

AI for Regulatory and Code Compliance

Sanjib Singha

M.Tech Civil Final Year,

Techno India University, West Bengal

Dr. Heleena Sengupta

Professor & HOD - Department of Civil Engineering

Techno India University, West Bengal

Abstract: Manual code compliance is an inefficient process vulnerable to human error. This paper addresses the challenge of systemic automation by introducing a two-stage NLP framework that processes entire building codes rather than fragmented sections. Through semantic hierarchy mapping and ontological rule generation, the proposed system translates natural language into verifiable logic for BIM integration. Comparative analysis indicates that this framework outperforms existing rule-based systems in handling complex cross-references and regulatory exceptions. This research provides architects and engineers with a transparent, trustworthy mechanism for automated decision-making, establishing a scalable foundation for the next generation of digital building regulations.

Keywords: Natural language processing, BIM, Automated code compliance.

1. INTRODUCTION

AI for Regulatory and Code Compliance refers to the application of artificial intelligence technologies—such as machine learning, natural language processing (NLP), and rule-based systems—to support, automate, and enhance compliance with laws, regulations, standards, and technical codes across industries. Traditional compliance processes are often manual, time-consuming, and prone to human error due to the growing volume, complexity, and frequent updates of regulatory requirements.

AI-based compliance systems can automatically interpret regulatory texts, map them to organizational policies or technical codes, and monitor activities or designs for potential violations in real time. For example, NLP enables AI to analyze legal and regulatory documents, extract relevant obligations, and track changes in regulations, while machine learning models can detect non-compliance patterns, predict risks, and recommend corrective actions. In domains such as finance, healthcare, construction, and software engineering, AI is increasingly used for automated audits, code compliance checking, risk assessment, and reporting.

Overall, AI for regulatory and code compliance improves efficiency, accuracy, and consistency, reduces compliance costs, and supports proactive risk management. However, challenges such as explainability, data quality, regulatory trust, and ethical considerations remain critical areas for ongoing research and development.

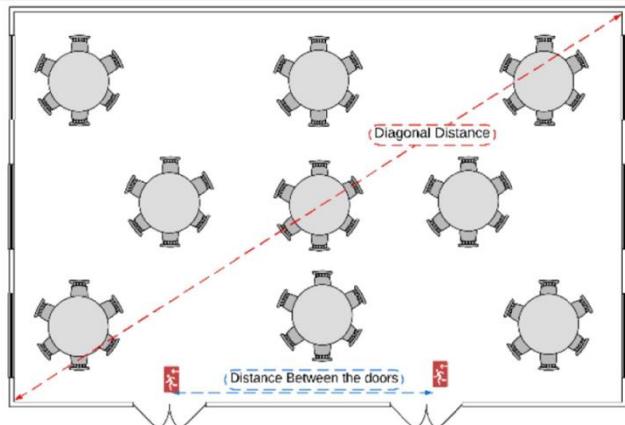
2. STIMULATING CASE STUDY

Building codes are written to be flexible so they can be applied to many different building designs and situations. Their main goal is to ensure safety by setting minimum requirements, while still allowing designers to adapt the rules to specific project conditions. Because of this flexibility, building codes often contain exceptions and conditions that can make interpretation complex.

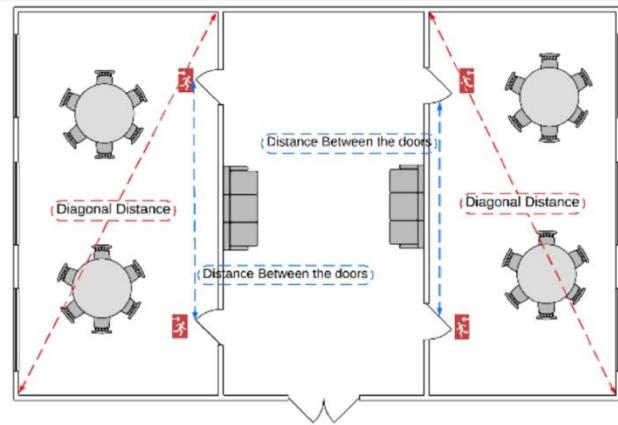
For example, when checking the required distance between two exits, the code states that exits must be separated by at least half of the room's maximum diagonal distance. However, this rule is not located in just one place in the code. Additional sections include exceptions and special conditions that the designer must also consider. Some exceptions may apply to a specific design, while others may not, requiring professional judgment to decide what is relevant.

Under the 2021 International Building Code (IBC), every space must have at least two means of egress, such as doors, stairs, or ramps. These exits must be properly separated. In general, the distance between exits must be at least half of the longest diagonal of the space. If the building has an automatic sprinkler system, the code allows an exception, reducing the required distance to one-third of the diagonal length. Special rules also apply to stairs and ramps, including cases where stairs are connected or where exits are on different levels.

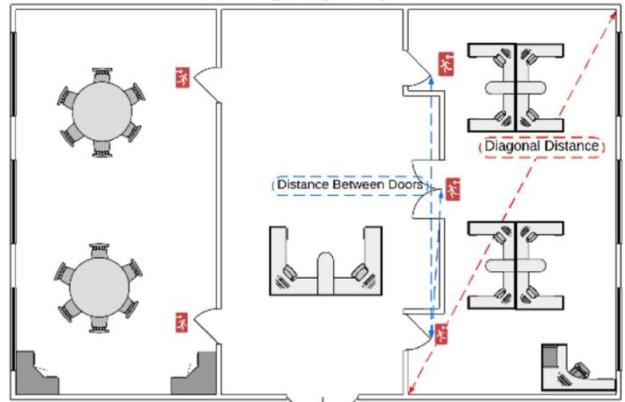
These requirements can lead to multiple design scenarios, where the same code rule is interpreted differently depending on layout, building systems, and exceptions. This demonstrates how building code compliance often involves nuanced judgment rather than simple rule checking.



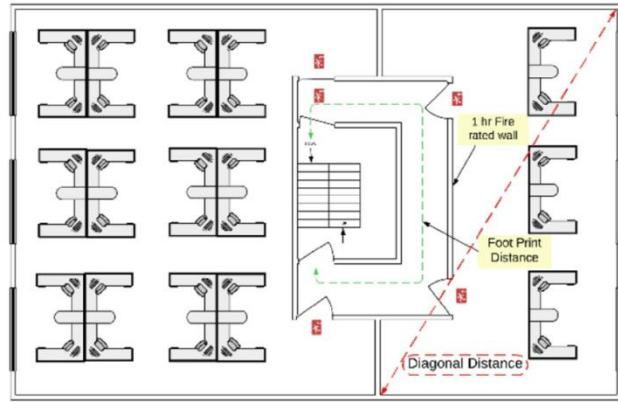
(a) Single-space floor



(b) Multiple spaces floor



(c) More than two exits



(d) Exits are connected to the corridor

This figure shows that:

(a) Single-space floor

- A large open area with multiple exits.
- Distance is measured diagonally from the farthest point to the nearest exit.
- Code intent: occupants anywhere in the space can reach an exit within allowed limits.

