2021 European Conference on Computing in Construction
Online eConference
July 26-28, 2021

# OBJECT-ORIENTED COMPLIANCE CHECKING: AN AUTOMATED APPROACH AND A CASE STUDY

Omar Doukari[1], David Greenwood[1], Kay Rogage[2], Mohamad Kassem1
[1] Department of Mechanical & Construction Engineering,
[2] Department of Computer and Information Sciences,
Northumbria University, Newcastle-Upon-Tyne, NE1 8ST, UK

## ABSTRACT

This paper proposes a novel 'bottom-up' object-oriented approach for automated model checking and the corresponding plugin prototype. The approach and the prototype enable four key processes: (i) define and interpret the requirement rules, (ii) prepare the BIM object data, (iii) execute the requirement rules, and (iv) report the results that were made available to the user. To demonstrate the feasibility and accuracy of the approach, we use a case study from the foodservice equipment industry using an existing object library of foodservice equipment created by a major French manufacturer. All four steps were successfully completed, and the results show savings of around 125 minutes per object between the automated approach and traditional manual methods of working.

## INTRODUCTION

Reuse of asset data throughout the design, construction and operational phases of buildings relies on the compliance of several components: building codes, normative standards, industry guidelines, and project requirements. Traditionally compliance to these components is achieved through a manual data checking process which is laborious, time consuming, and error-prone (Ghannad, et al., 2019). Errors relating to manual processing have been known to result in huge losses (e.g., £800,000 cost of construction and design changes within a UK's housing project, where the wheelchair ramps were found to be too steep and narrow (Ding, et al., 2006)).

Building Information Modelling (BIM) is part of a digital transition within the construction industry. It can be defined as a set of processes and technologies that support multiple stakeholders to collaboratively design, construct and operate a facility within a digital environment (Sacks, et al., 2018). The Building Information Model is an object-based, data rich, 3D digital model generated by each project participant through a BIM authoring platform. BIM is expected to significantly contribute to the development of automated checking of designs (McGraw-Hill, 2014). Indeed, with the development of computer-based BIM applications allowing multi-criteria parametric designs of a construction project through a set of objects defined by an identity, attributes, and relations, automatic compliance checking of building designs is becoming feasible (Choi & Kim, 2008) thus, contributing significantly to time and cost savings.

During the last four decades, more than 400 research works have considered automating compliance checking in the construction industry (Nawari, 2019a). Various model checking methodologies, platforms, and domain applications such as spatial assessment, structural integrity, safety, energy usage, etc., have been contemplated in the academic research literature and multiple industrial tools are now commercially available.

In one of the most referenced studies, Eastman et al. (Eastman, et al., 2009) presented a general functional architecture for rule checking and reporting systems. These authors structured a rule checking process into four main steps: (i) rule interpretation, (ii) building model preparation, (iii) rule execution, and (iv) rule check reporting. They discussed the shortcomings of each step and identified general requirements. Five rule checking systems were reviewed according to this structure: the CORENET project (Singapore), the HITOS project (Norwegian BIM project for Statsbygg), the Australian Building Codes Board project, the International Code Council project (US), and the General Services Administration project. As input, these systems rely on IFC-based building information models.

Based on the same rule checking systems, Greenwood et al. (Greenwood, et al., 2010) identified four key requirements to promote UK's BIM-based automated code checking. To overcome the lack of relevant information needed for enabling efficient compliance checking process, the authors proposed: (i) making programmed rules easily understandable and accessible by UK's Regulation authors, (ii) making the rule base independent of the rule checking software so different rule sets can be used with the same tool, (iii) complying with open standards like the IFC, and (iv) understanding and taking into account the model authoring process while developing such systems.

More recently, several model checking system classifications have been proposed. Hjelseth (Hjelseth, 2016) used an ontological framework based on the concepts of model checking to identify the different types of checking. He identified four different categories: (i) "*validation checking*" of the building information model's content with the rules set (regulation, standards, contract, etc.), (ii) "*model content checking*" of the completeness of the building information model's content with regards to a particular use-case, (iii) "*smart object checking*" of the model's objects with behavioural rules, and (iv) "*design option checking*" to support and guide the design process with respect to best practices.

Krijnen (Krijnen, 2016) gave an overview of technical solutions to automate data requirements checking and proposed a classification based on these technologies, namely: schema semantics and IFC [(Eastman, et al., 2009), (Terkaj & Šojić, 2015)], Model View Definitions [(Zhang, et al., 2014), (Solihin & Eastman, 2015), (Solihin, et al., 2015)], concept libraries [(Palos, et al., 2014), (Miller, 1995), (Navigli & Ponzetto, 2012)], query languages (Pauwels & Terkaj, 2016), and reasoners (Krijnen & Tamke, 2015).

Automated compliance checking methods can also be distinguished with respect to whether the encoding of the rules set is embedded in the BIM model [(Zhang & El-Gohary, 2016), (Hakim, et al., 2017), (Tan, et al., 2010)] or not [(Macit & Suter, 2015), (Dimyadi, et al., 2016), (Zhong, et al., 2018)]. To provide a bidirectional link and reduce the gap between design requirements and design solutions, Marchant (Marchant, 2015) suggested *as-briefed* and *as-designed* data could be correlated within integrated building information models and in a single IFC-based repository. To formalise the way requirements can be modelled using the same data schema as the design solution, Marchant (Marchant, 2016) suggested extending the IFC specification for managing properties, documents and adjacency requirements usually contained in codes, normative texts and client's requirements by using a new *IfcRelRequires* and related sub-classes. Furthermore, for some frameworks, the rules set can be either integrated, hard-encoded into the BIM-based code checking tool [(Benghi, 2019), (Zhang, et al., 2011)] or treated independently of it [(Nawari, 2019c), (Messaoudi & Nawari, 2020)] which could be costly to maintain and difficult to change (Zhang & El-Gohary, 2016).

