buildings up to 12 floors—main fire-resistance ratings requirements.

- 1.3.1. The floors shall form a fire separation that has a fire-resistance rating of at least 2 h.
 1.3.4. Walls, columns, and assemblies and fasteners shall have a fire-resistance rating of at least 2 h.
 1.3.7. Fire separations between suites and between a public corridor and a suite shall have a fire-resistance rating of at least 1 h.
 1.5.4. All vertical service spaces, including garbage chutes, shall be isolated from the rest of the building by fire separations. They shall have a fire-resistance rating of at least 2 h and be of incombustible construction.

The FRR concerns both structural and non-structural elements. This represents one of the examples of interfaces between fire-engineering, architectural, and structural domains. That means that one parameter can be retrieved from numerous linked models with different modeling practices. In the rule script (Figure 11), it is assumed that the information about the FRR parameter is contained either in the name parameter (containing, e.g., RAF 2 h for two-hour FRR) or in a specific fire rating parameter (automatically associated with its IFC format equivalent called FireRating). It is noted that the visual programming script is dependent on such details as units in which the FRR (minutes or hours) is expressed and that the building model must be prepared consistently regarding the units.

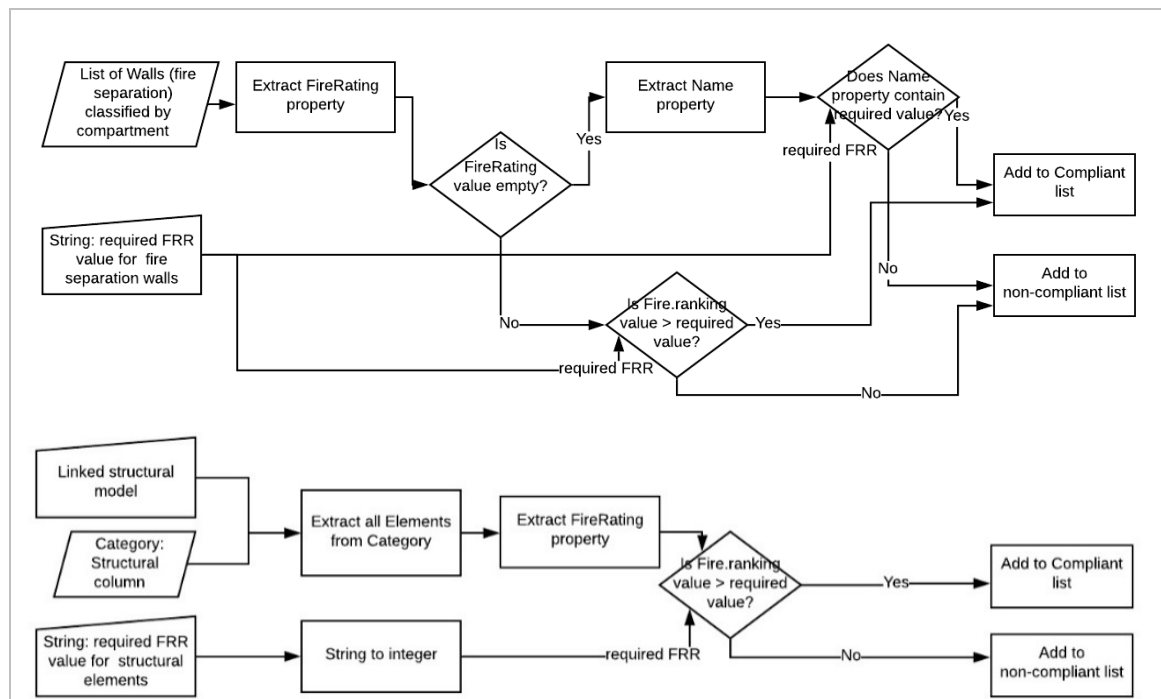


Figure 11. Logical representation of example rule for fire separation interior walls and structural columns.