(b) Multiple-space floor

- The floor is divided into rooms or compartments.
- Distance is measured through doors and circulation paths, not straight-line.
- Codes require checking:
 - Travel distance to exits
 - Distance between exit doors

(c) More than two exits

- When more than two exits are present, codes require:
 - Minimum separation between exits
 - Proper distribution
- AI systems check whether exits are too close or violate redundancy rules.

(d) Exits connected to a corridor

- Occupants must travel:
 - From workspace → corridor → exit
- Measurements include:
 - Footprint distance (actual walking path)
 - Diagonal distance
- Common in offices, hospitals, and schools.

3. APPROACH

The work proposes a semantic, space-centered framework to support designers in interpreting building codes during the design stage. By ontologically conceptualizing spaces and their relationships to building components, the framework enables human-like reasoning over complex, nested code requirements. Spaces serve as the core abstraction through which building components, user experience, and regulatory constraints are integrated. This approach allows high-level inference, efficient information retrieval, and early evaluation of code compliance within BIM-based ACC systems. By embedding compliance checking into the design process, the framework helps identify issues early, reduce review time, and streamline regulatory approval while preserving designers' functional and aesthetic intent.

4. LITERATURE REVIEW

A collective of researchers has explored the extraction of knowledge from normative codes and standards for automated code compliance (ACC) using machine learning (ML) and natural language processing (NLP) techniques. With due respect I studied all their paper and find method, contribution and limitation. Here I shortly mention those things:

5. METHODOLOGY

1. Zheng, et al. :

Title: A Text Classification-Based Approach for Evaluating and Enhancing the Machine Interpretability of Building Codes.
Method: A dataset and utilized pretrained language models to classify text based on criteria for clause classification.
Contribution: An improved text classification and interpretability of Chinese building codes, and a labeled dataset for future research.
Limitation: Only one-third of clauses could be interpreted automatically, and the findings may not apply beyond Chinese building codes.

2. Zhang and El-Gohary:

Title: Transformer-based approach for automated context-aware IFC-regulation semantic information alignment.
Method: A deep learning approach for aligning IFC and building regulations.
Contribution: An effective deep learning method, emphasizing domain-specific adaptation and leveraging contextual information.
Limitation: Limited refined classification and alignment methods and integrating with other ACC modules.

3. Wu, et al:

Title: Invariant signature, logic reasoning, and semantic natural language processing (NLP)-Based Automated Building Code Compliance Checking (I-SNACC).
Method: A framework that integrates computing modules for ACC prototype system.
Contribution: New framework for with high-precision ACC prototype, tested on Chapter 10 of the International Building Codes 2015.
Limitation: The framework and system have potential for automation, but manual effort is still required for some tasks due to machine and NLP limitations.

4. Peng and Liu:

Title: Automated code compliance checking research based on BIM and knowledge graph.
Method: ACC using BIM and knowledge graphs, transforming specifications via NLP and mapping rules to produce review reports.
Contribution: A system based on BIM and knowledge graph to improve accuracy and efficiency. It demonstrated feasible through a case study.
Limitation: BIM promotion and data transfer errors can hinder automatic review, along with semantic expansion and algorithm robustness.

5. Li Yuchao:

Title: A Semantic Representation Method of Building Codes Applied to Compliance Checking

Method: Code representation for compliance checking using Semantic Web technology for rule transformation.

Contribution: A code model for automatic rule generation, advancing Semantic Web use in compliance checking.

Limitation: A code model for automatic rule generation, advancing Semantic Web use in compliance checking.

6. Zou, et al :

Title: Investigating the New Zealand Off-Site Manufacturing Industry's Readiness for Automated Compliance Checking.

Method: Survey and interviews assessed NZ OSM industry's ACC awareness and readiness.

Contribution: Developed roadmap for ACC adoption based on survey and expert insights.

Limitation: Small sample size, geographic focus, subjective data, and roadmap implementation challenges.

7. Zhou, et al :

Title: Integrating NLP and context-free grammar for complex rule interpretation toward automated compliance checking.

Method: NLP and CFG for automated rule interpretation.

Contribution: High accuracy, broad application, and dataset publication.

Limitation: Parsing and semantic labeling improvement.

8. Zhang and El-Gohary:

Title: Natural language generation and deep learning for intelligent building codes.

Method: A deep learning-based method for generating intelligent building codes by defining a new semantic representation of requirements and developing a model for requirement sentence segment generation, semantic linking, and requirement configuration.

Contribution: A concept of intelligent building code, offering a solution to bypass error-prone information extraction processes, and achieves high performance in generating natural-language requirements and semantic linking.

Limitation: Future research needed to align the proposed semantic representation with industry standards, improve the deep learning model, and evaluate practical implementation for fully automated compliance checking in the AEC domain.

9. Xue and Zhang:

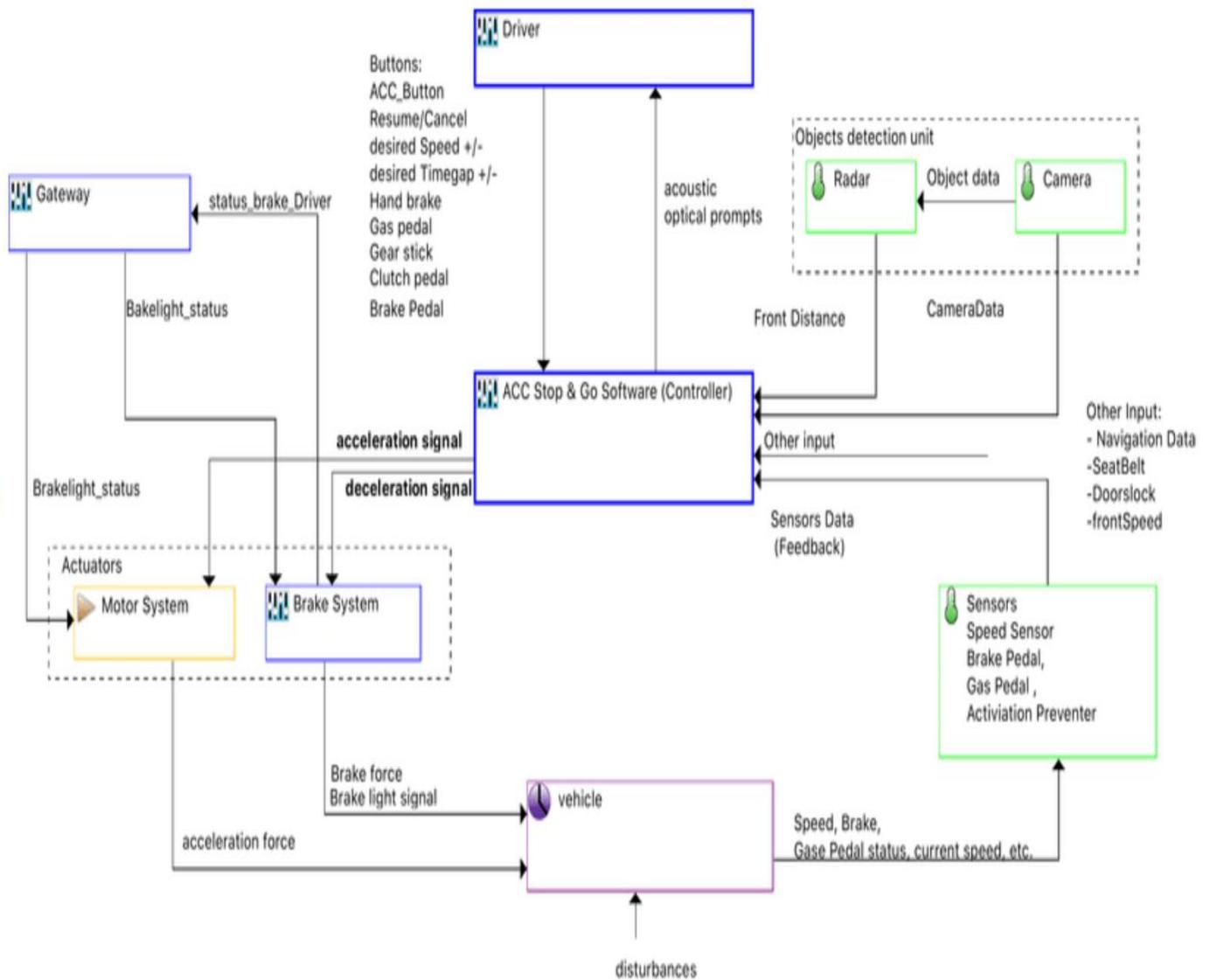
Title: Regulatory information transformation ruleset expansion to support automated building code compliance checking.

