Restructuring Elements in IFC Topology through Semantic Enrichment: A Case of Automated Compliance Checking

Ankan Karmakar[1] and Venkata Santosh Kumar Delhi[1]

[1]Department of Civil Engineering, Indian Institute of Technology Bombay, India
204046006@iitb.ac.in, venkatad@iitb.ac.in

Abstract –

The current trends exhibit that the construction industry is focusing on implementing the Building Information Model (BIM) to improve visual analyses, cost estimation, and information exchange. Though standards, requirements, and development control rules (DCR) govern the construction project lifecycle, only a few applications have been developed to investigate BIM's capability in automated code compliance checking (ACCC). A significant amount of the existing application limits its checking capabilities to direct parameter values, easily extractable from a Building Information Model (BIM). On the other hand, a few clauses require implicit information to be extensively expressed in machine language for automated checking. Industry Foundation Classes (IFC) derived from BIM models lacks the relational connections of building elements required for extracting property values for automated checking. This study develops a model to restructure a topologically complex element relationship (i.e., sunshade projection width) in an IFC schema. Different categories of semantic enrichment tasks are performed for restructuring and to represent the required clause test values explicitly. The model's performance was assessed on a test case concerning the clause requirements of the Unified Development Control and Promotion Regulations (UDCPR) code for Maharashtra, India. The compliance check method identified the sunshades in the developed model through simple if-then conditions. The method adopted in the study can be applied to similar code requirements to reduce the manual data preprocessing effort required from the architects and modelers, leading to enhanced penetration of ACCC in the industry.

Keywords –

Automatic Code Compliance Checking; Building Information Modeling; Industry Foundation Class; Semantic Enrichment; Artificial Intelligence

1 Introduction

The architectural, engineering, and construction (AEC) industry is increasingly becoming information intensive with the increased complexity of projects being executed in shorter timeframes. A direct implication is that the industry processes diverse information across projects. Such information is often encapsulated in a BIM model. BIM is now being used for but not limited to, data sharing, retrieval, cost estimation, energy consumption calculations, visual analyses, and building regulatory compliance checking [1]. Modern BIM tools successfully address most of these capabilities, except for automated code compliance checking (ACCC) [2].

Regulations, requirements, and standards administer a building project's lifecycle progress. For any new or extension to an existing project, the design has to go through code compliance requirements verification. This process is manual, time-consuming, and error-prone for local governing authorities in many countries [3]. Past research has been conducted in the domain of ACCC to improve its productivity. However, the currently available automatic code compliance checking systems remain restricted to a limited amount of code requirements, specifically to explicitly defined model attributes or numerical constraints accessible through direct rule-interfacing [4].

Further, with the increase in the industrial acceptance of BIM systems, the requirement for a standard data exchange format emerged. When models are exported from native BIM authoring tools to the open file format of Industry Foundation Class (IFC) [5], according to ISO 16739-1:2018, it loses the associational topological relationships. As a result, generic model review systems are unable to process complex code requirements that involve data processing through aggregations, connections, or physical topological structures.

The topological structure of IFC is illustrated in 'Figure 1'. The 3D object geometries, and their spatial connections are classified as the physical topology. The logical connections between the object classes are termed as the associative topology. Combination of both associative and physical topology creates relational topology of the IFC schema. Nodes represent the physical topology and edges signify the associative topology in 'Figure 1'. IFC models exported from BIM models suffer due to misclassification of object classes or logical connections. Such examples include internal walls tagged as 'IsExternal' [6], 'Skylight' tagged as 'Window' despite a presence of its predefined type, in the IFC schema. In such instances, the information is implicit in the model, even though the topological relationships are incorrect. These relationships can be re-structured the Semantic enrichment (SE) process.

