2026 European Conference on Computing in Construction
Corfu, Greece
July 12-15, 2026

TURNING CONSTRUCTION CHECKLISTS INTO MACHINE-READABLE INSPECTION SCHEMA

Till Weist, Leon Pfenning, and Prof. Sven Rogalski
Darmstadt University of Applied Sciences, Darmstadt, GER

Abstract

Construction verification currently relies on unstructured checklists, hindering integration with autonomous agents. To scalably convert these legacy instructions into machine-readable data, we propose a Hybrid Inspection Pipeline. This system uses LLMs and "Balanced Retrieval-Augmented Generation" to map text into an "Inspection Triad" aligned with IFC and AEC3PO ontologies. Achieving 92.5% semantic precision, the pipeline eliminates hallucinations using a deterministic fidelity loop. Ultimately, this framework provides the executable logic necessary for autonomous inspection agents, establishing a foundation for future construction research.

Introduction

The construction industry is characterized by a digital paradox, whereby buildings are designed in sophisticated three-dimensional (3D) environments yet are verified on-site using tools that have remained unchanged for decades (Nkosi, 2025). Contemporary digitization endeavors prioritize the capture of geometry (BIM) while the Quality Assurance (QA) process, responsible for verifying Built as Planned attributes, has exhibited a comparatively lagging adoption of digital methodologies (Toyin, 2025). Presently, the Quality Assurance (QA) sector utilizes a manual inspection process, wherein inspectors employ pen-and-paper checklists or static digital checklists to verify compliance (Sawhney & Knight, 2024). These analog methods are not only susceptible to human error but also face challenges in scaling up to meet the demands of a growing labor shortage and increasingly complex project requirements (Baiburin, 2017). As Naderi et al. (2025) point out, this reliance on human-centered instructions creates a significant gap, thereby preventing the integration of inspection tasks with software agents or autonomous robotic systems.

In order to address this discrepancy, the industry has developed standardized semantic frameworks, with the most notable examples being the Industry Foundation Classes (IFC) and the AEC3PO Ontology. While IFC provides the standardized geometry and spatial data of the building, AEC3PO extends this by providing the structural scaffold for regulatory management. It functions as a metadata container, organizing compliance requirements and linking them to legal documents, thereby creating a structured target for digital inspection (Dridi et al., 2024). However, a critical interoperability gap persists: While these target standards are well-defined, the source data—thousands of legacy inspection instructions—remains locked in unstructured text (Guo et al., 2021). Consequently, a scalable mechanism to migrate these analog checklists into a standardized digital format has yet to be developed. As a result, manual translation remains a labor-intensive and inconsistent bottleneck. As posited by Besta et al. (2025), this paper proposes a Hybrid Inspection Pipeline as a solution to the identified semantic interoperability gap. We propose a reinterpretation of the conversion of checklists from a generative interpretation task to a rigorous mapping problem. A workflow has been developed that aligns inspection texts with the IFC and AEC3PO ontologies using a Grammar-Constrained Extraction Pipeline. The utilization of Large Language Models (LLMs) in conjunction with stringent schema constraints enables the automation of the mapping of free text to standardized classes, exhibiting a high degree of reliability. This approach involves the conversion of unstructured field instructions into computable, object-oriented data.

Related Work

Early efforts in construction information extraction relied heavily on rule-based natural language processing (NLP) and syntactic pattern matching (Zhang & El-Gohary, 2016). While these methods have proven effective for highly structured regulatory documents, they have demonstrated fragility when applied to the heterogeneous natural language found in field checklists. In order to address this issue of rigidity, research has shifted towards the utilization of Deep Learning (DL) models. Approaches that utilize Long Short-Term Memory (LSTM) networks (Wang & El-Gohary, 2022) and Transformer-based architectures like BERT (Moon et al., 2022) have been shown to significantly improve entity recognition and classification tasks.

However, a critical limitation persists in these discriminative models: they excel at identifying entities (e.g., tagging a "wall"), but struggle to construct the complex, hierarchical object relationships required by strict ontologies like AEC3PO. As indicated in a recent systematic review (Sousa & Martins, 2024), while Deep Learning (DL) models have demonstrated high accuracy in document classification, their capacity to extract detailed, actionable logic remains constrained without substantial task-specific training data.