From a deployment point of view, Nawari (Nawari, 2019a) identified three model checking system categories: (i) add-in applications integrated within a specific authoring tool, (ii) desktop (or standalone) software independent of any authoring tool, and (iii) web-based platforms providing distant access via the internet. According to Nawari (Nawari, 2019a), all "*the cited methods for automated rules compliance auditing in building design are either based on proprietary frameworks, domain-specific areas, or hard-coded rule-based representations*". A Generalized Adaptive Framework for Building code compliance checking based on the IFC standard, was then proposed and developed in five steps: (i) analysis & classification of existing regulation, (ii) development of the Model View Definition, (iii) unambiguous data extraction, (iv) uncertain data extraction using fuzzy logic, and finally (v) rules execution (Nawari, 2019c).

To develop a roadmap for automated compliance checking adoption, Beach et al. (Beach, et al., 2020) conducted a detailed inventory of applicable industrial and academic developments. With the aid of industry partners, the authors identified and analysed the tools currently allowing model compliance checking. Ten existing industry tools were identified (AEC3 Require1, Autodesk Model Checker, Brief-Builder, CARS, GliderBIM, Xinaps, UpCodes AI, SMART review, Jotne EDMmodelchecker, and Solibri Site or Enterprise Versions). These were examined with respect to their: application domains (client requirements, fire safety, etc), capabilities for allowing digitisations, checking methodology, and input data format. Several other academic research platforms were analysed using these same criteria. Then, by conducting a survey with industry partners, the authors identified a list of obstacles to the adoption of automated compliance checking that led them to propose a roadmap that considered, concurrently, the political, commercial, and technological factors in the future development of automated compliance checking (Beach, et al., 2020). Although industrial partners were in favour of adopting automated compliance checking, they insisted on the necessity for designating a qualified human to supervise the whole process.

This paper proposes a bottom-up object-oriented approach for automated compliance checking using the foodservice equipment industry as a case study. A review of related studies is presented in the next section. This is followed by the presentation of our proposed compliance checking method and its demonstration in a case study of a large foodservice equipment manufacturer using an existing object library.

## RELATED STUDIES

To understand the application domain and different frameworks for automated compliance checking, a review of 34 research papers was conducted. The analysis was performed based on: (i) the technology/concept used to develop the framework, (ii) the domain application and case study covered by the research work, and (iii) whether the solution is based on open standards for representing BIM data (IFC, XML, ifcOWL…) and/or the rules set (LegalRuleML, RuleML…). We propose a classification of six existing methods:

### AI techniques

[ (Zhang & El-Gohary, 2016), (Hakim, et al., 2017), (Salama & El-Gohary, 2011), (Zhang & El-Gohary, 2013), (Zhang & El-Gohary, 2018)]: compliance checking systems based on implementing AI techniques, such as machine learning, generative design, pattern recognition, etc., for either rule detection and representation from regulation texts, or rule execution. They focus on the regulations and how it can be transformed into rule-based machine readable-format, but in some way, neglect the whole compliance checking process. They also propose AI-based algorithms with high level precision but still not sufficient to be suitable for industry application, where no degree of error is acceptable.

**Domain specific language**

[ (Ghannad, et al., 2019), (Macit & Suter, 2015), (Sydora & Stroulia, 2020), (Lee, et al., 2015), (Beach, et al., 2015), (Jiang, et al., 2018), (Soman, et al., 2020)]: based on domain specific languages, such as BERA language and SWRL, that are developed to allow representing rules in a machine-readable format, either for a specific domain application (e.g., interior design) or for a specific regulation type. While this category identifies practical compliance checking systems in their specific application domain, they are still non-extensible (or partially) to other application domains unless the language itself is extended which is not coherent with the definition of domain-specific languages. Therefore, they do not represent a holistic approach for all building codes and regulation domains so as to ensure a complete compliance checking process of a BIM model.

**Query language**

[ (Dimyadi, et al., 2016), (Zhong, et al., 2018), (Bus, et al., 2018), (Zhang, 2019)]: checker systems based on representing building knowledge, generally in the form of ontologies, then executing the rules using queries written in a query language (e.g., SPARQL). Generally, these have complex interfaces and necessitate a certain background in writing and using queries that limits their accessibility and usability. Furthermore, the cases considered are all focused on 'content validation checking' rather than model completeness, smart object (behavioural rules), or design option (best practices rules) checking.

**Reasoner**

[ (Tan, et al., 2010), (Zhang, et al., 2011), (Nawari, 2019c), (Messaoudi & Nawari, 2020), (Salama & El-Gohary, 2011), (Nawari, 2019b), (Zhang, 2019), (Zhang & El-Gohary, 2015), (Hjelseth, 2012), (Fenves, 1966)]: consisting of using logic, such as Deontic Logic, Fuzzy Logic, etc., to represent and/or execute the rules. A main difficulty with the compliance checking systems within this category is their complexity, and thus their suitability for industry application.

**Bespoke tool**

[ (Ghannad, et al., 2019), (Ding, et al., 2006), (Benghi, 2019), (Cheng & Das, 2014), (Ciribini, et al., 2015), (Zhang, et al., 2013), (Melzner, et al., 2013), (Han, et al., 1997), (Martins & Monteiro, 2013)]: checkers that are developed by encoding the rules as a desktop or a Web-based application using a programming language (Java, Marionette…), or as a plugin within an existing authoring platform (e.g., Revit, Tekla Structures). While these kinds of systems can be very useful in their implementation domain, they are still costly to maintain, difficult to change, and require high levels of computer programming skills.

**General architecture**

[ (Eastman, et al., 2009), (Nawari, 2019c), (Messaoudi & Nawari, 2020), (Nawari, 2019b), (Zhang, et al., 2013), (Melzner, et al., 2013)]: conceptual models and functional or modular abstractions to develop a theoretical background with a systemic view for automated compliance checking systems. As the name suggests, this category does not propose practical solutions, but only general definitions and recommendations for developing desirable compliance checking systems.

