ITcon
www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

OBJECT-CENTRED AUTOMATED COMPLIANCE CHECKING: A NOVEL, BOTTOM-UP APPROACH

SUBMITTED: July 2021
REVISED: October 2021
PUBLISHED: April 2022
GUEST EDITOR: Dana Vanier
DOI: 10.36680/j.itcon.2022.017

Omar Doukari, Dr,
Department of Mechanical & Construction Engineering, Northumbria University, Newcastle, UK;
omar.doukari@northumbria.ac.uk

David Greenwood, Prof.
Department of Mechanical & Construction Engineering, Northumbria University, Newcastle, UK;
david.greenwood@northumbria.ac.uk

Kay Rogage, Dr,
Department of Computer and Information Sciences, Northumbria University, Newcastle, UK;
kay.rogage@northumbria.ac.uk

Mohamad Kassem, Prof.
Department of Mechanical & Construction Engineering, Northumbria University, Newcastle, UK;
mohamad.kassem@northumbria.ac.uk

SUMMARY*: Building Information Modelling (BIM) is changing how built assets are delivered and operated. A built asset is represented as a set of objects, each with an identity, attributes, and relations. This object-oriented nature enables new approaches for ensuring compliance with a range of requirements: e.g. industry guidelines; project and client-specific requirements; and building codes and standards. Furthermore, bottom-up design approaches are known to be more suitable for quality control and design errors detection. Based on an adapted version of simulated annealing concept, this paper proposes an automated compliance checking classification and identifies a set of desired characteristics these methods should fulfil. It then demonstrates a bottom-up object-centred approach for automated model checking and the corresponding plugin prototype. The approach and the prototype enable four key processes and satisfy all desired characteristics of compliance checking methods including content validation, model completeness, smart object, and design option checking. To demonstrate the feasibility and accuracy of the approach, two case studies are processed using existing BIM objects libraries one of which is 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.*

KEYWORDS*: Compliance Checking, Model Checking, Simulated Annealing Optimisation, Artificial Intelligence, Automation, BIM, Case study*

REFERENCE: *Omar Doukari, David Greenwood, Kay Rogage, Mohamad Kassem (2022). Object-centred automated compliance checking: a novel, bottom-up approach. Journal of Information Technology in Construction (ITcon), Special issue: 'The Eastman Symposium', Vol. 27, pg. 335-362, DOI: 10.36680/j.itcon.2022.017*

1. INTRODUCTION

In the design, construction, and operation of built assets, it is important to fulfil clients' and users' needs and assure the desired level of quality by complying with several requirements: building codes, normative standards, industry guidelines, and project requirements. Traditional practice involves checking these requirements in a manual, laborious, time consuming, and error-prone process (Ghannad et al., 2019) that can result in financial losses (Ding et al., 2006) or worse, in project failure and loss of life (Zhang, 2014).

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 by using a BIM authoring platform. BIM is expected to contribute significantly 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, automated compliance checking (ACC) of building designs is becoming feasible (Choi & Kim, 2008) thus, contributing significantly to time and cost savings.

Numerous research works have considered automating compliance checking in the construction industry. Various model checking methodologies, platforms, and domain applications such as spatial assessment, structural integrity, safety, energy usage, etc., have been examined in the academic research literature and multiple industrial tools are now commercially available. The overall functional architecture can be structured through a sequential process based on four main steps: (i) rule interpretation, (ii) building model preparation, (iii) rule execution, and (iv) rule check reporting (Eastman et al., 2009). A set of desired characteristics and performances to be fulfilled by these systems have been identified as well, namely: (i) model checking-related characteristics including content validation, model completeness, smart object (behavioural rules checking), and design option checking to support and guide the design process with respect to best practices, and (ii) rules-related characteristics including rules encoding approach (manual, semi-automatic or automatic), independency of the BIM model and the model checking tool, extensibility of the rules set, and uncertainty handling.

Most of the existing approaches have been focusing 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 the best of our knowledge none of them ensures full object quality assurance with content validation, model completeness, smart object, and design option checking being simultaneously processed. Consequently, much of the valuable and detailed information concerned 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). Instead, bottom-up design approaches are known to be more suitable for quality control and design errors detection (United States Congress House Committe, 1986). In addition, few of these initiatives have endured the test of real industry applications.

Moreover, several standardisation initiatives (e.g., XP P07-150 (AFNOR, 2014), ISO 16757 (ISO, 2015), EN ISO 23386 (ISO, 2020a), EN ISO 23387 (ISO, 2020b), LEXiCON (Construction Innovation HUB & Construction Products Association, 2021)) for construction objects have been recently conducted to allow harmonised quality and performance information exchange between construction stakeholders for serving different purposes: digitalisation, international trade and operation and maintenance management. The aim is to define every construction object characteristic through a set of consistent attributes to allow capturing the data that describes the performance and quality of construction objects, systems and their components, and to ensure they are well-defined and structured so can be automatically read and processed. These standards allow to define the general structure, called Product Data Template, that can be used to provide digital description for any construction object. Based on these initiatives, a unified framework is being developed through a new standard (CEN, 2021), that will precise how data templates should be created, which will allow, inter alia, manufacturers to digitise their products and provide their data in a homogenised and agreed way. Undoubtedly, these objects-based standards will streamline the development of BIM data quality assurance and control which are two interlaced aspects of quality management processes (Ramesh, 2016), and allow to ensure high BIM data quality by adopting automated tools based on BIM object quality checklists (Zadeh et al., 2017).

The integration of best research practices and methodologies across different projects and sectors has fostered the development of several literature review approaches (Grant & Booth, 2009). Systematic literature review for instance (Swartz, 2011), seeks to systematically and extensively explore multiple research databases to answer clear formulated statements and problems by following a transparent, predefined and well-parametrised process. Although these approaches are very useful for both research questions identification and future research justification, they still need high skills for information searching and filtering (Torres-Carrión et al., 2018). In addition, they are time-consuming and complex to conduct, especially where important number of research works have to be considered, due to its global and complete search process. On the other hand, AI domain proposes a variety of methods in order to improve the search process such as metaheuristic optimization methods that have been developed based on some principles inspired by theory of evolution (Katoch et al., 2021), animal's behaviour (Eberhart & Kennedy, 1995), medicine and cell biology (Tschida & Silverberg, 2013), and natural phenomena ((Saka & Dogan, 2012), (Kaveh, 2017)). Simulated Annealing (Kirkpatrick et al., 1983) is one of the first metaheuristic methods that has been successfully applied in many case studies and domains ((Eglese, 1990), (Kalivas, 1992), (Román-Román & Torres-Ruiz, 2015), (Chebbi et al., 2017), (Bandyopadhyay et al., 2021)), and still utilised and developed up until now ((Ficarella et al., 2021), (Duan et al., 2021)). This method is inspired by a process of natural optimization phenomenon which is the physical annealing of materials. This latter consists in heating materials before crystallization with a suitable temperature and for an appropriate amount of time, then cooling them to create stable solids. This process was formalised into optimisation algorithm to enable local searches in the solution space and converge to global optimal solutions while avoiding local optima problems (Yang, 2014).

This paper has two main objectives. The first is to propose a new framework for bottom-up ACC that leverages the object-based representation of building information models. The framework should enable (a) a natural decomposition, hierarchical structuring and logical processing of the ACC operation, and (b) quality assurance of the information existing in the construction industry at the detailed level, where content validation, model completeness, smart object, and design option checking are simultaneously performed. The research methodology adopted and the ACC classification and analysis conducted are all based on an adapted version of the simulated annealing concept. The second objective is to enhance the understanding of the proposed ACC process and test it by implementing and analysing two case studies.

The remainder of this paper is structured as follows. A review of related studies is presented in Section 2. The research methodology of this paper is established in Section 3. This is followed, in sections 4 and 5 respectively, by the presentation of a simulated annealing-based classification of ACC approaches along with their desired characteristics, and a novel compliance checking method. In Section 6, this method is demonstrated in two case studies using existing BIM objects libraries: one created by a French foodservice equipment manufacturer and the other is the NBS National BIM library. Finally, Section 7 discusses the results obtained before concluding in Section 8.

2. RELATED STUDIES

During the last four decades, more than 400 research works have considered automating compliance checking in the construction industry (Nawari, 2019a). In one of the most referenced studies, (Eastman et al., 2009) presented a general functional architecture for rule checking and reporting systems, structuring a rule checking process into four main steps. The authors discussed the shortcomings of each step, identified general requirements, and reviewed five existing rule checking systems according to this structure. These were: the CORENET project (Singapore), the HITOS project (a Norwegian BIM project for Statsbygg), the Australian Building Codes Board project, the International Code Council project, and the General Services Administration project (both US). As input, these systems rely on Industry Foundation Classes (IFC)-based building information models. Based on the same rule checking systems, (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 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 such as 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, 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, 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 (MVD) ((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).

ACC 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 (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, 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* class and related sub-classes. Furthermore, for some frameworks, the rules set can be either integrated, hard-encoded into the BIM-based code checking tool ((Zhang et al., 2011), (Benghi, 2019)) 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, 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, 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*”. (Nawari, 2019c) developed a ‘generalized adaptive framework’ for building code compliance checking based on the IFC standard, comprising five steps: (i) analysis and 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.

To develop a roadmap for ACC adoption, (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, including 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 (e.g., fire safety), 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 ACC that led them to propose a roadmap that considered, concurrently, the political, commercial, and technological factors in the future development of ACC (Beach et al., 2020). Although industry partners were in favour of adopting ACC, they insisted on the necessity for designating a qualified human to supervise the whole process. Human intervention was also suggested for performance-based design where several simulations and computations are usually required (Dimyadi et al., 2017).

Recently, (Amor & Dimyadi, 2021) conducted a thorough literature review on existing approaches for ACC, to discuss the challenges faced in this topic and identify future pathways. They presented existing ACC approaches from a temporal perspective and proposed a classification into three different categories: pre-OpenBIM, OpenBIM and recent ACC approaches. Despites the developments achieved so far, there are still unresolved issues to be tackled to allow wider adoption of ACC systems. One of the main unresolved issues is related to the quality of the BIM model. Indeed, the data provided in the BIM model should be sufficient, correct, consistent and with high quality, to ensure efficient and accurate execution of ACC processes.