Method: A ruleset expansion method that extends checkable code requirements for ACC using iterative pattern matching-based rules.

Contribution: Demonstrates effectiveness, expands application scope, and releases logic clause dataset for further research.

Limitation: Errors attributed to language flexibility and missed patterns; solutions include expanded training data and stricter annotation guidelines.

ACC FRAMEWORK:



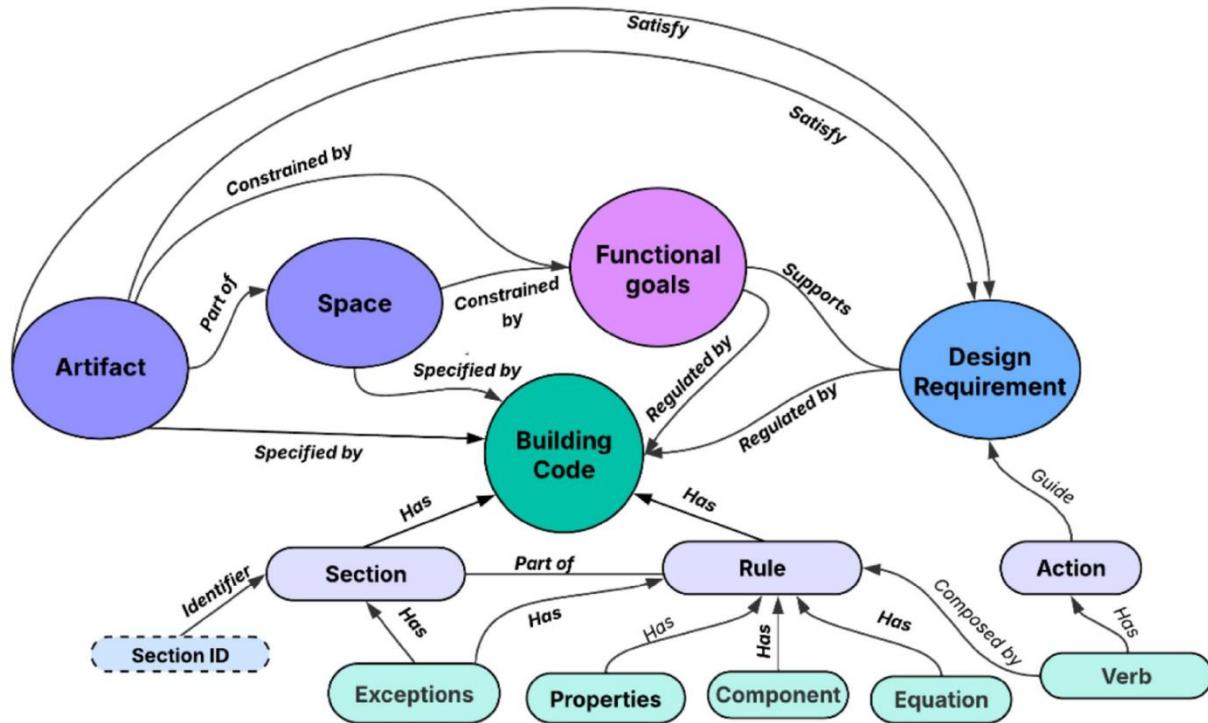
The safety control structure diagram of the ACC system with stop-and-go function

Feature	Manual Method	ACC Framework Involvement
Fire Egress	Manual measurement on 2D plans.	Automated 3D pathfinding & distance calculation.
Accessibility	Visual checks for ramps/clearance.	Automatic flagging of ADA/ISO violations.
Code Updates	Designers must "re-learn" new laws.	New digital rules are uploaded; models are re-scanned instantly.
Structural Integrity	Calculation based on disparate notes.	Cross-references structural codes directly with BIM data.

7.1 Knowledge about structure:

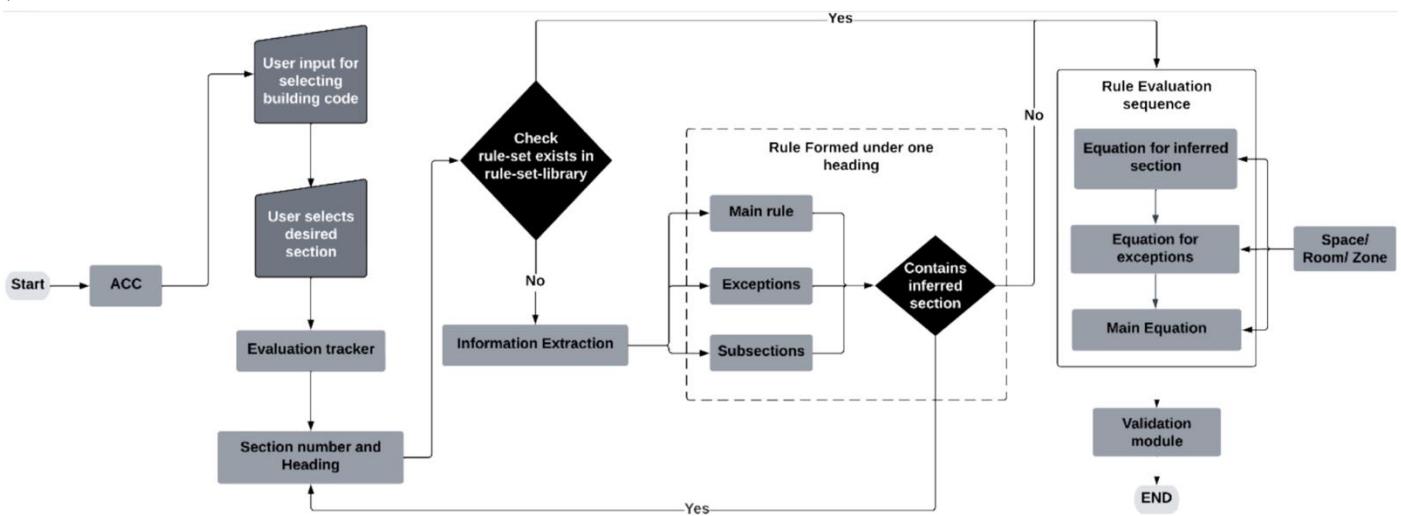
7.1.1 Paragraph boundaries and semantic schemas:

[Section] [Rule] [Component] [Properties] [Action Verb] [Equation] [Summary] [Exceptions] [Inferred Section]



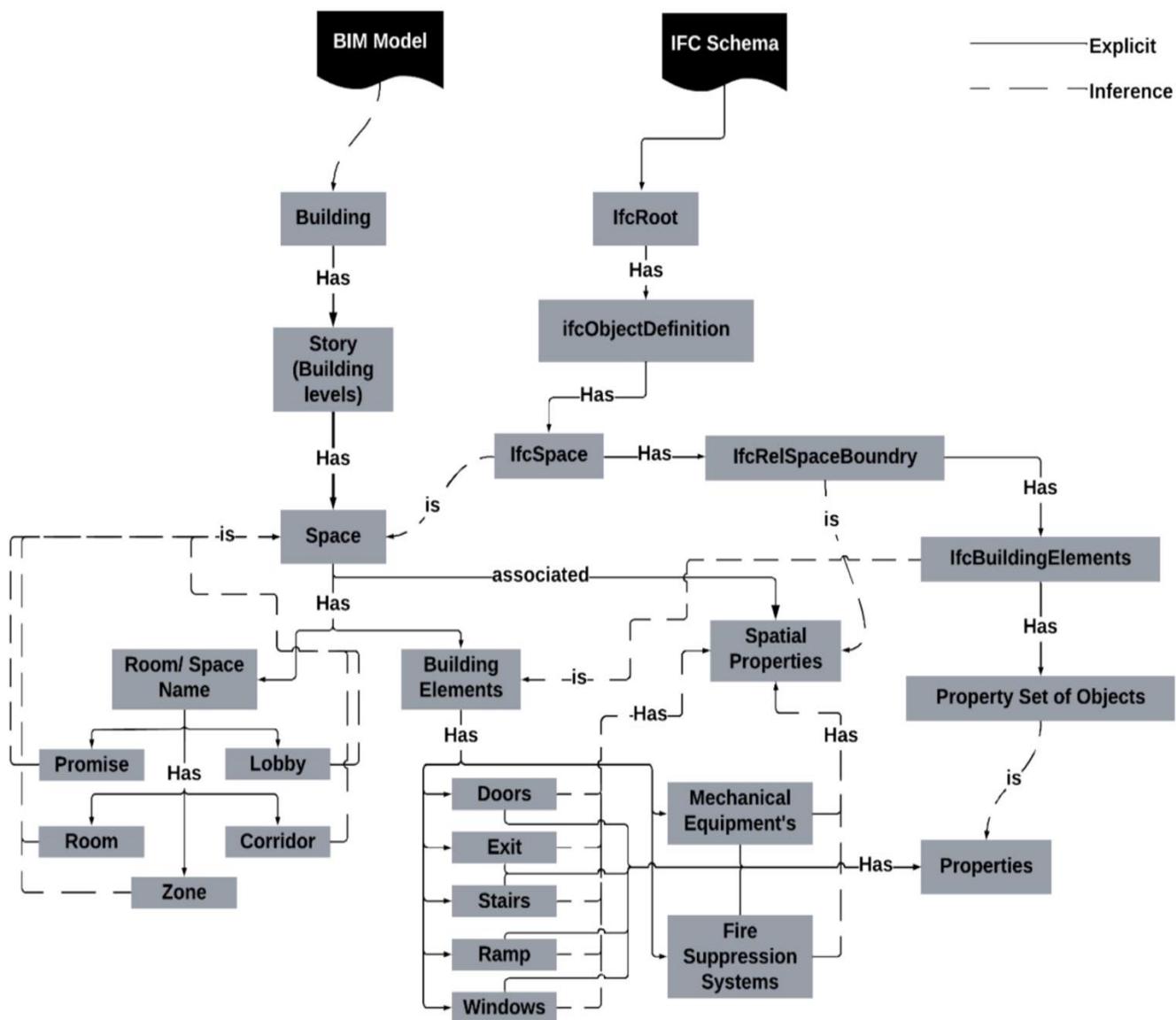
This section presents a user-guided approach to navigating the nested structure of building codes and resolving ambiguities in rule interpretation. Designers initiate compliance checking by selecting code sections from a semantic hierarchy, after which a Breadth-First Search (BFS) scans the text between headings to identify related sections, subsections, and exceptions. BFS systematically captures references such as sections, exceptions, and considerations, ensuring no relevant rules are missed. Identified references expand a hierarchical rule tree, allowing users to track rule creation and approve further semantic expansion when needed. Finally, extracted rules are converted into machine-readable rule sets and stored in CSV format for scalable and efficient computational analysis.

7.1.2 Semantic schema values for rule-set construction:



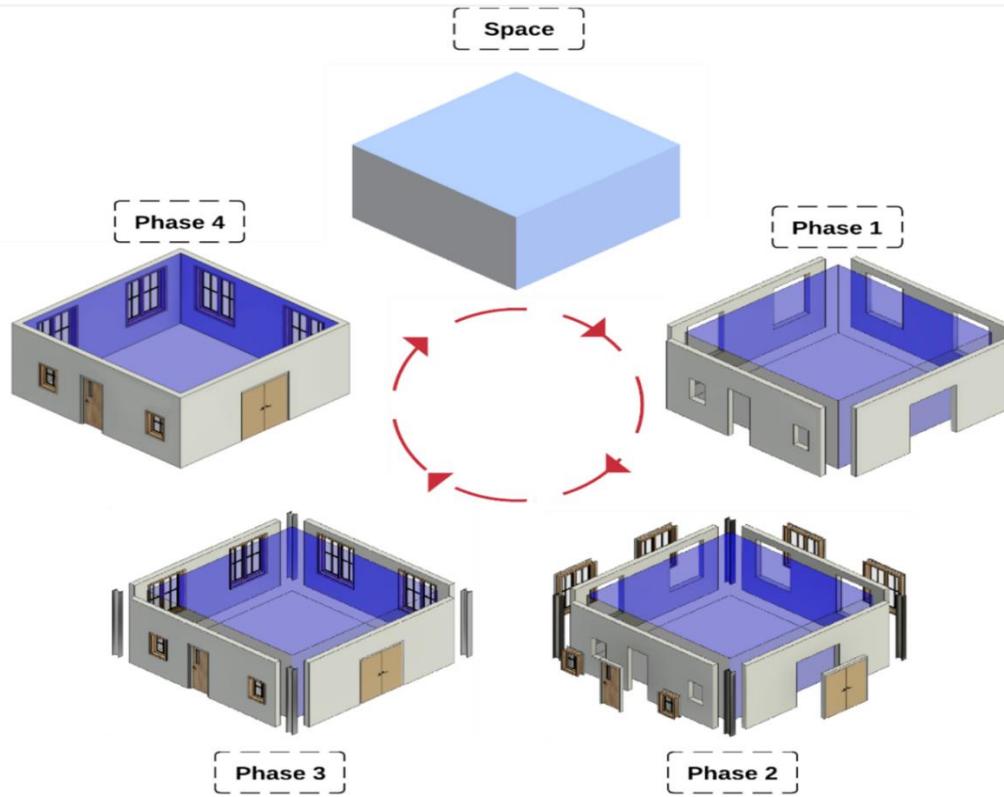
The generated rules formalize building code logic, specifying allowable states, actions, or relationships, and are represented in a semantic schema to make complex constraints explicit for both humans and machines. Rule sets are created by integrating information extraction, text summarization, and inference, structured with rule numbers, components, equations, exceptions, and summaries for clarity and easy editing. An Equation Mapping Dictionary (EMD) links mathematical expressions to their concepts and rules, preventing duplication and maintaining consistency. The taxonomical structure allows users to review, refine, and verify rules efficiently, and the resulting rules can be directly applied for compliance checking within BIM models.