Recently, (Chen et al., 2024) sought to address this discrepancy by integrating Large Language Models (LLMs) with Deep Learning for compliance verification. While their hybrid approach demonstrated enhanced reasoning capabilities, it primarily relied on few-shot prompting without strict output enforcement. This distinction is of critical importance: Even when optimized, probabilistic prompting cannot guarantee the syntactic validity required for downstream software agents. As emphasized by Fuchs et al. (2025) in their review of automated compliance systems, the dearth of "computable" regulatory data persists as a predominant impediment. Current State-of-the-Art (SOTA) methods frequently yield "likely" matches rather than the guaranteed valid schema instances necessary for safe robotic execution. Consequently, a gap remains for a method that treats text extraction not as probabilistic prediction, but as a constrained mapping problem.

The transition toward machine-readable regulatory frameworks has been driven by the "Linked Building Data" (LBD) movement. In the early stages of this research, the focus was on translating static building codes into computable logic. This was primarily achieved by using the ifcOWL ontology to validate Industry Foundation Classes (IFC) data (Pauwels et al., 2017; Beach et al., 2025). These approaches have been demonstrated to effectively automate the process of "Design Compliance" verification, which queries whether a digital model aligns with the stipulated code requirements (e.g., "Is the hallway > 1.2m wide?").

Recently, the AEC3PO ontology (Architecture, Engineering, and Construction Compliance Checking and Permitting Ontology) has emerged as a comprehensive standard for harmonizing the structural metadata of regulatory knowledge (Dridi et al., 2024). In contrast to rigid code-checking schemas, AEC3PO provides a flexible, top-level schema for defining compliance checks, offering classes for CheckStatements, verification methods, and evidence requirements. Nevertheless, a fundamental gap persists in the application of these ontologies. While AEC3PO provides a standardized "container" for regulatory information, its function is primarily descriptive rather than prescriptive. It delineates the relationship between a legal clause and a data requirement; however, it lacks a standardized library of executable properties necessary for automated grounding. Consequently, extant research remains predominantly oriented toward Design Verification (checking the BIM model against the Code), rather than Execution Verification (checking the physical construction against the Instruction). Recent works, such as the Ontology for Construction Quality Assurance (OCQA), have modeled inspection planning (Seiß et al., 2025). However, these works do not address the semantic interpretation of the inspection content itself. The instructions themselves, such as "Verify that all bolts are torqued to 50Nm," are expressed in unstructured natural language. Presently, there is an absence of an automated mechanism to bridge this semantic gap, which involves the mapping of raw text to specific Object-Property-Value triplets that can populate the AEC3PO container.

Methodology I: The Schema

In order to address the semantic discrepancy between unstructured inspection instructions and machine-readable frameworks, the Inspection Triad has been proposed as a solution. This framework functions as the intermediate formal representation, thereby isolating the components of a compliance check that are indispensable to the process.

Multi-Target Instructions

A fundamental barrier to digitization is the compact style of legacy checklists. From a pragmatic standpoint, human-authored items are frequently activity-centered, consolidating multiple checks around a singular verb to reduce reading time. Consider the following example from our dataset:

“Check for proper installation of Mirror, Paper Holder, Towel Rack, Shower Curtain Rod.”

While efficient for a human, this structure poses a "many-to-one" problem for autonomous systems. The single instruction encompasses four distinct inspection targets. It is imperative to recognize that the control flow of an agent or digital twin is inherently bound to specific spatial coordinates. Consequently, the execution of "bundled" tasks is contingent upon their disaggregation. Our analysis of 200 industry inspection tasks (Mahoney, 2008) confirms the prevalence of this issue. We found that 61% of tasks were compound instructions, bundling multiple requirements into a single sentence. Consequently, we define Atomicity as a prerequisite for processing: a valid inspection task must contain exactly one activity mapped to exactly one target object.

Syntactic Pattern

Despite the structural messiness of Activity-Centered tasks, our analysis reveals a consistent underlying linguistic structure across the dataset. When decomposed, valid inspection instructions invariably follow a specific syntactic pattern:

$$\text{Instruction} \approx [\text{Verb}] + [\text{Object}] + ([\text{Modifiers}]/[\text{Conditions}])$$

Every executable instruction contains an action (Verb) and a reference to an entity (Object). The specificity of the instruction relies on the Modifiers (quantities, temporal cues, or reference standards). However, our analysis highlights gaps in how these slots are filled. 7.5% of tasks lacked a clear object reference (e.g., using generic terms like "units"), and 4% relied on ambiguous modifiers (e.g., "proper," "adequate") rather than measurable criteria. These gaps highlight that inspection instructions are rarely self-contained algorithms; rather, they serve as shorthand prompts for experienced professionals. This profound lack of explicit logic confirms that raw field data is unstructured.