Completeness: The protection of assemblies is an essential aspect in passive protection, especially in wood construction, where metal assemblies are likely to compromise the insulating capacity of the charred layer (due to the heat conduction) of a solid wood element. Automating the verification of assemblies requires a level of detail of the order of LOD400. Furthermore, the assemblies are usually created in software that requires IFC import to Dynamo (e.g., Cadwork), but the reality today is that such details are not transferred through IFC format. This level of detailed information is also not possible to exploit in Dynamo today.

Usability: The Dynamo script is able to detect the FRR information by two means. This contributes to limiting the rigidity and the remodeling needs. Furthermore, all the necessary parameters for this rule are automatic or built in.

Accuracy: The results of the Dynamo script (Figure 12) show, that a wall’s FRR parameter containing an incorrect or no value will be detected as non-compliant.

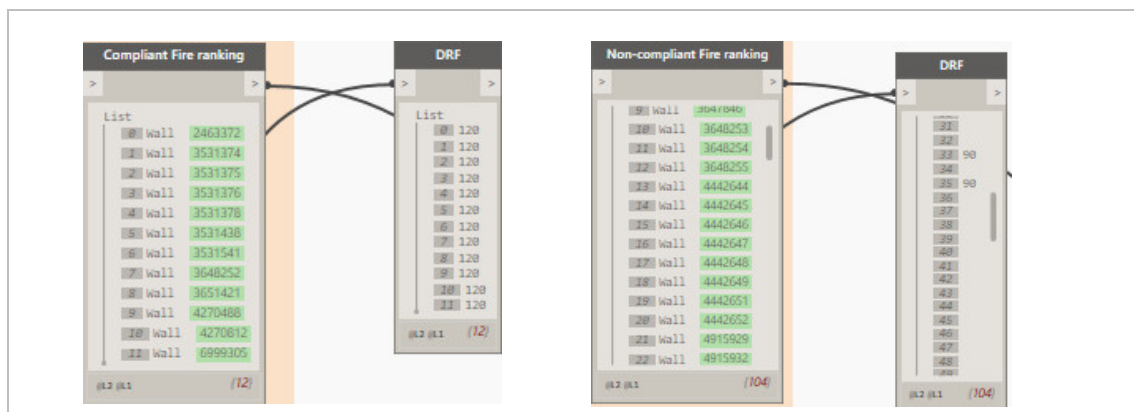


Figure 12. Complaint and non-compliant structural walls and their respective fire-resistance rating (FRR) values.

5.2. Flame Spread Rating

The goal is to verify that all the interior finishes have the proper flame spread rating based on the NBCC prescriptive requirements (Box 2). The required FSR of each object is defined by its spatial position in the NBCC (Table 3).

Box 2. Excerpt of NBCC 2010, amended for Québec, §. 3.1.3.12.1.

Except as otherwise required or permitted by this Subsection, the flame-spread rating of interior wall and ceiling finishes, including glazing and skylights, shall be not more than 150 and shall conform to Table 3.

Table 3. Excerpt of NBCC 2010, amended for Québec, §. 3.1.13.2. [4].

Space	Flame-Spread Rating—Walls and Ceilings (Sprinklered)
Exits	25
Vertical service spaces	25
Entrance halls	25
Other lining	150
Compartment doors	200

The elements and categories that are considered are the walls, ceilings, slabs, glass, curtain walls, windows (skylights) and doors. The flame spread rating (FSR) is related to the material and therefore it is a type parameter. An IFC parameter defined as SurfaceSpreadOfFlame exists in the PropertySet. However, it is not considered useful to enter the FSR manually for each family type. The material parameter is a more common and versatile parameter in the building models. The FSR can be automatically derived from the material parameter for generic values. Therefore, the compliance can be checked without adding any more data to the building model directly. The generic values of FSR in NBCC, amended Québec, table D.-3.1.1.A. [4] are used to associate FSR value to each material and this association is automated in Dynamo.

The information about materials of a specific element is stored either as the material parameter or in the name parameter. Therefore, the determination of the finish’s material will be done by examining both of these parameters. The goal is to associate each material with a corresponding FSR through

manipulation of String lists (Figure 4). The user can modify at any time the lists containing different materials of the same FSR, if necessary.