3. METHODOLOGY

To achieve the objectives of this work and propose an ACC approach that will enable the main ACC functionalities and satisfy all desired compliance checking method characteristics identified in previous research works, the following methodology was established (Fig. 1):

1. Conduct an extended literature review in order to understand the existing ACC frameworks and systems, and propose a classification based on a local search metaheuristic inspired by a process of natural optimization phenomenon which is the physical annealing of materials that consists in bringing a solid to a low energy, relatively stable state, after rising its temperature during an appropriate amount of time where more unstable states and random behaviours are likely to be faced (Algorithm 1). The ACC classification proposed is based on different criteria, such as the concepts used to develop the ACC framework, the domain and suitability for industry application, open standards compatibility and usage, etc.;
2. Identify from the literature, all desired ACC characteristics required to perform suitable compliance checking processes in the construction industry, and analyse in light of these characteristics, the existing ACC approaches;
3. Propose a bottom-up ACC approach that leverages the object-based representation of building information models and enables full integration and satisfaction of desired ACC characteristics;
4. Demonstrate the feasibility of the proposed approach by implementing a real case study using an existing BIM objects library and analyse the results to identify unresolved issues and define future improvements and perspectives to be iteratively integrated.

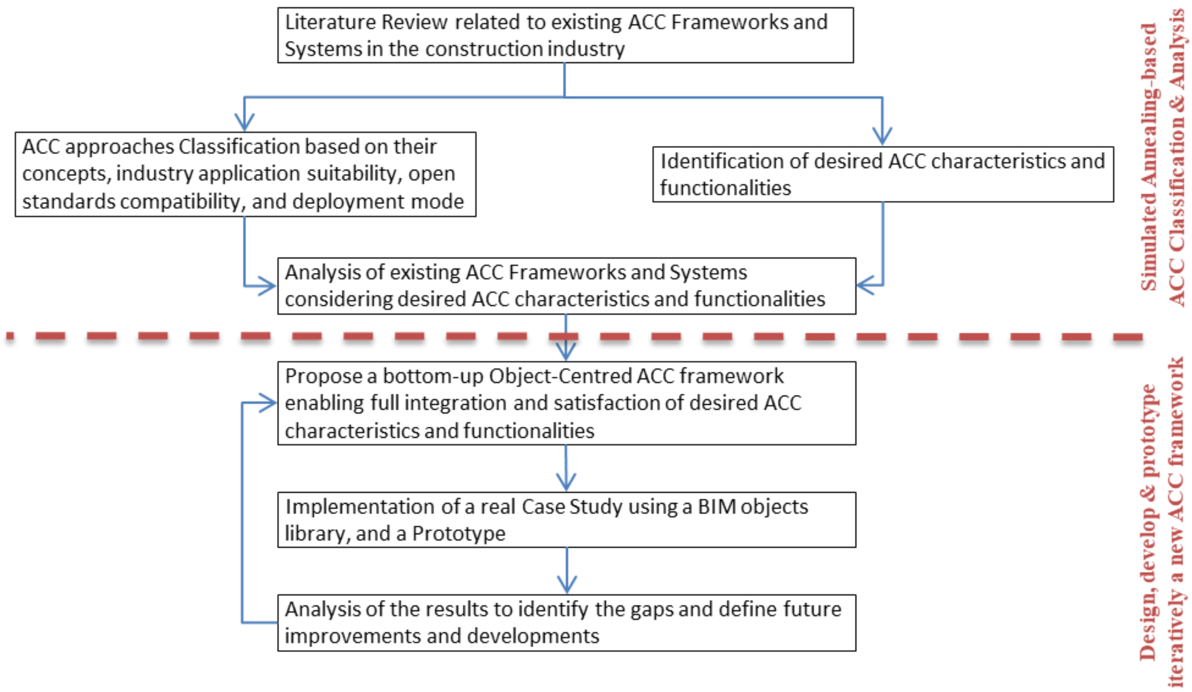

FIG. 1: Research methodology

ALGORITHM 1: Simulated Annealing-based classification method

```
Input:

Ω: ACC papers; /* 626 journal and conference papers from Scopus */
Φ: papers dealing with ACC approaches classification; /* 19 classification works */

Process:

For all x in Φ do:
   ACC-Classification' = Classification proposed in x;
   If ACC-Classification does not integrate ACC-Classification' then:
      ACC-Classification.expand(ACC-Classification');
      ACC-Classification.manageInconsistency();
   If ACC-DesiredCriteria(x) is not in ACC-DesiredCriteria then:
      ACC-DesiredCriteria.expand(ACC-DesiredCriteria(x));
      ACC-DesiredCriteria.manageInconsistency();
End For

Repeat:
   Initialise randomly x from (Ω – Φ);
   If ACC-Classification does not include ACC-Framework(x) then:
      ACC-Classification.update(ACC-Framework(x));
   If ACC-DesiredCriteria(x) is not in ACC-DesiredCriteria then:
      ACC-DesiredCriteria.update(ACC-DesiredCriteria(x));
Until ACC-Classification and ACC-DesiredCriteria are stabilised;

Output:
Return ACC-Classification and ACC-DesiredCriteria;
```

4. SIMULATED ANNEALING-BASED ACC APPROACHES CLASSIFICATION AND ANALYSIS

To understand the application domain and different frameworks for compliance checking methods, and propose a classification system for ACC, a review of more than thirty research papers was conducted. This first analysis, summarised in Table 1., was performed based on the following criteria: (i) the mode of system deployment of each solution; (ii) the various technologies/concepts used to develop each framework; (ii) the suitability for industrial application based on the presented prototype if any; and, (iv) whether the solution is based on open standards for representing BIM data (e.g. IFC, XML, ifcOWL…) and/or the rules set (e.g. LegalRuleML, RuleML…).

It is worth to mention that even though the literature review did not cover all existing ACC related research works, since only 37 papers (from a total of 626 ACC works) have been considered while building the classification and the set of desired ACC characteristics, but the way in which this has been conducted, should guarantee the optimality and genericity of the solution to be inclusive of all kinds of ACC frameworks and systems. Indeed, papers dealing with ACC classification have been considered first to develop a trade-off classification, i.e. a classification that enable generalising the existing classifications to allow representing them in one consistent classification without redundancy. Then, this latter has been adapted and adjusted by studying several other works proposing different ACC approaches. This process is repeated until the structures corresponding to the proposed classification and the set of desired ACC characteristics are stabilised (Algorithm 1). The research method implemented here could be seen as an adapted version of the simulated annealing concept that is largely used in combinatorial optimisation (Kirkpatrick et al., 1983). This concept is based on nature-inspired optimization phenomenon which is the physical annealing of materials. The corresponding algorithms allow developers to conduct local searches in the solution space and converge to global optimal solutions while avoiding local optima inconvenient characterising local search methods (Yang, 2014). This local research methodology and classification approach based on the simulated annealing metaheuristic will be detailed in another research work.

TABLE 1: Analysis of reviewed ACC approaches

Research works (with reference numbers)	(i) Deployment	(ii) Technology concept	(iii) Industry application	(iv) Open standards	Classification
(Zhang & El-Gohary, 2013) (Salama & El-Gohary, 2016) (Zhang & El-Gohary, 2016) (Hakim et al., 2017) (Zhang & El-Gohary, 2018)	/	Machine-learning, rule classification, semantic text classification, natural language processing, ontology, pattern-matching, clustering.	No	No, except (Zhang & El-Gohary, 2016), (Hakim et al., 2017)	AI Techniques
(Beach et al., 2015) (Lee et al., 2015) (Macit & Suter, 2015) (Jiang et al., 2018) (Ghannad et al., 2019) (Dimyadi et al., 2020) (Soman et al., 2020) (Sydora & Stroulia, 2020)	Add-in, web-based	The four-level paradigm, semantic modelling, SMARTcodes, domain-specific language, BERA Language, Semantic Web Rule Language, semantic modelling, LegalDocML, LegalRuleML, GBEOntology, Shapes Constraint Language, data modelling language	Yes, except (Ghannad et al., 2019)	Yes (OWL, LegalDocML, Legal-RuleML)	Domain-specific language
(Dimyadi et al., 2016) (Bus et al., 2018) (Zhong et al., 2018) (Zhang, 2019)	Web-based	LegalDocML, LegalRuleML, semantic web, MVD, SPARQL, RDF query language, IfcOWL ontology.	Yes, except (Dimyadi et al., 2016), (Zhong et al., 2018)	Yes (RuleML, IFC, RDF, XML)	Query Language
(Fenves, 1966) (Tan et al., 2010) (Salama & El-Gohary, 2011) (Zhang et al., 2011) (Hjelseth, 2012) (Zhang & El-Gohary, 2015) (Nawari, 2019b) (Nawari, 2019c) (Zhang, 2019) (Messaoudi & Nawari, 2020) (Li et al., 2020)	Add-in, desktop, web-based	Fuzzy logic, uncertainty, Expert system, spatial reasoning, decision tables, deontic logic, data fusion, Extended BIM, FOL logic, Answer Set Programming, knowledge representation.	No, except (Zhang et al., 2011), (Zhang, 2019)	Yes, except (Fenves, 1966)	Reasoner
(Han et al., 1997) (Ding et al., 2006) (Clayton, 2013) (Martins & Monteiro, 2013) (Melzner et al., 2013) (Zhang et al., 2013) (Cheng & Das, 2014) (Ciribini et al., 2015) (Benghi, 2019) (Ghannad et al., 2019)	Add-in, desktop, web-based	Xbim Toolkit, DesignCheck, LicA tool, SMARTreview APR, Web service, SMC, Revit, Web-based tool, Java, VPL, marionette API, Tekla Structures, Level of detail, model completeness.	Yes, except (Ghannad et al., 2019), (Han et al., 1997)	Yes, except (Han et al., 1997)	Bespoke tool
(Eastman et al., 2009) (Melzner et al., 2013) (Zhang et al., 2013) (Nawari, 2019b) (Nawari, 2019c) (Messaoudi & Nawari, 2020)	Add-in	System functional capabilities, GAF process, IDM, MVD, WBS, if-then based rules.	No, except (Zhang et al., 2013)	Yes	General Architecture