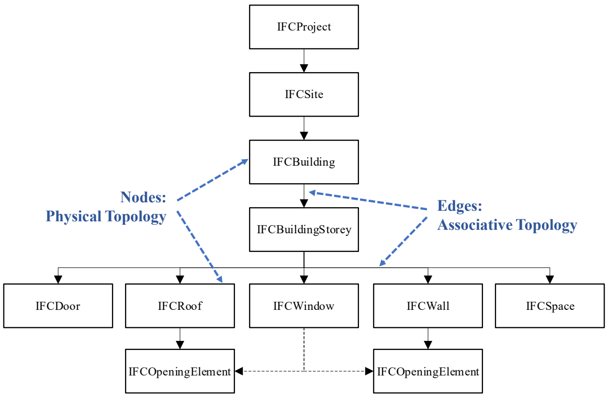


Figure 1. Topological relationships in IFC schema

SE automatically normalizes the implicit information to an explicit representation, thus supplementing BIM models for domain-specific applications [1]. Automatic data augmentation reduces the manual preprocessing effort required by the architects and modelers while submitting the model for verification. SE reinforces the automatic verification process and lays the path for the industry's digital transformation through enhanced transparency and information transfer [7]. Through a test case, this study explores the data enrichment process for compliance checking of topologically complex model elements. An Indian DCR code clause for Maharashtra was used to identify the semantic enrichment tasks and steps involved in the ACCC process.

2 Literature Review

Fenves first ideated the concept of automated rule checking for structural design checks through a logic table in 1966 [8]. 'DesignCheck' brought the next revolution in the ACCC domain by using IFC as a bridging tool between CAD systems [9]. Thereafter IFC started marking its presence in the open-building model schema.

2.1 IFC in ACCC

The object-oriented data schema for building product modeling was first envisioned by Eastman [10] in 1999. Object-oriented modeling (OOM) was established as an essential function for digitally representing building components, their function, and forms. Guiding principles for modeling space and space boundaries were originated by Bjork [11]. Further, these concepts were enriched through the 'RATAS' project model [12] and 'CIMSteel' model for constructional steelwork [13]. As a result of these developments, the thrust towards open building model schema resulted in IFC. The current format addressed in this paper is IFC4, according to ISO 16739-1:2018 [14].

The academic and automated model-checking applications developed chiefly depend on the information extracted from the IFC model. The first significant tool officially used in compliance checking was the Singapore e-CORENET project, with enriched IFC through the FORNAX engine [15]. Table 1 shows a plethora of code compliance checking applications developed subsequently based on the IFC schema.

Table 1. ACCC tools developed based on IFC schema

Name	Input Data Format	Subject of Compliance
CORENET e-PlanCheck [15]	IFC model enriched with FORNAX engine	Fire, water, and energy design for Singapore
LiCA [24]	Process Converted IFC	Water distribution system
Melzner [25]	IFC	Site Safety
RegBIM [26]	RASE-based IFC	UK building regulations
Dimyadi [27]	IFCOwl	New Zealand building code
Bus [28]	IFCOwl	French fire safety and accessibility
Zhong [29]	NLP and IFC	Environmental check
Nawari [30]	IFCXml	Florida building code
Messaoudi [31]	IFC	Permits for the state of Florida
Solibri Model Checker [19]	IFC	Rules for accessibility and intersections
BIMDCR [32]	Enriched IFC	Indian building code and DCR
EDMmodel-Checker [18]	IFC	None

2.2 Requirement for Semantic Enrichment

Automated compliance checking for complex code clauses requires explicit representation of enriched semantic information [16]. The existing automated code-checking applications face challenges in retrieving essential information in its proper representation [17]. Current ACCC applications like EDMmodelchecker [18] and Solibri [19] use hard-coded rule sets to evaluate test values. The explicit requirement of information requires manual preprocessing of data. A spectrum of semantic enrichment for code-checking strategy was proposed by Sacks et al. [20]. Complete enrichment through rule-inferencing is at one end of the spectrum [21], whereas the other end directs toward the application of deep learning. As per Bloch et al., the classification of a pass or fail for each class can be achieved by feeding a complete building model directly to the model-checking deep learning algorithm [4]. However, the idea of replacing the requirement of semantic enrichment entirely through ML is an unexplored territory.

The SE is used in two areas of ACCC, i.e., pre-processing of data before submission of the IFC model for verification, and automatic verification stage. As shown in ‘Figure 1’ currently the pre-processing task are done manually, and model verification is done for a few rules that require explicit information. SE is a multi-layered process requiring ordered steps of concept development, which leads to acquiring desired property values from the information model [22]. Additional steps of classification and geometric calculations are sometimes required to develop the missing concepts in an IFC model, as explained in their study by Bloch et al. [4]. The task sequence depends on the characteristics and structures of the semantic enrichment problems. When missing topological relationships are understood, it is preferable to create a rule-based solution as ML-based solution provide false positive and false negative results. Specially in the case of ML-based verification, the incorrect results can lead to legal and authorization issues. However, where the topology identification is difficult and a pattern exists among the semantic entities, ML-based approaches perform better in such cases.