The material property designation lack of standard in the BIM and therefore their names are not unified. It can be expected that from one model to another, the names of materials will not be represented in the same way, given that there is a choice of diacritics, language, capitals etc. The retrieved finish material names from each object are normalized by removing the capitals and the diacritics, through a Python script. This is to identify more easily the name in the building model with the corresponding name in the Dynamo script. Once the proper value of FSR is associated by the visual program to each material, it is necessary to verify that the FSR value is correct with the NBCC requirements. Based on the regulation text analysis, a Dynamo script is developed to pair to each individual space the prescribed corresponding flame spread rating (Table 3). The verification of the flame spread rating of the finish materials is performed with a short hard-coded Python script, which is needed in order to handle two lists of different dimensions. The output of this node is a true/false/unknown list, corresponding to the compliance of each finish material associated with each object.

In regards to the FSR verification, there is no need for new parameters, every information is either automatically set in during modeling (material parameter), or the parameter can be derived from available data. It is still to be noted that there are specific needs in terms of modeling. All the rooms must be assigned a name, especially the ones that are subjected to a certain constraint in terms of FSR (escape routes, vertical service spaces, and entrance halls) and that are easily non identified (vertical service spaces). If the rooms are not identified in any way by the modeler, the rule checking code cannot detect an anomaly.

Completeness: Two types of elements were not detected by the Dynamo script in the real building model used for the proof of concept: the shaft opening and the garbage chute. The garbage chute (in the real project building mode, the garbage chute was tagged, but only its entry and not the chute itself). However, these elements must be carefully designated, because they are going through several floors and therefore are a privileged way of fire propagation.

Usability: The Dynamo script is able to detect the FSR information by two means. This contributes to limit the rigidity and the remodeling needs. Furthermore, all the necessary parameters for this rule are automatic or built-in.

Accuracy: The accuracy of the rule was re-checked, visually verifying the Material names and their values.

5.3. Continuity of Fire Separations

The integrity of the compartment is respected if every vertical fire separation has a connecting element above and below (Box 3). For horizontal compartments, the fire separation pairs are comprised of a wall and a slab above and below. In order to simplify the study, suspended ceilings are not considered as fire separations (Figure 5). For the vertical compartments, the fire separation pairs are comprised of walls and a slab on the side of the wall.

Box 3. Excerpt from NBCC 2010, amended Québec, § 3.1.8.3.4.

The continuity of a fire separation shall be maintained where it abuts another fire separation, a floor, a ceiling, a roof, or an exterior wall assembly.

The following categories of elements are classified by compartment (Figure 13): the walls at floor level n , slabs at floor level n (below wall), and slabs at floor level $n + 1$ (above the wall). The upper surface of a wall and the bottom surface of a slab are checked for intersection in order to associate a slab paired to it, below and above (Figure 14).