As shown in Table 1, and informed by the review of the extant literature, a classification system of six (06) existing methods is proposed. They are here explained in further detail:

- *AI techniques* are compliance checking systems based on implementing AI techniques, such as machine learning, pattern recognition, generative design, 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: more than 90% in the case of (Salama & El-Gohary, 2016) but still not sufficient to be suitable for industry application, where no degree of error is acceptable;
- Techniques based upon *Domain-specific Language* (such as BERA language and SWRL) are developed to allow representing rules in a machine-readable format, either for a specific domain application (e.g., interior design, by (Sydora & Stroulia, 2020)) or for a specific regulation type (Ghannad et al., 2019). While this category identifies practical compliance checking systems in their specific application domain, they are still non-extensible (or partially: only within their definition domain) 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* checker systems are 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* solutions use logic, such as Deontic Logic (Salama & El-Gohary, 2011), Fuzzy Logic (Nawari, 2019b), or Answer Set Programming (Li et al., 2020), to represent and/or execute the rules. A difficulty with the compliance checking systems within this category is their complexity, and thus their suitability for industry application. Indeed, logic reasoning operators while dealing with a large amount of information (in our case building codes) and infinite number of rules that can be defined, are known to be intractable (Doukari & Jeansoulin, 2012). In this category, the two cases that succeeded in reaching proof-of-concept stage ((Zhang et al., 2011), (Zhang, 2019)) were only tested on a limited number of rules;
- *Bespoke tools* are checkers that are developed by encoding the rules as a desktop or a Web-based application using a programming language, e.g. Java (Cheng & Das, 2014) or Marionette (Ghannad et al., 2019), or as a plugin within an existing authoring platform, e.g., Revit ((Clayton, 2013), (Ciribini et al., 2015)), or Tekla Structures (Zhang et al., 2013). 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 (Zhang & El-Gohary, 2016);
- In the *General architecture* category, conceptual models and functional (Eastman et al., 2009) or modular (Nawari, 2019c) abstractions that develop a theoretical background with a systemic view for ACC systems are listed. As the name suggests, this category does not propose practical solutions, but only general definitions and recommendations for developing desirable compliance checking systems.

Parallelly to the proposed ACC classification, a second literature review analysis was conducted in order to identify what are considered to be essential or desired ACC characteristics. Three (03) categories of desired characteristics were identified dealing with model checking, compliance checking processes and rules criteria (Table 2). The model checking desired characteristics category consists in checking basic quality assurance operations, and include content validation and completeness, smart object behaviour checking, and best practices design integration. The compliance checking essential processes consist in checking whether fundamental compliance checking processes are adopted, including rules interpretation, BIM data preparation, rules execution, and results and reporting. Finally, desired rules criteria concern the properties of the rules set and how it is implemented with regard to the BIM model and the compliance checking tool (or the rules execution module), its extensibility, and how the rules uncertainty is handled while they are interpreted and coded to be thereafter executed.

Furthermore, existing ACC frameworks and their corresponding classes identified in the first analysis, were subjected to scrutiny based upon these performance criteria. The results are summarised in Table 2.

TABLE 2: Analysis of reviewed ACC approaches

ACC desired characteristics and functionalities		AI techniques	Domain-specific language	Query language	Reasoner	Bespoke tool	General architecture
Model checking criteria	Content validation	Yes	Yes	Yes	Yes	Yes (except (Benghi, 2019))	Yes
	Content completeness	No	No (except (Soman et al., 2020))	No	No	No (except (Benghi, 2019))	No
	Smart object	No	No	No	No	No	No
	Design option	No	No	No	No (except (Zhang et al., 2011))	No (except (Zhang et al., 2013))	No (except (Zhang et al., 2013))
Compliance checking criteria	Preparation	No (except (Zhang & El-Gohary, 2016), (Hakim et al., 2017))	Yes (except (Ghannad et al., 2019))	Yes (except (Dimyadi et al., 2016))	Yes (except (Zhang et al., 2011), (Zhang & El-Gohary, 2015), (Hjelseth, 2012))	Yes (except (Ghannad et al., 2019), (Cheng & Das, 2014))	Yes
	Interpretation	Yes	Yes	Yes	Yes	Yes	Yes
	Execution	No	Yes	Yes	Yes	Yes	Yes
	Reporting	No	Yes	Yes (except (Dimyadi et al., 2016))	Yes (except (Salama & El-Gohary, 2011), (Zhang & El-Gohary, 2015), (Hjelseth, 2012), (Fenves, 1966))	Yes (except (Ciribini et al., 2015))	Yes
Rules criteria	Encoding approach	(Semi-) Automatic	Manual, semi-automatic	Manual, semi-automatic	Manual, (semi-) automatic	Manual, semi-automatic	Semi-automatic
	Independency of the BIM model	No	Yes	Yes	Yes (except (Tan et al., 2010))	Yes (except (Ding et al., 2006))	Yes
	Independency of the model checking tool	/	Yes (except (Ghannad et al., 2019))	Yes	Yes (except (Zhang et al., 2011), (Fenves, 1966))	No	No (except (Nawari, 2019c))
	Extensibility/sustainability	Yes	Partially	Yes / Partially	Yes/ Partially (except (Fenves, 1966))	No / Partially	Yes / Partially
	Uncertainty handling approach	/	None (except (Beach et al., 2015))	/	Fuzzy logic, approximate reasoning methods	/	Fuzzy logic, approximate reasoning methods

5. PROPOSED APPROACH

The ultimate aim of a construction process should be to fulfil client requirements, satisfy 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 buildings must comply. Depending on the governing authority (country, state, province, city, etc.), they typically 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 and 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 (Designing Buildings, 2021).

BIM has been changing the way that 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.

5.1 Object-oriented approach principles

Objects are the real-world entities that exist around us. An object is defined through its properties (or data that represents the state of the object such as dimensions, localisation, fire rating...) and behaviours (or functions such as light, open, secure...) that manipulate and control the data (André et al., 2015). Object-oriented analysis, design, modelling and programming approaches are all characterised by the four following principles (Fig. 2):

- Encapsulation: allows defining objects as entities with well-defined boundaries and identities that encapsulates data and behaviours. This enables, for instance, compliance checking of simple/elementary objects with related regulations and codes (Ex. a Fire Door with Fire Safety requirements), since they encapsulate all necessary data, without considering additional data or objects. In object-oriented programming, encapsulation also enables hiding of system's data implementation and making them private. To access data, accessor methods are defined and made public.
- Abstraction: enables defining 'abstract' objects that are not associated with any concrete instance such as spaces. In object-oriented programming, these objects are used to express the intent rather than the actual implementation. They constitute the interface of inner objects without communicating their content or details. Abstraction is also a relationship that relates two objects or sets of objects representing the same concept but at different levels of abstraction or from different perspectives. Abstraction relationships include different types such as derivation, realisation, refinement, and trace relationships.
- Inheritance: is a mechanism by which more specific objects, called 'child objects', acquire all the properties and behaviours of the more general objects, called 'parent objects'. Inheritance express subsumption (i.e. 'is-a') and/or composition (i.e. 'has-a') relationships between objects. For example, a wall can be load bearing or non-load bearing wall; internal or external, etc.; a building is composed of a set of storeys, and each storey is made of walls, beams, columns, slabs, doors, etc.
- Polymorphism: enables defining different object's behaviours using the same symbol, i.e. one symbol many forms. Thus, it allows different objects to respond to the same process in a different way according to their properties and definition. For instance, a compliance checking process can be triggered for a building as a whole object but conducted differently over each building's object (Ex. doors, walls, spaces, etc.) with respect to this latter's requirements and codes it should comply with (Ex. fire safety requirement, uniformity and structural integrity, access and circulation, etc.).