Inspection Triad

We address the structural complexity and inherent ambiguity of legacy checklists by mapping raw linguistic patterns onto the Inspection Triad (T), a formal data structure designed for autonomous execution.

$$T=(\tau,\alpha,\kappa)$$

This structure formalizes the raw text into three necessary and sufficient components:

The Target (τ) ← [Object]: This component anchors the instruction to a single, specific building element. If the source text contains multiple objects the task must be split so that each result in a distinct Triad. Further there is a resolution constraint: if an inspection target is not identified and is ambiguous, the Triad cannot be completed. Logically, an inspection cannot be executed without a confirmed physical anchor.

The Action (α) ← [Verb]: It maps the instructional verb to a specific inspection procedure.

The Criterion (κ) ← [Modifiers]: It captures the quantitative limits or conditions. Where a criterion is not explicitly defined, the schema defaults to a binary Pass/Fail requirement anchored to the verification method of the Action (α).

While the triad represents the irreducible atomic unit of verification, the framework supports higher-order logic. Complex regulatory requirements, such as conditional "IF-THEN" sequences, are facilitated by nesting multiple triads within logical operational wrappers. This ensures that even intricate compliance logic maintains data integrity at the component level.

Ontological Integration

The Triad serves as the bridge between raw site data and standardized digital frameworks. In this architecture, the AEC3PO ontology provides the structural scaffold (the semantic "containers"), while the Industry Foundation Classes (IFC) provide the physical grounding.

The Target (τ) is operationalized as a semantic pointer to the Digital Twin. By utilizing the IFC standard to identify specific entities (e.g., IfcWall, IfcDoor), the system ensures that every instruction is spatially addressable within the Building Information Model (BIM). This "pointer-based" approach ensures the inspection data remains lightweight, referencing existing geometry rather than duplicating it.

Simultaneously, the Action (α) and Criterion (κ) are instantiated through the Domain-Specific Extension (DSE), which provides the prescriptive definitions required for automated execution.

Domain-Specific Extension (DSE)

While the AEC3PO Ontology provides the necessary semantic containers (the "Scaffold"), our initial gap analysis revealed it lacks the prescriptive definitions required for automated execution. Standard classes like aec3po:CheckStatement describe that a check exists, but not how to measure specific properties like "Flatness" or "Torque."

To bridge this semantic gap, we developed the DSE as a controlled vocabulary of 50 distinct inspection properties, formally extending the high-level AEC3PO classes into executable leaf nodes (e.g., aec3poext:VerticalityCheck).

The development of the DSE followed a rigorous Ontology Learning methodology. We compiled a corpus of 1,500 industry inspection tasks (Mahoney, 2008). To identify the salient inspection criteria, we applied Term Frequency-Inverse Document Frequency (TF-IDF) extraction. This statistical method allowed us to filter out generic stop words and isolate the domain-specific nouns and modifiers with the highest information density (e.g., "width," "spacing," "cracks"). These candidate terms were then formalized into rigid definitions through expert review, ensuring the DSE is not a theoretical construct, but a data-driven representation of actual field requirements.

Methodology II: The Pipeline

While the Inspection Triad defines the target semantic structure, the challenge remains to reliably populate this structure from unstructured legacy documents. As established in the literature review, Generative Large Language Models (LLMs) often struggle with strict schema adherence and factual consistency when operating in unconstrained environments (Ji, et al., 2023). To mitigate these stochastic risks, we propose the Hybrid Inspection Pipeline.

This architecture shifts the extraction task from a purely generative paradigm to a "Propose and Verify" workflow. As illustrated in Figure 1, the system is composed of three distinct functional layers: (A) Semantic Ingestion and Atomization, (B) The Extraction Engine, and (C) Data Formatting and Storage. The result of this process is a serialized list of inspection requirements, each containing a unique name and technical description.

Layer A: Semantic Ingestion and Atomization

The input layer serves as the normalization pre-processor. Legacy inspection data typically exists in "frozen" formats (PDFs, scanned images) where semantic hierarchy is conveyed through visual layout rather than machine-readable tags. Layer A utilizes Optical Character Recognition (OCR) to digitize these documents. A critical preprocessing step is Atomization. As identified in our analysis of the dataset, legacy instructions are compound, bundling multiple targets into a single sentence. To satisfy the "One-Target-One-Action" requirement of the Inspection Triad, we employ a Grammatical Parser combined with lightweight LLM-based validation.