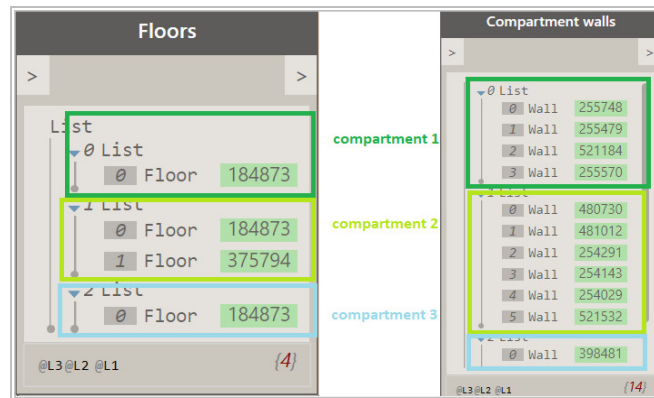
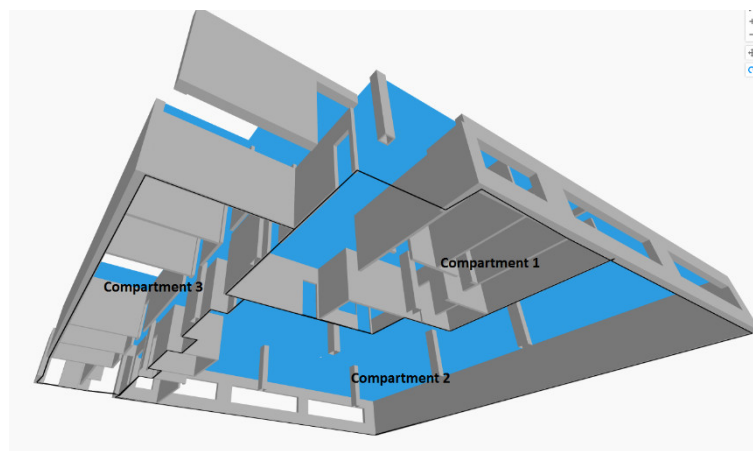
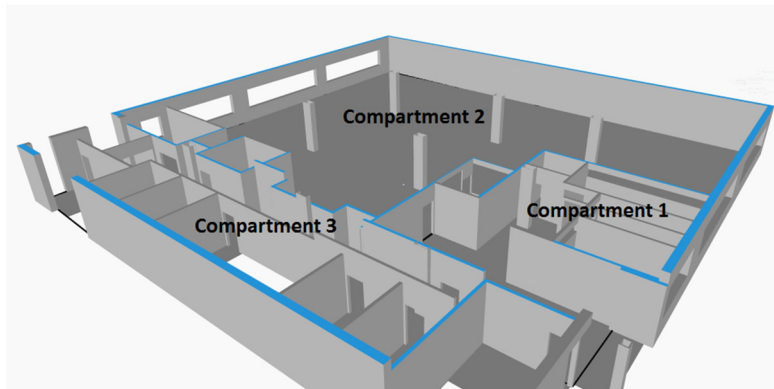


Figure 13. Corresponding levels of sublists for geometric study for the floor level n.



(a)



(b)

Figure 14. The highlighted geometries are checked if they overlap to respect fire separation continuity. (a) Slab on floor level n + 1 lower surface; (b) Fire separating walls highlighted on level n upper surface with visible slab (floor level n).

Completeness: This Dynamo script carries out a complex geometrical study, selecting only compartment boundaries and inspecting their intersection. The Dynamo script managed to cover both structural and non-structural partitions and walls. However, it could not take into consideration the continuity of suspended ceilings or other vertical elements. This script is limited to checking the continuity of fire separation boundaries. However, it does not yet address the integrity of the fire separations in terms of protection of openings (e.g., dampers). Automating the verification of fire

separation assemblies at construction joints requires a level of detail of the order of LOD400. Similar to the case of metal assemblies, joint details are usually not modeled by an architectural model.

Usability: The Dynamo script is able to detect the based on the geometry of the model, all the necessary parameters for this rule are automatic or built in.

Accuracy: Details regarding a partition's height are sometimes neglected. Therefore, there are situations, where the gypsum board, with an encapsulating function for a timber wall, does not reach the ceiling above and therefore exposes the timber structure. The rule check accuracy is validated as even such small errors were detected by the script.

5.4. Summary of the Evaluation of the Proposed Method

To validate the proposed method in a design science perspective, it is important to evaluate the completeness, the consistency, the usability, and the accuracy of its main components [48]. Based on the criteria developed, the four different rule domains have been tested on two models in order to obtain the resulting evaluation. The results of this evaluation, detailed in the previous Sections, are summarized in Table 4.

It can be noted that the completeness of the rules could not be entirely met, this is due mainly to a required higher LOD than what is currently usual in practice, and therefore, more research is required in terms of the implementation of other rules. The consistency of the rule implementation is generally successful; this can be justified by the constructed MAT. In terms of usability, a limited number of new manually added parameters had to be created. Finally, the accuracy of the implemented rules was generally successful, even though more research is required for fire separations retrieval, more specifically, regarding the vertical service spaces.