Most of these approaches have the disadvantage of being focused on the regulations (building codes, normative standards, project requirements, etc.) and how they can be represented into rule-based machine readable-format, rather than taking full advantage of BIM and its object-oriented nature for representing construction projects, to ensure full object quality assurance including 'content validation', 'model completeness', 'smart object', and 'design option' checking. Consequently, much of the valuable and detailed information does not benefit from a suitable and relevant compliance checking process. This is in part constrained by the top-down functional decomposition, that usually characterises process-based approaches and structuring of the design elements as functional primitives (Gorti, et al., 1998). In addition, few of these works have endured the test of real industry applications.

This paper has the following four objectives:

- Propose a new framework for automated compliance checking that leverages the object-based representation of building information models;
- Enable a natural decomposition, hierarchical structuring and logical processing of the automated compliance checking operation;
- Enable quality assurance of the complex information existing in the construction industry at the detailed level; and
- Enhance the understanding of the automated compliance checking process by analysing and implementing a real-world case study.

## PROPOSED APPROACH

The ultimate aim of a construction process should be to fulfil the clients' requirement, satisfy the end users' needs, and ensure a level of performance in accordance with applicable regulations. A set of requirements, such as client's requirements, building codes, normative standards, industry guidelines, and project requirements, should be checked and satisfied as soon as a design solution is available.

Building codes represent a set of minimum rules to protect the health, safety, environmental impacts, etc., and with which all buildings must comply. They cover the whole life cycle of a building from design through construction to operation and maintenance. They include general requirements structured into chapters dealing with: stability of structures, fire safety, site preparation and resistance, toxic substances, sound insulation, ventilation, sanitation, hot water safety & water efficiency, drainage, heat appliances and fuel system, falling, collision and impact protection, energy, access and use, electrical safety, security, communication networks and construction materials (Regulation, 2020).

BIM has been changing the way built assets are designed, constructed, and operated and maintained. At the centre of these changes is the building information model which is now represented as a set of objects where each object is defined with an identity, attributes, and relations. The method proposed takes advantage of the BIM object-oriented nature and focus on the notion of object to define an iterative bottom-up approach for enabling automatic compliance checking processes (see Figure 1).

The object-oriented compliance checking approach is based on a four-step process (Eastman, et al., 2009):

(i) *Requirement rules interpretation* consists of transformation of building requirements, principally represented in natural language in the form of texts, charts, tables, and mathematical expressions (Nawari, 2019c), into machine-readable rules to allow their automatic execution. Unfortunately, none of the existing frameworks have been able to propose an efficient and automatic approach to allow this transformation automatically. Existing Natural Language Processing algorithms such as semantic-based, syntactic-based, and recent AI techniques-based, cannot correctly handle ambiguous, uncertain and domain specific knowledge characterising building requirement texts. Consequently, in this study, we adopt an intuitive and logical approach consisting of converting these requirements through collaborative work meetings with domain experts, into 'if-then' logical rules. These are then encoded into a computer-readable format; in our case into a computer code using C# as a programming language (see next Section).

*(ii) BIM objects data preparation* consists of simplifying the BIM model data so that it can be checked efficiently, without any loss of relevant information. Our compliance checking approach is object-oriented which makes IFC the file format of choice as it is neutral and an open schema enabling use and interchange across a wide range of BIM authoring platforms. However, due to its highly complex data structuring, its suitability is questionable and it has performed poorly when faced with the huge number of rules inherent in an iterative compliance checking process (Ghang, et al., 2014). To overcome this problem, the IFC-based building information models can be parsed and all its objects prepared and pre-treated before being processed. For example, it may not be sufficient to collect only easily-noticeable, physical objects identified within the *ifcProduct* class (defined as "*any object that relates to a geometric or spatial context*" (Standard, 2020)) but also to construct implicit and virtual objects, such as *ifcBuilding*, *ifcBuildingStorey*, and *ifcSpace*, with all their required/related information extracted from the building information model. These objects are defined within the *IfcSpatialStructureElement* class and are usually used to structure and organize a building project. This should improve the efficiency of processing IFC-based building information models (Sydora & Stroulia, 2020) and ensure the BIM data completeness before executing the rules set.

*(iii) Requirement rules execution* consists of executing each rule from the rules set converted in the first step, on each relevant object prepared in the second step. The result can be either 'Pass' or 'Fail' for each pair <rule, object>. An overall evaluation of the BIM model compliance is based on a concatenation of all these elementary evaluations. The rules set is structured in multiple rule subsets according to their application domain (energy, accessibility, safety, etc.) and to the types of objects to which they are relevant. As shown in Figure 1, the execution is processed bottom-up according to this proposed rationale for rules and objects structuring. Indeed, basic and explicit objects (e.g., floor, wall, etc.) are checked before more complex and implicit ones (e.g., roof structures, stair… defined by the *IfcRelAggregates* relationship), and