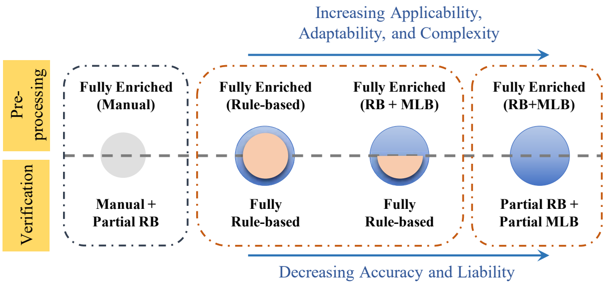


Figure 2. Hybrid approach of SE for ACCC

A test case of topologically complex code requirements is considered in this paper to identify the intertwined relationship between the semantic enrichment task types proposed by Bloch [23]. The application of these tasks is tested on elements misplaced in the IFC Schema. The topological connection restructuring is illustrated through SE tasks on the code requirements for a chajja (sunshade) projection according to an Indian DCR.

3 Chajja Projection Code Requirements

A chajja means a horizontal or sloping structural overhang, usually provided over windows and openings on external walls with the purpose of protecting from sun and rain. In a few cases, it is used for architectural appearance as well. Due to weather conditions, chajja is generally provided in every apartment building in India. According to clause 6.7 (a) of UDCPR, Maharashtra, the external marginal open spaces of a building should be kept free from any erection. Therefore, the maximum permissible width for chajja is 0.75 m over the open space. However, the chajja provided over a balcony is permitted up to balcony projections at a horizontal level [33].

The difficulty in solving this clause requirement arises as the IFC does not have any specific class dedicated to chajja elements. Further, the current modeling practices show the use of slab elements to create chajja projection. As slabs can have any arbitrary shape in a plan view, the width parameter of chajja cannot be explicitly extracted from the model. Hence, several semantic enrichment tasks are executed before the automated code compliance checking.

3.1 Semantic Enrichment Task Types

An automated code-checking process requires understanding all such objects' relations and attributes, which might be used during compliance checking. The IFC format is a standard representation for collecting these properties from a BIM model. The process aims to enrich a model to the extent that a chajja object explicitly contains its width value. This requirement dictates the semantic enrichment steps.

According to Bloch and Sacks [22], semantic enrichment tasks are categorized under broad heads of properties and concepts. The tasks defined for concept development are creation and association, whereas the final properties of an object are extracted through classification and clustering. Among these, the classification task can be addressed through a machine learning (ML)-based approach. Whereas the other three are better solvable through rule inferencing. However, these tasks are not independent and are intertwined for complex code requirements like chajja identification.

As architectural projection does not have any specified class in the IFC, it is created with a slab element. Therefore, the chajja classification of 'IFCSlab' elements is necessary for further calculation tasks. A model consists of numerous slab elements such as floors, stair landings, sunshades, cornices, and lofts. Thus, a slab must satisfy the conditions for chajja as defined in section 3, according to the UDCPR rulebook. However, an 'IFCSlab' element does not contain any inverse relationship between connected walls and associated windows in those walls. For classification through ML, a feature vector must be generated for every slab element. Among these features, two key features are the proximity relationship of the slab with the nearby window and the slab should be placed in an external space. Due to these features' absence, an association task is required to develop the relationship between 'IFCSlab' and 'IFCWindow'. Further, the association task was conducted depending upon the creation of an abstract bounding box element. The steps followed to ensure a successful enrichment of the test case for code requirements are as follows –

1. Identify all windows with precisely one related space, and mark those as external windows.
2. Create a bounding box around the window and create an upwards and outwards buffer to filter 'IFCSlab' elements intersecting with the developed bounding box.
3. Mark the slab element filtered as chajja if it is not at the structural slab level and is above the window.
4. Calculate the max perpendicular projection of the chajja from the associated wall.

Finally, the test value acquired for each chajja is checked against the permissible limit in the compliance checking step. The semantic enrichment stages from this case are explained in detail in the following sections.