Table 4. Results of the evaluation by four defined criteria of the example scripts.

Criteria	Non-Geometrical Parameters		Parameters with Geometrical Study	
	Fire-Resistance Rating	Flame Spread Rating	Fire Separations Retrieval	Continuity
Completeness	≈	✓	✓	≈
Consistency	✓	✓	✓	✓
Usability	✓	✓	✓	✓
Accuracy	✓	✓	≈	✓

✓ the evaluation criterion is entirely satisfied; ≈ criterion is not entirely satisfied.

6. Discussion

The challenges of information transfer and integration were identified in this paper. During the rule construction, it turned out that the steps for data retrieval and parameter derivation rely heavily on how the information is stored in the model. This shows that it can be important that the modeling standards remain vague in order to allow the modelers freedom in the input of the semantics; however, it makes the data retrieval more difficult. Nevertheless, if a more generalized tool for code checking was to be executed, specific modeling practice standards would be needed that do not exist today.

The formalization in Dynamo for geometrical requirements needs to be further studied. As an example, the method of vertical shaft compartment is only adapted to vertical shafts that keep a similar geometry between levels. For a more complex geometry, such as vertical service spaces for ventilation ducts that can change direction, a different approach would have to be developed.

Furthermore, this study solely focuses on the architectural and structural interfaces of the fire concepts, more specifically, on fire-resistance rating, flame spread rating, and continuity examples. The study chose to limit its work to these main concepts. The continuity and integrity of fire separations could be further developed in future works, as well as other interfaces (HVAC, plumbing, electrical engineering) [50]. Mechanical systems can contribute to fire propagation, especially if ventilation is assured mechanically. Therefore, NBCC requirements emit specifications related to service equipment

design such as limiting the amount of exhaust air between floors, smoke extraction etc. These aspects of fire protection cannot be addressed by the automated verification of the model, because they require a performance-based design and calculations that a simple compliance-checking tool cannot assure. The retrieval of data for BIM rule checking with data obtained from a performance-based external analysis is yet to be integrated [7,25].

7. Conclusions

The method's intent is to build an alternative global solution using different tools that are accessible to everyone and that could enable architecture, engineering and construction (AEC) professionals to conduct a code checking evaluation of their design. This study's goal was to determine whether an automated code checking method was adaptable to the fire protection design. The five steps of the whole process were described in a detailed way. Examples of both non-geometrical and geometrical studies to verify compliance of a building model regarding NBCC fire protection requirements were shown. It was demonstrated in the complete study of the method for compliance rule construction, that fire safety checking requires a lot of extensive data manipulation. Therefore, it is concluded that the fire safety compliance check is possible at extensive building model preparation during the design phase.

Compared to recent similar research works [25,26,31], the proposal is more comprehensive and provides structured knowledge based on different scientific domains and a bottom-up understanding of the specific context of Quebec. Moreover, the proposed method relies on a use of visual programming approach, the interface of which "is more in line with architects' design habits" [45]. Indeed, the visual programming has proven to be a powerful tool in regards to the data derivation and retrieval of specific information from the building model; it has performed well in the context of the employed method especially for the non-geometrical elements.

However, several domains related to fire protection requirements cannot be covered by this method. The parameters required to run a compliance check can be too detailed and complicated to input. In many situations, it would require adding a new parameter, which serves only one specific purpose and therefore adds a burden on the modelling. This can be insufficiently beneficial to justify the additional work. It remains, therefore, challenging to check more complicated aspects related to the geometrical study, such as fire dampers position or determination of radiant heat view factor for limiting distance calculation. This kind of data manipulation can be done if hard-coded classification and manipulation of information is conducted. Another limitation of the research may be related to the fact that a proprietary visual programming tool has been used, even though this choice is justified by the fact that Revit is by far the most used BIM design authoring tool in the Canadian construction industry.