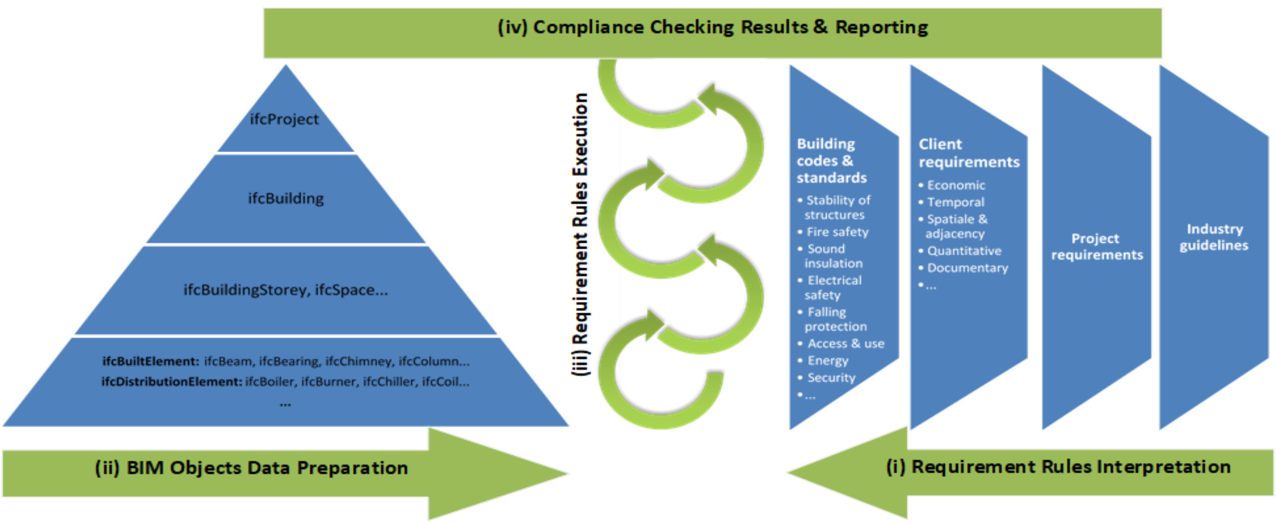


*Figure 1: BIM objects-based automated compliance checking method*

mandatory requirements are also scanned first before guidelines and non-mandatory requirements.

*(iv) Requirement rules checking results and reporting* informs the user about the outcomes of the compliance checking execution step, i.e., Pass or Fail result, and produces a detailed report containing each elementary check result with explanations of the reasons for non-compliance where applicable.

## CASE STUDY: SYNEG COMPLIANCE CHECKING

A foodservice equipment objects library, created by BONNET THIRODE, is demonstrated through the four-step approach: (i) Requirement rules interpretation, (ii) BIM objects data preparation, (iii) Requirement rules execution, and finally (iv) Compliance checking results and reporting.

### Requirement rules interpretation

The definition of Catering Equipment Manufacturer requirements is based on the FCSI (Foodservice Consultants Society International) Foodservice Equipment Standards document (Foodservice, 2020). This document aims to define a set of guidelines and recommendations for model content creation within an object-oriented authoring platform, such as Autodesk Revit, for use in the food service equipment industry (e.g., refrigerators, ranges, ice makers, etc.). It is composed of three files: (i) the Foodservice modelling standards, (ii) the International Foodservice Equipment (IFSE) parameter list, and the FCSI materials library.

The Foodservice modelling standards is the main document containing a set of guidelines and standards enabling the portability, compatibility and performance of the BIM objects data created in the Foodservice Equipment Industry (Foodservice, 2020). It covers various themes of BIM modelling guidelines such as: the software authoring tool version, recommended object template (hosted versus free-standing), object representation and level of detail, object file size, visibility settings, nested object-groups-voids recommendations, imported geometry and linked files, manufacturer logos, object and object type naming, categories and subcategories, parameters, type catalogues, materials, and recommendations about connectors.

To clarify and clear some of the uncertainties included in the recommendations of these standards, several meetings were organised with SYNEG, the national association of French catering equipment manufacturers, which is one of the contributors to the definition of FCSI Foodservice equipment standards. For example, regarding ‘Connectors’ the team needed to understand which controls to create and what input information to use to enable them. Furthermore, considering the complexity of foodservice equipment and the necessary skills level required to model them using BIM authoring platforms, some of the recommendations have been adapted (e.g., max object file size is set to 3 MB instead of 0,75 MB). Also, provision is needed to be made for rules classified as ‘mandatory’ and ‘non-mandatory’ so that rules in both categories are checked but that failure on a non-mandatory rule does not result in overall failure.

All these rules are interpreted into computer-readable code using the C# programming language within the SYNEG plugin.

### BIM objects data preparation

BONNET THIRODE has developed a complete BIM object library of its products. In terms of BIM adoption, BONNET is known to be one of the earliest and most advanced adopters amongst French foodservice equipment manufacturers.

In this case study, the BIM data preparation step entailed inputting both the BIM library with its 49 objects to be checked, plus the manufacturer’s library description file in a single repository. All the objects are RFA (Revit Family) files representing foodservice equipment objects. Moreover, as they are independent from each other, and defined at the same level of complexity, their processing was executed without a predefined order.

### Requirement rules execution

To allow automated compliance checking for SYNEG BIM objects library, a SYNEG plugin was developed using C# programming language, within the Autodesk Revit 2020 environment. Figure 2 shows the architecture of this plugin. It is composed of (i) a GUI (Graphical User Interface) to simplify its utilisation by non-expert users (Figure 3); (ii) the SYNEG compliance checking Rules module which consists of implementing a computer-readable format of the adapted version of the Foodservice modelling standards; (iii) Autodesk Revit API; (iv) the FCSI Materials library; and (v) other resources (templates, logos, etc.).

The plugin inputs are the BIM objects library to be checked alongside the manufacturer’s library description. The outputs from the plugin are the automated compliance checking reports: one overall for all the BIM objects, and individual reports for each object family.

It should be noted that although this kind of rules implementation is generally costly to maintain and difficult to update or change (Zhang & El-Gohary, 2016), the classification of the rules and their encoding in separate modules could greatly help reduce this complexity. That is the case here, where the Foodservice modelling standards rules have been encoded in a single module so that if in future, there is any need to further enrich these rules, more specific rules modules could be added without changing the general architecture of the application.