3.1.1 Association Stage

In this stage, the windows of the model are filtered, and the association of the window with related spaces is checked. If more than one case of space boundary is found, the window is tagged as an external window. However, IFC4 needs to be created from the model with second-level space boundaries for this relationship to work properly. Second-level space boundaries are generated by walls, doors, windows, and slabs associated with an abstract entity, i.e., space. 'Figure 3' illustrates an example of an external window associated with a single space boundary. Having more than one space boundary associated with a window means there is room on both sides of the room. This case is only possible for internal windows.

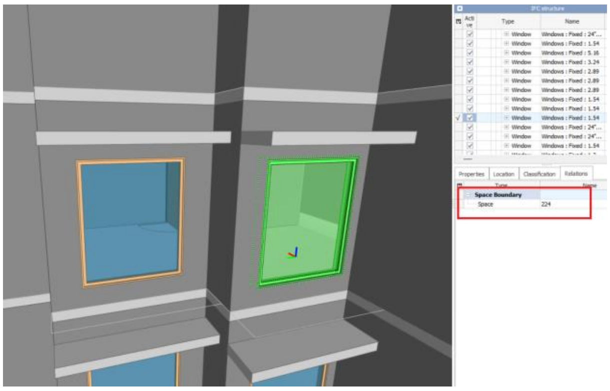


Figure 3. Example of an external window associated with a single Space Boundary

3.1.2 Creation and Classification Stage

Once the external windows are identified, a bounding box for each window is generated. According to the definition, a chajja should be placed above the window. Hence, a buffered bounding box is created, with upwards and outward extensions of 0.5 m. Next, any slab elements are filtered that intersect with the buffered bounding box. The subsequent test checks whether the slab bottom elevation is higher than the window top elevation.

Further check ensures that these slabs are not on the structural floor level. If a chajja is designed at the structural floor level, it is considered a slab projection or cornice, not a sunshade. Once a slab passes all the checks, it is classified as a chajja. 'Figure 4' depicts a buffered bounding box created around an external window and an 'IFCSlab' element (marked in red) intersecting with the bounding box.

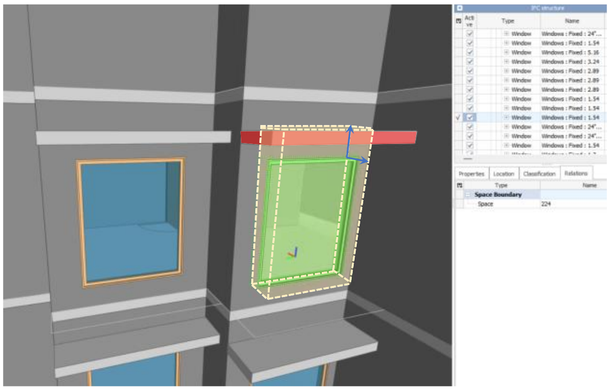


Figure 4. Creation of a buffered bounding box to identify any overlapping slab element in the box

3.1.3 Calculation Stage

The next step is to calculate the width of the chajja projection. First, the host wall of a window is identified for this computational task. Next, the shape representation of the chajja element is checked. If the chajja is rectangular, the X and Y dimensions of the slab are extracted. The dimension perpendicular to the host wall line is considered the chajja width. On the other hand, if the slab is L-shaped or of any arbitrary polygon shape, the edge line properties of the slab polygon are extracted. The straight-line properties slope (m) and intercept (c) are generated. Edge lines that are nearest and parallel to the wall line are considered the baseline. For parallelity check, the slope' m' of both lines is compared, and the difference between the intercept 'c' of the lines is used for finding the nearest distance. 'Figure 5' highlights the associated wall line and the nearest parallel chajja edge considered as a baseline for width computations. The shortest distance of other parallel edges from the baseline is calculated in the next step. The maximum value acquired among the perpendicular distances calculated is returned as the maximum width of the chajja. 'Figure 6' illustrates the parallel dimensions computed from the baseline for width calculation.