Through this research, it is possible to present some recommendations regarding fire protection needs, identifying the gaps in development and identifying domains that are not yet fully integrated in today's BIM models. The MAT can also be considered as a proposal for family creators such as manufacturers for additional parameters to be included in their modeling practices for doors or lifts among others. In this work, example rules are offered that were proofed in practice on two different models in order to demonstrate that this method seeks mainly to withstand the challenges of different modeling practices. This has helped us to issue recommendations concerning data input consistency and reusability.

To conclude, this study offers a new view on the BIM requirements for automated rule checking and in the context of fire protection, showing especially that detailed rule checks ask for very precise information input, which is not considered a standard for everyone today.

Author Contributions: Conceptualization, K.K., C.B. and P.B.; Methodology, K.K. and C.B.; Investigation, K.K.; resources, C.B., P.B. and C.D.; Writing—original draft preparation, K.K.; Writing—review and editing, K.K., C.B., P.B. and C.D.; Supervision, C.B. and P.B.; Validation C.B., Project administration, K.K. and C.B.; Funding acquisition, P.B. All authors have read and agreed to the published version of the manuscript.

Funding: The authors are grateful to Natural Sciences and Engineering Research Council of Canada for the financial support through its ICP and CRD programs (IRCPJ 461745-12 and RDCPJ 445200-12) as well as the industrial partners of the NSERC industrial chair on eco-responsible wood construction (CIRCERB).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. British Columbia Laws. *Wood First Act*; Queen's Printer: Victoria, BC, Canada, 2009.
2. Ministry of Forests, Wildlife and Parcs. *The Wood Charter*; MFWP: Quebec City, QC, Canada, 2017.
3. Veilleux, L.; Gagnon, S.; Dagenais, C. *Bâtiments De Construction Massive En Bois D'au Plus 12 Étages*; Régie du bâtiment du Québec: Montreal, QC, Canada, 2015.
4. CNRC. *Code National Du Bâtiment-Canada, Division B-Partie 3: Protection Contre L'incendie, Sécurité Des Occupants Et Accessibilité*; Conseil national de recherches: Ottawa, ON, Canada, 2010.
5. Dagenais, C.; Desjardins, R. Cases Studies of Performance-Based Design for Mid-Rise Wood Constructions in Quebec (Canada). In Proceedings of the World Conference on Timber Engineering, Auckland, New Zealand, 15–19 July 2012.
6. Eastman, C.M.; Lee, J.-M.; Jeong, Y.-S.; Lee, J.-K. Automatic rule-based checking of building designs. *Autom. Constr.* **2009**, *18*, 1011–1033. [CrossRef]
7. BIMForum. Level of Development Specification: For Building Information Models. Available online: <http://www.bimforum.org/lod> (accessed on 20 June 2018).
8. Eastman, C. Automated Assessment of Early Concept Designs. *Archit. Des.* **2009**, *79*, 52–57. [CrossRef]
9. Beach, T.; Rezgui, Y.; Li, H.; Kasim, T. A rule-based semantic approach for automated regulatory compliance in the construction sector. *Expert Syst. Appl.* **2015**, *42*, 5219–5231. [CrossRef]
10. Norén, J.; Strömberg, M.; Nystedt, F. *Fire Safety Design and Building Information Modelling (Bim)–Needs and Expectations in the Building Industry*; Briab Brand & Riskingenjörerna AB: Malmö, Sweden, 2018.
11. SFPE. Building Information Modeling and Fire Protection Engineering. In *Position Statement P-05-11*; Society of Fire Protection Engineers: Gaithersburg, MD, USA, 2011.
12. AgaCAD. Fire Sprinklers. Available online: <http://www.aga-cad.com/products/bim-solutions/mep-engineering-smart-sprinklers/features> (accessed on 10 June 2020).
13. AutoSPRINK. Fire Sprinkler Design in Revit. Available online: <https://autosprinkrvt.com/> (accessed on 10 June 2020).
14. Dimyadi, J.; Solihin, W.; Amor, R. Using Ifc to Support Enclosure Fire Dynamics Simulation. In Proceedings of the 25th EG-ICE International Workshop on the Advanced Computing Strategies for Engineering, Cham, Switzerland, 10–13 June 2018.
15. Dimyadi, J.; Amor, R.; Spearpoint, M. Using Bim to Support Simulation of Compliant Building Evacuation. In Proceedings of the 11th European Conference on Product and Process Modelling (ECPMM), Limassol, Republic of Cyprus, 7–9 September 2016.
16. Porto, M.F.; Franco, J.R.Q.; Vianna, B.F.; Porto, R.M.A.B. Automatic Code Checking Applied to Fire Fighting and Panic Projects in a Bim Environment-Bimscip. *Syst. Cybern. Inform.* **2017**, *15*, 86–90.
17. University of Toronto. *Canada Wide Survey, 2nd Annual Bim Report*; Building Tall Research Centre: Toronto, ON, Canada, 2019.
18. Buchanan, A.H.; Abu, A.K. *Structural Design for Fire Safety*; John Wiley & Sons, Inc.: Chichester, UK, 2001.
19. Canadian Wood Council. *Fire Safety Design in Buildings*; Canadian Wood Council: Ottawa, ON, Canada, 1996.
20. Östman, B.; Brandon, D.; Frantzich, H. Fire safety engineering in timber buildings. *Fire Saf. J.* **2017**, *91*, 11–20. [CrossRef]
21. Narayanaswamy, H.; Liu, H.; Al-Hussein, M. BIM-Based Automated Design Checking for Building Permit in the Light-Frame Building Industry. In Proceedings of the 36th International Symposium on Automation and Robotics in Construction, ISARC 2019, Banff, AB, Canada, 21–24 May 2019.
22. Botton, C.; Forgues, D.; Halin, G. A framework for building information modeling implementation in engineering education. *Can. J. Civ. Eng.* **2018**, *45*, 866–877. [CrossRef]
23. Nawari, N.O. A Generalized Adaptive Framework (GAF) for Automating Code Compliance Checking. *Building* **2019**, *9*, 86. [CrossRef]