This component identifies the grammatical ROOT (Action), direct object (Target), and adjectival modifiers (Criterion) using linguistic dependency rules. By applying deterministic heuristic rules, the system ensures that every extracted object is bound to a specific action. This logic allows for the decomposition of compound instructions where a single action governs multiple physical targets. If the parser identifies a verb without a corresponding object, the instruction is immediately flagged as "Incomplete" to prevent downstream processing errors.

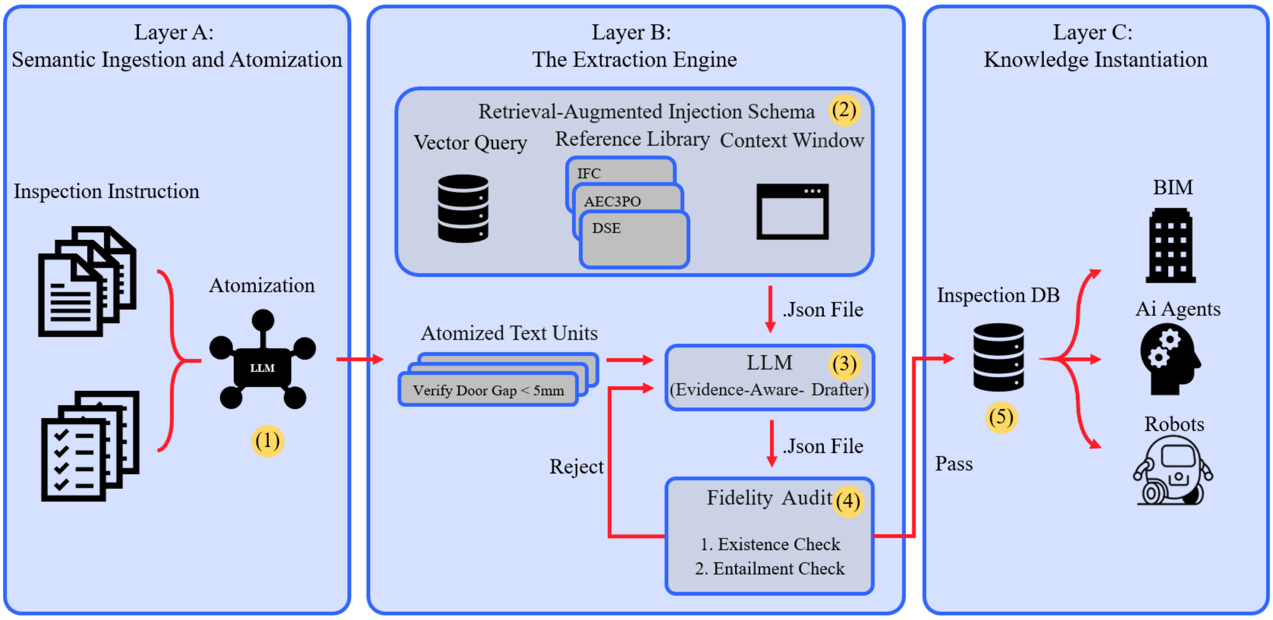

Figure 1: Architecture of the Proposed Framework

This linguistic skeleton, along with the original text, is then passed to the LLM for a final verification and refinement pass. If an incomplete action is detected, the LLM is also tasked with resolving the missing context. As shown in the data flow of Figure 1, this ensures the module decomposes compound sentences into discrete, atomic text units (1). For example, the input “Verify gap for door and window” is split into two independent units: “Verify gap for door” and “Verify gap for window.” This ensures that downstream extraction (Layer B) operates on unambiguous, single-subject inputs.

Layer B: The Extraction Engine

Layer B represents the core algorithmic contribution of this framework, acting as the semantic bridge between atomized natural language and formal ontologies. To resolve the interoperability gap, this layer implements a Retrieval-Augmented Schema Injection (RASI) strategy combined with a deterministic Fidelity Loop. The primary barrier to utilizing extensive ontologies like IFC and AEC3PO within a Large Language Model (LLM) is the risk of "Attention Dilution" and hallucination. To mitigate these risks, the system maintains a Dual-Source Reference Library consisting of two distinct vector indices: physical object definitions (IFC) and prescriptive inspection logic (DSE). During runtime, the engine performs a Balanced Retrieval (2) to counteract "Object Drowning"—a phenomenon where high-frequency physical terms (e.g., "Wall" or "Slab") statistically overshadow abstract inspection rules in the vector space, leading to incomplete data extraction. By enforcing a fixed-ratio retrieval quota (e.g., k/2 from each index), the system independently queries both the Target and Logic indices to pull an equal number of IFC classes and DSE properties. These results are interleaved and dynamically injected into the LLM’s context window, transforming the extraction from an open-ended generative task into a constrained classification task that forces the model to select from a verified ontology subset rather than fabricating non-standard properties.