### Requirement rules results and reporting

By executing the SYNEG plugin on the BONNET BIM objects library, a detailed automated compliance checking report with respect to the Foodservice modelling standards, is automatically generated. It consists of an Excel

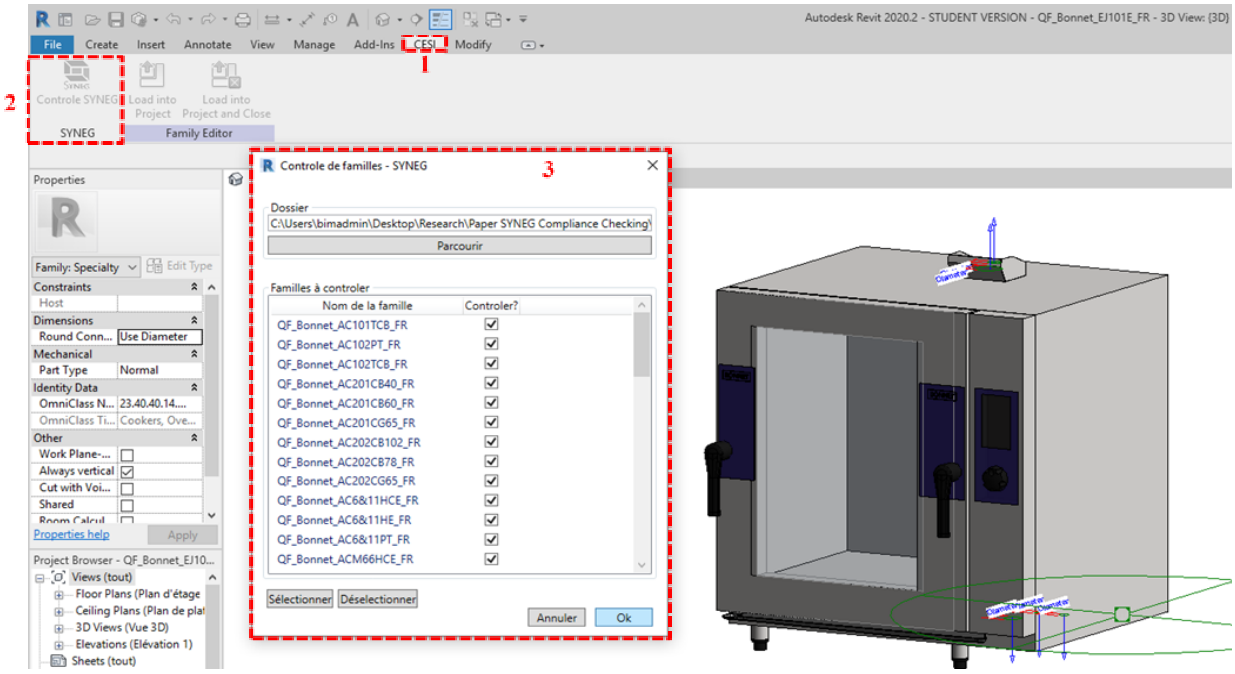


*Figure 2: SYNEG plugin Graphical User Interface*

file with multiple sheets (as many as the object families within the BIM library). The first sheet indicates the overall compliance checking result depending on the average of all the objects' compliance results.

Furthermore, the result of the compliance checking of each BIM object is given in a separate sheet. It presents the result corresponding to each object family, where the satisfied rules are highlighted in green and unsatisfied rules are left as are. To justify the compliance checking result and enable future correction, an automated justification for each 'Failed'/'Passed' rule result is given in this report.

## DISCUSSION

The bottom-up approach for automated compliance checking proposed in this paper takes advantage of the object-based nature of building information models and ensures full compliance checking processing including content validation, model completeness, smart object, and design option checking. Through 4 steps: requirement rules interpretation, BIM objects data preparation, requirement rules execution, and compliance checking results & reporting, it allows verification of a design solution with respect to a set of construction requirements.

To enable requirement rules interpretation, this approach uses an intuitive and logic framework consisting of converting the requirements into if-then logical rules which will be encoded thereafter, using a computer programming language such as C#. Although this could be justified by the fact no efficient approach exists to allow this interpretation automatically, this approach is useful in its specific domain implementation. However, according to Zhang et al. (Zhang & El-Gohary, 2016), it is still costly to maintain, difficult to change and needs high level of computer programming skills. To address this challenge and satisfy evolving requirements as well as answering the question of regulations sustainability as pointed out in (Hakim, et al., 2017), a modular architecture has been

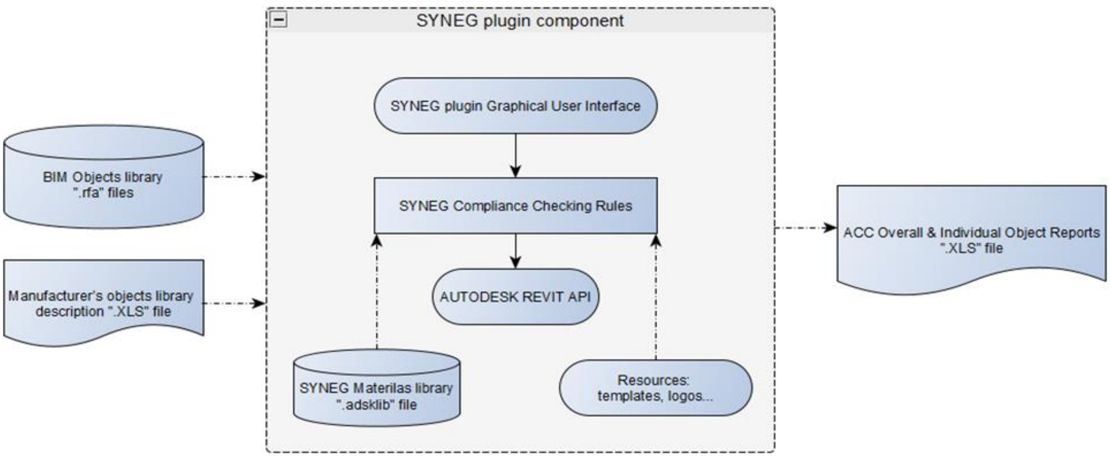


*Figure 3: SYNEG plugin Architecture*

adopted and implemented. This architecture enables each requirements class to be integrated into an independent module. As a result, extensions become possible without changing the whole application architecture; and in case of requirements change, updates can be performed with the minimal change to the existing information. Furthermore, to address the problem of dealing with uncertain and ambiguous building requirements characterising these kind of approaches (Zhang & El-Gohary, 2016), various meetings with domain experts have been organised to allow the best possible interpretations for the Foodservice Equipment Standards.