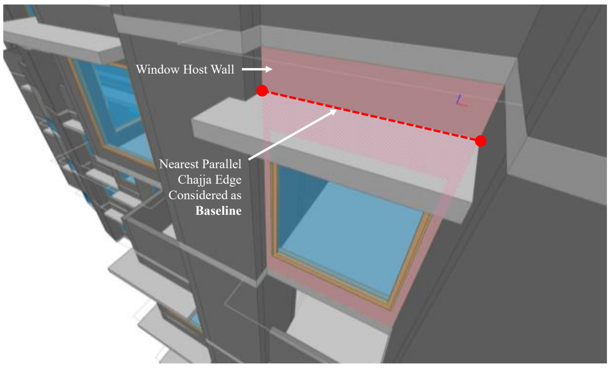


Figure 5. Host identification and Chajja baseline marking for perpendicular distance calculation

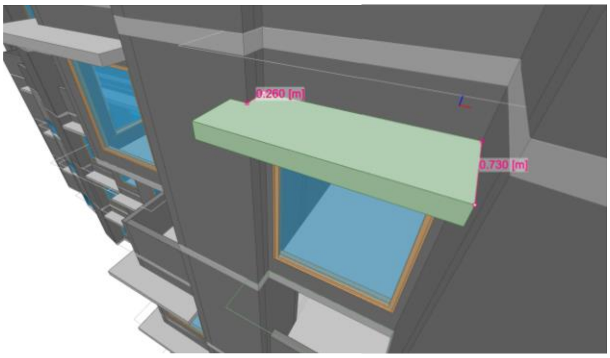


Figure 6. Shortest distance computation between chajja edges. The maximum computed value is returned as the optimal result.

3.2 Code Compliance Checking

As the semantic enrichment task is conducted to contain a single explicit width value of the chajja projection, a single rule check is conducted to check if the projection is greater than 0.75 m. The current result is returned in the form of a dictionary that contains 'Pass' or 'Fail' outcomes under the unique GUID (Global ID) of chajja elements as the key values.

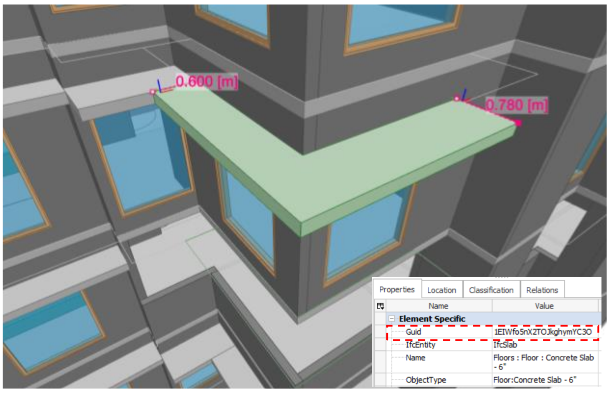
```
IF
    object X is a IFCSlab AND
    object X tag = “Chajja” AND
    perpendicular dist. P for object X ≤ 0.75
THEN
    object X flag = “Pass”
ELSE
    object X flag = “Fail”
```

4 Test Implementation

The semantic enrichment and compliance checking process was applied to a model of an existing residential building project in Mumbai to validate the correctness of outcomes. As the model was taken from a real-life project, all the chajja widths were expected to meet the code clause. Hence, a few changes were introduced in the model to simulate cases where the compliance check will fail. The modified building contained seven floors with seventy chajja slabs. The fourth floor of the building consisted of a refuge area, leading to only six windows on the specified floor. The model was exported as an IFC4 design transfer view for further processing. 'Figure 7' shows an overview of the sample building model in an IFC viewer software.

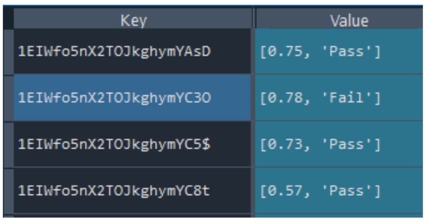
The code compliance check was conducted on a python-based platform. As expected, the code evaluated the chajjas correctly that were compliant with the code requirements. Five L-shaped chajjas were reported as a violation of the 0.75 m maximum permissible width. 'Figure 8' illustrates the case where one dimension (0.6 m) of the chajja satisfies the requirements. However, the check fails due to higher width (0.78 m) along another direction. A compliance report view with "Pass" and "Fail" results, along with the reported maximum width of chajja, is demonstrated in 'Figure 9'.