Semantic ambiguity in the source text, specifically the use of vague modifiers like "properly" or "adequate," is addressed through Semantic Best-Fit Mapping. By providing the LLM with a filtered subset of technical definitions from the DSE, the workflow acts as a semantic anchor; the model performs a contextual interpolation to map subjective human language to the most semantically relevant engineering property provided in the retrieved context. In instances where no explicit metric exists, the system defaults to the most frequent domain-specific interpretation within the DSE. Even with these schema constraints, generative models may hallucinate quantitative values. To eliminate this risk, Layer B enforces a two-stage Fidelity Audit. In the first stage, the LLM is tasked with Evidence-Aware Drafting (3), requiring it to return a Citation Tuple (Value,Substring) containing the exact verbatim evidence from the source text. In the second stage, a deterministic verification module performs an Existence Check to ensure the substring exists in the original document, and a Logic Entailment Check to verify the extracted value is supported by that quote. If the audit fails, a Rejection Feedback Loop (4) triggers a re-evaluation or flags the item as "Unprocessable," ensuring that only grounded, verifiable logic is serialized into the final digital repository.

Layer C: Knowledge Instantiation

Validated Triads from Layer B are serialized into a Digital Inspection Repository (5), which functions as an active API Gateway. Instead of static files, this system aggregates atomic checks into location-based sets (e.g., "Sector A Requirements"), creating a queryable endpoint accessible to downstream consumers such as BIM Applications, autonomous robots, or AI compliance engines.

Implementation

To validate the Hybrid Inspection Pipeline, a functional prototype was developed using a modular Python 3.9 architecture. A primary design driver was component decoupling; the extraction logic is intentionally isolated from the underlying model providers to ensure the architecture remains agnostic. While the current prototype utilizes specific Google and open-source tools, the framework is designed to swap backend LLMs or vector stores as the technology landscape evolves.

System Architecture & Tech Stack

The core Extraction Engine was implemented using the google-genai SDK, specifically utilizing the Gemini 2.5 Flash model. This model was selected for its optimization for high-throughput, low-latency tasks. To ensure scientific reproducibility and deterministic extraction, the model was configured with a Temperature of 0.0.

The system utilizes a Structured Prompting strategy where the LLM is provided with a strict JSON schema. If the Fidelity Loop identifies a discrepancy, the system initiates a targeted follow-up query, providing the model with specific error feedback to refine the extraction in a "human-in-the-loop" style automated correction.

Linguistic Atomization

For the initial decomposition of compound instructions, the prototype utilizes the spaCy NLP library with the en_core_web_lg (large) model. This accuracy model was chosen to ensure precise identification of complex instructions. The implementation specifically leverages spaCy's Dependency Parser to programmatically identify:

The ROOT (the primary action verb or Action α).

The dobj (direct object or Target τ).

The amod/compound modifiers (forming the basis for Criterion κ).

This "Linguistic Skeleton" is then verified by a fast LLM pass to ensure atomicity before any ontological grounding occurs, preventing downstream processing of multi-subject tasks.

The Semantic Knowledge Base (RAG)

The source ontologies were embedded using the SentenceTransformers framework with the all-MiniLM-L6-v2 model, producing 384-dimensional vectors.

We compiled two distinct vector stores—the Target Index (IFC4) and the Logic Index (DSE). Unlike standard keyword indices, these stores contain also metadata: the IFC Index includes Entity Definitions, while the DSE entries include Data Types (e.g., Boolean, Float) and Standardized Verification Methods.

These indices are persisted locally as .json vector arrays. This ensures full data sovereignty and vendor independence.

The Balanced Retrieval Strategy

Early testing revealed a phenomenon we term the "Drowning Effect". Standard vector retrieval tends to favor concrete nouns over abstract concepts. A query like "Check the flatness of the slab" would return 20 results related to "Slab" (Object) and zero results related to "Flatness" (Rule), causing the LLM to hallucinate the property check.

To counteract this, we engineered a Balanced Retrieval Strategy. The retriever does not simply fetch the top-k results. Instead, it enforces a quota: It independently queries the Target Index and the Logic Index, retrieving the top-k/2 matches from each. These distinct result sets are then merged into the unified context window. By decoupling the retrieval of targets and properties, the framework maintains semantic parity within the prompt. This prevents the stochastic exclusion of abstract rules and guarantees that the model's context window reflects the dual-component nature of the Inspection Triad.