For the BIM objects data preparation step, the choice of using a BIM file format other than IFC (RFA format

in our case) was justified by two main reasons. Firstly, it was the decision of the users (SYNEG and all the French foodservice manufacturers) who have already chosen to work on the same authoring platform, Autodesk Revit, using the same set of modelling standards. Secondly, some of the foodservice manufacturers have already created their own BIM object libraries in Revit, therefore, converting their BIM data into IFC would require a lot of unnecessary work caused by information losses while converting from a proprietary BIM format to IFC (Turk, 2020).

The two last steps, namely, requirement rules execution and results & reporting, have been formalised by considering the users' needs. Indeed, foodservice manufacturers use the Autodesk Revit platform when creating their BIM object libraries, so the most efficient, suitable, and obvious approach to check their BIM objects is to conduct it using the same authoring environment via a new integrated functionality – a plugin in our case. Reporting the compliance checking results and identifying any change or correction to be made via an Excel file is also a practical way to check and correct the BIM library object by object. The BONNET library did not pass the test, the overall result was '0%', principally due to the authoring tool version used to create this library which was Revit 2015.

In terms of performance, compared with the manual approach, Table 1 presents the results of execution times corresponding to the automated compliance checking approach using the SYNEG plugin, and the traditional approach by checking the rules manually. A laptop with Intel(R) Core(TM) i7-4610 CPU @ 3.00 GHz 3.00 GHz Processor, and 16.0 GB of RAM, has been used to run the plugin. The results show a drastic time saving in processing the library of 49 BIM objects. The automated compliance checking processing with the SYNEG plugin required 197.09 seconds (~ 3.3 minutes). This time is principally due to 'loading' operations of RFA files (112.26 seconds) into the Autodesk Revit 2020 environment, and 'upgrading' operations of BIM objects (81.63 seconds) since they were created in a previous version of Revit (Revit 2015). Concerning the manual approach, execution times have been calculated by considering estimations of an expert Revit user who is capable of realising the whole compliance checking process manually. For only 20 elementary rules, the time to check and report one BIM object was estimated to be 74 901 seconds (~ 125 minutes), whereas the 49 objects library would take 367 010 seconds which is equivalent to at least 2 working weeks.

## CONCLUSION & PERSPECTIVES

The objectives of this research have been completely achieved.

Firstly, after an extensive literature review of 34 papers dealing with automated compliance checking, a classification into 6 categories of existing approaches (AI techniques, domain specific language, query language, reasoner, bespoke tool, and general architecture) has been proposed and a new framework for automated compliance checking based on BIM models has been defined and structured over a four-step process.

Secondly, this framework proposes a natural and hierarchical decomposition of the rules by structuring them into multiple subsets according to their application domain (energy, accessibility, safety...) and to the types of objects to which they are relevant.

The BIM objects are also organised according to their types (implicit, explicit) and complexity (basic, complex/aggregated). The execution of the rules is then processed from bottom to top, starting with basic and explicit objects to more complex and implicit ones.

Thirdly, this structuration of BIM objects and rules allows verification of both simple and complex information. For example, elementary information such as

Table 1: Comparison of ACC execution times (seconds) using SYNEG plugin and manual approach

| | Manual approach | | | Automated compliance checking (SYNEG plugin) | | | | |
|---|---|---|---|---|---|---|---|---|
| | Rules execution | Reporting | **Total** | Object loading | Objects upgrading | Rules execution | Reporting | **Total** |
| 1 BIM Object | 5 690 | 1 800 | **7 490** | 2.29 | 1.67 | 0.05 | 0.02 | **4.02** |
| BONNET's Library: 49 BIM Objects | 27 8810 | 88 200 | **367 010** | 112.26 | 81.63 | 2.32 | 0.87 | **197.09** |

those related to manufacturer logo and its visibility settings, i.e. the existence of a parameter named 'Show Logo' to control the visibility of the logo, can be considered and checked with regard to existing standards and norms. In addition, behavioural (such as, using free-standing and not hosted object) and composition (or structuring, like limiting nested object levels, and using groups and/or voids) BIM-object information can also be checked thanks to this framework.

Finally, a real case study consisting of a foodservice equipment objects library of a French Catering equipment manufacturer, has been presented and processed according to the four-step process approach, and a new plugin application for automated compliance checking has been developed and tested. This plugin assists foodservice equipment designers and helps them to check and correct their BIM object libraries according to the Foodservice modelling standards and requirements.

In future work, we aim to further develop this application to address the general compliance checking framework presented in this paper. This would enable: (i) processing of other requirement rules domains, (ii) adopting a neutral and open standard BIM format, such as IFC, and (iii) implementing the pyramidal BIM objects data structuration allowing the full implementation of the bottom-up rules execution, with elementary and explicit objects processed first, followed by complex and implicit ones.

Ultimately, to allow SYNEG to approve created BIM libraries and play the role of a certification organisation, we aim to extend the SYNEG plugin to be a Web-based application, so that foodservice equipment manufacturers would be able to submit their BIM libraries to obtain a 'BIM Approved by SYNEG' certification.

## ACKNOWLEDGEMENTS

This paper is supported by European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 892071.