Figure 7. Perspective view of developed model



Name	Value
Element Specific	
Guid	1EIWfo5nX2TOJkghymYC3O
IfcEntity	IfcSlab
Name	Floors : Floor : Concrete Slab - 6"
ObjectType	Floor:Concrete Slab - 6"

Figure 8. A view of an L-shaped chajja that violated code requirements in one direction



Key	Value
1EIWfo5nX2TOJkghymYAsD	[0.75, 'Pass']
1EIWfo5nX2TOJkghymYC3O	[0.78, 'Fail']
1EIWfo5nX2TOJkghymYC5$	[0.73, 'Pass']
1EIWfo5nX2TOJkghymYC8t	[0.57, 'Pass']

Figure 9. A view of the compliance report window with "Pass" and "Fail" tags

A similar approach can be adopted for identification of external walls, and dead walls on the building depending on the semantic relationships among IFC classes. The location of walls, connectivity to habitable and non-habitable spaces, window and door locations are correct in such an example. However, the due to absence of correct logical connections in IFC schema, information required for rules related to such objects cannot be directly extracted. Similar SE steps can be used in combinations to restructure the relational topology.

5 Discussion and Conclusion

In this study, semantic enrichment tasks were performed through a set of rules performed in sequence for checking code compliance requirements of chajja projections. This task required the evaluation of topologically complex model elements. The task stages involved identifying objects with no pre-defined IFC Class and performing geometrical calculations on these objects. The challenge faced while tackling this problem is to generate the associative topology among building objects. In BIM, the model consists of relational topologies among building elements. Relational topologies consist of associative topology and physical topology. Physical topology refers to the physical layout of the building, including the arrangement of its components and systems, focusing on the position and location of elements in the physical space. Associative topology refers to how different objects or elements in a building model are related or connected. The physical layout and logical connections together produce logical relationships, which is the essence of the relational topology. Even though there are relationships like walls terminating below a structural slab in BIM authoring tools (Revit 2023), the intelligent constraints are not transferred while exported as an IFC file. As a result, the IFC file lacks enrichment beyond the physical topology of the objects. Hence, an automated semantic enrichment was required to replace the necessity of manual preprocessing of data.

The semantic enrichment steps are composed of a progressive chain of rule-based calculations. The entire cycle of rules works on every element of the models that satisfy the eligibility criteria. The object classification task can be solved through an ML-based approach. However, notice that the classification task of Chajja identification was performed solely through rule inferencing in this example. The ML-based approach was not used as the relationship between wall and slab objects is missing in the IFC environment. Once the associational relationship data is established, the chajja classification task is reduced to simple IF-THEN rule evaluation. However, a challenge might arise during identifying chajja through the 'IFCSlab' method. As no standardized BIM modeling practice is imposed in the industry, the feasibility of purely rule-based classification can raise questions [22]. In the cases of architectural awning sunshades, the objects can be modeled through the generic model-in-place function of Revit. Thus, this work's future scope includes checking the machine learning model's capabilities for dealing with diverse object compositions in the case of chajja classification.

Once the classification task is completed, the code requirement values for width are generated through geometric computational methods. The final rule evaluation was conducted for a single test case value generated through semantic enrichment. This result emphasizes the importance of mixed multi-method enrichment processes on IFC models exported from BIM. Through the combination of creation and association (concepts) alongside classification and computational (properties) methods, implicit model data can be expressed as a single explicit data point for enhanced machine readability.

The concept and property type tasks are performed in alternate sequences to achieve the final attribute data. Defining concepts can be classified as a high-level task, as it converts implicit information to explicit data through relational topologies. Once the relations are established, property tasks generate the final single test value for verification. In complex code requirements, property tasks might be required first to generate an associative relationship between the undefined class of objects in the IFC schema. However, as we progress toward the "no semantic enrichment" end of the spectrum [20], the alternate semantic task sequences can be merged and skipped through the application of ML algorithms. In a recent research on room type classification, the classification task of occupancy type identification and association type task of apartment room clustering was performed in a single step through the graph neural networks [34].

The adaptation of SE can be through both rule-based and ML-based approaches. However, the rule-based approach illustrated in the paper may face limitations due to complex relationship among building elements. When the required relationships are not defined in the DCR rulebooks, and rules are subjected to reviewer's discretion, in such scenarios ML-based SE approaches may perform better. It is important to understand the optimum complexity level of the relational restructuring required, to decide on the most suitable approach. Future work in this domain will focus on understanding the advancements in the relationships of semantic task types layers through ML applications. Also, a research work is under progress to identify the level of 'relational restructuring complexity' for application of rule-based and ML-based methods. These levels will inform the concerned authority (ULBs) about the trade-off to consider between adaptability and accuracy of the ACCC systems. Finally, from a comprehensive stakeholder's perspective, the adaptation of AI/ML-based semantic enrichment removes the additional efforts required from the end users, leading to an organic growth of Automatic code compliance checking in the industry.