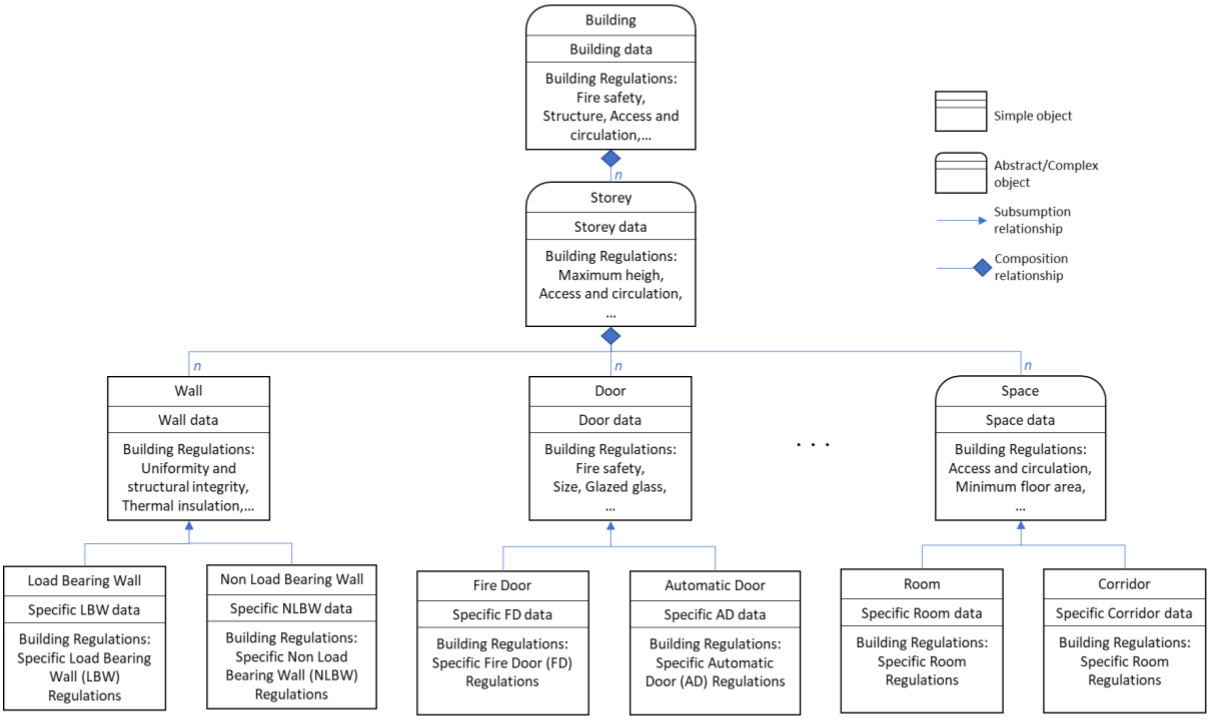

FIG. 2: Object-oriented approach principles

5.2 A novel bottom-up ACC approach

As stated by the investigation commission of the Challenger Accident, a bottom-up design approach is more suitable for quality control and design errors detection (United States Congress House Committe, 1986). Assembling objects in a bottom-up way with a good understanding of their properties and limitations, allows a high degree of confidence and quality of the whole model. Any further compliance or quality issue would more likely be the result of the interactions between objects rather than the objects themselves. However, a top-down approach is usually known to be expensive, with lack of knowledge about the objects, subsystems and their interactions, and presents less flexibility when errors are detected. In AI-based methods, top-down approaches aim to reproduce human intelligence independently of the biological structure of the brain and adopt macro rule-based descriptions to allow implementing inference operators, whereas bottom-up approaches are more interested in developing, training and incrementally tuning elementary brain components such as artificial neuronal networks, to ensure an overall intelligence and consistency of the system (Sinz et al., 2019). In this context, bottom-up approaches are more suitable to enable capturing and representing AI human-like intelligence (Brey & Søraker, 2009) in order to build Strong AI-based systems.

The method proposed takes advantage of the BIM object-oriented nature and focus on the notion of object to define a bottom-up approach for enabling desired ACC model checking characteristics including content validation, model completeness, smart object, and design option checking (see Fig. 3). Additionally, the object-centred ACC approach integrates all compliance checking essential processes, that are rules interpretation, BIM data preparation, rules execution, and results and reporting. These steps are now explained, with additional comments on the limitations of the previously reviewed work and how these have been addressed.

(i) Requirement rules interpretation, consisting 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, most of the existing frameworks fail to propose an efficient and automatic approach to allow this transformation automatically. Existing Natural Language Processing algorithms such as sematic-based, syntactic-based, and recent AI techniques-based, could not correctly handle ambiguous, uncertain and domain specific knowledge characterising building requirement texts, in a suitable way for industry applications where no degree of error is acceptable. Consequently, in this study, an intuitive and logical approach consisting of converting and representing these requirements through collaborative

work meetings with domain experts, into 'if-then' logical rules using XML language (Extensible Markup Language) allowing both human-readability and machine-readability, transparency and extensibility for the requirement rules set.

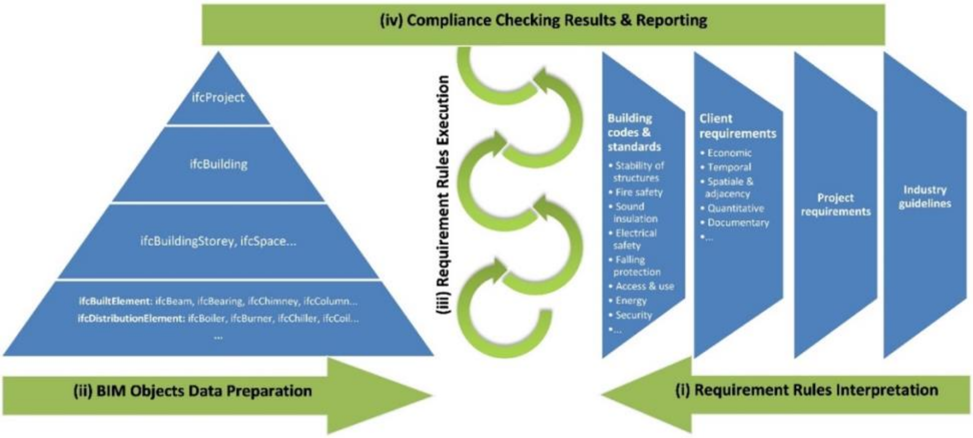

FIG. 3: BIM objects-based ACC method

Although if-then logical rules encoding has a limited expressive power to capture all kinds of expert knowledge, especially uncertain and modal knowledge (Garson, 2010), but it is still one of the most efficient and powerful knowledge representation formalisms that have been used in AI (Doukari & Greenwood, 2020). In addition, it provides a good trade-off between expressive power and analysability, while allowing tractability and decidability for decision problems (Doukari et al., 2009). Collaborative work meetings can be organised to help with understanding and clarifying the fuzziness of the rules as it is the case in this work.

(ii) BIM objects data preparation, consisting 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-centred which makes IFC the file format of choice as it is neutral and 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 (Lee 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 *ifcProduct* class (defined as "*any object that relates to a geometric or spatial context*" (IFC 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 *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, consisting 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 Fig. 3, 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, etc. defined by the *IfcRelAggregates* relationship), and mandatory requirements are also scanned first before guidelines and non-mandatory requirements.

(iv) Requirement rules checking results and reporting, that 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.

Table 3 summarises the characterisation of the proposed approach based on the desired ACC rule characteristics and performance.

TABLE 3*: Object-centred ACC approach characterisation*

Research works	Deployment	Technology concept	Industry application	Open standards	Classification	Model checking criteria				Compliance checking criteria				Rules criteria				
						Content validation	Content completeness	Smart object	Design option	Interpretation	Preparation	Execution	Reporting	Encoding approach	Independency of the BIM model	Independency of the model checking tool	Extensibility/sustainability	Uncertainty handling approach
Current paper	Add-in	Bottom-up approach, BIM objects, four-steps process, modular rules encoding	Yes	Yes	Object-centred	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Manual (modular 'if-then' XML rules encoding)	Yes	Yes	Yes	Collaborative workshops

Standardisation initiatives for construction objects (or products), such as XP P07-150 (AFNOR, 2014), ISO 16757 (ISO, 2015), EN ISO 23386 (ISO, 2020a), EN ISO 23387 (ISO, 2020b), and LEXiCON (Construction Innovation HUB & Construction Products Association, 2021), aim to harmonise data quality and information exchange between construction stakeholders. In these frameworks, construction objects are defined and characterised through a set of consistent attributes to allow capturing the data that describe their performance and quality, systems and their components. The general structure, called Data Template, can be used to provide digital description for any construction object. A unified framework is currently being developed (CEN, 2021) and will precise how data templates should be created, which will allow manufacturers to digitise their products and provide their data in a homogenised and agreed way. These objects-based standards and efforts will certainly simplify the implementation of the proposed ACC approach and the development of BIM data quality assurance and control processes (Ramesh, 2016) through automated tools based on, for example, BIM object quality checklists like the ones proposed in (Zadeh et al., 2017).

Section 6 presents a real-world case study including an ACC checklist defined with SYNEG (the national association of French foodservice equipment manufacturers) based on the 'Foodservice Equipment Standards Document' (Foodservice Consultants Society International, 2020) that aims to define a set of guidelines and recommendations for BIM model content creation.

6. AUTOMATED COMPLIANCE CHECKING APPLICATION

In this section, two case studies are presented and processed to demonstrate the efficiency of the object-centred approach. The first one illustrates the proposed approach on a real-world case study using a BIM objects library created by a French BIM adopter manufacturer, and the second one presents an additional example relating to building codes using Fire Door objects to further clarify and prove the applicability of the object-centred approach to other requirement domains and types. The two applications are illustrated through the four-step method: (i) Requirement rules interpretation, (ii) BIM objects data preparation, (iii) Requirement rules execution, and finally (iv) Compliance checking results and reporting.

6.1 Case study 1: SYNEG compliance checking

To demonstrate the feasibility and accuracy of the proposed approach, a first case study from the foodservice equipment industry using an existing object library of foodservice equipment created by BONNET THIRODE, a major French manufacturer, is processed.

a) Requirement rules interpretation

The definition of requirements is based on the '*Foodservice Equipment Standards Document*' (Foodservice Consultants Society International, 2020) that aims to define a set of guidelines and recommendations for BIM model content creation within an object-oriented authoring platform. This document 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 cover various themes of BIM modelling guidelines related to object templates (hosted *versus* free-standing), object representation and level of detail, visibility settings, nested object-groups-voids, imported geometry and linked files, manufacturer logos, object and object type naming, categories and subcategories of BIM components, parameters, materials, connectors, etc. (an extract is given in Table 4).

These guidelines and standards have been developed and interpreted with SYNEG, the national association of French catering equipment manufacturers which is one of the contributors to the definition of FCSI Foodservice equipment standards, within Autodesk Revit as an object-oriented authoring platform.

To clarify and detail some of the fuzzy rules given in these standards, several meetings were organised to help with understanding these documents. For example, regarding "*Connectors*" (see Table 4) the team needed to understand which controls to create and what input information to use to enable them. Furthermore, considering the foodservice equipment complexity and so the high skills level required to model them using BIM authoring platforms, some of the rules have been adapted (e.g., Max object file size is set to 3 MB instead of 0,75 MB). Also, provision 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.