Part of this work was conducted at CESI Ecole d'Ingénieurs of Paris-Nanterre. The authors would like to gratefully acknowledge the useful discussions and collaborative work during the development of the SYNEG plugin, with CESI, SYNEG and BONNET THIRODE teams including Lionel POISSON, André-Pierre DOUCET, Laurent GODARD, and Géraldine LARRIVEE.

## REFERENCES

Beach, T. H., Hippolyte, J.-L. & Rezgui, Y., 2020. Towards the adoption of automated regulatory compliance checking in the built environment. *Automation in Construction,* 118(0926-5805), p. 103285.

Beach, T. H., Rezgui, Y., Li, H. & Kasim, T., 2015. A rule-based semantic approach for automated regulatory compliance in the construction sector. *Expert System Application,* 42(12), p. 5219–5231.

Benghi, C., 2019. Automated verification for collaborative workflows in a Digital Plan of Work. *Automation in Construction,* 107(0926-5805), p. 102926.

Bus, N., Muhammad, F. & Fies, B., 2018. *Semantic topological querying for compliance checking.* s.l., eWork and eBusiness in Architecture, Engineering and Construction, No. European Conference on Product and Process Modelling, CRC Press, pp. 459–464.

Cheng, J. C. P. & Das, M., 2014. A BIM-based web service framework for green building energy simulation and code checking. *Journal Information Technology in Construction ,* 19(19), p. 150–168.

Choi, J. & Kim, I., 2008. An Approach to Share Architectural Drawing Information and Document Information for Automated Code Checking System. *Tsinghua Science & Technology,* Volume 13 (1), pp. 171-178.

Ciribini, A. L. C., Bolpagni, M. & Oliveri, E., 2015. An innovative approach to e-public tendering based on model checking". Proc. Econ. Financ.. *Procedia Economics and Finance,* 21(2212-5671), p. 32–39.

Dimyadi, J., Pauwels, P. & Amor, R., 2016. Modelling and accessing regulatory knowledge for computer-assisted compliance audit. *Information Technology in Construction,* Volume 21 (21), p. 317–336.

Ding, L., Drogemuller, R., Rosenman, M. & Marchant, D., 2006. *Automating code checking for building designs – DesignCheck.* s.l., Clients Driving Innovation: Moving Ideas into Practice, Cooperative Research Centre (CRC) for Construction Innovation.

Eastman, C., Lee, J.-m., Jeong, Y.-s. & Lee, J.-k., 2009. Automatic rule-based checking of building designs. *Automation in Construction,* 18(8), pp. 1011-1033.

Fenves, S. J., 1966. Tabular decision logic for structural design. *Journal of Structural Division,* 92(6), p. 473–490.

Foodservice, I. C. S., 2020. Foodservice Consultants Society International. Available at: <http://www.fcsi.org/?page=REVITStandards>. [Accessed 19 November 2020].

Ghannad, P., Lee, Y.-C., Dimyadi, J. & Solihin, W., 2019. Automated BIM data validation integrating open-standard schema with visual programming language. *Advanced Engineering Informatics,* 40(1474-0346), pp. 14-28.

Gorti, S. et al., 1998. An object-oriented representation for product and design processes. *Computer-Aided Design,* 30(7), pp. 489-501.

Greenwood, D., Lockley, S., Malsane, S. & Matthews, J., 2010. *Automated compliance checking using building information models.* Dauphine Université, Paris, A. Ganah, & A. Rennie (Eds.), The Construction, Building and Real Estate Research Conference of the Royal Institution of Chartered Surveyors.

Hakim, S., Cecconi, F. R., Dejaco, M. C. & Maltese, S., 2017. *Rules complexity classification for automated compliance checking.* s.l., Digital Proceedings of the 24th European Group for Intelligent Computing in Engineering International Workshop on Intelligent Computing in Engineering 2017, pp. 136–145.

Han, C. S., Kunz, J. C. & Law, K. H., 1997. Making automated building code checking a reality. *Management Journal,* p. 22–28.

Hjelseth, E., 2012. *Experiences on converting interpretative regulations into computable rules.* Beirut, Lebanon, Proceedings of the 29th International Conference of CIB W78, ITC Digital Library, pp. 4–13.

Hjelseth, E., 2016. Classification of BIM-based model checking concepts. *Information Technology in Construction,* 21 (Special issue CIB W78 2015 Special track on Compliance Checking), p. 354–369.

International, F. C. S., 2020. http://www.fcsi.org/?page=REVITStandards.

Jiang, S., Wang, N. & Wu, J., 2018. Combining BIM and ontology to facilitate intelligent green building evaluation. *Journal of Computing in Civil Engineering,* 32(5).

Krijnen, T., 2016. *Methodologies for requirement checking on building models.* s.l., 13th International Conference on Design and Decision Support Systems in Architecture and Urban Planning, June 27–28.

Krijnen, T. & Tamke, M., 2015. Assessing Implicit Knowledge in BIM Models with Machine Learning. *Modelling Behaviour, Springer International Publishing,* pp. 397- 406.

Lee, Y. C., Eastman, C. M. & Lee, J. K., 2015. Automated rule-based checking for the validation of accessibility and visibility of a building information model. *Computing in Civil Engineering 2015, American Society of Civil Engineers, Reston, VA,* p. 572–579.

Macit, S. & Suter, G., 2015. *A Hybrid Model for Building Code Representation Based on Four- Level and Semantic Modeling Approaches.* s.l., CIB W78 Conference 2015, pp. 517–526 .

Marchant, D., 2015. *Capturing and Integrating the Design Brief in Building Information Models.* Eindhoven, The Netherlands, 27-29 October, 538-548, Proceeding of the 32nd CIB W78 Conference.

Marchant, D., 2016. The design brief: requirements and compliance. *Journal of Information Technology in Construction,* 21(Special track on Compliance Checking), pp. 337-353.