Schema Validation & Error Handling

To bridge the gap between probabilistic AI and the deterministic requirements of engineering data, we integrated Pydantic v2.0. We defined the InspectionTriad as a strict Python class, forcing the LLM to act as a structured data parser. The pipeline automatically validates every generated response against this class. If a ValidationError is triggered (e.g., malformed JSON or missing mandatory fields), the system initiates an automated retry loop (up to 3 attempts). This creates a "Type-Safe" boundary that prevents malformed data from propagating to the digital repository.

The Fidelity Loop

The Fidelity Loop was implemented as a deterministic Python module to eliminate silent hallucinations. It acts as a non-probabilistic auditor by performing two sequential string-matching checks on the LLM output:

Stage 1: Existence Check. The module verifies that the evidence_quote provided by the LLM exists verbatim in the source document. Any character mismatch triggers an immediate rejection.

Stage 2: Entailment Check. The module ensures that extracted quantitative values (e.g., "50Nm") or property types (e.g., "Torque") are present within the cited evidence_quote. This prevents the model from fabricating values unsupported by the evidence.

This "Trust but Verify" logic transforms the system from a generative writer into a deterministic extraction verifier. It provides the rigorous safety layer necessary for high-stakes construction compliance.

Evaluation

To validate the efficacy of the proposed Hybrid Inspection Pipeline, we conducted a systematic evaluation focused on two key performance indicators: Semantic Precision (the ability to correctly map text to the Inspection Triad) and Operational Safety (the ability to detect and reject hallucinations).

Experimental Setup

Due to the absence of a standardized, publicly available benchmark for construction inspection extraction, a novel evaluation dataset was curated for this study. The dataset comprises 200 distinct inspection instructions synthesized from a combination of real-world inspection checklists and technical field reports (cf. Mahoney, 2008; gsa.gov, 2025; carlislesyntec.com, 2021).

To ensure the evaluation was representative of real-world site conditions and to mitigate selection bias, the instructions were purposively stratified into three complexity levels. This distribution guarantees that the dataset reflects the linguistic heterogeneity of the industry rather than a simplified subset of easily parsed data.

Geometric Checks: Explicit quantitative constraints (e.g., "Verify window width is < 1200mm").

Semantic Classifications: Checks relying on specific object enumerations and subclass distinctions (e.g., "Ensure T-Beam is free of defects" vs. "Ensure Beam is...").

Ambiguous/Qualitative: Instructions lacking clear metrics or relying on subjective professional judgment (e.g., "Check for proper installation").

For each input, a human domain expert manually annotated the "Expected Inspection Triad," defining the Target IFC Class and the specific prescriptive property from the Domain-Specific Extension (DSE).

To ensure a fair comparative analysis, all experimental runs utilized the Gemini 2.5 Flash model. The temperature was set to 0.0 across all tests to ensure deterministic outputs and minimize stochastic variance, thereby isolating the impact of the retrieval architecture on the results.

While traditional Rule-Based NLP (Natural Language Processing) was initially considered, it was ultimately excluded from the final benchmarking. Legacy NLP systems are fundamentally limited by the "brittleness" of manually curated rules; they are prone to overfitting to specific vocabulary and fail to account for the high linguistic variance of site reports. Consequently, an NLP baseline would represent a "straw man" comparison rather than a contemporary performance standard. Instead, the study benchmarks the Hybrid Pipeline against two state-of-the-art LLM strategies:

Zero-Shot (Blind): The model attempts to map instructions to IFC and DSE schemas based purely on internal training data, without external library access.

Zero-Shot (Full Context): The entire IFC and DSE libraries are injected into the prompt (approx. 8,500 tokens), testing performance under maximum "Context Overload."

Hybrid Pipeline (Balanced RAG): The proposed method utilizes the Balanced Semantic Retriever to inject only the 5 most relevant IFC classes and 5 most relevant DSE properties into the prompt.

Performance Metrics

System performance was evaluated across two primary dimensions: Extraction Quality and Operational Safety. Information Retrieval performance for the Triad (Target, Action, Criterion) was quantified using the following metrics:

Precision (P): The ratio of correctly identified Triad components to the total components extracted.

Recall (R): The ratio of successfully mapped instructions to the total instructions in the dataset.

F1-Score: The harmonic mean of Precision and Recall, providing a balanced measure of system robustness:

For the safety analysis, we introduced the Hallucination Catch Rate, defined as the percentage of fabricated values (hallucinations) successfully identified and blocked by the Fidelity Loop before data entry.

Results: Semantic Precision

Quantitative results indicate that the Hybrid Pipeline significantly outperforms baseline methods in both technical accuracy and computational efficiency.