7.1.3 Exceptions and inferred section exploration and tracking:

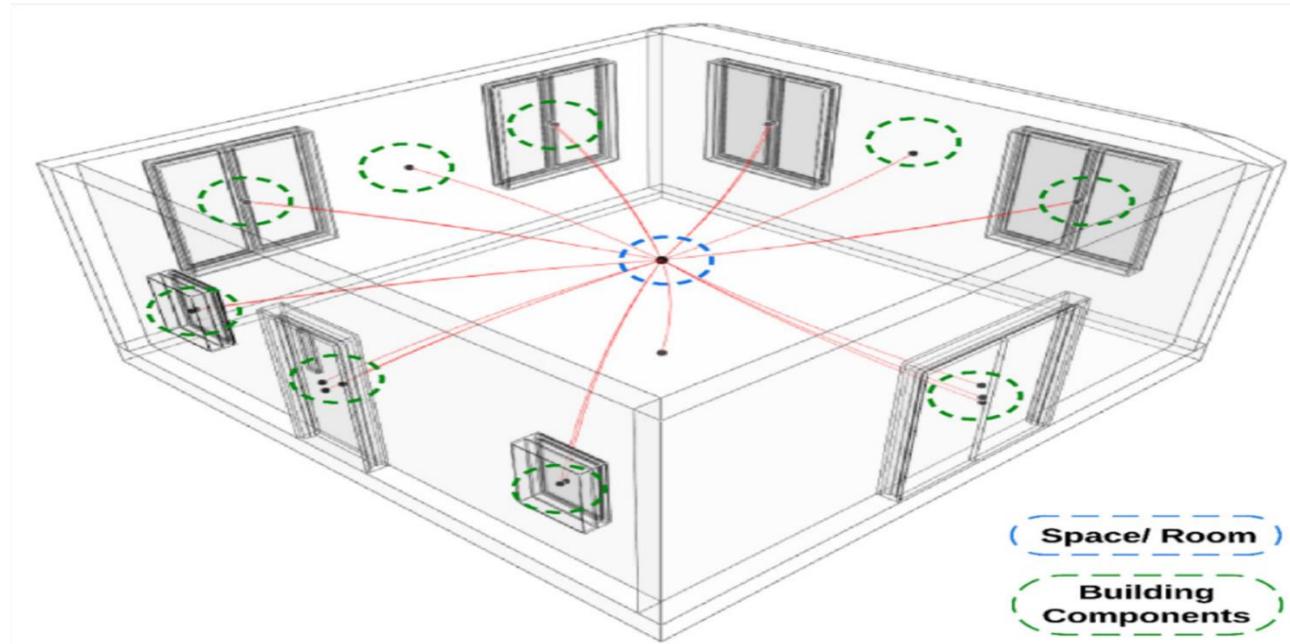


The **IfcSpace** entity represents specific spatial areas within a building, such as rooms, hallways, or other enclosed volumes, and is linked to building storeys for hierarchical organization. It can be refined into partial spaces for granular definitions in complex designs. The **IfcRelSpaceBoundary** connects these spaces to bounding building elements like walls, floors, and ceilings, managing attributes such as area, volume, and occupancy. **IfcBuildingElement** subclasses, including doors, walls, and slabs, define the physical components of a building, supporting structural integrity and functionality. This IFC-based dictionary enables precise spatial and component representation for BIM-based design and analysis.

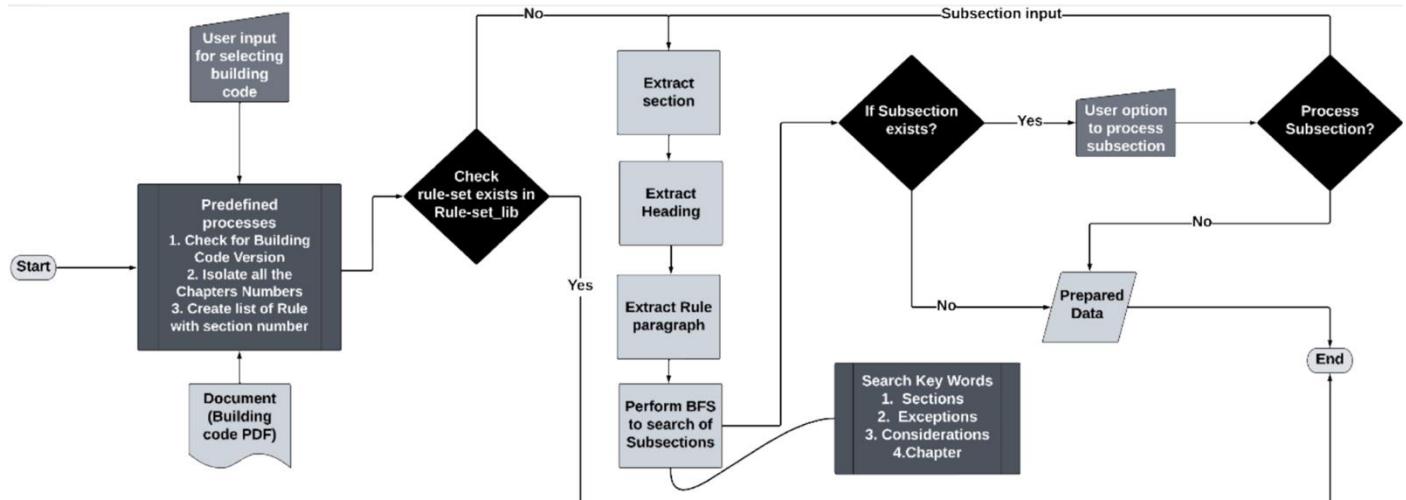
7.1.4 Conceptualization of space framework:



7.1.5 Space and component relationships:



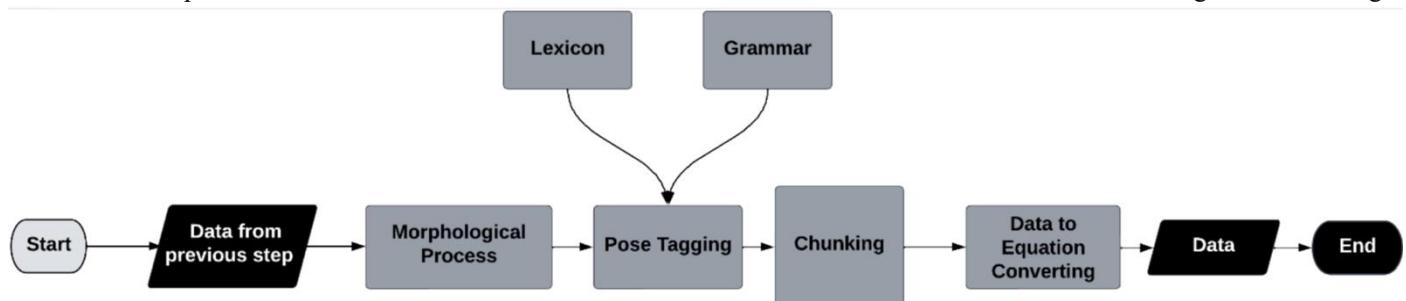
7.2 Information extraction module:



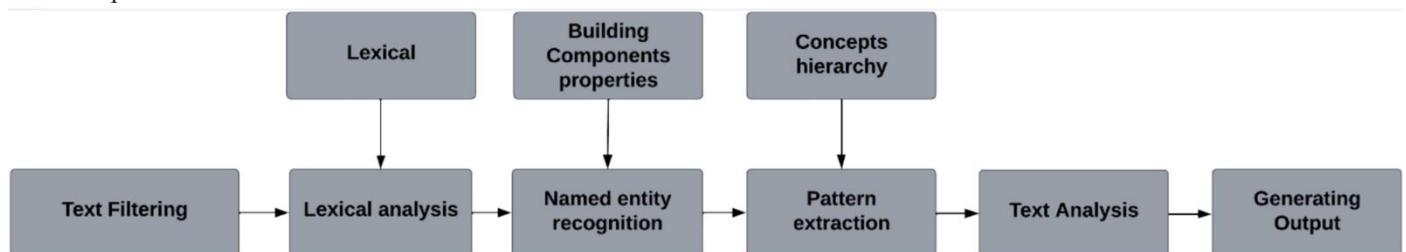
This module is responsible for validating and refining the extracted data by detecting and correcting errors, inconsistencies, and irregularities. It systematically breaks down the build-aligned code document, originally in PDF format, into its core components. NLP and text summarization techniques are applied to improve clarity, scalability, and usability, while standardizing text features such as case sensitivity, formatting styles (e.g., bold or regular), and overall correction status.

7.2.1 Natural language processing: In the context of regulatory and code compliance, Natural Language Processing (NLP) is the subfield of AI that enables computers to "read," "understand," and "convert" human-written laws into digital, machine-executable rules.

Building codes are written for humans—they are full of technical jargon, cross-references, and conditional logic (e.g., "If X is true, then Y must be done, unless Z occurs"). NLP serves as the translator that turns these messy paragraphs into a structured format that a computer can use to check a building design.



NLP compute node data flow:



7.2.2 Rule equation extraction: The system includes a computational module that reads, understands, and simplifies architectural equations. It converts shorthand and complex expressions into a standard format that is easier to analyze and compare. The system consists of four parts: a preprocessor that expands shorthand terms using dictionaries, a parser that breaks equations into components

and properties, an optimizer that standardizes the equation structure, and a calculator that performs required calculations, such as distances between elements.

All equations are converted into a uniform format: [constant] [component] [property] = [constant] [component] [property], which improves clarity and supports automated analysis.

In building design equations: constants are numerical values, components are building elements like doors or rooms, and properties are measurable attributes such as length, area, or diagonal.

Example: This is a code to use room safety

```
def check_room_safety(room):
    """
    room: dictionary containing room properties
    """
    violations = []
    # Minimum room area (example: 9 m2)
    if room["area"] < 9:
        violations.append("Room area is below minimum requirement.")
    # Occupancy density check
    max_occupancy = room["area"] / 2 # 2 m2 per person
    if room["occupants"] > max_occupancy:
        violations.append("Occupancy exceeds safe limit.")
    # Exit requirement
    if room["occupants"] > 50 and room["number_of_exits"] < 2:
        violations.append("At least two exits are required.")
    # Exit separation rule (½ diagonal rule)
    required_separation = 0.5 * room["diagonal"]
    if room["exit_separation"] < required_separation:
        violations.append("Exit separation distance is insufficient.")
```

```
# Ceiling height check

if room["ceiling_height"] < 2.4:
    violations.append("Ceiling height is below minimum safety requirement.")

# Final result

if not violations:
    return "Room is compliant with safety requirements."

else:
    return violations
```

Input:

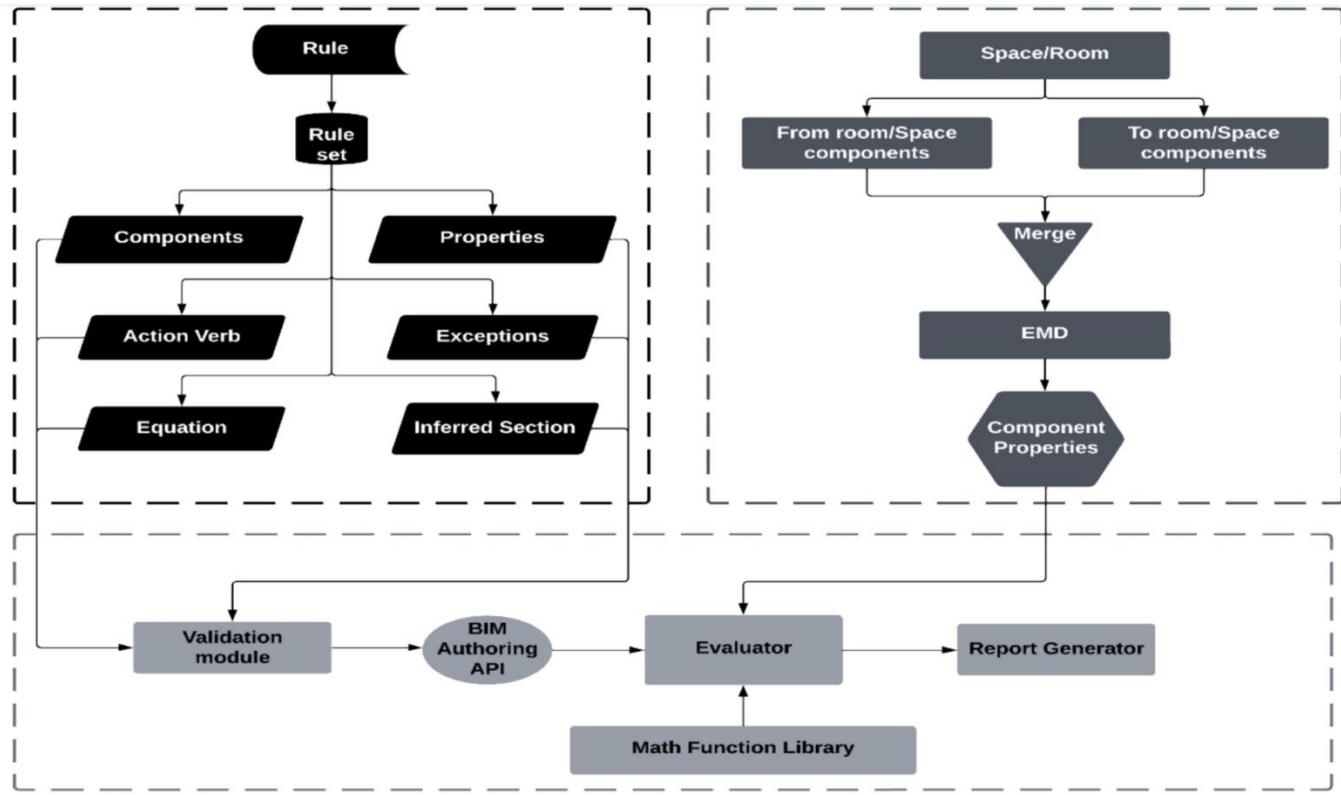
```
room_data = {
    "area": 20,
    "occupants": 12,
    "number_of_exits": 2,
    "diagonal": 7.0,
    "exit_separation": 4.0,
    "ceiling_height": 2.6
}

print(check_room_safety(room_data))
```