24. Dimyadi, J.; Amor, R. Automated Building Code Compliance Checking—Where Is It At. In Proceedings of the CIB WBC, Brisbane, Australia, 5–9 May 2013.
25. Li, Y. Automated Code-Checking of Bim Models. Master's Thesis, Universidad de Cantabria, Cantabria, Spain, 2015.
26. Taciuc, A.; Karlshøj, J.; Dederichs, A. Development of Ifc Based Fire Safety Assessment Tools. In Proceedings of the International RILEM Conference Materials, Systems and Structures in Civil Engineering, Lyngby, Denmark, 15–18 August 2016.
27. Solihin, W.; Shaikh, N.; Rong, X.; Lam, K. Beyond Interoperability of Building Model: A Case for Code Compliance Checking. Available online: https://www.researchgate.net/publication/280598933_BEYOND_INTEROPERATIBILITY_OF_BUILDING_MODEL_A_CASE_FOR_CODE_COMPLIANCE_CHECKING (accessed on 3 April 2019).
28. Ding, L.; Drogemuller, R.; Rosenman, M.; Marchant, D.; Gero, J. Automating Code Checking for Building Designs—Designcheck. *Clients Driv. Innov. Mov. Ideas into Pract.* **2006**, *2016*, 1–16.
29. Kincelova, K.; Botton, C.; Blanchet, P.; Dagenais, C. Bim-Based Code Compliance Checking for Fire Safety in Timber Buildings: A Comparison of Existing Tools. In Proceedings of the CSCE/CRC Construction Specialty Conference, Montréal, QC, Canada, 12–15 June 2019.
30. Greenwood, D.; Lockley, S.; Malsane, S.; Matthews, J. Automated Compliance Checking Using Building Information Models. In Proceedings of the The Construction, Building and Real Estate Research Conference of the Royal Institution of Chartered Surveyor, Paris, France, 2–3 September 2010.
31. Preidel, C.; Borrmann, A. Automated Code Compliance Checking Based on a Visual Language and Building Information Modeling. In Proceedings of the International Symposium on Automation and Robotics in Construction, Oulu, Finland, 24 July 2015.
32. Nawari, N.O. *Building Information Modeling: Automated Code Checking and Compliance Processes*; CRC Press: Cleveland, OH, USA, 2018.
33. Malsane, S.; Matthews, J.; Lockley, S.; Love, P.E.; Greenwood, D. Development of an object model for automated compliance checking. *Autom. Constr.* **2015**, *49*, 51–58. [[CrossRef](#)]
34. Kim, H.; Lee, J.K.; Shin, J.; Choi, J. Bim-Supported Visual Language to Define Building Design Regulations. In Proceedings of the 22nd International Conference on Computer-Aided Architectural Design Research in Asia: Protocols, Flows and Glitches, Hong Kong, China, 1 January 2017.
35. Hjelseth, E.; Nisbet, N. Exploring Semantic Based Model Checking. In Proceedings of the 27th CIB W78 international conference, Cairo, Egypt, 16–18 November 2010.