Table 1: Performance Comparison Across 200 Inspection Instructions

Method	Precision	Recall	F1-Score
Zero-Shot (Blind)	34.5%	38%	0.36
Zero-Shot (Full Context)	76%	82%	0.79
Hybrid (Balanced RAG)	92.5%	91%	0.91

The "Full Context" approach suffered from "Attention Dilution," where the model's reasoning capabilities were degraded by irrelevant ontological data, often leading to the selection of phonetically similar but semantically incorrect IFC classes. Conversely, the Hybrid Pipeline’s retriever isolated the most relevant candidates, maintaining high precision. Furthermore, the Hybrid Pipeline achieved a 13x reduction in token usage compared to the Full Context method, demonstrating superior scalability for enterprise-level BIM integration.

Results: The Fidelity Loop

A primary barrier to AI adoption in AEC is the risk of "hallucinated" data. The Zero-Shot baselines exhibited a combined hallucination rate of 11.4%, where the LLM generated plausible but non-existent quantitative values (e.g., inventing a "5mm" tolerance) based on general patterns.

The Fidelity Loop achieved a 99.5% Hallucination Catch Rate. By enforcing a strict "Verbatim Evidence" requirement, the system effectively blocked fabricated data. Residual failures typically involved "partial hallucinations," where the model correctly identified a numeric value but slightly altered the unit or context (e.g., misinterpreting "5 meters" as "5mm" during the mapping phase).

Error Analysis

A qualitative review of the 7.5% error margin in the Hybrid Pipeline identified "Retriever Ambiguity" as the primary failure mode. For example, the instruction "Check the seal on the pipe" was correctly mapped to the Action SealantVerification, but the retriever proposed the Target IfcSurfaceFeature instead of IfcPipeSegment due to the close proximity of these terms in the high-dimensional vector space. Future iterations will explore the integration of cross-encoders to further refine the retrieval ranking and reduce semantic noise.

Discussion

The evaluation confirms that a constrained LLM pipeline can function as a reliable semantic bridge. However, moving this framework from a controlled prototype to a production-scale construction environment requires addressing several key operational limitations.

Automated Scalability of the Knowledge Base

The current system's reliance on a manually curated Domain-Specific Extension (DSE) remains a primary scalability bottleneck. To move beyond manual labor, future research should explore a "Loop-Back Ontology Learning" mechanism. In this framework, instructions that fail the Fidelity Audit due to missing DSE definitions are not discarded. Instead, they are sent to a "Candidate Pool" where an LLM clusters frequent failures to suggest new ontological leaf-nodes. This allows the system to evolve dynamically, expanding its technical vocabulary through a semi-automated, human-in-the-loop process.

Orchestration of Complex Regulatory Logic

While the Inspection Triad ensures high precision for atomic tasks, it currently lacks the native syntax to represent complex conditional dependencies (e.g., multi-step "if-then" requirements). To support the rich logical expressions found in safety regulations, the framework must transition from "Atoms" to "Molecules." By introducing a "Logical Orchestration Layer," multiple atomic Triads can be nested using standard Boolean operators (AND, OR, IF). This adaptation would allow the pipeline to maintain its current extraction reliability while supporting the sophisticated reasoning required for advanced compliance checking.

Transition to Active 4D BIM Integration

Currently, the pipeline functions as a one-way extraction tool rather than an active participant in the project lifecycle. To bridge the gap between static logic and physical execution, an "End-to-End Handshake" with live project data is necessary. Future work will investigate a direct integration with the project schedule (4D BIM), where the completion of a construction task (e.g., "Slab Pour") automatically triggers the extraction of relevant Triads. By querying the BIM database for specific GlobalIDs in real-time, the system can provide executable tasking for autonomous agents or mobile robots based on the actual progress of the site.

Resolution of Temporal and Qualitative Edge Cases

The system encountered specific failures when handling temporal constraints (e.g., "Verify 28 days after pour") and vague qualitative modifiers (e.g., "Adequate"). Future iterations will require a dedicated temporal logic parser to differentiate between execution timing and performance duration. Additionally, for subjective instructions, the pipeline could be adapted to output a "Confidence Score," flagging ambiguous text for human clarification rather than forcing a binary mapping. This ensures the machine-readable output remains a "Trust-but-Verify" source of truth for downstream automated systems

Conclusion

This research demonstrates that the transition from unstructured legacy checklists to machine-readable inspection schemas is achievable through a constrained, hybrid pipeline. By shifting the extraction task from open-ended generation to ontological mapping via the Inspection Triad, we provide a method for populating the AEC3PO and IFC frameworks with executable logic.