TABLE 4: Extract from the final checklist validated with SYNEG

Themes	Rules	Checking type
Authoring tool version	Objects should be created in a '*software release*' >= '*software current release'* - 3	Metadata
Object template	a) Objects should be created under the '*Specialty Equipment*' category b) It is recommended to use free-standing and not hosted object templates	Content validation, smart object, design option
Object representation	Coarse and fine levels of detail are used in 2D and 3D representations	Model completeness
Object file size	Object file size is no more than 3 MB	Metadata
Imported geometry	It is recommended to not use imported/linked CAO files	Content validation
Manufacturer logo	a) Existence of a parameter named '*Show Logo*' to control the visibility of the logo b) All logo geometry should be assigned to the *QF_Logo* subcategory	Model completeness, content validation, smart object
Object naming	a) Naming objects should respect '*QF_Brand_CATALOGREFERENCE*' b) Object or category name should not be used as an object type name	Metadata, content validation, design option
Categories and subcategories	a) The following subcategories should exist in each object: *QF_Logo* b) The following subcategories should respect the following line types: *QF_Clearances_Code Dash 1/16"* *QF_Clearances_Door_Drawer Dash 1/16"* *QF_Clearances_Service Dash 1/16"* *QF_Details_Cutout Solid* *QF_Details_Overcounter Dash Dot 3/16"* *QF_Details_Surface Solid* *QF_Details_Undercounter Dash 1/16"* *QF_Logo Solid*	Model completeness, content validation, design option
Parameters	Not all the IFSE parameter list should be added to all objects, but only those indicated by 'X' in the manufacturer's library description file	Model completeness, content validation
Materials	Objects should use the FCSI Material Library	Content validation
Connectors	Type and instances number should comply with the manufacturer's library description file	Model completeness, content validation
Type catalogues	A catalogue of types should be created if more than 6 types exist	Metadata

Table 4 shows a subset of the final version of SYNEG checklist including recommendations as adapted and validated with SYNEG. These rules are interpreted into if-then logical rules using XML language in a format that is both human-readable and machine-readable, thus enabling transparency and extensibility for the set of rules. Fig. 4 illustrates the XML structure adopted and an example of a rule encoded to allow checking the existence of a parameter named "*Show Logo*".

```xml
<?xml version="1.0" encoding="UTF-8" standalone="true"?>

<RulesSet SetName="SYNEG Rules" Description="Foodservice Modelling Standards">
  <Rule RuleName="" Description="" FailureMessage="" Condition="" IsMandatory="">
    <Filter Category="" Property="" Condition="" Value="" Operator="" />
    <Filter ... />
    <Filter ... />
  </Rule>
  <Rule RuleName="Show Logo Parameter" Description="Checks if there exists a parameter called Show Logo"
    FailureMessage="Show Logo Parameter does not exist" Condition="FailNoElements" IsMandatory="True">
    <Filter Category="Parameter" Property="Name" Condition="Equal" Value="Show Logo" />
  </Rule>

  <Rule ...>
  </Rule>

</RulesSet>
```

Structure of the rules

Example of a rule

FIG. 4: XML structure for the Foodservice Modelling Standards Rules Set

The rules set will be then loaded and checked by the Rule Execution Engine which is an application developed as an Autodesk Revit plugin using the C# programming language (see Section 6.3).

b) BIM objects data preparation

A complete BIM objects library was developed by the French manufacturer for its products. This library served as a case study of our approach, while each object is a Revit Family (RFA) file representing a different type of a foodservice equipment along with all its required information details. Consequently, in this case study, the BIM data preparation step consists simply in putting both the BIM objects to be checked and the Manufacturer's objects description file (Fig. 5) in a single repository. All the objects are physical and there was no need to construct implicit or virtual ones. Moreover, as they are independent from each other, and defined at the same level of complexity, their processing was straightforward and without predefined order.

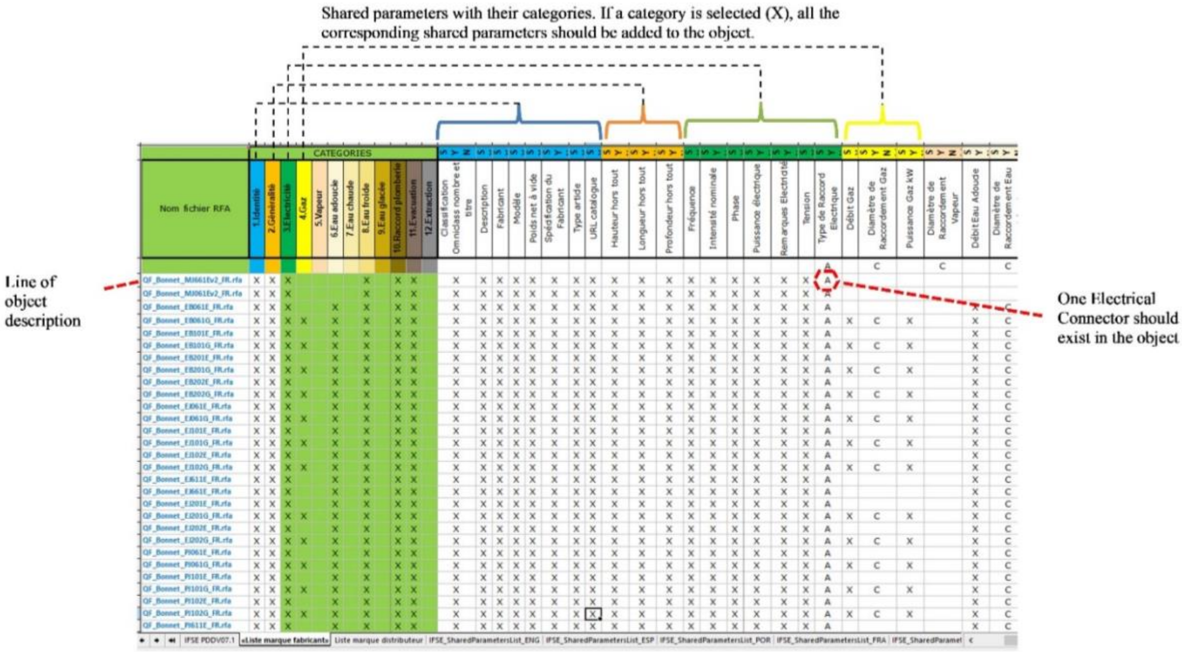

FIG. 5: Example of Manufacturer's library description file

c) Requirement rules execution

To allow ACC for the BIM objects library, a SYNEG plugin was developed using C# programming language, within the Autodesk Revit 2020 environment. As shown in Fig. 6, this plugin including the Rules Execution Engine, enables loading and executing the XML file containing the Foodservice Modelling Standards rules set, by using (i) a GUI (Graphical User Interface) to simplify its utilisation by non-expert users; (ii) Autodesk Revit API; (iii) the FCSI Materials library; and (iv) other resources (file templates, metadata, etc.). The plugin inputs are the BIM objects library to be checked alongside the rules set (XML file) and the manufacturer's objects description (XLS file). The outputs from the plugin are the ACC reports: one overall for all the BIM objects and individual reports for each object family.

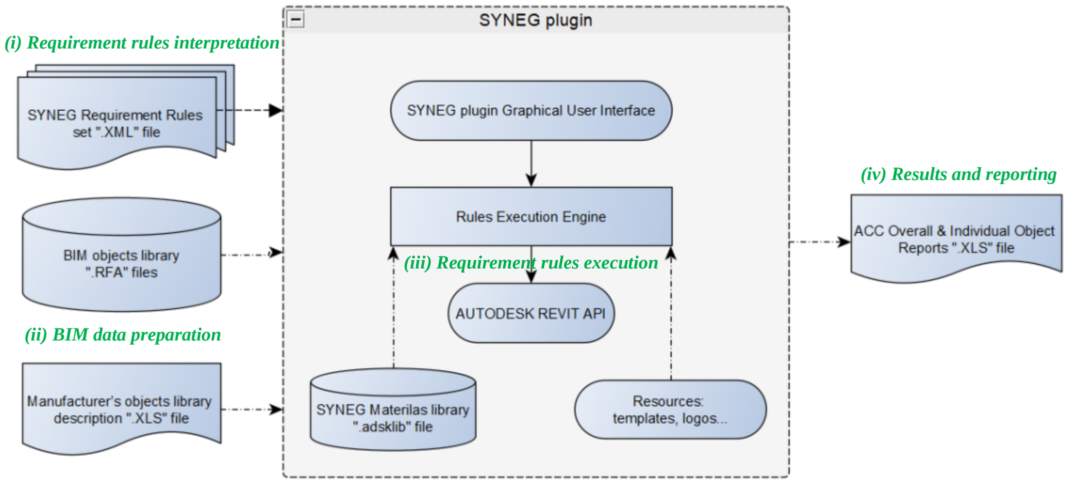

FIG. 6: SYNEG plugin – Component and Functional Architecture

The process of using this plugin consists of 3 steps (Fig. 7):

1. Launch Autodesk Revit 2020 and accept loading the SYNEG plugin (ACC Menu/functionality);
2. Launch ACC process by clicking on SYNEG button to enable loading the SYNEG rules set;
3. Select the library or BIM objects to be checked and validate the selection over the browser window.