References

[1] M. Belsky, R. Sacks, and I. Brilakis, "Semantic Enrichment for Building Information Modeling," *Comput. Civ. Infrastruct. Eng.*, vol. 31, no. 4, pp. 261–274, Apr. 2016, doi: 10.1111/mice.12128.

[2] R. Sacks, C. Eastman, G. Lee, and P. Teicholz, *BIM Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers*, 3rd ed. John Wiley & Sons, 2018.

[3] W. Solihin and C. Eastman, "Classification of rules for automated BIM rule checking development," *Autom. Constr.*, vol. 53, pp. 69–82, May 2015, doi: 10.1016/j.autcon.2015.03.003.

[4] T. Bloch, M. Katz, R. Yosef, and R. Sacks, "Automated Model Checking for Topologically Complex Code Requirements – Security Room Case Study," in *European Conference on Computing in Construction*, 2019, pp. 48–55. doi: 10.35490/EC3.2019.157.

[5] buildingSMART, "Industry Foundation Classes (IFC)." https://www.buildingsmart.org/standards/bsi-standards/industry-foundation-classes/ (accessed Jul. 10, 2022).

[6] H. Ying and S. Lee, "A rule-based system to automatically validate IFC second-level space boundaries for building energy analysis," *Autom. Constr.*, vol. 127, no. August 2020, p. 103724, 2021, doi: 10.1016/j.autcon.2021.103724.

[7] A. Karmakar and V. S. K. Delhi, "Construction 4.0: what we know and where we are headed?," *J. Inf. Technol. Constr.*, vol. 26, no. July, pp. 526–545, 2021, doi: 10.36680/j.itcon.2021.028.

[8] S. J. Fenves, "Tabular Decision Logic for Structural Design," *J. Struct. Div.*, vol. 92, no. 6, pp. 473–490, Dec. 1966, doi: 10.1061/JSDEAG.0001567.

[9] L. Ding, R. Drogemuller, M. Rosenman, D. Marchant, and J. Gero, "Automating code checking for building designs -DesignCheck," *Fac. Eng. - Pap.*, Jan. 2006, Accessed: Mar. 21, 2022. [Online]. Available: https://ro.uow.edu.au/engpapers/4842

[10] C. M. Eastman, *Building Product Models: Computer Environments, Supporting Design and Construction*. Boca Raton: CRC Press, 1999. doi: 10.1201/9781315138671.

[11] B. C. Björk, "A conceptual model of spaces, space boundaries and enclosing structures," *Autom. Constr.*, vol. 1, no. 3, pp. 193–214, Dec. 1992, doi: 10.1016/0926-5805(92)90013-A.

[12] B. Björk, "RATAS ProjectDeveloping an Infrastructure for ComputerIntegrated Construction," *J. Comput. Civ. Eng.*, vol. 8, no. 4, pp. 401–419, Oct. 1994, doi: 10.1061/(ASCE)0887-3801(1994)8:4(401).

[13] A. J. Crowley and A. S. Watson, "Representing Engineering Information for Constructional Steelwork," *Comput. Civ. Infrastruct. Eng.*, vol. 12, no. 1, pp. 69–81, Jan. 1997, doi: 10.1111/0885-9507.00047.

[14] ISO, *ISO 16739-1:2018 Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries — Part 1: Data schema*. 2018.

[15] B. H. Goh, "E-Government for Construction: The Case of Singapore's CORENET Project,"

IFIP Adv. Inf. Commun. Technol., vol. 254 VOLUME 1, pp. 327–336, 2007, doi: 10.1007/978-0-387-75902-9_34.

[16] W. Solihin, N. Shaikh, X. Rong, and L. K. Poh, “Beyond Interoperatibility of Building Model: A Case for Code Compliance,” in *Proceedings of the BP-CAD Workshop*, 2004, no. July.

[17] C. Preidel and A. Borrmann, “Automated code compliance checking based on a visual language and building information modeling,” *32nd Int. Symp. Autom. Robot. Constr. Min. Connect. to Futur. Proc.*, 2015, doi: 10.22260/isarc2015/0033.