7.3 Reasoning and inference: code checking: The Code Checker bridges the Rule Generation components, which include the Data Preparation module, the NLP module, and the rule set library, with the design represented by the BIM model. The Code Checker module carries out two primary functions.

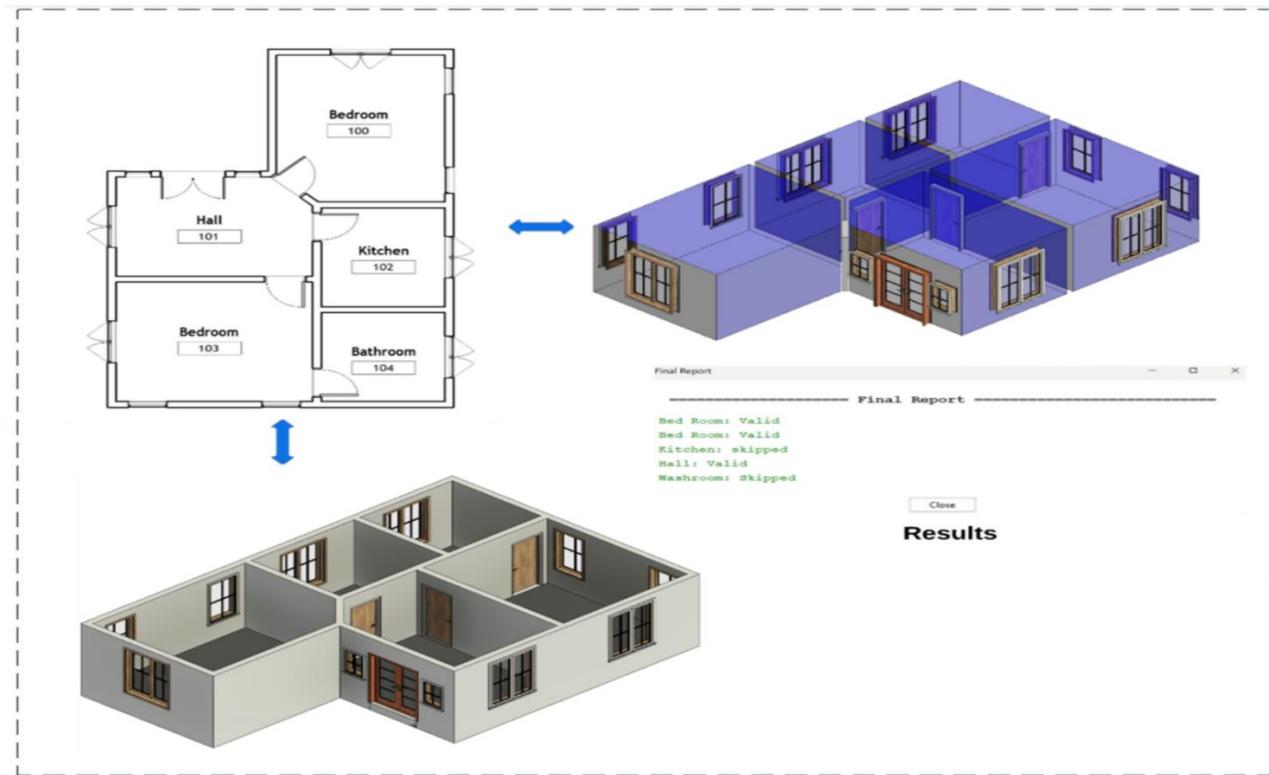
7.3.1 Mapping building components in spatial terms: This function maps all building components based on their spatial dimensions by linking each element to its spatial context and storing relevant properties directly in the BIM model using the EMD. The module first identifies all spaces and organizes them into a hierarchical tree for structured management. It then populates spatial properties such as area, length, width, height, and XYZ coordinates for each space. Using this space hierarchy, the module assigns related building components and their properties accordingly. This space-centered organization ensures consistent, efficient, and structured data acquisition for further analysis and compliance checking.