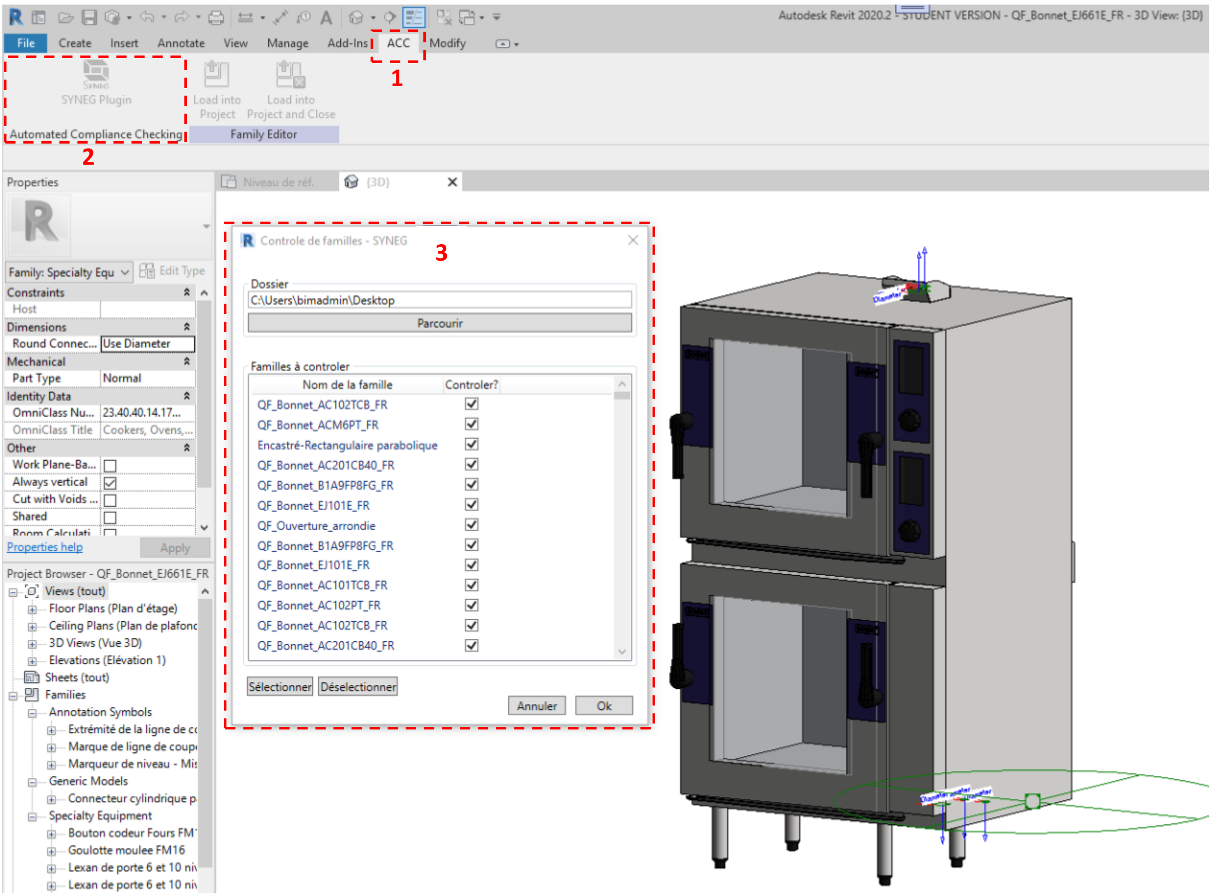

FIG. 7: SYNEG plugin Graphical User Interface

At the end of the process, the compliance checking result is generated, and a complete report with various Excel sheets is displayed automatically. It should be noted that the classification of the rules and their encoding in separate sets according to their application domains will greatly simplify future extension and enrichment of the rules base.

d) Results and reporting

By executing the SYNEG plugin using the XML rules set on the BIM objects library, a detailed ACC report with respect to the Foodservice Modelling Standards, is automatically generated. It consists of an Excel file with multiple sheets as many as there are objects family within the BIM library. The first sheet ("*Recapitulatif* ": *Summary* in English), indicates the overall compliance checking result depending on the average of all the objects' compliance results. Fig. 8 shows the average results per rule, and an overall result of all these averages to decide whether the checked BIM library "Passes" or "Fails". In our case, the BIM library does not pass the test since the overall result is "0%".

Furthermore, the result of the compliance checking of each BIM object is given in a separate sheet. To justify the compliance checking result and enable future correction, an automated justification for each "Failed"/ "Passed" rule result is given in the individual reports.

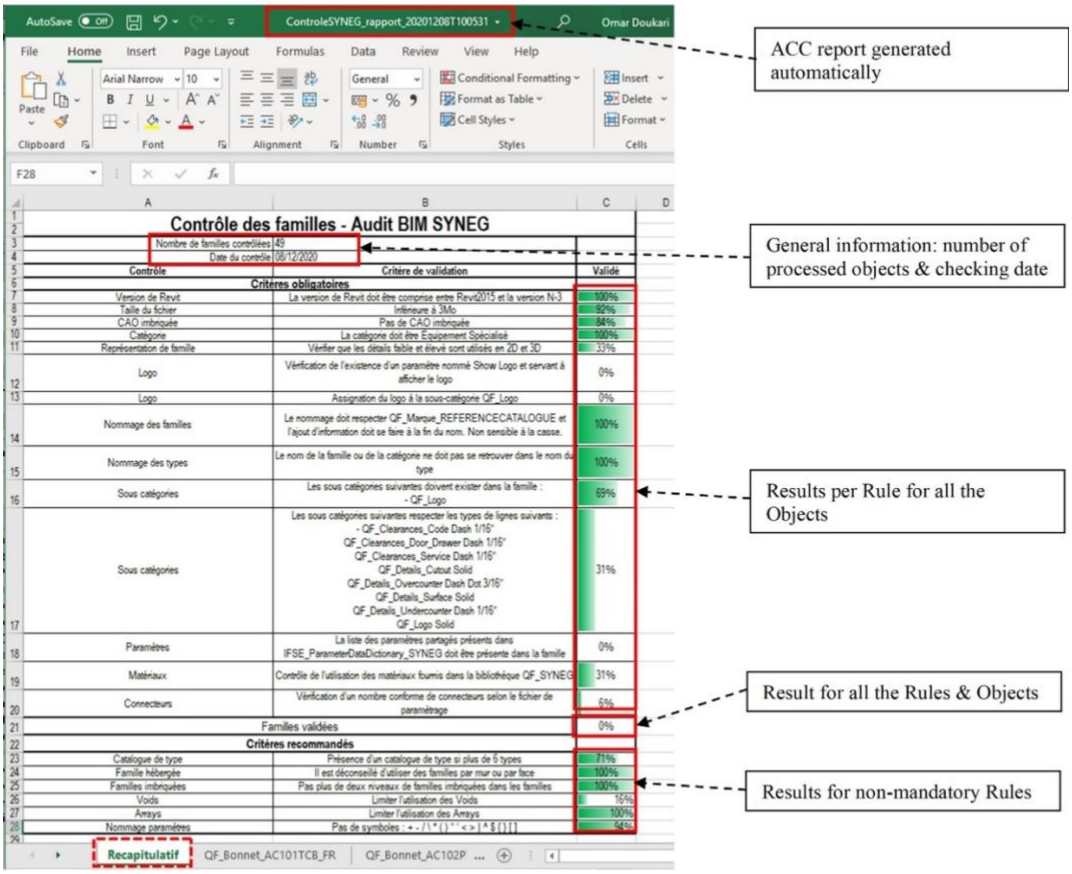

FIG. 8: Case study 1 – ACC results and reporting

6.2 Case study 2: Fire safety compliance checking

Building regulation includes managing fire safety precautions within and around buildings. Building operators must ensure reducing the risk of fire spreading within the premises (Regulatory Reform, 2005). Fire doors play a key role in achieving this goal and ensure occupants and buildings safety. They are a legal requirement in all non-domestic properties as well as in residential flats and houses of multiple occupancy that have three or more floors.

To clarify and prove the applicability of the proposed approach to other requirement domains, a Fire Door BIM object example taken from the NBS National BIM Library (NBS National BIM Library, 2015) is processed according to the four-step method.

a) Requirement rules interpretation

Design specifications and requirements of Fire Doors should be checked and certified before manufacture. Table 5 presents an extract of these requirements in a checklist form. This list of rules is interpreted and translated into if-then logical rules following the XML schema illustrated in Fig. 4. The resulting XML file is independent of the ACC plugin. It will be loaded as external input and checked using the Rule Execution Engine during the Requirement rules execution step (Section 6.2.c). Fig. 9 illustrates the 'Fire Door Leaf's Rating rule' (the first rule in Table 5) represented in XML format to enable checking the existence and value of the '*Fire Rating*' parameter of a 'Door Leaf' element.

```
<Rule RuleName="Fire Door Leaf Rating" Description="Checks if the door leaf rating value exists and valid"
  FailureMessage="Fire Door Leaf Rating is not valid" Condition="FailNoElements" IsMandatory="True">
  <Filter Category="Type" Property="Name" Condition="Equal" Value="Door Leaf" Operator="And"/>
  <Filter Category="Parameter" Property="Name" Condition="Equal" Value="Fire Rating" Operator="And"/>
  <Filter Category="Parameter" Property="Value" Condition="GreaterOrEqual" Value="30" />
</Rule>
```

FIG. 9: Fire Door Leaf's Rating rule expressed in XML

Table 5: Extract from Fire Doors requirements checklist (Saunders, 2021)

Themes	Rules	Checking type
Fire rating	a) Fire door leaf's rating >= 30 min b) Fire door frame's rating >= 30 min c) Fire door hinge's rating >= 30 min d) Fire door frame's rating = Fire door leaf's rating	Content validation, model completeness
Leaves	a) 2mm <= Gap between leaf and frame <= 4mm b) Leaves should contain 'intumescent strips' and 'cold smoke seals' on the top and sides	Content validation, model completeness
Glazing	All should be fire resistant	Content validation, model completeness, smart object
Hinges	3 sets of door hinges are recommended for each door leaf	Content validation, model completeness
Closer	1 automatic door closer should exist	Content validation, model completeness
Retainer	1 door retainer is recommended	Content validation, model completeness
Signs	2 fire door signs should be fitted both sides of the door	Content validation, model completeness

b) BIM objects data preparation

A Fire Door BIM object was downloaded from the NBS National BIM Library platform and used as a second case study (NBS National BIM Library, 2015). The BIM object was originally created by BIMBox with Autodesk Revit as a RFA file (Fig. 10), and the physical Fire Door is currently manufactured by Dortek Ltd.

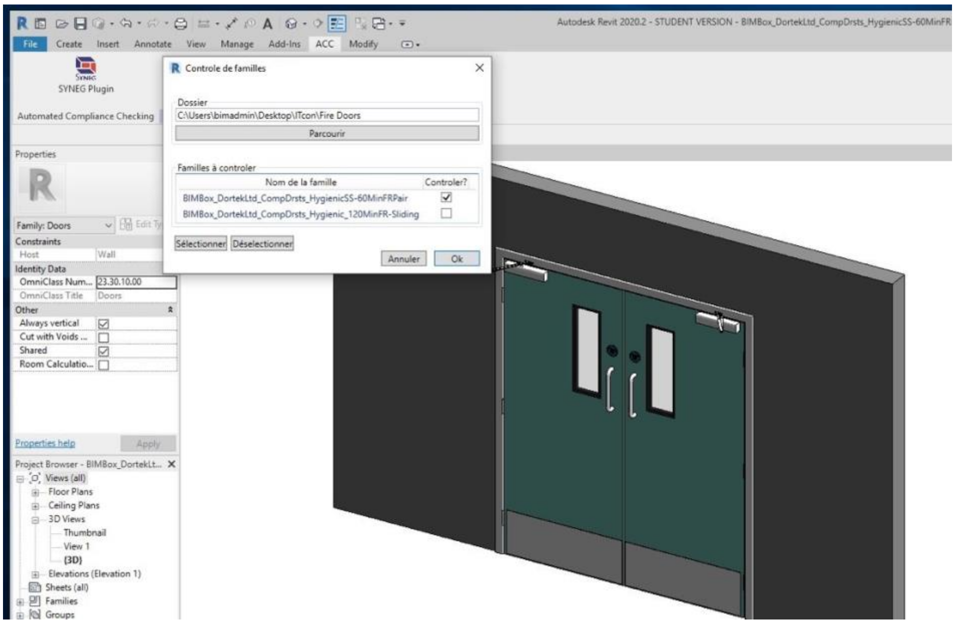

FIG. 10: A Fire Door BIM object (NBS National BIM Library, 2015)

Since the BIM object already exists, the BIM data preparation step will, as in the first case study, simply consist in putting the Fire Door BIM object in the repository where it should be uploaded then checked by the ACC plugin.