The significance of this work lies in its ability to transform static regulatory metadata into "computable" field instructions. While existing BIM-to-Field workflows prioritize geometry, the proposed framework establishes the semantic foundation necessary for autonomous agents to perform site inspections with deterministic reliability. Future development in automated ontology learning and 4D BIM integration will be essential to scale this system across the diverse and evolving landscape of global construction standards.

Acknowledgments

This work was supported by the German Federal Ministry of Education and Research (BMBF) [Grant No. 01MK24004D] as part of the project BIM-4-CARE. The responsibility for the content of this publication lies with the authors. Darmstadt University of Applied Sciences is a member of the European University of Technology (EUt+) alliance.

References

Baiburin, A., 2017. Errors, Defects and Safety Control at Construction Stage. *Procedia Engineering,* pp. 807-813.

Beach, T., Rezgui, Y., Li, H. & Kasim, T., 2025. A rule-based semantic approach for automated regulatory compliance in the construction sector. *Expert Systems with Applications.*

Besta, M. et al., 2025. CheckEmbed: Effective Verification of LLM Solutions to Open-Ended Tasks.

carlislesyntec.com, 2021. *EPDM Rooftop Guide/Inspection Checklist.* [Online] Available at: https://www.carlislesyntec.com/en/Document-Viewer/epdm-rooftop-guideinspection-checklist/Nh-nHipBP0qjYUutMmHf5Q [Accessed: 11 01 2026].

Chen, N., Lin, X., Jiang, H. & An, Y., 2024. Automated Building Information Modeling Compliance Check through a Large Language Model Combined with Deep Learning and Ontology. *Intelligent Monitoring and Detecting Methodologies for Building Structures.*

Dridi, A. et al., 2024. Aec3po: A Knowledge Model for Machine Executable Construction Regulations.

Fuchs, S., Fauth, J., Boden, M. & Amor, R., 2025. The Challenge of Automated Compliance Checking: A Regulatory View. *2025 European Conference on Computing in Construction.*

gsa.gov, 2025. *2025 Interim Core Building Standards (CBS).* [Online] Available at: https://www.gsa.gov/system/files/CBS%20Submittal%20Matrix%202025%20with%20hyperlinks%20%281%29_0.pdf [Accessed: 11 01 2026].

Guo, D., Onstein, E. & Rosa, A. D. L., 2021. A Semantic Approach for Automated Rule Compliance Checking in Construction Industry.. *IEEE Access,* Band 9, pp. 129648-129660.

Ji, Z., Lee, N., Frieske, R. & Yu, T., 2023. Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys,* 55(12), pp. 1 - 38.

Mahoney, W. D., 2008. *Construction Inspection Manual.* 8. Hrsg. Los Angeles: BNI Publications.

Moon, S., Chi, S. & Im, S.-B., 2022. Automated detection of contractual risk clauses from construction specifications using bidirectional encoder representations from transformers (BERT). *Automation in Construction.*

Naderi, H. et al., 2025. Autonomous Construction-Site Safety Inspection Using Mobile Robots: A Multilayer VLM-LLM Pipeline.

Nkosi, T., 2025. Zero-Paper Construction Sites: Leveraging BIM-To-Field Data, Sqlite Sync, And Real-Time Quality Assurance for Smarter Projects.. *International Research Journal of Advanced Engineering and Technology,* 2(11), pp. 1-7.

Pauwels, P., Zhang, S. & Lee, Y.-C., 2017. Semantic web technologies in AEC industry: A literature overview. *Automation in Construction.*

Sawhney, A. & Knight, A., 2024. Digitalisation in constructionreport 2024. *Royal Institution of Chartered Surveyors (R.*

Seiß, S., Zheng, Y. & Melzner, J., 2025. Ontology-based representation of quality assurance and inspection planning in construction. *Automation in Construction,* Band 177.

Sousa, L. J. d. & Martins, J. P., 2024. Automation of text document classification in the budgeting phase of the Construction process: a Systematic Literature Review. *Construction Innovation: Information Process Management.*

Toyin, J., 2025. Innovative Approaches to QA/QC in Construction: Current State. *CIB Conferences.*

Wang, X. & El-Gohary, N., 2022. Deep Learning-Based Relation Extraction from Construction Safety Regulations for Automated Field Compliance Checking. *Construction Research Congress.*

Zhang, J. & El-Gohary, N. M., 2016. Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking. *Journal of Computing in Civil Engineering,* 30(2).