7.3.2 Code verification and report generation:

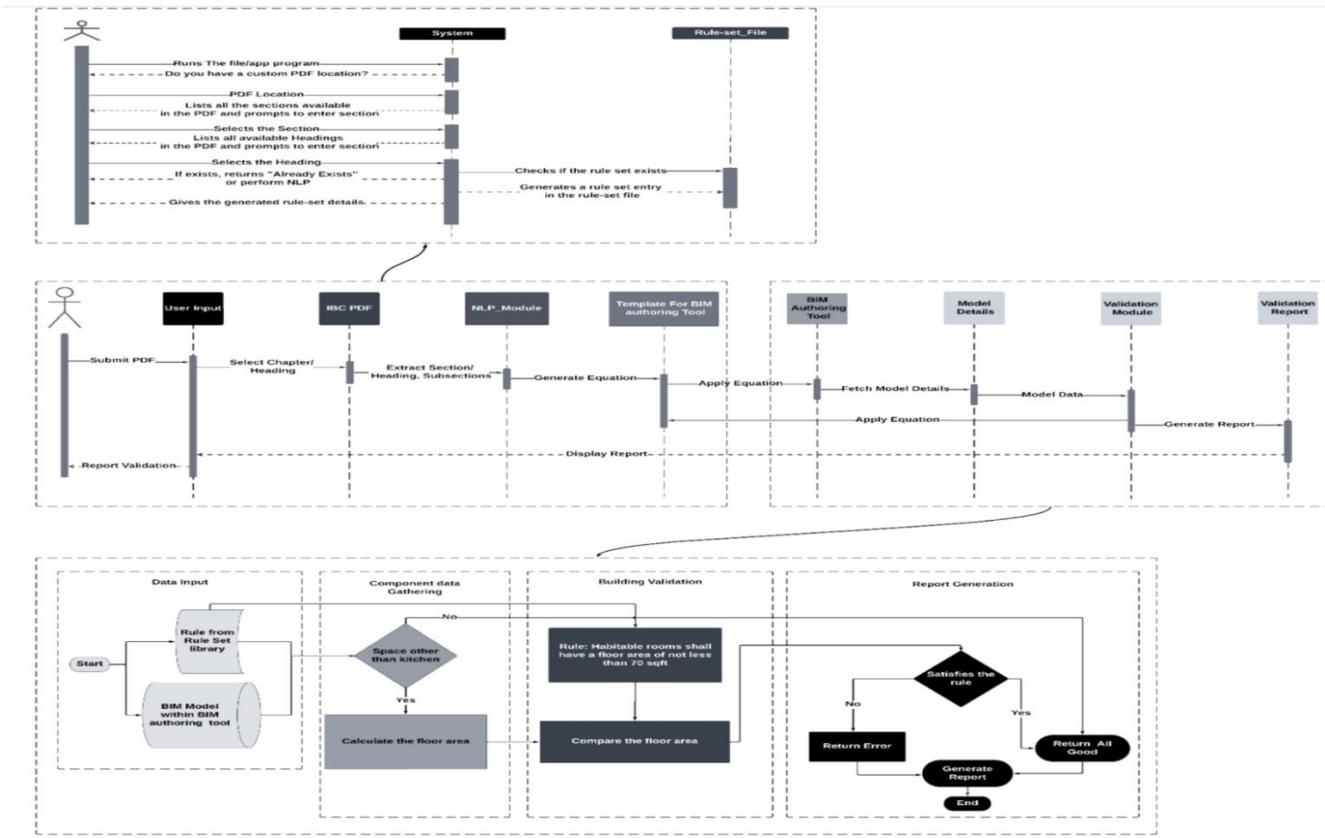


The second function checks the BIM model against predefined building codes and generated rules to identify compliance or violations. The Code Checker module serves as a middle layer that links rule generation with the BIM model, validates components using spatial logic, and reports compliance results for review and action.

6. CASE EXAMPLE:



The case pertains to an examination of a single-story, single-family residence featuring two bedrooms, a hall, and a kitchen, as depicted in The study relies on the 2021 International Residential Code (IRC) as the primary point of reference, which is acknowledged for its comparatively lower intricacy in contrast to the IRC. Section “R304.1 Minimum area” is specifically chosen, stipulating that habitable rooms must encompass a floor area of at least 70 square feet (6.5 m^2).



When ACC is initiated, the user loads the IRC 2021 PDF, after which the system performs a data screening process similar to previous cases and displays all headings with their section numbers. Once the user selects Section R304.1, the ACC framework begins generating a new rule set library for the IRC. Since no inferred sections are identified during the search, the system skips the iterative inference step and directly creates a rule set consisting of the primary rule and its exceptions.

7. CONTRIBUTION TO THIS RESEARCH AREA

This research advances Automated Code Compliance (ACC) by applying knowledge-based reasoning to better interpret complex building regulations. Its key contributions are summarized as follows:

Human-in-the-loop compliance: The framework integrates expert involvement in rule validation and exception handling, increasing trust, accountability, and practical reliability in automated compliance checking.

Generalizable ACC framework: A modular and adaptable ACC architecture is developed that can support different codes and versions without altering core logic, making the system scalable and robust to regulatory changes.

Semantic rule modeling: The study introduces a semantic parsing approach that converts complex, hierarchical code text into clear, machine-readable rules, improving transparency and ease of refinement compared to traditional ACC methods.

8. CONCLUSION

Over the last decade, research has focused on using NLP to convert building codes into machine-readable rules for automated compliance checking. This approach emphasizes rule interpretation with user involvement to handle complex, nested codes, context, and non-quantitative text. It generates clear rule sets that can be applied in BIM, allowing designers to check compliance early and reduce review time. Future work will address tables, figures, and fragmented PDF text to further improve rule accuracy and completeness.

9. REFERENCES:

- [1] Q.Z. Yang, X. Xu Design knowledge modeling and software implementation for building code compliance checking Build. Environ., 39 (6) (2004), pp. 689-698, 10.1016/j.buildenv.2003.12.004
View PDFView articleView in ScopusGoogle Scholar
- [2] D.W. Thornburg, J.R. Henry Exit and exit access doorway configuration 2015 International Building Code Illustrated Handbook (First edition), McGraw-Hill Education (in en), New York (2015) (ISBN: 9781259586125)
Google Scholar
- [3] N. F. P. Association NFPA 101: Life Safety Code 2018 National Fire Protection Association (2017) (ISBN:9781455916832)
Google Scholar
- [4] J. Zhang, N.M. El-Gohary Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking Automat. Construct., 73 (2017), pp. 45-57, 10.1016/j.autcon.2016.08.027
View PDFView articleGoogle Scholar
- [5] E.A. Delis, A. Delis Automatic fire-code checking using expert-system technology J. Comput. Civil Eng., 9 (2) (1995), pp. 141-156, 10.1061/(ASCE)0887-3801(1995)9:2(141)
View at publisherView in ScopusGoogle Scholar
- [6] J. Wu, X. Xue, J. Zhang Invariant signature, logic reasoning, and semantic natural language processing (NLP)-based automated building code compliance checking (I-SNACC) framework J. Inform. Technol. Construct., 28 (2023), pp. 1-18, 10.36680/j.itcon.2023.001
View at publisherGoogle Scholar
- [7] T.H. Beach, Y. Rezgui, H. Li, T. Kasim A rule-based semantic approach for automated regulatory compliance in the construction sector Expert Syst. Appl., 42 (12) (2015), pp. 5219-5231, 10.1016/j.eswa.2015.02.029
View PDFView articleView in ScopusGoogle Scholar
- [8] N.O. Nawari A generalized adaptive framework (GAF) for automating code compliance checking Buildings, 9 (4) (2019), 10.3390/buildings9040086
View at publisherGoogle Scholar
- [9] Y. Ding, J. Ma, X. Luo Applications of natural language processing in construction Autom. Construct., 136 (2022), 10.1016/j.autcon.2022.104169
View at publisherGoogle Scholar
- [10] J. Zhang, N.M. El-Gohary Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking J. Comput. Civ. Eng., 30 (2) (2016), 10.1061/(ASCE)CP.1943-5487.0000346
View at publisherGoogle Scholar
- [11] Y.-C. Zhou, Z. Zheng, J.-R. Lin, X.-Z. Lu Integrating NLP and context-free grammar for complex rule interpretation towards automated compliance checking Comput. Ind., 142 (2022), 10.1016/j.compind.2022.103746
View at publisherGoogle Scholar
- [12] C. Manning, H. Schütze Foundations of Statistical Natural Language Processing MIT press (1999)
(ISBN:0262303795)
Google Scholar
- [13] Author links open overlay panelNishanth Purushotham a, Chethan Kailashnath b, Ivan Mutis a
Show more
<https://doi.org/10.1016/j.autcon.2025.106598>