To identify BIM objects, their parameters and specific properties in a unified way, a modelling chart including objects and parameters naming, and related semantics, should be adopted when creating BIM objects. An objects library description file, as in the first case study, is required in order to specify objects names, parameters, and properties, and connect or map them to their corresponding meaning. For instance, a 'Fire Rating' parameter of a door leaf or frame must have a unique name and refer to the same meaning in all BIM objects. This will particularly enable unifying the way the rules are expressed in XML and so simultaneously processing many BIM objects libraries. On the other hand, this process will certainly improve the level of collaboration and communication between clients, designers, product manufacturers and suppliers, and allow them to easily share and exchange product information. Data Template standardisation frameworks such as (AFNOR, 2014), ISO 16757 (ISO, 2015), EN ISO 23386 (ISO, 2020a), EN ISO 23387 (ISO, 2020b), LEXi-CON (Construction Innovation HUB & Construction Products Association, 2021), and (CEN, 2021) will play a decisive role in achieving this goal.

Fire Safety Requirements Control		
# BIM objects controlled	1	
Control date	29/01/2022	
Theme	**Control**	**Validated**
Fire rating	Fire door leaf's rating >= 30 min	0%
Fire rating	Fire door frame's rating >= 30 min	0%
Fire rating	Fire door hinge's rating >= 30 min	0%
Fire rating	Fire door frame's rating = Fire door leaf's rating	0%
Leaves	2mm <= Gap between leaf and frame <= 4mm	100%
Leaves	Leaves should contain 'intumescent strips' AND 'cold smoke seals'	0%
Glazing	Fire resistant	0%
Hinges	3 sets of door hinges are recommended for each door leaf	100%
Closer	Automatic door closer should exist	100%
Retainer	Door retainer is recommended	0%
Signs	Fire door signs should be fitted both sides of the door	100%

FIG. 11: Case study 2 – ACC results and reporting

c) Requirement rules execution

The ACC plugin presented in Section 6.1.c enables automatically checking the Fire Door BIM object. To do so, the Rules Execution Engine loads the BIM object and executes the XML file comprising the Fire Door rules. In this case, the inputs are the Fire Door BIM object, the Fire Door requirements checklist (XML file) and the XLS file template where the results will be reported. The outputs are the ACC reports of the Fire Door BIM object given in an XLS file as illustrated in Fig. 11.

d) Results and reporting

The ACC plugin automatically generates a detailed ACC report regarding the compliance of the Fire Door BIM object with the Fire Doors requirements (Fig. 11). The output shows the compliance checking result of each rule, where '100%' stands for rule 'Passes' and '0%' value means rule 'Fails'. Indeed, a quick manual check of the Fire Door BIM object can clearly show that no 'Fire rating' data, for both Door leaves, Door frame and Glazing elements were integrated in the BIM object. In addition, some BIM elements such as 'Door retainer', 'Intumescent strips', and 'Smoke seals' have not been modelled and integrated into the BIM model.

7. DISCUSSION

The bottom-up approach for ACC 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 four steps: requirement rules interpretation, BIM objects data preparation, requirement rules execution, and compliance checking results and 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 using XML language in a format that is both human-readable and machine-readable to enable transparency and extensibility for the set of rules which will be loaded and checked thereafter by the Rules Execution Engine. Although the limitations that such an encoding approach presents in terms of language expressiveness, it remains one of the most intuitive and efficient formalisms that have been used in AI (Doukari & Greenwood, 2020), especially in the context of industry application where no degree of error is acceptable. Indeed, representing standards into if-then logical rules provides a good trade-off between expressive power, tractability and decidability for decision problems (Doukari et al., 2009). Furthermore, to address the problem of dealing with uncertain and ambiguous building requirements characterising compliance checking application domains (Zhang & El-Gohary, 2016), various meetings with domain experts have been organised to allow the best possible interpretations for the standards. Moreover, to answer the challenge of regulations extensibility as pointed out in (Hakim et al., 2017), the rules base has been interpreted in XML language, independently from the ACC software (the Rules Execution Engine), alongside a modular structure that has been adopted and implemented. This structure enables each requirements class to be represented into an independent rules set.

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 and 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, train and support them in the design correction, is to conduct it using the same authoring environment via a new integrated functionality – a plugin in this case study. 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.

In terms of performance, compared with the manual approach, Table 6 presents the results of execution times corresponding to the ACC 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 BIM library including 49 BIM objects. The ACC 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 version).

Concerning the manual approach, execution times have been calculated by considering estimations of a Revit modeller expert who conducted the whole compliance checking process manually. For only 20 elementary rules, the time to check and report one BIM object was estimated to be 7 490 seconds (~ 125 minutes), whereas the 49 objects library would take 367 010 seconds which is equivalent to at least 2 working weeks.

TABLE 6: Comparison of ACC execution times (in seconds) using SYNEG plugin and manual approach

	Manual approach			ACC (SYNEG plugin)				
	Rules execution	Reporting	**Total**	Objects 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**
BIM library: 49 objects	27 8810	88 200	**367 010**	112.26	81.63	2.32	0.87	**197.09**

8. CONCLUSION AND PERSPECTIVES

This research work aims to propose a new framework for bottom-up ACC approach that leverages the object-based representation of building information models. This framework develops a hierarchical structuring and logical processing of the ACC operation, while enabling quality assurance of the information existing in the construction industry at the detailed level, where content validation, model completeness, smart object, and design option checking are simultaneously performed. It also targets to enhance the understanding of the ACC process and test it by implementing and analysing two case studies. The first application illustrates the feasibility and the accuracy of the proposed approach using a real-world case study, i.e. a BIM objects library created by a French BIM adopter manufacturer, whereas the second application demonstrates and proves its applicability to other requirement domains using a Fire Door BIM object downloaded from the NBS National BIM library (NBS National BIM Library, 2015).

To achieve that, an extensive literature review of more than thirty papers dealing with ACC frameworks and systems has been conducted, leading to identify a classification into six categories of existing ACC approaches (AI techniques, domain specific language, query language, reasoner, bespoke tool, and general architecture), and a set of desired ACC characteristics and performance using an incomplete search method, but 'intelligent' since based on a nature-inspired optimisation metaheuristic known as simulated annealing method. Then, a new framework for bottom-up object-centred ACC approach has been defined, structured over a four-step process, and characterised according to these desired ACC criteria and performance. This framework satisfies all desired ACC characteristics and performance identified form the literature review and 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. This structuration of BIM objects and rules allows verification of both simple and complex information. For example, as seen in the case study (Section 6), elementary information such as those related to manufacturer logo and its visibility settings can be considered and checked with regard to existing standards and norms. Furthermore, behavioural (such as, using free-standing and not hosted object) and complex (or structuring, such as 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 BIM objects library created by a French manufacturer, has been presented and processed by implementing the ACC four-step desired process, and a new plugin application for ACC has been developed and tested. This plugin assists BIM object designers and modellers and helps them to check and correct their BIM object libraries according to the Foodservice modelling standards and requirements.

As stated in the Methodology section (Section 3), the implementation of the proposed ACC framework is expected to be done iteratively by integrating, testing and analysing its different functionalities and building regulation and standards, one by one until achieving satisfactory results. This paper presents the first iteration of the research project process, where the definition of a general architecture of the bottom-up object-centred approach, in addition to enabling quality assurance including content validation, model completeness, smart object, and design option checking, were the main objectives. In future work, this application will be further developed to address the general compliance checking framework presented in this paper. This would enable: (i) processing of other regulatory 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. The unified standards (CEN, 2021) that will define homogeneously and in an agreed way, product data templates could be adopted to streamline the development of BIM data quality assurance and data control processes that allow to ensure high BIM data quality using BIM object quality checklists (Zadeh et al., 2017).

Furthermore, to allow SYNEG to approve created BIM libraries and play the role of a certification organisation, it may be possible 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 (see Fig. 12).

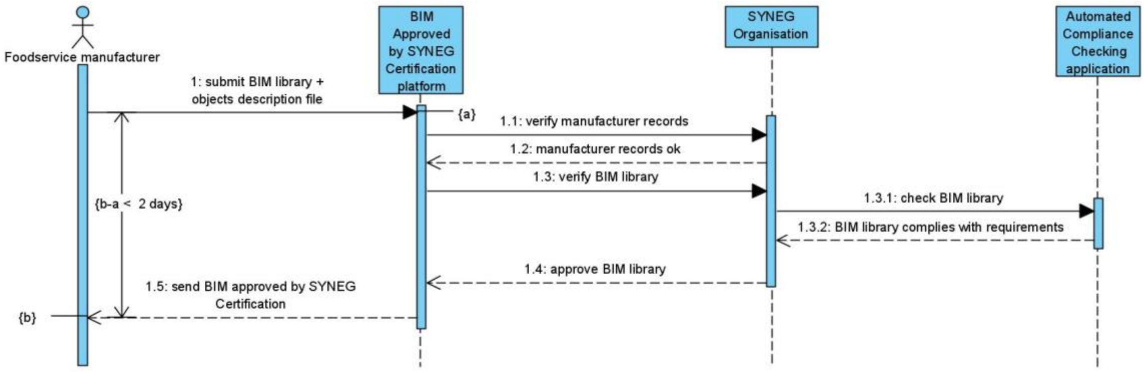

FIG. 12: BIM Approved by SYNEG Certification scenario

ACKNOWLEDGEMENTS

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, Louis CHAUVEAU, Brice CANTENEUR, André-Pierre DOUCET, Laurent GODARD, and Géraldine LARRIVEE.