36. Ilal, S.M.; Günaydın, H.M. Computer representation of building codes for automated compliance checking. *Autom. Constr.* **2017**, *82*, 43–58. [[CrossRef](#)]
37. Solihin, W.; Eastman, C. Classification of rules for automated BIM rule checking development. *Autom. Constr.* **2015**, *53*, 69–82. [[CrossRef](#)]
38. buildingSMART. *Report on Open Standards for Regulations, Requirements and Recommendations Content*; Regulatory Room Working Group: Geneva, Switzerland, 2017.
39. Ghannad, P.; Lee, Y.-C.; Dimyadi, J.; Solihin, W. Automated BIM data validation integrating open-standard schema with visual programming language. *Adv. Eng. Inform.* **2019**, *40*, 14–28. [[CrossRef](#)]
40. Nguyen, T.-H.; Kim, J.-L. Building Code Compliance Checking Using BIM Technology. In Proceedings of the 2011 Winter Simulation Conference (WSC), Phoenix, AZ, USA, 11–14 December 2011; pp. 3395–3400.
41. Porto, M.; Franco, J.; Correa, L.; Alves, L.; Baracho, R. BIM as a Structural Safety Study Tool in Case of Fire-BIMscip. *Syst. Cybern. Inform.* **2018**, *16*, 81–86.
42. Lee, H.; Lee, J.-K.; SeoKyung, P.; Kim, I. Translating building legislation into a computer-executable format for evaluating building permit requirements. *Autom. Constr.* **2016**, *71*, 49–61. [[CrossRef](#)]
43. Preidel, C.; Borrmann, A. Towards Code Compliance Checking on the Basis of a Visual Programming Language. *J. Inf. Technol. Constr. ITcon* **2016**, *21*, 402–421.
44. Boeykens, S.; Neuckermans, H. Visual Programming in Architecture: Should Architects Be Trained as Programmers? In Proceedings of the CAAD Futures Conference, Montréal, QC, Canada, 17–19 July 2009.
45. Wu, N.; Wang, M. The Integration of Procedural Information in Traditional Architectural Design. *Comput. Des. Appl.* **2018**, *16*, 16. [[CrossRef](#)]
46. Spearpoint, M. Fire Engineering Properties in the Ifc Building Product Model and Mapping to Branzfire. *Int. J. Eng. Perform. Based Fire Codes* **2005**, *7*, 134–147.

47. Lee, Y.-C.; Eastman, C.M.; Solihin, W. An ontology-based approach for developing data exchange requirements and model views of building information modeling. *Adv. Eng. Inform.* **2016**, *30*, 354–367. [[CrossRef](#)]
48. Hevner, A.R.; March, S.T.; Ram, J.P. Design Science in Information Systems Research. *MIS Q.* **2004**, *28*, 75. [[CrossRef](#)]
49. Autodesk CFD. Building Hvac: Software in Action. Available online: <https://www.autodesk.com/solutions/simulation/building-hvac> (accessed on 3 April 2019).
50. Lataille, J.I. *Fire Protection Engineering in Building Design*; Elsevier: Amsterdam, The Netherlands, 2002.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).