Martins, J. P. & Monteiro, A., 2013. LicA: A BIM based automated code-checking application for water distribution systems. *Automation in Construction,* 29(0926-5805), pp. 12-23.

McGraw-Hill, 2014. *The Business Value Of BIM,* s.l.: Global Markets 2014, Smart Market Report, Design and construction Intelligence, McGraw-Hill Construction.

Melzner, J., Zhang, S., Teizer, J. & Bargstädt, H. J. J., 2013. A case study on automated safety compliance checking to assist fall protection design and planning in building information models. *Construction Management and Economics,* 31(6), p. 661–674.

Messaoudi, M. & Nawari, N. O., 2020. BIM-based Virtual Permitting Framework (VPF) for post-disaster recovery and rebuilding in the state of Florida. *International Journal of Disaster Risk Reduction,* 42(2212-4209), p. 101349.

Miller, G. A., 1995. *WordNet: a lexical database for English.* s.l., Communications of the ACM 1995, 38.11, p. 39-41.

Navigli, R. & Ponzetto, S. P., 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence,* Volume 193, pp. 217-250.

Nawari, N. O., 2019a. A generalized adaptive framework (GAF) for automating code compliance checking, Buildings. *Buildings,* Volume 9 (4):86.

Nawari, N. O., 2019b. A Generalized Adaptive Framework for Automating Design Review Process: Technical Principles. In: I. Mutis, ed. *Advances in Informatics and Computing in Civil and Construction Engineering.* Cham: Springer International Publishing, p. 405–414.

Nawari, N. O., 2019c. Generalized adaptive framework for computerizing the building design review process. *Journal of Architecture Engineering,* 26(1).

Palos, S., Kiviniemi, A. & Kuusisto, J., 2014. Future perspectives on product data management in building information modelling. *Construction Innovation,* Volume 14(1), p. 52–68.

Pauwels, P. & Terkaj, W., 2016. EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Automation in Construction,* 63(0926-5805), pp. 100-133.

Regulation, B., 2020. Building Regulation. Available at: <https://www.designingbuildings.co.uk/wiki/Building_regulations> [Accessed 1 December 2020].

Sacks, R., Eastman, C., Lee, G. & Teicholz, P., 2018. *BIM Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers.* 978-1-119-28753-7 ed. Hoboken, NJ: John Wiley and Sons.

Salama, D. M. & El-Gohary, N., 2016. Semantic text classification for supporting automated compliance checking in construction. *Journal of Computing in Civil Engineering,* 30(1).

Salama, D. M. & El-Gohary, N. M., 2011. *Semantic modeling for automated compliance checking.* Miami, Florida, United States, International Workshop on Computing in Civil Engineering, American Society of Civil Engineers, p. 641–648.

Solihin, W. & Eastman, C., 2015. Classification of rules for automated BIM rule checking development. *Automation in Construction,* 53(0926-5805), pp. 69-82.

Solihin, W., Eastman, C. & Lee, Y. C., 2015. Toward robust and quantifiable automated IFC quality validation. *Advanced Engineering Informatics,* Volume 29(3), pp. 739-756.

Soman, R. K., Molina-Solana, M. & Whyte, J. K., 2020. Linked-Data based Constraint-Checking (LDCC) to support look-ahead planning in construction. *Automation in Construction,* 120(0926-5805), p. 103369.

Standard, I., 2020. IFC Standard. Available at: <https://standards.buildingsmart.org/IFC/DEV/IFC4_3/RC1/HTML/> [Accessed 19 November 2020].

Sydora, C. & Stroulia, E., 2020. Rule-based compliance checking and generative design for building interiors using BIM. *Automation in Construction,* 120(0926-5805), p. 103368.

Tan, X., Hammad, A. & Fazio, P., 2010. Automated code compliance checking for building envelope design. *Journal of Computing in Civil Engineering,* Volume 24 (2) , p. 203–211.

Terkaj, W. & Šojić, A., 2015. Ontology-based representation of IFC EXPRESS rules: An enhancement of the ifcOWL ontology. *Automation in Construction,* 57(0926-5805), pp. 188-201.

Turk, Ž., 2020. Interoperability in construction – Mission impossible?. *Developments in the Built Environment,* 4(2666-1659).

Zhang, C., 2019. *Requirement Checking in the Building Industry: Enabling Modularized and Extensible Requirement Checking Systems Based on Semantic Web Technologies,* Eindhoven: Doctor of Philosophy, Department of the Built Environment.

Zhang, C., Beetz, J. & Weise, M., 2014. *Model view checking: automated validation for IFC building models.* s.l., eWork and eBusiness in Architecture, Engineering and Construction: ECPPM 2014.

Zhang, J. & El-Gohary, N. M., 2013. Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking. *Journal of Computing in Civil Engineering,* 30(2).

Zhang, J. & El-Gohary, N. M., 2015. Automated information transformation for automated regulatory compliance checking in construction. *Journal of Computing in Civil Engineering,* 29(4).

Zhang, J. & El-Gohary, N. M., 2016. Extending building information models semiautomatically using semantic natural language processing techniques. *Journal of Computing in Civil Engineering,* Volume 30(5).

Zhang, R. & El-Gohary, N. M., 2018. *A clustering approach for analyzing the computability of building code requirements.* s.l., Construction Research Congress, pp. 86–95.

Zhang, S. et al., 2011. *Integrating BIM and Safety: An Automated Rule-Based Checking System for Safety Planning and Simulation.* s.l., Proceedings of CIB W99 Conference, pp. 1–13.

Zhang, S. et al., 2013. Building Information Modeling (BIM) and Safety: Automatic Safety Checking of Construction Models and Schedules. *Automation in Construction,* 29(0926-5805), pp. 183-195.

Zhong, B., Gan, C., Luo, H. & Xing, X., 2018. Ontology-based framework for building environmental monitoring and compliance checking under BIM environment. *Building Environment,* Volume 141, p. 127–142.