DATA AVAILABILITY

All data, models, and code generated or used during the study appear in the submitted article.

REFERENCES

AFNOR, 2014. PPBIM - XP P07-150. *AFNOR Boutique Editions*, December, pp. 07-150.

Amor, R. & Dimyadi, J., 2021. The promise of automated compliance checking. *Developments in the Built Environment,* 5(100039, ISSN 2666-1659).

André, D., Charles, J.-L., & Iordanoff, I., 2015. Object Oriented Approach and UML. *3D Discrete Element Workbench for Highly Dynamic Thermo-Mechanical Analysis*, John Wiley & Sons, Ltd, pp. 1–14. doi: 10.1002/9781119116356.ch1.

Bandyopadhyay, R., Basu, A., Cuevas, E. & Sarkar, R., 2021. Harris Hawks optimisation with Simulated Annealing as a deep feature selection method for screening of COVID-19 CT-scans. *Applied Soft Computing,* 111(107698).

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.

Brey, P. & Søraker, H. J., 2009. Philosophy of Computing and Information Technology. In: *Handbook of the Philosophy of Science, Philosophy of Technology and Engineering Sciences.* North-Holland: s.n., pp. 1341-1407.

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

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.

CEN, E. C. f. S., 2021. Building information modelling (BIM) - Data templates for construction objects used in the life cycle of built assets - Data templates based on European standards and technical specifications. *CEN/TC 442 - Building Information Modelling (BIM)*, 25 01, pp. https://standards.cen.eu/dyn/www/f?p=204:110:0::::FSP_PROJECT,FSP_LANG_ID:73441,25&cs=1C565BFE2CF5C2B8BABE0494591EC9A02.

Chebbi, A., Bargaoui, Z. K., Abid, N. & Cunha, M. d. C., 2017. Optimization of a hydrometric network extension using specific flow, kriging and simulated annealing. *Journal of Hydrology,* Volume 555, pp. 971-982.

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.

Clayton, M. J., 2013. *Automated plan review for building code compliance using BIM.* Vienna, Austria, 20th International Workshop: Intelligent Computing in Engineering, pp. 1–10. EG-ICE 2013..

Construction Innovation HUB & Construction Products Association, 2021. *LEXiCON Methodology: Creating Relevant Authorities and achieving consensus,* UK: Construction Innovation HUB, https://constructioninnovationhub.org.uk/.

Designing Buildings, 2021. *Building Regulations,* s.l.: Available at: https://www.designingbuildings.co.uk/wiki/Building_regulations (Accessed 8 February 2021)..

Dimyadi, J., Fernando, S., Davies, K. & Amor, R., 2020. *Computerising the New Zealand Building Code for Automated Compliance Audit.* Auckland, New Zealand , 6th New Zealand Built Environment Research Symposium, 6, pp. 39–46..

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.

Dimyadi, J., Thomas, G. & Amor, R., 2017. *Enabling automated compliance audit of architectural designs.* Wellington, New Zealand, The 51th International Conference on Architectural Science.

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.

Doukari, O. & Greenwood, D., 2020. Automatic generation of building information models from digitized plans. *Automation in Construction,* 113(103129).

Doukari, O. & Jeansoulin, R., 2012. *A Logic-based Model for uncertainty reduction: parallel processing.* Leipzig, Germany, Environmental Modelling and Software 2012, July 1-5.

Doukari, O., Wurbel, E. & Jeansoulin, R., 2009. Distance-Based Semantics for C-Structure Belief Revision. In: S. B. Heidelberg, ed. *Symbolic and Quantitative Approaches to Reasoning with Uncertainty.* Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 359-370.

Duan, Q. et al., 2021. Improved Salp Swarm Algorithm with Simulated Annealing for Solving Engineering Optimization Problems. *Symmetry,* 13(6).

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.

Eberhart, R. & Kennedy, J., 1995. *A new optimizer using particle swarm theory.* Nagoya, Japan, IEEE - MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science.

Eglese, R., 1990. Simulated annealing: A tool for operational research. *European Journal of Operational Research,* 46(3), pp. 271-281.

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

Ficarella, E., Lamberti, L. & Degertekin, S., 2021. Comparison of three novel hybrid metaheuristic algorithms for structural optimization problems. *Computers & Structures,* 244(106395).

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

Garson, J. W., 2010. Expressive Power and Incompleteness of Propositional Logics. *Journal of Philosophical Logic,* 39(2), p. 159–171.

Ghang, L. et al., 2014. Query Performance of the IFC Model Server Using an Object-Relational Database Approach and a Traditional Relational Database Approach. *Journal of Computing in Civil Engineering,* Volume 28, pp. 210-222.

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.

Grant, M. J. & Booth, A., 2009. A typology of reviews: an analysis of 14 review types and associated methodologies. *Health Information and Libraries Journal,* 26(2), pp. 91-108.

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.

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

ISO, 2015. ISO 16757-1:2015 - Data structures for electronic product catalogues for building services — Part 1: Concepts, architecture and model. *Technical Committee : ISO/TC 59/SC 13 Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM)*, 02, p. 33.

ISO, 2020a. ISO 23386:2020 - Building information modelling and other digital processes used in construction — Methodology to describe, author and maintain properties in interconnected data dictionaries. *Technical Committee : ISO/TC 59/SC 13 Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM)*, 03, p. 40.

ISO, 2020b. ISO 23387:2020 - Building information modelling (BIM) — Data templates for construction objects used in the life cycle of built assets — Concepts and principles. *Technical Committee : ISO/TC 59/SC 13 Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM)*, 07, p. 17.

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).

Kalivas, J. H., 1992. Optimization using variations of simulated annealing. *Chemometrics and Intelligent Laboratory Systems,* 15(1), pp. 1-12.

Kassem, M. et al., 2019. *Measuring and Improving the Productivity of Construction's Site Equipment Fleet: An integrated IoT and BIM System.* Newcastle, UK, 36th CIB W78 2019 Conference Advances in ICT Design, Construction & Management in Architecture, Engineering, Construction and Operations (AECO), p. 901-911.

Katoch, S., Chauhan, S. S. & Kumar, V., 2021. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications,* Volume 80, p. 8091–8126.

Kaveh, A., 2017. *Applications of Metaheuristic Optimization Algorithms in Civil Engineering.* 1 ed. Springer International Publishing Switzerland 2017: Springer, Cham.

Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P., 1983. Optimization by Simulated Annealing. *Science,* 220(4598), pp. 671-680.

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, G. et al., 2014. Query performance of the IFC model server using an object-relational database approach and a traditional relational database approach. *Journal of Computing in Civil Engineering,* 28(2), pp. 210-222.

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.

Li, B., Dimyadi, J., Amor, R. & Schultz, C., 2020. *Qualitative and traceable calculations for building codes.* São Paulo, 37th CIB W78 Information Technology for Construction Conference, pp. 69–84..

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).

NBS National BIM Library, 2022. Hygienic Hinged GRP Fire Doors - 60 Min FR - Single leaf (GRP frame) - NBS National BIM Library. Available at: <https://www.nationalbimlibrary.com/en-gb/objects/?id=235370&cid=10317&lo=en-GB> [Accessed 29 January 2022].

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.

Ramesh, A., 2016. *A Procedure For Planning The Quality Assurance And Control For Facility Information Handover,* Pennsylvania, USA: Master Thesis, Penn State University.

Román-Román, P. & Torres-Ruiz, F., 2015. A stochastic model related to the Richards-type growth curve. Estimation by means of simulated annealing and variable neighborhood search. *Applied Mathematics and Computation,* Volume 266, pp. 579-598.

Regulatory Reform, 2015. The Regulatory Reform (Fire Safety) Order 2005. Available at: < https://www.legislation.gov.uk/uksi/2005/1541/contents/made> [Accessed 29 January 2022].

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.

Saka, M. & Dogan, E., 2012. Recent Developments in Metaheuristic Algorithms: A Review. *Computational Technology Reviews,* 5(doi:10.4203/ctr.5.2), pp. 31-78.

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.

Saunders, A. M., 2021. 10 Things You Should Know About Fire Doors. *Safelincs Blog*. Available at: <https://www.safelincs.co.uk/blog/2021/07/07/10-fire-door-questions-answered/> [Accessed 29 January 2022].

Sinz, F. H. et al., 2019. Engineering a Less Artificial Intelligence. *Neuron,* 103(6), pp. 967-979.

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.

Swartz, M. K., 2011 . The PRISMA statement: a guideline for systematic reviews and meta-analyses. *J Pediatr Health Care,* 25(1).

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.

Torres-Carrión, P. V., González-González, C. S., Aciar, S. & Rodríguez-Morales, G., 2018. *Methodology for Systematic Literature Review.* Santa Cruz de Tenerife, Spain, 2018 IEEE Global Engineering Education Conference (EDUCON).

Tschida, C. E. & Silverberg, L. M., 2013. Cellular growth algorithms for shape design of truss structures. *Computers & Structures,* Volume 116, pp. 1-6.

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

United States Congress House Committe, 1986. *Investigation of the Challenger Accident: Report of the Committee on Science and Technology,* Ninety-Ninth Congress, Second Session: House Report.

Yang, X.-S., 2014. Chapter 4 - Simulated Annealing. In: X. Yang, ed. *Nature-Inspired Optimization Algorithms.* s.l.:Elsevier, pp. 67-75.

Zadeh, P. A. et al., 2017. Information Quality Assessment for Facility Management. *Advanced Engineering Informatics,* Volume 33, pp. 181-205.

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., 2014. *Integrating safety and BIM: automated construction hazard identification and prevention,* Georgia, USA: (PhD Thesis) Georgia Institute of Technology.

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.