[18] Jotne, “Jotne IT.” https://jotneit.no/ (accessed May 13, 2022).

[19] “Solibri | Solibri Office.” https://www.solibri.com/solibri-office (accessed Mar. 23, 2022).

[20] R. Sacks, T. Bloch, M. Katz, and R. Yosef, “Automating Design Review with Intelligence and BIM: State of the Art and Research Framework,” *Comput. Civ. Eng.*, no. Mvd, pp. 353–360, 2019.

[21] C. Eastman, J. min Lee, Y. suk Jeong, and J. kook Lee, “Automatic rule-based checking of building designs,” *Automation in Construction*, vol. 18, no. 8. Elsevier, pp. 1011–1033, Dec. 01, 2009. doi: 10.1016/j.autcon.2009.07.002.

[22] T. Bloch and R. Sacks, “Clustering Information Types for Semantic Enrichment of Building Information Models to Support Automated Code Compliance Checking,” *J. Comput. Civ. Eng.*, vol. 34, no. 6, pp. 1–11, Nov. 2020, doi: 10.1061/(ASCE)CP.1943-5487.0000922.

[23] T. Bloch, “An Evaluation of Computational Approaches for Semantic Enrichment of BIM models,” Technion - Israel Institute of Technology, 2020.

[24] J. P. Martins and A. Monteiro, “LicA: A BIM based automated code-checking application for water distribution systems,” *Autom. Constr.*, vol. 29, pp. 12–23, Jan. 2013, doi: 10.1016/j.autcon.2012.08.008.

[25] J. Melzner *et al.*, “A case study on automated safety compliance checking to assist fall protection design and planning in building information models,” *https://doi.org/10.1080/01446193.2013.780662*, vol. 31, no. 6, pp. 661–674, Jun. 2013, doi: 10.1080/01446193.2013.780662.

[26] T. H. Beach, Y. Rezgui, H. Li, and T. Kasim, “A rule-based semantic approach for automated regulatory compliance in the construction sector,” *Expert Syst. Appl.*, vol. 42, no. 12, pp. 5219–5231, Jul. 2015, doi: 10.1016/J.ESWA.2015.02.029.

[27] J. Dimyadi, P. Pauwels, and R. Amor, “Modelling and accessing regulatory knowledge for computer-assisted compliance audit,” *J. Inf. Technol. Constr.*, vol. 21, no. July, pp. 317–336, 2016, [Online]. Available: https://www.itcon.org/2016/21

[28] N. Bus, F. Muhammad, B. Fies, and A. Roxin, “Semantic topological querying for compliance checking,” in *eWork and eBusiness in Architecture, Engineering and Construction*, CRC Press, 2018, pp. 459–464. doi: 10.1201/9780429506215-57.

[29] B. Zhong, C. Gan, H. Luo, and X. Xing, “Ontology-based framework for building environmental monitoring and compliance checking under BIM environment,” *Build. Environ.*, vol. 141, no. December 2017, pp. 127–142, 2018, doi: 10.1016/j.buildenv.2018.05.046.

[30] N. O. Nawari, “A Generalized Adaptive Framework (GAF) for Automating Code Compliance Checking,” *Build. 2019, Vol. 9, Page 86*, vol. 9, no. 4, p. 86, Apr. 2019, doi: 10.3390/BUILDINGS9040086.

[31] M. Messaoudi and N. O. Nawari, “BIM-based Virtual Permitting Framework (VPF) for post-disaster recovery and rebuilding in the state of Florida,” *Int. J. Disaster Risk Reduct.*, vol. 42, p. 101349, Jan. 2020, doi: 10.1016/J.IJDRR.2019.101349.

[32] SoftTech Engineers Limited, “BIMDCR.” https://softtech-engr.com/bimdcr/ (accessed May 13, 2022).

[33] Government of Maharashtra Urban Development Department, *Unified Development Control and Promotion Regulations for Maharashtra State*. 2020.

[34] Z. Wang, R. Sacks, and T. Yeung, “Exploring graph neural networks for semantic enrichment: Room type classification,” *Autom. Constr.*, vol. 134, no. October 2021, p. 104039, Feb. 2022, doi: 10.1016/j.autcon.2021.104039.